

**CNC**

**8070**

Programming manual.

(Ref: 2008)



**FAGOR AUTOMATION**

---

## TRANSLATION OF THE ORIGINAL MANUAL

This manual is a translation of the original manual. This manual, as well as the documents derived from it, have been drafted in Spanish. In the event of any contradictions between the document in Spanish and its translations, the wording in the Spanish version shall prevail. The original manual will be labeled with the text "ORIGINAL MANUAL".

---

## MACHINE SAFETY

It is up to the machine manufacturer to make sure that the safety of the machine is enabled in order to prevent personal injury and damage to the CNC or to the products connected to it. On start-up and while validating CNC parameters, it checks the status of the following safety elements. If any of them is disabled, the CNC shows the following warning message.

- Feedback alarm for analog axes.
- Software limits for analog and sercos linear axes.
- Following error monitoring for analog and sercos axes (except the spindle) both at the CNC and at the drives.
- Tendency test on analog axes.

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC resulting from any of the safety elements being disabled.

---

## HARDWARE EXPANSIONS

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC resulting from any hardware manipulation by personnel unauthorized by Fagor Automation.

If the CNC hardware is modified by personnel unauthorized by Fagor Automation, it will no longer be under warranty.

---

## COMPUTER VIRUSES

FAGOR AUTOMATION guarantees that the software installed contains no computer viruses. It is up to the user to keep the unit virus free in order to guarantee its proper operation. Computer viruses at the CNC may cause it to malfunction.

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC due a computer virus in the system.

If a computer virus is found in the system, the unit will no longer be under warranty.

---

## DUAL-USE PRODUCTS

Products manufactured by FAGOR AUTOMATION since April 1st 2014 will include "-MDU" in their identification if they are included on the list of dual-use products according to regulation UE 428/2009 and require an export license depending on destination.



FAGOR AUTOMATION

All rights reserved. No part of this documentation may be transmitted, transcribed, stored in a backup device or translated into another language without Fagor Automation's consent. Unauthorized copying or distributing of this software is prohibited.

The information described in this manual may be subject to changes due to technical modifications. Fagor Automation reserves the right to change the contents of this manual without prior notice.

All the trade marks appearing in the manual belong to the corresponding owners. The use of these marks by third parties for their own purpose could violate the rights of the owners.

It is possible that CNC can execute more functions than those described in its associated documentation; however, Fagor Automation does not guarantee the validity of those applications. Therefore, except under the express permission from Fagor Automation, any CNC application that is not described in the documentation must be considered as "impossible". In any case, Fagor Automation shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC if it is used in any way other than as explained in the related documentation.

The content of this manual and its validity for the product described here has been verified. Even so, involuntary errors are possible, hence no absolute match is guaranteed. However, the contents of this document are regularly checked and updated implementing the necessary corrections in a later edition. We appreciate your suggestions for improvement.

The examples described in this manual are for learning purposes. Before using them in industrial applications, they must be properly adapted making sure that the safety regulations are fully met.

# INDEX

About the product - CNC 8070 .....	9
Declaration of CE conformity and warranty conditions .....	13
Version history - CNC 8070 .....	15
Safety conditions .....	25
Returning conditions .....	29
CNC maintenance .....	31

**CHAPTER 1 CREATING A PROGRAM.**

1.1 Programming languages .....	33
1.2 Program structure .....	34
1.2.1 Program body .....	35
1.2.2 The subroutines .....	36
1.3 Program block structure .....	37
1.3.1 Programming in ISO code .....	38
1.3.2 High-level language programming .....	40
1.4 Programming of the axes .....	41
1.5 List of "G" functions .....	42
1.6 List of auxiliary (miscellaneous) M functions .....	45
1.7 List of statements and instructions .....	46
1.8 Programming the block labels .....	49
1.9 Comment programming .....	50
1.10 Variables and constants .....	51
1.11 Arithmetic parameters .....	52
1.12 Arithmetic and logic operators and functions .....	53
1.13 Arithmetic and logic expressions .....	55

**CHAPTER 2 MACHINE OVERVIEW**

2.1 Axis nomenclature .....	57
2.2 Coordinate system .....	59
2.3 Reference systems .....	60
2.3.1 Origins of the reference systems .....	61
2.4 Home search .....	62
2.4.1 Definition of "Home search" .....	62
2.4.2 "Home search" programming .....	63

**CHAPTER 3 COORDINATE SYSTEM**

3.1 Programming in millimeters (G71) or in inches (G70) .....	65
3.2 Absolute (G90) or incremental (G91) coordinates .....	66
3.2.1 Rotary axes .....	67
3.3 Absolute and incremental coordinates in the same block (I) .....	69
3.4 Programming in radius (G152) or in diameters (G151) .....	70
3.5 Coordinate programming .....	71
3.5.1 Cartesian coordinates .....	71
3.5.2 Polar coordinates .....	72
3.5.3 Angle and Cartesian coordinate .....	74

**CHAPTER 4 WORK PLANES.**

4.1 About work planes on lathe and mill models .....	78
4.2 Select the main new work planes .....	79
4.2.1 Mill model or lathe model with "trihedron" type axis configuration .....	79
4.2.2 Lathe model with "plane" type axis configuration .....	80
4.3 Select any work plane and longitudinal axis .....	81
4.4 Select the longitudinal axis of the tool .....	83

**CHAPTER 5 ORIGIN SELECTION**

5.1 Programming with respect to machine zero .....	86
5.2 Set the machine coordinate (G174) .....	88
5.3 Fixture offset .....	89
5.4 Coordinate preset (G92) .....	90
5.5 Zero offsets (G54-G59/G159) .....	91



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

5.5.1	Variables for setting zero offsets.....	93
5.5.2	Incremental zero offset (G158).....	94
5.5.3	Excluding axes in the zero offset (G157).....	96
5.6	Zero offset cancellation (G53).....	97
5.7	Polar origin preset (G30).....	98

**CHAPTER 6 TECHNOLOGICAL FUNCTIONS**

6.1	Machining feedrate (F).....	101
6.2	Feedrate related functions.....	103
6.2.1	Feedrate programming units (G93/G94/G95).....	103
6.2.2	Feedrate blend (G108/G109/G193).....	104
6.2.3	Constant feedrate mode (G197/G196).....	106
6.2.4	Cancellation of the % of feedrate override (G266).....	108
6.2.5	Acceleration control (G130/G131).....	109
6.2.6	Jerk control (G132/G133).....	111
6.2.7	Feed-Forward control (G134).....	112
6.2.8	AC-Forward control (G135).....	113
6.3	Spindle speed (S).....	114
6.4	Tool number (T).....	115
6.5	Tool offset number (D).....	118
6.6	Auxiliary (miscellaneous) functions (M).....	120
6.6.1	List of "M" functions.....	121
6.7	Auxiliary functions (H).....	122

**CHAPTER 7 THE SPINDLE. BASIC CONTROL.**

7.1	The master spindle of the channel.....	124
7.1.1	Manual selection of a master spindle.....	126
7.2	Spindle speed.....	127
7.2.1	G192. Turning speed limitation.....	128
7.2.2	Constant surface speed.....	129
7.3	Spindle start and stop.....	130
7.4	Gear change.....	132
7.5	Spindle orientation.....	134
7.5.1	The turning direction for spindle orientation.....	136
7.5.2	M19 function with an associated subroutine.....	138
7.5.3	Positioning speed.....	139
7.6	M functions with an associated subroutine.....	140

**CHAPTER 8 PATH CONTROL.**

8.1	Rapid traverse (G00).....	141
8.2	Linear interpolation (G01).....	143
8.3	Circular interpolation (G02/G03).....	149
8.3.1	Cartesian coordinates (Arc center programming).....	151
8.3.2	Cartesian coordinates (arc radius programming).....	153
8.3.3	Cartesian coordinates (arc radius pre-programming) (G263).....	155
8.3.4	Polar coordinates.....	156
8.3.5	Programming example (M model). Polar coordinates.....	158
8.3.6	Programming example (M model). Polar coordinates.....	159
8.3.7	Programming example (T model). Programming examples.....	160
8.3.8	Polar coordinates. Temporary Polar origin shift to the center of arc (G31).....	161
8.3.9	Cartesian coordinates. Arc center in absolute coordinates (no-modal) (G06).....	162
8.3.10	Cartesian coordinates. Arc center in absolute coordinates (modal) (G261/G262).....	163
8.3.11	Arc correction (G264/G265).....	165
8.4	Arc tangent to previous path (G08).....	167
8.5	Arc defined by three points (G09).....	169
8.6	Helical interpolation (G02/G03).....	171

**CHAPTER 9 TOOL PATH CONTROL. MANUAL INTERVENTION.**

9.1	Additive manual intervention (G201/G202).....	174
9.2	Exclusive manual intervention (G200).....	175
9.3	Jogging feedrate.....	176
9.3.1	Feedrate in continuous jog (#CONTJOG).....	176
9.3.2	Feedrate in incremental jog (#INCJOG).....	177
9.3.3	Feedrate in incremental jog (#MPG).....	178
9.3.4	Manual path movement limits (#SET OFFSET).....	179
9.3.5	Synchronization of coordinates and additive manual offset (#SYNC POS).....	180
9.4	Variables.....	181



CNC 8070

(REF: 2008)

<b>CHAPTER 10</b>	<b>ELECTRONIC THREADING AND RIGID TAPPING.</b>	
10.1	Electronic threading with constant pitch (G33) .....	183
10.1.1	Programming examples (·M· model).....	186
10.1.2	Programming examples (·T· model) .....	187
10.2	Electronic threading with variable pitch (G34) .....	189
10.3	Rigid tapping (G63).....	193
10.4	Withdraw the axes after interrupting an electronic threading (G233).....	195
10.4.1	Variables related to G233. ....	198
10.4.2	Programming example.....	198
<b>CHAPTER 11</b>	<b>GEOMETRIC ASSISTANCE.</b>	
11.1	Semi-rounded corner (G50).....	199
11.2	Square corner (G07/G60).....	200
11.3	Controlled corner rounding (G05/G61).....	201
11.3.1	Edge rounding. #ROUNDPAR [1].....	202
11.3.2	Edge rounding. #ROUNDPAR [2].....	203
11.3.3	Edge rounding. #ROUNDPAR [3].....	204
11.3.4	Edge rounding. #ROUNDPAR [4].....	205
11.3.5	Edge rounding. #ROUNDPAR [5].....	206
11.4	Corner rounding, radius blend, (G36).....	209
11.5	Corner chamfering, (G39).....	211
11.6	Tangential entry (G37).....	213
11.7	Tangential exit (G38).....	214
11.8	Mirror image (G10, G11, G12, G13, G14).....	215
11.8.1	Activating the mirror image (G11, G12, G13, G14).....	215
11.8.2	Mirror image cancellation (G10).....	218
11.8.3	Summary of the variables.....	218
11.9	Pattern rotation (G73).....	219
11.10	Scaling factor (G72#SCALE).....	222
11.10.1	General scaling factor (G72#SCALE).....	222
11.10.2	Scaling factor per axis (G72).....	226
11.10.3	Summary of the variables.....	227
11.11	Work zones (G120/G121/G122/G123).....	228
11.11.1	CNC behavior when there are active work zones.....	229
11.11.2	Defining the linear limits of the work zone (G120/G121).....	230
11.11.3	Set circular limits of the work zone (G123).....	232
11.11.4	Enable/disable the work zones (G122).....	234
11.11.5	Summary of the variables.....	237
<b>CHAPTER 12</b>	<b>ADDITIONAL PREPARATORY FUNCTIONS</b>	
12.1	Dwell (G04 / #TIME).....	239
12.2	Software limits.....	241
12.2.1	Define the first software limit (G198/G199).....	242
12.2.2	Define the first software limit via variables.....	244
12.2.3	Define the second software limit via variables.....	245
12.2.4	Variables associated with the software limits.....	246
12.3	Turn Hirth axis on and off (G170/G171).....	247
12.4	Set and gear change.....	248
12.4.1	Change parameter set of an axis (G112).....	248
12.4.2	Change the gear and set of a Sercos drive using variables.....	249
12.4.3	Variables related to set and gear change.....	250
12.5	Smooth the path and the feedrate.....	251
12.5.1	Smooth the path (#PATHND).....	251
12.5.2	Smooth the path and the feedrate (#FEEDND).....	252
<b>CHAPTER 13</b>	<b>TOOL COMPENSATION</b>	
13.1	Tool radius compensation.....	255
13.1.1	Location code (shape or type) of the turning tools.....	256
13.1.2	Functions associates with radius compensation.....	259
13.1.3	Beginning of tool radius compensation.....	262
13.1.4	Sections of tool radius compensation.....	265
13.1.5	Change of type of radius compensation while machining.....	269
13.1.6	Cancellation of tool radius compensation.....	271
13.2	Tool length compensation.....	274
13.3	3D tool compensation.....	276
13.3.1	Programming the vector in the block.....	278
<b>CHAPTER 14</b>	<b>CONTROLLING THE EXECUTION AND DISPLAYING THE PROGRAM.</b>	
14.1	Conditional block skip (I).....	279

14.2	Abort the execution of the program and resume it in another block or program. ....	280
14.2.1	Define the execution resuming block or program (#ABORT).....	281
14.2.2	Default point to continue the execution (#ABORT OFF).....	282
14.3	Block repetition (NR).....	283
14.3.1	Movement block repetition n times (NR/NR0).....	283
14.3.2	Prepare a subroutine without executing it (NR0).....	284
14.4	Block group repetition (#RPT).....	285
14.4.1	Programming example.....	287
14.5	Interrupt block preparation until an event is caused (#WAIT FOR).....	288
14.6	Interrupt block preparation (#FLUSH).....	289
14.7	Enable/disable the single-block treatment (#ESBLK/ #DSBLK).....	290
14.8	Enable/disable the stop signal (#DSTOP/#ESTOP).....	291
14.9	Enable/disable the feed-hold signal (#DFHOLD/#EFHOLD).....	292
14.10	Block skip (\$GOTO).....	293
14.11	Conditional execution (\$IF).....	294
14.11.1	Conditional execution (\$IF).....	294
14.11.2	Conditional execution (\$IF - \$ELSE).....	295
14.11.3	Conditional execution (\$IF - \$ELSEIF).....	296
14.12	Conditional execution (\$SWITCH).....	297
14.13	Block repetition (\$FOR).....	298
14.14	Conditional block repetition (\$WHILE).....	300
14.15	Conditional block repetition (\$DO).....	301

## CHAPTER 15 SUBROUTINES.

15.1	Executing subroutines from RAM memory. ....	305
15.2	Definition of the subroutines.....	306
15.3	Subroutine execution.....	307
15.3.1	LL. Call to a local subroutine.....	308
15.3.2	L. Call to a global subroutine.....	308
15.3.3	#CALL. Call to a global or local subroutine.....	309
15.3.4	#PCALL. Call to a global or local subroutine initializing parameters.....	310
15.3.5	#MCALL. Modal call to a local or global subroutine.....	311
15.3.6	#MDOFF. Turning the subroutine into non-modal.....	313
15.3.7	#RETDSBLK. Execute subroutine as a single block.....	314
15.4	#PATH. Define the location of the global subroutines.....	315
15.5	OEM subroutine execution.....	316
15.6	Generic user subroutines (G500-G599).....	318
15.7	Assistance for subroutines.....	321
15.7.1	Subroutine help files.....	321
15.7.2	List of available subroutines.....	323
15.8	Interruption subroutines.....	324
15.8.1	Repositioning axes and spindles from the subroutine (#REPOS).....	325
15.9	Subroutine associated with the start.....	326
15.10	Subroutine associated with the reset.....	327
15.11	Subroutines associated with the kinematics calibration cycle.....	328

## CHAPTER 16 EXECUTING BLOCKS AND PROGRAMS

16.1	Executing a program in the indicated channel.....	329
16.2	Executing a block in the indicated channel.....	331

## CHAPTER 17 C AXIS

17.1	Activating the spindle as "C" axis.....	334
17.2	Machining of the face of the part.....	336
17.3	Machining of the turning side of the part.....	338

## CHAPTER 18 ANGULAR TRANSFORMATION OF AN INCLINE AXIS.

18.1	Turning angular transformation on and off.....	343
18.2	Freezing (suspending) the angular transformation.....	344
18.3	Obtaining information on angular transformation.....	345

## CHAPTER 19 TANGENTIAL CONTROL.

19.1	Turning tangential control on and off.....	349
19.2	Freezing tangential control.....	352
19.3	Obtaining information on tangential control.....	354

## CHAPTER 20 KINEMATICS AND COORDINATE TRANSFORMATION

20.1	Coordinate systems.....	356
------	-------------------------	-----



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

20.2	Movement in an inclined plane. ....	357
20.3	Select a kinematics (#KIN ID). ....	358
20.4	Coordinate systems (#CS / #ACS). ....	359
20.4.1	Define a coordinate system (MODE1). ....	363
20.4.2	Define a coordinate system (MODE2). ....	364
20.4.3	Define a coordinate system (MODE3). ....	365
20.4.4	Define a coordinate system (MODE4). ....	366
20.4.5	Define a coordinate system (MODE5). ....	367
20.4.6	Define a coordinate system (MODE6). ....	368
20.4.7	Operation with 45° spindles (Huron type). ....	370
20.4.8	How to combine several coordinate systems. ....	372
20.5	Tool perpendicular to the inclined plane (#TOOL ORI). ....	374
20.5.1	Programming examples. ....	375
20.6	Using RTCP (Rotating Tool Center Point). ....	377
20.6.1	Programming examples. ....	379
20.7	Correct the implicit tool length compensation of the program (#TLC). ....	381
20.8	How to withdraw the tool when losing the plane. ....	382
20.9	Tool orientation in the part coordinate system. ....	383
20.9.1	Activate tool orientation in the part coordinate system. ....	383
20.9.2	Cancel tool orientation in the part coordinate system. ....	384
20.9.3	How to manage the discontinuities in the orientation of rotary axes. ....	385
20.9.4	Screen for choosing the desired solution. ....	387
20.9.5	Execution example. Selecting a solution. ....	388
20.10	Selecting the rotary axes that position the tool in type-52 kinematics. ....	389
20.11	Transform the current part zero considering the position of the table kinematics. ....	390
20.11.1	Process of saving a part zero with the table axes in any position. ....	391
20.11.2	Example to maintain the part zero without rotating the coordinate system. ....	392
20.12	Summary of kinematics related variables. ....	393

**CHAPTER 21 HSC. HIGH SPEED MACHINING.**

21.1	Recommendations for machining. ....	398
21.2	User subroutines G500-G501 to turn HSC on/off. ....	399
21.2.1	Alternative example for functions G500-G501 supplied by Fagor. ....	401
21.3	HSC SURFACE mode. Optimization of surface finish. ....	403
21.4	HSC CONTERROR mode. Optimizing the contouring error. ....	406
21.5	HSC FAST mode. Optimizing the machining feedrate. ....	408
21.6	Canceling the HSC mode. ....	410

**CHAPTER 22 VIRTUAL TOOL AXIS.**

22.1	Activate the virtual tool axis. ....	412
22.2	Cancel the virtual tool axis. ....	413
22.3	Variables associated with the virtual tool axis. ....	414

**CHAPTER 23 DISPLAYING MESSAGES, WARNINGS AND ERRORS.**

23.1	#ERROR. Display an error on the screen. ....	416
23.2	#WARNING / #WARNINGSTOP. Displaying a warning on the screen. ....	418
23.3	#MSG. Display a message on the screen. ....	420
23.4	Format identifiers and special characters. ....	422
23.5	cncError.txt file. List of OEM and user errors and warnings. ....	423
23.6	cncMsg.txt file. List of OEM and user messages. ....	424
23.7	Summary of the variables. ....	425

**CHAPTER 24 DMC (DYNAMIC MACHINING CONTROL).**

24.1	Activating the DMC. ....	428
24.2	Deactivating the DMC. ....	430
24.3	Summary of the variables. ....	431
24.4	Operating with DMC. ....	433
24.4.1	DMC operation. ....	433
24.4.2	DMC status and progress. Automatic mode. ....	435
24.4.3	Percentage of feedrate (feedrate override). ....	435

**CHAPTER 25 OPENING AND WRITING FILES.**

25.1	#OPEN. Open file for writing. ....	437
25.2	#WRITE. Writing in a file. ....	439
25.3	#CLOSE. Close a file. ....	441
25.4	cncWrite.txt file. List of OEM and user messages. ....	442



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**CHAPTER 26 PROGRAMMING STATEMENTS.**

---

26.1	Display instructions. Define the size of the graphics area .....	443
26.2	ISO generation.....	446
26.3	Electronic axis slaving .....	449
26.4	Axis parking .....	450
26.5	Modifying the configuration of the axes of a channel.....	452
26.6	Modifying the configuration of the spindles of a channel .....	457
26.7	Spindle synchronization .....	460
26.8	Selecting the loop for an axis or a spindle. Open loop or closed loop .....	464
26.9	Collision detection.....	466
26.10	Spline interpolation (Akima).....	468
26.11	Polynomial interpolation.....	471
26.12	Acceleration control .....	472
26.13	Definition of macros .....	474
26.14	Communication and synchronization between channels.....	476
26.15	Movements of independent axes.....	479
26.16	Electronic cams. ....	483
26.17	On line modification of the machine configuration in HD graphics (xca files) .....	486

**CHAPTER 27 CNC VARIABLES.**

---





# ABOUT THE PRODUCT - CNC 8070

## BASIC CHARACTERISTICS.

Basic characteristics.	·BL·	·OL·	·L·
Number of axes.	3 to 7	3 to 31	3 to 31
Number of spindles.	1	1 to 6	1 to 6
Number of tool magazines.	1	1 to 4	1 to 4
Number of execution channels.	1	1 to 4	1 to 4
Number of interpolated axes (maximum).	4	3 to 31	3 to 31
Number of handwheels.	1 to 12		
Type of servo system.	Analog / Digital Sercos Digital Mechatrolink		Analog Sercos Digital
Communications.	RS485 / RS422 / RS232 Ethernet		
PCI expansion.	No	Option	No
Integrated PLC. PLC execution time. Digital inputs / Digital outputs. Marks / Registers. Timers / Counters. Symbols.	< 1ms/K 1024 / 1024 8192 / 1024 512 / 256 Unlimited		
Block processing time.	< 1 ms		< 1 ms

Remote modules.	RIOW	RIO5	RIOR	RCS-S	RIOW-E Inline
Communication with the remote modules.	CANopen	CANopen	CANopen	Sercos	EtherCAT
Digital inputs per module.	8	24 / 48	48	---	8
Digital outputs per module.	8	16 / 32	32	---	8
Analog inputs per module.	4	4	---	---	4
Analog outputs per module.	4	4	---	4	2
Inputs for PT100 temperature sensors.	2	2	---	---	---
Feedback inputs.	---	---	---	4 (*)	---

(\*) TTL / Differential TTL / Sinusoidal 1 Vpp / SSI protocol / FeeDat / EnDat

### Customizing.

PC-based open system, fully customizable.  
 INI configuration files.  
 Tool for display configuration FGUIM.  
 Visual Basic®, Visual C++®, etc.  
 Internal databases in Microsoft® Access.  
 OPC compatible interface



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## SOFTWARE OPTIONS.

Some of the features described in this manual are dependent on the acquired software options. The active software options for the CNC can be consulted in the diagnostics mode (accessible from the task window by pressing [CTRL] [A]), under software options.

Configuration	Software options
	<p>Options installed</p> <p>Number of channels: 1</p> <p>Number of axes: 4</p> <p>Number of interpolated axes: 4</p> <p>Number of spindles: 1</p> <p>Number of magazines: 1</p> <p>Y axis</p> <p>C axis</p> <p>Tandem axes</p> <p>Probe</p> <p>High speed machining II</p> <p>Digital SERCOS</p> <p>Conversational</p>

Consult the ordering handbook for information on the software options available for your model.

### SOFT ADDIT AXES

#### Additional shaft.

Add axes to the default configuration.

### SOFT ADDIT SPINDLES

#### Additional spindle.

Add spindles to the default configuration.

### SOFT ADDIT TOOL MAGAZ

#### Additional tool magazine.

Add tool magazines to the default configuration.

### SOFT ADDIT CHANNELS

#### Additional channel.

Add channels to the default configuration.

### SOFT 4 AXES INTERPOLATION LIMIT

#### Limited to 4 interpolated axes.

It limits the number of axes to 4, where the CNC can also interpolate these at the same time.

### SOFT i4.0 CONNECTIVITY PACK

#### Industry 4.0.

This option allows for the use and capture of data using the FSYS.

### SOFT OPEN SYSTEM

#### Open system.

The CNC is a closed system that offers all the features needed to machine parts. Nevertheless, at times there are some customers who use third-party applications to take measurements, perform statistics or other tasks apart from machining a part.

This feature must be active when installing this type of application, even if they are Office files. Once the application has been installed, it is recommended to close the CNC in order to prevent the operators from installing other kinds of applications that could slow the system down and affect the machining operations.

### SOFT DIGITAL SERCOS

#### Sercos digital bus.

Sercos digital bus.

### SOFT DIGIT NO FAGOR

#### Non-Fagor digital servo system.

Mechatrolink digital bus.

### SOFT EDIT/SIMUL

#### EDISIMU mode (editing and simulation).

It allows for the editing, modification and simulation of a part-program.

### SOFT IEC 61131 LANGUAGE

#### IEC 61131 language

IEC 61131 is a PLC programming language that is very popular in alternative markets, which is slowly entering into the machine-tool market. With this feature, the PLC may be programmed either in the usual Fagor language or in IEC 61131 format.

### SOFT TOOL RADIUS COMP

#### Compensación de radio.

Tool compensation allows programming the contour to be machined based on part dimensions of the and without taking into account the dimensions of the tool that will be used later on. This avoids having to calculate and define the tool path based on the tool radius.

### SOFT PROFILE EDITOR

#### Profile editor.

Allows for the part profiles to be edited graphically and to import dxf files.

### SOFT RTCP

#### Dynamic RTCP (Rotating Tool Center Point).

The dynamic RTCP option is required for interpolation machining with 4, 5 or 6 axis.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## SOFT C AXIS

### C axis.

It activates the kinematics for working with the C axis and the associated canned cycles. The CNC can control several C axes. The parameters of each axis indicate if it will function as a C axis or not, where it will not be necessary to activate another axis for the machine parameters.

## SOFT TANDEM AXES

### Tandem axes.

A tandem axis consists in two motors mechanically coupled (slaved) and making up a single transmission system (axis or spindle). A tandem axis helps provide the necessary torque to move an axis when a single motor is not capable of supplying enough torque to do it.

When activating this feature, it should be kept in mind that for each tandem axis of the machine, another axis must be added to the entire configuration. For example, on a large 3-axis lathe (X Z and tailstock), if the tailstock is a tandem axis, the final purchase order for the machine must indicate 4 axes.

## SOFT SYNCHRONISM

### Synchronization of axes and spindles.

The axes and ballscrews may be synchronized in two ways: in terms of speed or position. The CNC configuration takes into consideration the synchronization of 2 axes or 2 spindles. Once synchronized, only the master displays and programs the element.

## SOFT HSSA II MACHINING SYSTEM

### High Speed Surface Accuracy II.

Option to enable the HSSA-II (High Speed Surface Accuracy) algorithm for high-speed machining (HSC), which has the following advantages compared to the HSSA-I algorithm.

- Advanced algorithm for point preprocessing in real time.
- Extended curvature algorithm with dynamic limitations. Improved acceleration and jerk control.
- Greater number of pre-processed points.
- Filters to smooth out the dynamic machine behavior.

## SOFT TANGENTIAL CONTROL

### Tangential control.

"Tangential Control" maintains a rotary axis always in the same orientation with respect to the programmed tool path. The machining path is defined on the axes of the active plane and the CNC maintains the orientation of the rotary axis along the entire tool path.

## SOFT DRILL CYCL OL

### Drilling ISO cycles for the OL model.

Drilling ISO cycles for the OL model (G80, G81, G82, G83).

## SOFT PROBE

### Probing canned cycles.

The CNC may have two probes; usually a tabletop probe to calibrate tools and a measuring probe to measure the part.

This option activates the functions G100, G103 and G104 (for probe movements); probe canned cycles are not included.

## SOFT THIRD PARTY I/Os

### Third-party CANopen.

Enables the use of non-Fagor CANopen modules.

## SOFT FVC STANDARD

### SOFT FVC UP TO 10m3

### SOFT FVC MORE TO 10m3

### Volumetric compensation.

5-axis machines are generally used during the manufacturing of large parts. The accuracy of the parts is limited by the machine manufacturing tolerances and is effected by temperature variations during machining.

In sectors such as the aerospace industry, machining demands mean that classic compensation tools are becoming suboptimal. Volumetric compensation FVC comes in to complement the machine adjusting tools. When mapping the total work volume of the machine, the CNC knows the exact position of the tool at all times. After applying the required compensation, the resulting part is made with the desired precision and tolerance.

There are 3 options, which depend on the size of the machine.

- FVC STANDARD: 25-point compensation on each axis. Quick calibration (time), but less precise than the other two, but sufficient for the desired tolerances.
- FVC UP TO 10m3: Volume compensation up to 10 m<sup>3</sup>. More accurate than FVC STANDARD, but requires a more accurate calibration using a Tracer or Tracker laser).
- FVC MORE TO 10m3: Volume compensation greater than 10 m<sup>3</sup>. More accurate than FVC STANDARD, but requires a more accurate calibration using a Tracer or Tracker laser.

## SOFT 60 PWM CONTROL

### Pulse-Width Modulation.

This function is only available for Sercos bus controlled systems. It is mostly oriented toward laser machines for the cutting of very thick sheets, where the CNC generates a series of PWM pulses to control the power of the laser when drilling the starting point.

This feature is essential for cutting very thick sheets and it requires two quick digital outputs located on the central unit. With this new feature, the OEM does not need to install or program any external device, which reduces machine costs and installation times. The end user also benefits, since the "Cutting with PWM" feature is much easier to use and program.

## SOFT 60 GAP CONTROL

### Gap control.

This is mostly oriented toward laser machines. Gap control makes it possible to maintain a set distance between the laser nozzle and the surface of the sheet. This distance is calculated by a sensor connected to the CNC, so that the CNC offsets the sensor variations on the distance programmed with additional movements in the axis programmed for the gap.

## SOFT DMC

### Dynamic Machining Control.

DMC adapts the feedrate during machining to maintain the cutting power as close as possible to ideal machining conditions.

## SOFT FMC

### Fagor Machining Calculator.

The FMC application consists of a database of materials to be machined and machining operations (milling and turning) with an interface to choose suitable cutting conditions for these operations.

## SOFT FFC

### Fagor Feed Control.

During the execution of a canned cycle of the editor, the FFC function makes it possible to replace the feedrate and speed programmed in the cycle with the active values of the execution, which are acted upon by the feed override and speed override.

**SOFT 60/65/70 OPERATING TERMS****Temporary user license.**

The "Operating Terms" option activates a temporary user license for the CNC, which is valid until the date determined by the OEM.

**SOFT MANUAL NESTING****Manual nesting**

Nesting consists of creating a pattern on the sheet material using previously defined figures (in dxf, dwg or parametric files), so as to use most of the sheet as possible. Once the pattern has been defined, the CNC creates a program. During manual nesting, the operator distributes the parts on top of the sheet material.

**SOFT AUTO NESTING****Automatic nesting**

Nesting consists of creating a pattern on the sheet material using previously defined figures (in dxf, dwg or parametric files), so as to use most of the sheet as possible. Once the pattern has been defined, the CNC creates a program. During automatic nesting, the application distributes the figures on the sheet material and optimizes the spaces.

# DECLARATION OF CE CONFORMITY AND WARRANTY CONDITIONS

## DECLARATION OF CONFORMITY

The declaration of conformity for the CNC is available in the downloads section of FAGOR'S corporate website. <http://www.fagorautomation.com>. (Type of file: Declaration of conformity).

## WARRANTY TERMS

The warranty conditions for the CNC are available in the downloads section of FAGOR's corporate website. <http://www.fagorautomation.com>. (Type of file: General sales-warranty conditions).

BLANK PAGE

# VERSION HISTORY - CNC 8070

Here is a list of the features added to each manual reference.

## Ref. 0201

<b>Software V01.00</b>
First version. Milling model.

## Ref. 0212

<b>Software V01.10</b>	
New repositioning feedrate after tool inspection.	• Machine parameter: REPOSFEED.
New treatment of the JOG keys. Different keys to select the axis and the direction.	• Machine parameter: JOGKEYDEF.
Know the dimensions of the kinematics on an axis.	• Variable: (V.)A.HEADOF.xn
Keyboard simulation from the PLC.	• Variable: (V.)G.KEY
General scaling factor.	• Instruction: #SCALE.
Probe selection.	• Instruction: #SELECT PROBE.
Probing canned cycles.	• Instruction: #PROBE.
Programming of warnings.	• Instruction: #WARNING.
Block repetition.	• Instruction: #RPT.
Know the active general scaling factor.	• Variable: (V.)G.SCALE
Knowing which is the active probe.	• Variable: (V.)G.ACTIVPROBE
Improved programming of high speed machining.	• Instruction: #HSC.
Improved programming of axis swapping.	• Instructions: #SET #CALL #FREE #RENAME
The number of macros in a program is now limited to 50.	• Macros.

## Ref. 0501

<b>Software V02.01</b>	
Windows XP operating system.	
Emergency shutdown with battery (central unit PC104).	
Multi-channel system, up to 4 channels. Swapping of axes and spindles, communication and synchronization between channels, common arithmetic parameters, access variables by channel, etc.	
Multi-spindle system, up to 4 spindles.	
Tool management with up to 4 magazines.	
Tool radius compensation mode (G136/G137) by default	• Machine parameter: IRCOMP.
New behavior for rotary axes.	
The "(V.)TM.MZWAIT" variable is not necessary in the subroutine associated with M06.	• Subroutine associated with M6. • Variable: (V.)TM.MZWAIT
Know the software version.	• Variable: (V.)G.SOFTWARE
Variables related to loop adjustment. Gain setting via PLC.	• Variables: (V.)A.PLCFFGAIN.xn (V.)A.PLCACFGAIN.xn (V.)A.PLCPROGAIN.xn
Variables related to loop adjustment. Position increment and sampling period.	• Variables: (V.)A.POSINC.xn (V.)A.TPOSINC.xn (V.)A.PREVPOSINC.xn
Variables related to loop adjustment. Fine adjustment of feedrate, acceleration and jerk.	• Variables: (V.)A.FEED.xn (V.)A.TFEED.xn (V.)A.ACCEL.xn (V.)A.TACCEL.xn (V.)A.JERK.xn (V.)A.TJERK.xn
Variables related to the feedback inputs.	• Variables: (V.)A.COUNTER.xn (V.)A.COUNTERST.xn (V.)A.ASINUS.xn (V.)A.BSINUS.xn



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Software V02.01	
Optimize the reading and writing of variables from the PLC. Only the access to the following variables will be asynchronous. <ul style="list-style-type: none"> <li>The tool variables will be read asynchronously when the tool is neither the active one nor in the magazine.</li> <li>The tool variables will be written asynchronously whether the tool is the active one or not.</li> <li>The variables referred to local arithmetic parameters of the active levels will be read and written asynchronously.</li> </ul>	<ul style="list-style-type: none"> <li>Reading and writing of variables from the PLC.</li> </ul>
Spindle parking and unparking.	<ul style="list-style-type: none"> <li>Instructions: #PARK #UNPARK</li> </ul>
Tool radius compensation. <ul style="list-style-type: none"> <li>Behavior of the beginning and end of tool radius compensation when not programming a movement.</li> <li>Changing the type of radius compensation while machining.</li> </ul>	
Via program, loading a tool in a specific magazine position.	
Programming of modal subroutines.	<ul style="list-style-type: none"> <li>Instruction: #MCALL.</li> </ul>
Executing a block in a channel.	<ul style="list-style-type: none"> <li>Instruction: #EXBLK.</li> </ul>
Programming the number of repetitions in the block.	<ul style="list-style-type: none"> <li>NR command.</li> </ul>

**Ref. 0504**

Software V02.03	
Electronic cam programming (real coordinates).	<ul style="list-style-type: none"> <li>Instruction: #CAM.</li> </ul>
Synchronization of independent axis (real coordinates).	<ul style="list-style-type: none"> <li>Instruction: #FOLLOW.</li> </ul>
Movement of the independent axis.	<ul style="list-style-type: none"> <li>Instruction: #MOVE.</li> </ul>
G31. Temporary polar origin shift to the center of interpolation.	<ul style="list-style-type: none"> <li>Function G31.</li> </ul>
G112. Change the drive's parameter set.	<ul style="list-style-type: none"> <li>Function G112.</li> </ul>

**Ref. 0509**

Software V03.00	
Lathe model. Machining canned cycles, lathe tool calibration, variables to consult the geometry of lathe tools, etc.	
Incline axis.	
Permit using the G95 function in jog mode.	<ul style="list-style-type: none"> <li>Machine parameter: FPRMAN.</li> </ul>
"C" axis maintained.	<ul style="list-style-type: none"> <li>Machine parameter: PERCAX.</li> </ul>
Magazine-less system.	
Ground tools for a turret magazine.	
Variable to read the accumulated PLC offset.	<ul style="list-style-type: none"> <li>Variable: (V.)A.ACTPLCOF.xn</li> </ul>
Variable to obtain a linear estimation of the following error.	<ul style="list-style-type: none"> <li>Variable: (V.)A.FLWEST.xn</li> </ul>
Variables to read the instant value of feed-forward or AC-forward.	<ul style="list-style-type: none"> <li>Variables: (V.)A.ACTFFW.xn (V.)A.ACTACF.xn</li> </ul>
Variable to know the line number of the file being executed.	<ul style="list-style-type: none"> <li>Variable: (V.)G.LINEIN</li> </ul>
Variable to know what kind of cycle is active.	<ul style="list-style-type: none"> <li>Variable: (V.)G.CYCLETYPEON</li> </ul>
Variable to know the tool orientation.	<ul style="list-style-type: none"> <li>Variable: (V.)G.TOOLDIR</li> </ul>
Variable to know whether the HSC mode is active or not.	<ul style="list-style-type: none"> <li>Variable: (V.)G.HSC</li> </ul>
Variable to know the theoretical feedrate on 3D path.	<ul style="list-style-type: none"> <li>Variable: (V.)G.F3D</li> </ul>
Variable to know the number of the warning being displayed.	<ul style="list-style-type: none"> <li>Variable: (V.)G.CNCWARNING</li> </ul>
The variable (V.)G.CNCERR is now per channel.	<ul style="list-style-type: none"> <li>Variable: (V.)G.CNCERR</li> </ul>
Select the type of loop, open or closed, for the spindle.	<ul style="list-style-type: none"> <li>Instruction: #SERVO.</li> </ul>
Spindle synchronization.	<ul style="list-style-type: none"> <li>Instruction: #SYNC.</li> </ul>
Spindle synchronization.	<ul style="list-style-type: none"> <li>Instruction: #TSYNC.</li> </ul>
Spindle synchronization.	<ul style="list-style-type: none"> <li>Instruction: #UNSYNC.</li> </ul>
Select milling cycles at a lathe model.	<ul style="list-style-type: none"> <li>Instruction: #MILLCY.</li> </ul>
Select turning cycles at a milling model.	<ul style="list-style-type: none"> <li>Instruction: #LATHECY.</li> </ul>
Define a kinematics when activating the C axis.	<ul style="list-style-type: none"> <li>#CYL instruction.</li> </ul>
Define a kinematics when activating the C axis.	<ul style="list-style-type: none"> <li>#FACE instruction.</li> </ul>
Improved coordinate transformation (#CS#ACS). <ul style="list-style-type: none"> <li>Keep the part zero when deactivating the transformation.</li> <li>Working with 45° spindles. Select between the two choices.</li> <li>Keep the rotation of the plane axes with MODE 6.</li> </ul>	<ul style="list-style-type: none"> <li>Instructions #CS #ACS.</li> </ul>
G33. New parameter (Q1) to define the entry angle.	<ul style="list-style-type: none"> <li>Function G33.</li> </ul>
G63. Tool inspection is possible during rigid tapping.	<ul style="list-style-type: none"> <li>Function G63.</li> </ul>
Function G112 is not valid for the spindle.	<ul style="list-style-type: none"> <li>Function G112.</li> </ul>
New criteria when assuming a new master spindle in the channel.	

**Ref. 0601**

Software V03.01	
Axis slaving. Configuring the default status of an axis slaving (coupling).	<ul style="list-style-type: none"> <li>Machine parameters: LINKCANCEL.</li> </ul>
Tool radius compensation. The way tool radius is canceled.	<ul style="list-style-type: none"> <li>Machine parameters: COMPCANCEL.</li> </ul>
Using the ":" character to program a comment in a part-program.	
Variables. Geometry of the lathe tools.	
Variables. Number of the tool in the claws of the changer arm.	<ul style="list-style-type: none"> <li>Variables: (V.)TM.TOOLCH1[mz] (V.)TM.TOOLCH2[mz]</li> </ul>

(REF: 2008)



CNC 8070



Software V03.01	
The instruction #EXEC does not issue an error if the channel is busy; the instruction waits for the operation in progress to end.	• #EXEC instruction.
The instruction #EXBLK does not issue an error if the channel is busy; the instruction waits for the operation in progress to end.	• #EXBLK instruction.

**Ref. 0606**

Software V03.10	
Feedrate. Maximum machining feedrate.	• Machine parameter: MAXFEED.
Feedrate. Default machining feedrate when none has been programmed.	• Machine parameter: DEFAULTFEED.
The CNC allows changing the spindle override during electronic threading (G33) and in the threading canned cycles of the T model (G86, G87 and their equivalent of the cycle editor).	• Machine parameters: THREADOVR, OVRFILTER.
"Retrace" function.	
Tangential control.	
The CNC checks whether the programmed turning direction (M3/M4) matches the one preset in the tool table.	
M02/M30. There is no need to program M02 or M30 to end a part program.	• Functions M02/M30.
Canceling the preset turning direction of a tool.	• Variables: (V.)G.SPDLTURDIR
Change the maximum feedrate allowed in the channel from the PLC.	• Variables: (V.)PLC.PLCG00FEED
Show the status of the emergency relay.	• Variables: (V.)G.ERELAYST
HSC. New FAST mode.	• #HSC instruction.
C axis. The #CYL instruction requires programming the radius.	• #CYL instruction.

**Ref. 0608**

Software V03.11	
"Retrace" function. Several improvements to the retrace function.	
HSC. New command CORNER.	• #HSC instruction.
G33. The override limitation is maintained while returning to the beginning of the thread.	• Function G33.
RTCP. Home search is now possible on the axes that are not involved in RTCP.	
Abort the execution of the program and resume it somewhere else.	• Instruction: #ABORT.

**Ref. 0704 / Ref. 0706**

Software V03.13	
Define the tool wear with incremental or absolute values.	
Variables (V.)TM.TOOLCH1[mz] / (V.)TM.TOOLCH2[mz] may be written from the PLC.	• Variables: (V.)TM.TOOLCH1[mz] (V.)TM.TOOLCH2[mz].

Software V03.14	
MCU and ICU central unit. battery powered RAM. Connecting handwheels to the central unit. local I/O. Local feedback inputs. Loca probes.	
The turning speed limitation (G192) is also applied when the spindle is working at constant turning speed (G97)	• Function G192.

**Ref. 0707**

Software V03.15	
Know the type of hardware.	• Variable: (V.)G.HARDTYPE
Theoretical tool feedrate along the path	• Variable: (V.)G.PATHFEED
Zero offsets for the C axis.	
The CNC shows a warning when a channel is expecting a tool that is being used in another channel.	

**Ref. 0709**

Software V03.16	
Tandem spindles.	
The CNC does not assume any kinematics on power-up.	
The CNC allows modifying the override while threading if it detects that the feed forward (parameter FFWTYPE) is not active in a gear or if the active feed forward is lower than 90%	

**Ref. 0712**

Software V03.17	
C axis maintained after executing M02, M30 or after an emergency or reset.	• Machine parameter: PERCAX.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Ref. 0801****Software V03.20**

Set change. The CNC lets change the gear of the slave axis or spindle of a tandem.	
Coordinate latching with the help of a probe or a digital input.	• Variables: (V.)A.LATCH1.xn (V.)A.LATCH2.xn
Status of the local probes.	• Variables: (V.)G.PRBST1 (V.)G.PRBST2.
Axis synchronization. Managing a rotary axis as an infinite axis making it possible to increase the feedback count of the axis indefinitely (without limits) regardless of the value of the module.	• Variables: (V.)A.ACCUDIST.xn
Show a warning and interrupt program execution.	• Instruction: #WARNINGSTOP.
Electronic cam programming (theoretical coordinates).	• Instruction: #TCAM.
Dynamic distribution of the machining operations between channels.	• Instruction: #DINDIST.
The CNC can park the main axes.	
The axes may be programmed using the "?" wild card that refers to the axis position in the channel.	• Wild card "?".
Functions G130 (percentage of acceleration) and G132 (percentage of jerk) may be applied to the spindles.	• Functions G130 and G132.
Interface related variables.	

**Ref. 0809****Software V04.00 (it does not include the features of version V03.21)**

Unicode.	
Cancel spindle synchronization after executing M02, M30 or after an error or a reset.	• Instructions #SYNC and #TSYNC.
Positioning a turret magazine whether there is a tool in the indicated position or not.	• #ROTATEMZ instructions.
A channel maintains its master spindle after executing M02, M30 or after an emergency or a reset or restarting the CNC.	• #MASTERinstruction.
Force the change of gears and/or of the parameter set of a Sercos drive	• Variable: (V.)A.SETGE.xn
Set a machine coordinate.	• Function G174.
There can now be up to 99 zero offsets.	• Function G159.
There can now be up to 100 synchronization marks.	• Instructions #MEET, #WAIT and #SIGNAL.
Select a turret position.	• #ROTATEMZ instructions.
Axis synchronization. Managing a rotary axis as an infinite axis making it possible to increase the feedback count of the axis indefinitely (without limits) regardless of the value of the module.	• Variables: (V.)A.PREVACCUDIST.xn
Variables. The variable (V.)E.PROGSELECT can be written via part-program, PLC and interface. This variable can only be written with the value of '0'	• Variables: (V.)E.PROGSELECT
Variables. The following variables are valid for the spindle.	• Variables: (V.)A.MEAS.sn (V.)A.ATIPMEAS.sn (V.)A.MEASOF.sn (V.)A.MEASOK.sn (V.)A.MEASIN.sn
Handwheels. Number of pulses sent by the handwheel since the system was started up.	• Variables: (V.)G.HANDP[hw]
Feed handwheel.	

**Ref. 0811****Software V03.21 (features not included in version V04.00)**

There can now be up to 1024 PLC messages.	• PLC resources: MSG.
There can now be up to 1024 PLC errors.	• PLC resources: ERR.

**Ref. 0907****Software V04.01**

Define the maximum acceleration and jerk allowed on the tool path.	• Variables: (V.)G.MAXACCEL (V.)G.MAXJERK
Variable to know the following error (lag) when feedback combination is active.	• Variables: (V.)A.FLWE.xn (V.)A.FLWACT.xn
Variable to know the position value of the first feedback when feedback combination is active.	• Variable: (V.)A.POSMOTOR.xn

**Ref. 1007****Software V04.10 (it does not include the features of version V04.02)**

New languages (Russian and Czech).	• Machine parameter: LANGUAGE.
Cancel the inclined plane on start-up.	• Machine parameter: CSCANCEL.
M functions with an associated subroutine.	
The CNC admits function G174 for axes in DRO mode and spindles.	• Function G174.
Detailed CNC status in jog mode.	• Variable: (V.)G.CNCMANSTATUS
Detailed CNC status in automatic mode.	• Variable: (V.)G.CNCAUTSTATUS
Know the axes selected for home search, repositioning, coordinate preset or movement to a coordinate.	• Variable: (V.)G.SELECTEDAXIS

Software V04.10 (it does not include the features of version V04.02)	
Know the current position of the main rotary axes of the kinematics (third axis).	• Variable: (V.)G.POSROTT
Know the target position of the main rotary axes of the kinematics (third axis).	• Variable: (V.)G.TOOLORIT1 (V.)G.TOOLORIT2
Cancel the name change for axes and spindles (#RENAME) after executing M02 or M30, after a reset or at the beginning of a new part-program in the same channel.	• #RENAME instruction.

**Ref. 1010**

Software V04.02 (features not included in version V04.10)	
New language (Russian).	• Machine parameter: LANGUAGE.
The CNC admits function G174 for axes in DRO mode and spindles.	• Function G174.
Detailed CNC status in jog mode.	• Variable: (V.)G.CNCMANSTATUS
Detailed CNC status in automatic mode.	• Variable: (V.)G.CNCAUTSTATUS
Know the axes selected for home search, repositioning, coordinate preset or movement to a coordinate.	• Variable: (V.)G.SELECTEDAXIS
Know the current position of the main rotary axes of the kinematics (third axis).	• Variable: (V.)G.POSROTT
Know the target position of the main rotary axes of the kinematics (third axis).	• Variable: (V.)G.TOOLORIT1 (V.)G.TOOLORIT2
Know the status of a cam.	• Variable: (V.)G.CAMST[cam]
Modify the range of the slave axis when activating the cam.	• Variable: (V.)G.CAM[cam][index]
Set 0% feedrate override via PLC.	• Variable: (V.)PLC.FRO
Cancel the name change for axes and spindles (#RENAME) after executing M02 or M30, after a reset or at the beginning of a new part-program in the same channel.	• #RENAME instruction.

**Ref. 1107**

Software V04.11	
Synchronized switching.	• Variables: (V.)G.TON (V.)G.TOF (V.)G.PON (V.)G.POF • Statement: #SWTOUT

**Ref. 1304**

Software V04.20	
Maximum safety limit for feedrate.	• Machine parameter: FLIMIT.
Maximum safety speed limit.	• Machine parameter: SLIMIT.
Interruption subroutines per channel.	• Programming instructions: #REPOS.
There may be up to 30 OEM subroutines per channel now (G180-G189/G380-G399).	
The OEM subroutines may be executed either in a non-modal (G180, G181, etc) or in a modal way (MG180, MG181, etc).	
The operation of M19 with subroutine has changed.	• Function: M19.
Know the status of a cam.	• Variable: (V.)G.CAMST[cam]
Modify the range of the slave axis when activating the cam.	• Variable: (V.)G.CAM[cam][index]
Set 0% feedrate override via PLC.	• Variable: (V.)PLC.FRO
Detailed CNC status in automatic mode. New values.	• Variable: (V.)G.CNCAUTSTATUS
Active zero offset.	• Variable: (V.)G.EXTORG
The CNC can execute programs of the 8055 MC and 8055 TC models made up with conversational canned cycles including geometric assistance.	

Software V04.21	
New model LCD-10K.	• Variables: (V.)MPMAN.JOGKEYDEF[jk] (V.)MPMAN.USERKEYDEF[uk]

Software V04.22	
Set the zero offsets with a coarse part and a fine part.	• Variables: (V.)A.ADDORG.xn (V.)A.COARSEORG.xn (V.)A.FINEORG.xn (V.)A.COARSEORG[nb].xn (V.)A.FINEORG[nb].xn
Cancel mirror image (G11/G12/G13/G14) after M30 and reset.	

Software V04.24	
Additional negative command pulse for analog axes.	• Variable: (V.)MPA.BAKANOUT[set].xn
The SPDLEREV mark (reverse turning direction) affects the spindle in M19.	• Variable: (V.)MPA.M19SPDLEREV.xn
Functions M02, M30 and reset do not cancel the speed limit function G192.	• Function G192.
Functions M02, M30 and reset do not cancel the constant surface speed (CSS) function.	• Function G96.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Software V04.25	
Synchronized switching.	<ul style="list-style-type: none"> <li>Variables: (V.)G.TON (V.)G.TOF (V.)G.PON (V.)G.POF</li> <li>Statement: #SWTOUT</li> </ul>
Error programmed in HSC mode.	<ul style="list-style-type: none"> <li>Variable: (V.)G.CONTERROR</li> </ul>
The HSC FAST mode may be used to adjust the chordal error (parameter E).	<ul style="list-style-type: none"> <li>Statement: #HSC</li> </ul>
The CNC will load into RAM memory the subroutines having the extension .fst.	
If function G95 is active and the spindle does not have an encoder, the CNC will use the programmed theoretical rpm to calculate the feedrate.	<ul style="list-style-type: none"> <li>Function G95.</li> </ul>

**Ref. 1305**

Software V04.26	
New model LCD-10K.	<ul style="list-style-type: none"> <li>Variables: (V.)MPMAN.JOGKEYDEF[jk] (V.)MPMAN.USERKEYDEF[uk]</li> </ul>
New LCD-15 model.	
New keyboard VERTICAL-KEYB.	
New keyboard HORIZONTAL-KEYB.	
New operator panel OP-PANEL.	
Keep the longitudinal axis when changing planes (G17/G18/G19).	<ul style="list-style-type: none"> <li>Function G17/G18/G19.</li> </ul>
The M3/M4/M5 functions cancel the C axis and set the spindle in open loop.	
Programs with ".mod" extension may be modified when they are interrupted using "cancel and resume".	

**Ref. 1309**

Software V04.27	
Virtual tool axis.	<ul style="list-style-type: none"> <li>Statement: #VIRTAX</li> <li>Variable: (V.)G.VIRTAXIS (V.)G.VIRTAXST (V.)A.VIRTAXOF.xn</li> </ul>
PWM (Pulse-Width Modulation)	<ul style="list-style-type: none"> <li>Statement: #PWMMOUT</li> <li>Variable: (V.)G.PWMMON (V.)G.PWMFREQ (V.)G.PWMDUTY (V.)PLC.PWMFREQ (V.)PLC.PWMDUTY</li> </ul>
Modify the simulation speed via PLC.	<ul style="list-style-type: none"> <li>Variable: (V.)PLC.SIMUSPEED</li> </ul>
Execute subroutine as a single block.	<ul style="list-style-type: none"> <li>Statement: #RETDSBLK</li> </ul>

**Ref. 1405**

Software V04.27.10	
HSC. New SURFACE mode.	<ul style="list-style-type: none"> <li>#HSC instruction.</li> </ul>
Generic user subroutines.	<ul style="list-style-type: none"> <li>Functions G500-G599.</li> </ul>
Generic user subroutines pre-configured by Fagor.	<ul style="list-style-type: none"> <li>G500-G501 functions.</li> </ul>
"program-start" subroutine.	
Override of the dynamics for HSC.	<ul style="list-style-type: none"> <li>Variable: (V.)G.DYNOVR</li> </ul>
New name for the (V.)G.CONTERROR variable	<ul style="list-style-type: none"> <li>Variable: (V.)G.ACTROUND</li> </ul>
Maximum frequency generated on the machining path.	<ul style="list-style-type: none"> <li>Variable: (V.)MPG.MAXFREQ</li> </ul>

Software V05.01	
ModBUS server.	<ul style="list-style-type: none"> <li>Variables: (V.)MPG.MODBUSVRTCP (V.)MPG.MODBUSVRRS (V.)MPG.MODSVRID (V.)MPG.MODBRATE</li> </ul>
CANopen bus communication frequency.	<ul style="list-style-type: none"> <li>Variable: (V.)MPG.CANOPENFREQ</li> </ul>
Feedback type associated with the handwheel input.	<ul style="list-style-type: none"> <li>Variable: (V.)MPMAN.HWFBTYPE[hw]</li> </ul>
Detailed CNC status in jog mode. New values.	<ul style="list-style-type: none"> <li>Variable: (V.)G.CNCMANSTATUS</li> </ul>
Activate the Mechatrolink drive options.	<ul style="list-style-type: none"> <li>Variable: (V.)MPA.OPTION.xn</li> </ul>
Enable the hardware alarm (alarm pin) of the local feedback.	<ul style="list-style-type: none"> <li>Variable: (V.)MPA.HWFBACKAL[set].xn</li> </ul>
Maximum position difference allowed to consider that there is no need to home again.	<ul style="list-style-type: none"> <li>Variable: (V.)MPA.MAXDIFREF[set].xn</li> </ul>

**Ref. 1408**

Software V05.10	
Orient the tool in the part coordinate system.	<ul style="list-style-type: none"> <li>Instructions #CSROT, #DEFROT.</li> </ul>
Select onto which rotary axes of the kinematics the tool orientation is calculated for a given direction on the work piece (part).	<ul style="list-style-type: none"> <li>Instruction #SELECT ORI.</li> <li>Variable: (V.)G.SELECTORI</li> </ul>
Transform the current part zero considering the position of the table kinematics.	<ul style="list-style-type: none"> <li>Instruction #KINORG.</li> </ul>
Type of the active kinematics.	<ul style="list-style-type: none"> <li>Variable: (V.)G.KINTYPE</li> </ul>
Number of axes of the active kinematics.	<ul style="list-style-type: none"> <li>Variable: (V.)G.NKINAX</li> </ul>
Current position of the fourth rotary axis of the kinematics.	<ul style="list-style-type: none"> <li>Variable: (V.)G.POSROTO</li> </ul>

(REF: 2008)



Software V05.10	
Position to be occupied by the fourth rotary axis of the kinematics in order to position the tool perpendicular to the inclined plane (solution 1 and 2).	• Variable: (V.)G.TOOLORIO1 (V.)G.TOOLORIO2
Status of the #CSROT function.	• Variable: (V.)G.CSROTST
Position (machine coordinates) calculated for the rotary axis of the kinematics at the beginning of the block, for solution 1 of the #CSROT mode.	• Variables: (V.)G.CSROTF1[1] (V.)G.CSROTS1[1] (V.)G.CSROTT1[1] (V.)G.CSROTO1[1]
Position (machine coordinates) calculated for the rotary axis of the kinematics at the end of the block, for solution 1 of the #CSROT mode.	• Variables: (V.)G.CSROTF1[2] (V.)G.CSROTS1[2] (V.)G.CSROTT1[2] (V.)G.CSROTO1[2]
Position (machine coordinates) calculated for the rotary axis of the kinematics at the beginning of the block, for solution 1 of the #CSROT mode.	• Variables: (V.)G.CSROTF2[1] (V.)G.CSROTS2[1] (V.)G.CSROTT2[1] (V.)G.CSROTO2[1]
Position (machine coordinates) calculated for the rotary axis of the kinematics at the end of the block, for solution 1 of the #CSROT mode.	• Variables: (V.)G.CSROTF2[2] (V.)G.CSROTS2[2] (V.)G.CSROTT2[2] (V.)G.CSROTO2[2]
Position (machine coordinates) to be occupied by the rotary axis of the kinematics at the beginning of the block, for the #CSROT mode.	• Variables: (V.)G.CSROTF[1] (V.)G.CSROTS[1] (V.)G.CSROTT[1] (V.)G.CSROTO[1]
Position (machine coordinates) to be occupied by the rotary axis of the kinematics at the end of the block, for the #CSROT mode.	• Variables: (V.)G.CSROTF[2] (V.)G.CSROTS[2] (V.)G.CSROTT[2] (V.)G.CSROTO[2]
Position of the part zero transformed by the instruction #KINORG, considering the table position, on the first three axes of the channel.	• Variable: (V.)G.KINORG1 (V.)G.KINORG2 (V.)G.KINORG3
Allow the user modify the kinematics parameters.	• Variable: (V.)MPK.TDATAFkin[nb] (V.)G.OFTDATAkin[nb] (V.)G.OFTDATAFkin[nb] (V.)G.OFTDATA_ikin[nb] (V.)MPK.MAXOFTDATAkin[nb] (V.)MPK.MAXOFTDATAFkin[nb] (V.)MPK.MAXOFTDATA_ikin[nb]

**Ref. 1501**

Software V05.20	
New options in graphics. <ul style="list-style-type: none"> <li>Define whether the part is cylindrical or rectangular.</li> <li>Define up to four parts.</li> <li>Assign a part to one or more channels.</li> </ul>	• Statement: #DGWZ.
On line modification of the machine configuration in HD graphics (xca files). 3D tool compensation.	• Statement: #DEFGRAPH. • Statement: #COMP3D.
HSC. SURFACE mode. New commands RE, SF and AXF.	• #HSC instruction.
HSC. FAST mode. New commands RE, SF and AXF.	• #HSC instruction.
HSC. CONTERROR mode. New commands RE and AXF.	• #HSC instruction.

**Ref. 1505**

Software V05.31	
Coordinate programming. Angle and Cartesian coordinate.	
Electronic threading with variable pitch.	• Function G34.
Withdraw the axes after interrupting an electronic threading.	• Function G233. • Variable: (V.)G.RETRETJ
Assume IPLANE as active plane after M30/RESET or keep the active one.	
Detailed CNC status in automatic mode. New value \$100000.	• Variable: (V.)G.CNCAUTSTATUS
Volts of output [n] of the RCS-S module.	• Variable: (V.)G.ANASO[n]
HSC. SURFACE mode. New command OS.	• Statement: #HSC.
HSC. If the RE command is not programmed, the amount of error allowed on rotary axes will be the maximum between parameter MAXERROR and the E command.	• Statement: #HSC.
If no resume point has been defined, the execution continues in the #ABORT OFF instruction; if the instruction has not been defined, the program will jump to the end of the program (M30).	• Statement: #ABORT
ISO generation.	• Statement: #ISO
System spindles involved in the subroutine associated with M3, M4, M5, M19 and M41-M44.	• Variable: (V.)G.SUBMSPDL
Active canned cycle.	• Variable: (V.)G.ACTIVECYCLE
Status of probe -1.	• Variable: (V.)G.PRBST
Probing movement. Value measured at the master spindle of the channel.	• Variable: (V.)G.PLMEAS4
End of axis and spindle repositioning at the starting point.	• Variable: (V.)G.ENDREPINI
End of axis and spindle repositioning at the interruption point.	• Variable: (V.)G.ENDREPINT
Time remaining to activate the laser output.	• Variable: (V.)G.LASEROTMON
Time remaining to deactivate the laser output.	• Variable: (V.)G.LASEROTMOFF
Amount of time that PWM stays active in burst mode.	• Variable: (V.)G.PWMBTIME
Final PWM status once burst mode is over.	• Variable: (V.)G.PWMBEND



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Software V05.31	
Percentage of loop time (cycle time) used by the PLC.	• Variable: (V.)G.PLCTIMERATE
Percentage of loop time (cycle time) used by the dynamic preparation of the tool path.	• Variable: (V.)G.TRAYTIMERATE
Value of the local count-up 1 input.	• Variable: (V.)G.LCOUNTER1
Value of the local count-up 2 input.	• Variable: (V.)G.LCOUNTER2
Actual (real) CNC feedrate in G95.	• Variable: (V.)G.FREALPR
Real feedrate on the tool path.	• Variable: (V.)G.ACTFEED
Feedrate active in the block.	• Variable: (V.)G.IPOFEED
Active tool. Code of the tool offset type.	• Variable: (V.)TM.TOOLTYP[ofid]
Tool being prepared. Code of the tool offset type.	• Variable: (V.)G.TOOLTYP
Tool being prepared. Tool-holder orientation.	• Variable: (V.)G.FIXORI
Solution 2 is selected in instruction #CS or #ACS.	• Variable: (V.)G.TORISOL2
CNC model.	• Variable: (V.)G.CNCMODEL
CNC sub-version number (decimal value).	• Variable: (V.)G.SUBVERSIO
Number of the line of the program where the cursor is.	• Variable: (V.)G.CURSORLINE
Orientation smoothing of the rotary axes working with RTCP.	• Variable: (V.)MPG.ORISMOOTH
Amount of error allowed on the axis for the HSC mode.	• Variable: (V.)A.ACTROUND.xn

**Ref. 1512**

Software V05.40	
Work zones.	<ul style="list-style-type: none"> <li>• Function: G120, G121, G122, G123.</li> <li>• Variables: (V.)MPA.ZONELIMITTOL.xn (V.)G.ZONEST[k] (V.)G.ZONETOOLWATCH[k] (V.)G.ZONEWARN[k] (V.)A.ZONELIMITTOL.xn (V.)A.ZONELOWLIM[k].xn (V.)A.ZONEUPLIM[k].xn (V.)G.ZONECIR1[k] (V.)G.ZONECIR2[k] (V.)G.ZONER[k] (V.)G.ZONECIRAX1[k] (V.)G.ZONECIRAX2[k]</li> </ul>
Smooth the path.	• Statement: #PATHND
Smooth the path and the feedrate.	• Statement: #FEEDND

**Ref. 1604**

Software V05.50	
The CNC permits setting the machine coordinate for gantry axes.	• Function: G174.
The CNC permits executing seven subroutines per block.	

**Ref. 1709**

Software V05.60.00	
Subroutine associated with the reset.	• Subroutine: PROGRAM_RESET
Subroutine associated with the tool calibration cycle.	• Subroutine: KinCal_Begin.nc KinCal_End.nc
New search criteria for help files associated with the subroutines.	<ul style="list-style-type: none"> <li>• File: pcall.txt</li> <li>• File: subroutine_name.txt subroutine_name.bmp</li> </ul>
The subroutines associated with functions G500-599 may also have help files which are displayed during editing.	• File: G500.txt, G501.txt, etc. G500.bmp, G501.bmp, etc.
The subroutines associated with functions G8000-8999 may also have help files which are displayed during editing.	• File: G8000.txt, G8001.txt, etc. G8000.bmp, G8001.bmp, etc.
New radius compensation algorithm, optimized to solve stepped profiles.	
User subroutines (G500-G599) and modal subroutines.	<ul style="list-style-type: none"> <li>• Functions: G500, G501, etc.</li> <li>• Instructions: #MCALL</li> </ul>
User subroutines (G8000-G8999) and modal subroutines.	<ul style="list-style-type: none"> <li>• Functions: G8000, G8001, etc.</li> <li>• Instructions: #MCALL</li> </ul>
Calls to subroutines with parameter initialization allow the programming of 32 additional parameters (P26 to P57), which can also be defined as "D0=" to "D31=", so that "D0=" is equal to P26, "D1=" to P27 and so forth.	<ul style="list-style-type: none"> <li>• Functions: G500, G501, etc. G180, G181, etc. G380, G381, etc.</li> <li>• Instructions: #PCALL, #MCALL</li> </ul>
Non-modal G02 and G03 functions.	• Functions: G2, G3
Non-modal function G00.	• Functions: G0
Being the CNC in SETUP mode, the OEM subroutines with an fst extension will be loaded into RAM memory during the preparation of the blocks.	



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Ref. 1807**

Software V01.70	
DMC function.	• Instructions: #DMC ON, #DMC OFF
Opening and writing files. <ul style="list-style-type: none"> <li>• Open file for writing.</li> <li>• Writing in a file.</li> <li>• Close a file.</li> <li>• The OEM or the user can define a text group for the cncWrite.txt file.</li> </ul>	• Instructions: #OPEN, #WRITE, #CLOSE
CNC messages. <ul style="list-style-type: none"> <li>• Format identifiers (%D, %i, %u, etc).</li> <li>• Define messages using a number.</li> <li>• The OEM or the user can define a text group for the cncMsg.txt file.</li> </ul>	• Statement: #MSG
Errors and warnings. <ul style="list-style-type: none"> <li>• Format identifiers (%D, %i, %u, etc).</li> <li>• Change location of cncError.txt file.</li> <li>• Programming a warning with value 0 clears all displayed warnings</li> </ul>	• Instructions: #ERROR #WARNING #WARNINGSTOP
Work zones. The command "E" must always be placed after the function G122; otherwise the CNC will interpret it as the name of an axis.	• Function G122.

**Ref. 1901**

Software V06.00	
New command to select the file type (TYPE command).	• Statement: #OPEN

**Ref. 2008**

Software V06.01.06	
Scaling factor per axis.	• Function: G72.



BLANK PAGE



# SAFETY CONDITIONS

Read the following safety measures in order to prevent harming people or damage to this product and those products connected to it. Fagor Automation shall not be held responsible of any physical or material damage originated from not complying with these basic safety rules.



*Before start-up, verify that the machine that integrates this CNC meets the 2006/42/EC Directive.*

## PRECAUTIONS BEFORE CLEANING THE UNIT

- Do not get into the inside of the unit.** Only personnel authorized by Fagor Automation may access the interior of this unit.
- Do not handle the connectors with the unit connected to AC power.** Before handling these connectors (I/O, feedback, etc.), make sure that the unit is not powered.

## PRECAUTIONS DURING REPAIRS

In case of a malfunction or failure, disconnect it and call the technical service.

- Do not get into the inside of the unit.** Only personnel authorized by Fagor Automation may access the interior of this unit.
- Do not handle the connectors with the unit connected to AC power.** Before handling these connectors (I/O, feedback, etc.), make sure that the unit is not powered.

## PRECAUTIONS AGAINST PERSONAL HARM

- Interconnection of modules.** Use the connection cables provided with the unit.
- Use proper cables.** To prevent risks, only use cables and Sercos fiber recommended for this unit.  
To prevent a risk of electrical shock at the central unit, use the proper connector (supplied by Fagor); use a three-prong power cable (one of them being ground).
- Avoid electric shocks.** To prevent electrical shock and fire risk, do not apply electrical voltage out of the indicated range.
- Ground connection.** In order to avoid electrical discharges, connect the ground terminals of all the modules to the main ground terminal. Also, before connecting the inputs and outputs of this product, make sure that the ground connection has been done.  
In order to avoid electrical shock, before turning the unit on verify that the ground connection is properly made.
- Do not work in humid environments.** In order to avoid electrical discharges, always work with a relative humidity (non-condensing).
- Do not work in explosive environments.** In order to avoid risks, harm or damages, do not work in explosive environments.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## PRECAUTIONS AGAINST DAMAGE TO THE PRODUCT

<b>Work environment.</b>	This unit is ready to be used in industrial environments complying with the directives and regulations effective in the European Community. Fagor Automation shall not be held responsible for any damage suffered or caused by the CNC when installed in other environments (residential, homes, etc.).
<b>Install this unit in the proper place.</b>	It is recommended, whenever possible, to install the CNC away from coolants, chemical product, blows, etc. that could damage it. This unit meets the European directives on electromagnetic compatibility. Nevertheless, it is recommended to keep it away from sources of electromagnetic disturbance such as: <ul style="list-style-type: none"> <li>Powerful loads connected to the same mains as the unit.</li> <li>Nearby portable transmitters (radio-telephones, Ham radio transmitters).</li> <li>Nearby radio / TC transmitters.</li> <li>Nearby arc welding machines.</li> <li>Nearby high voltage lines.</li> </ul>
<b>Enclosures.</b>	It is up to the manufacturer to guarantee that the enclosure where the unit has been installed meets all the relevant directives of the European Union.
<b>Avoid disturbances coming from the machine.</b>	The machine must have all the interference generating elements (relay coils, contactors, motors, etc.) uncoupled.
<b>Use the proper power supply.</b>	Use an external regulated 24 Vdc power supply for the keyboard, operator panel and the remote modules.
<b>Connecting the power supply to ground.</b>	The zero Volt point of the external power supply must be connected to the main ground point of the machine.
<b>Analog inputs and outputs connection.</b>	Use shielded cables connecting all their meshes to the corresponding pin.
<b>Ambient conditions.</b>	Maintain the CNC within the recommended temperature range, both when running and not running. See the corresponding chapter in the hardware manual.
<b>Central unit enclosure.</b>	To maintain the right ambient conditions in the enclosure of the central unit, it must meet the requirements indicated by Fagor. See the corresponding chapter in the hardware manual.
<b>Power switch.</b>	This switch must be easy to access and at a distance between 0.7 and 1.7 m (2.3 and 5.6 ft) off the floor.

## SAFETY SYMBOLS

### Symbols that may appear in the manual.



*Danger or prohibition symbol.*  
This symbol indicates actions or operations that may hurt people or damage products.



*Warning or caution symbol.*  
This symbol indicates situations that certain operations could cause and the suggested actions to prevent them.



*Obligation symbol.*  
This symbol indicates actions and operations that must be carried out.



*Information symbol.*  
This symbol indicates notes, warnings and advises.



*Symbol for additional documentation.*  
This symbol indicates that there is another document with more detailed and specific information.

**Symbols that the product may carry.**



*Ground symbol.*

*This symbol indicates that that point must be under voltage.*



*ESD components.*

*This symbol identifies the cards as ESD components (sensitive to electrostatic discharges).*



BLANK PAGE

## RETURNING CONDITIONS

Pack it in its original package along with its original packaging material. If you do not have the original packaging material, pack it as follows:

- 1 Get a cardboard box whose 3 inside dimensions are at least 15 cm (6 inches) larger than those of the unit itself. The cardboard being used to make the box must have a resistance of 170 Kg (375 lb.).
- 2 Attach a label to the device indicating the owner of the device along with contact information (address, telephone number, email, name of the person to contact, type of device, serial number, etc.). In case of malfunction also indicate symptom and a brief description of the problem.
- 3 Protect the unit wrapping it up with a roll of polyethylene or with similar material. When sending a central unit with monitor, protect especially the screen.
- 4 Pad the unit inside the cardboard box with polyurethane foam on all sides.
- 5 Seal the cardboard box with packaging tape or with industrial staples.

BLANK PAGE

# CNC MAINTENANCE

## CLEANING

The accumulated dirt inside the unit may act as a screen preventing the proper dissipation of the heat generated by the internal circuitry which could result in a harmful overheating of the unit and, consequently, possible malfunctions. Accumulated dirt can sometimes act as an electrical conductor and short-circuit the internal circuitry, especially under high humidity conditions.

To clean the operator panel and the monitor, a smooth cloth should be used which has been dipped into de-ionized water and /or non abrasive dish-washer soap (liquid, never powder) or 75° alcohol. Never use air compressed at high pressure to clean the unit because it could cause the accumulation of electrostatic charges that could result in electrostatic shocks.

The plastics used on the front panel are resistant to grease and mineral oils, bases and bleach, dissolved detergents and alcohol. Avoid the action of solvents such as chlorine hydrocarbons, benzole, esters and ether which can damage the plastics used to make the unit's front panel.

## PRECAUTIONS BEFORE CLEANING THE UNIT

Fagor Automation shall not be held responsible for any material or physical damage derived from the violation of these basic safety requirements.

- Do not handle the connectors with the unit supplied with power. Before handling these connectors (I/O, feedback, etc.), make sure that the unit is not powered.
- Do not get into the inside of the unit. Only personnel authorized by Fagor Automation may access the interior of this unit.

BLANK PAGE



# CREATING A PROGRAM.

# 1

## 1.1 Programming languages.

The CNC has its own programming language described in this manual. The program is edited block by block and each one may be written in ISO language or in High level language. See "[1.3 Program block structure.](#)" on page 37.

When editing high level commands, the editor offers a list of available commands.

### **8055 language.**

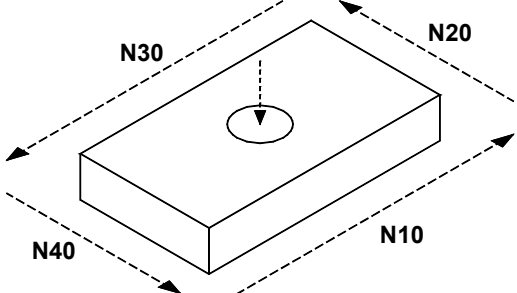
Programs can also be edited in the 8055 CNC language. Programming in 8055 CNC language is enabled from the part-program editor. Refer to the operating manual to enable this option.

This manual does not describe the 8055 language; refer to the specific documentation for this product. Obviously, since this CNC and the 8055 are two functionally different products, some concepts may be different.

## 1.2 Program structure.

A CNC program consists of a set of blocks or instructions that properly organized, in subroutines or in the program body, provide the CNC with the necessary data to machine the desired part.

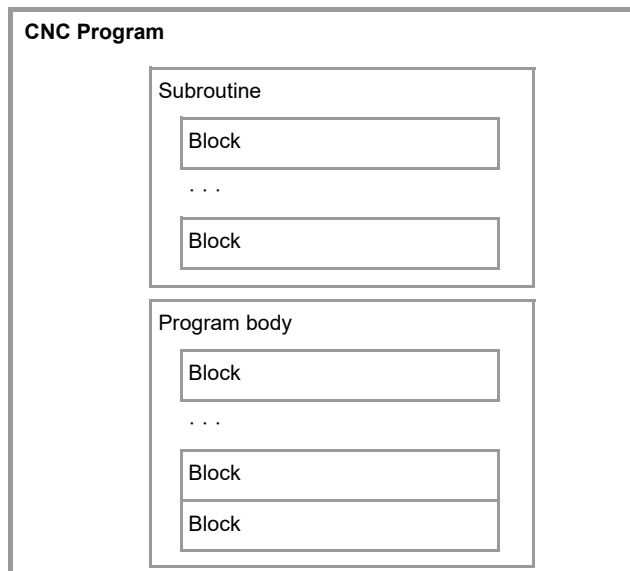
Each block contains all the functions or command necessary to execute an operation that may be machining, preparing the cutting conditions, controlling the elements of the machine, etc.



```

%example
(Name of the program)
N5 F550 S1000 M3 M8 T1 D1
(Sets the machining conditions)
N6 G0 X0 Y0
(Positioning)
N10 G1 G90 X100
N20 Y50
N30 X0
N40 Y0
(Machining)
N50 M30
(End of program)
    
```

The CNC program may consist of several local subroutines and the body of the program. The local subroutines must be defined at the beginning of the program.



1.

CREATING A PROGRAM.  
Program structure.

## 1.2.1 Program body.

The body of the program has the following structure.

Header	The header indicates the beginning of the body of the program. The header must be programmed when the program has local subroutines.
Program blocks	It is the main part of the program, the one containing movements, operations, etc.
End of program	

### Program header.

The header of the program is a block consisting of the "%" character followed by the name of the program. The name of the program may be up to 14 characters long and may consist of uppercase and lowercase characters as well as numbers (no blank spaces are allowed).

```
%0123
%PROGRAM
%PART923R
```

The header must be programmed when the program contains local subroutines; otherwise, programming the header is optional.

The name defined in the header has nothing to do with the name of the file. The two may be different.

### Program body.

The body of the program consists of blocks in charge of executing operations, movements, etc.

### End of the program.

The end of the program body is defined by functions "M02" or "M30" and they are equivalent. There is no need to program these functions; when reaching the end of the program without executing any of them, the CNC ends the execution and shows a warning indicating that they are missing.

```
M30
M02
```

The CNC behaves differently when reaching the end of the program depending on whether the M02 / M30 has been programmed or not

	With M02/M30	Without M02/M30
The CNC selects the first block of the program.	Yes	Yes
The CNC stops the spindle.	Yes	No
The CNC assumes the initial conditions.	Yes (*)	No
The CNC initializes the cutting conditions.	Yes	No

(\*) Stopping the spindle depends on the setting of machine parameter SPDLSTOP.

# 1.

**CREATING A PROGRAM.**  
Program structure.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 1.2.2 The subroutines.

A subroutine is a set of blocks that, once properly identified, may be called upon several times from another subroutine or from the program. Subroutines are normally used for defining a bunch of operations or movements that are repeated several times throughout the program. See chapter "15 Subroutines."

### Types of subroutines.

The CNC has two types of subroutines, namely local and global. There is also a third type available, OEM subroutines, that are a special case of a global subroutine defined by the OEM.

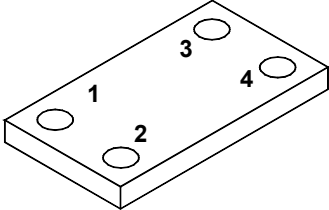
#### Global subroutines.

The global subroutine is stored in CNC memory as an independent program. This subroutine may be called upon from any program or subroutine being executed.

#### Local subroutines.

The local subroutine is defined as part of a program. This subroutine may only be called upon from the program where it has been defined.

A program can have several local subroutines; but they all must be defined before the body of the program. A local subroutine can call a second local subroutine with the condition that the calling subroutine be defined after the one being called.



```

%L POINTS
G01 X·· Y·· (Point 2)
G01 X·· Y·· (Point 3)
G01 X·· Y·· (Point 4)
M17

%PROGRAM
G81 X·· Y·· (Point 1. Center punching definition)
LL POINTS (call to a subroutine)
G81 X·· Y·· (Point 1. Center punching definition)
LL POINTS (call to a subroutine)
G84 X·· Y·· (Point 1. Center punching definition)
LL POINTS (call to a subroutine)
G80
M30
    
```

1.  
**CREATING A PROGRAM.**  
 Program structure.

## 1.3 Program block structure.

The blocks comprising the subroutines or the program body may be defined by commands in ISO code or in high-level language. Each block must be written in either language but not mixed; a program may combine blocks written in both languages. Empty blocks (empty lines) are also allowed.

In either language, it is also possible to use any type of arithmetic, relational or logic expression.

### Programming in ISO code.

It is especially designed to control the movement of the axes because it provides movement data and conditions as well as feedrate and speed. Some of the available commands are:

- Preparatory functions for movement establishing the geometry and work conditions such as linear and circular interpolations, threading, canned cycles, etc.
- Functions to control cutting conditions such as feedrate of the axes, spindle speed and accelerations.
- Functions to control the tools.
- Complementary functions, with technological instructions.
- Definition of position values.

### High-level language programming.

This language provides the user with a set of control commands with a terminology similar to the one used by other languages, such as \$IF, \$GOTO, #MSG, #HSC, etc. Some available commands are:

- Programming instructions.
- Flow controlling instructions to make loops and jumps within the program.
- To define and call upon subroutines with local parameters where a local variable is the one only known to the subroutine where it has been defined.

It is also possible to use any type of arithmetic, relational or logic expression.

### Arithmetic parameters, variables, constants and arithmetic expressions.

Constants, arithmetic parameters, variables and arithmetic expressions may be used from ISO blocks as well as from high level commands.

1.

**CREATING A PROGRAM.**  
Program block structure.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 1.3.1 Programming in ISO code.

ISO-coded functions consist of letters and numbers. The letters are "N", "G", "F", "S", "T", "D", "M", "H", "NR" plus those identifying the axes.

The numbers include digits "0" through "9", the "+" and "-" signs and the decimal point ".". Likewise, the numerical format may be replaced by a parameter, variable or arithmetic expression whose result is a number.

Programming allows blank spaces between letters, numbers and a sign as well as not using the sign with positive values.

#### Block structure.

A block may have the following functions, but needs not contain all of them. The data has no set order, it may be programmed anywhere in the block. The only exception being the block-skip condition and the block identification which must always be programmed at the beginning.

```
/ N— G— G— X..C— F— S— T— D— M— H— NR—
```

#### ·/· Block skip condition.

The block skip condition is governed by the BLKSKIP1 mark of the PLC. If the mark is active, the CNC will skip the blocks where it is programmed (not executing them) and will go on to the next block.

The control reads various blocks ahead of the one being executed in order to calculate the tool path in advance. The block skip condition will be analyzed at the time when the block is read, i.e. various blocks before execution. If the block skip needs to be analyzed at the time of execution, it is necessary to interrupt the block preparation, by programming the instruction #FLUSH in the previous block.

#### [LABEL] ·N· Block label.

The labels allow the blocks to be identified. The programming of labels assists in the monitoring of the program and allows block skips and repetitions to be programmed. In this last case, it is recommended to program the labels alone in the block. See "[1.8 Programming the block labels.](#)" on page 49.

#### ·G· Preparatory functions.

G functions set the geometry and work conditions such as linear and circular interpolations, chamfers, canned cycles, etc. See "[1.5 List of "G" functions.](#)" on page 42.

#### ·X..C· Coordinates of the point.

These functions set the movement of the axes. See "[1.4 Programming of the axes.](#)" on page 41.

Depending on the units, the programming format will be:

- In millimeters, format  $\pm 5.4$  (5 integers and 4 decimals).
- In inches, format  $\pm 4.5$  (4 integers and 5 decimals).

#### ·F· Axis feedrate.

The feedrate is represented by the letter "F" followed by the desired feedrate value.

#### ·S· Spindle speed.

This function sets the spindle speed.

The spindle name is defined by 1 or 2 characters. The first character is the letter S and the second character is optional, it must be a numerical suffix between 1 and 9. This way, the name of the spindles may be within the range S, S1 ... S9.

1.

CREATING A PROGRAM.  
Program block structure.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

The feedrate is represented by the axis letter followed by the target position for the axis. For spindles like S1, S2, etc. the "=" sign must be included between the axis name and the speed.

```
S1000  
S1=334
```

**·T· Tool number.**

This function selects the tool to be used to carry out the programmed machining operation. The tool is represented by the letter "T" followed by the tool number (0-4294967295).

**·D· Tool offset number.**

This function selects the tool offset. The tool offset is represented by the letter "D" followed by the tool offset number. The number of offsets available for each tool is defined in the tool table.

**·M H· Auxiliary functions.**

With the auxiliary functions, it is possible to control machine elements such as spindle turning direction, coolant, etc. These functions are represented by the letters "M" or "H" followed by the function number (0-65535)

**·NR· Number of block repetitions.**

The NR command indicates the number of times a block is to be executed, and it can only be added to blocks when a movement, a canned cycle or a modal subroutine has been programmed. See "[14.3 Block repetition \(NR\)](#)." on page 283.

**Block comment .**

Any comment may be associated with the blocks. When executing the program, the CNC ignores this information.

The CNC offers various methods to include comments in the program. See "[1.9 Comment programming](#)." on page 50.

# 1.

**CREATING A PROGRAM.**  
Program block structure.

## 1.3.2 High-level language programming.

The commands of high level language are made up of control instructions "#" and flow control instructions "\$".

### Block structure.

A block may have the following commands, but needs not contain all of them.

```
/ N— <rest of commands>
```

#### ·/· Block skip condition.

The block skip condition is governed by the BLKSKIP1 mark of the PLC. If the mark is active, the CNC will skip the blocks where it is programmed (not executing them) and will go on to the next block.

The control reads various blocks ahead of the one being executed in order to calculate the tool path in advance. The block skip condition will be analyzed at the time when the block is read, i.e. various blocks before execution. If the block skip needs to be analyzed at the time of execution, it is necessary to interrupt the block preparation, by programming the instruction #FLUSH in the previous block.

#### [LABEL] ·N· Block label.

The labels allow the blocks to be identified. The programming of labels assists in the monitoring of the program and allows block skips and repetitions to be programmed. In this last case, it is recommended to program the labels alone in the block. See "[1.8 Programming the block labels.](#)" on page 49.

#### ·# \$· High-level language commands.

The high-level commands comprise the instructions and flow control instructions.

- Instructions are programmed preceded by the "#" sign and they can only be programmed one per block. They are used to carry out various functions.
- Flow control instructions are programmed preceded by the "\$" sign and can only be programmed one per block. They are used to make loops and program jumps.

Assigning values to parameters and variables can also be considered as high-level commands.

#### Block comment .

Any comment may be associated with the blocks. When executing the program, the CNC ignores this information.

The CNC offers various methods to include comments in the program. See "[1.9 Comment programming.](#)" on page 50.

1.

CREATING A PROGRAM.  
Program block structure.



## 1.4 Programming of the axes.

### Programming using the name of the axis.

The axis name is defined by 1 or 2 characters. The first character must be one of the letters X - Y - Z - U - V - W - A - B - C - E. The second character is optional and will be a numerical suffix between 1 and 9. This way, the name of the spindles may be within the range X, X1...X9,...C, C1...C9, E, E1...E9.

The movements are represented by the axis letter followed by the target position for the axis. For axes like X1, Y2, etc. the "=" sign must be included between the axis name and the coordinate.

```
X100
Z34.54
X2=123.4
A5=78.532
```

### Programming using wild cards.

The axes can also be programmed using wild cards. The wild cards may be used to program and refer to the axes of the channel using their position in it, including the empty spaces. The wild card is represented by the "?" character followed by the position number of the axis, ?1 for the first axis, ?2 for the second one, and so forth. If the position of a gap is programmed, the CNC will display an error message.

```
Y 00000.0000
X 00000.0000
? * * * * . * * * *
Z 00000.0000
```

In a channel with the following distribution of axes, the wild cards refer to the following axes.

- The ?1 wild card corresponds to the Y axis.
- The ?2 wild card corresponds to the X axis.
- The ?3 wild card issues an error, there is no axis in that position.
- The ?4 wild card corresponds to the Z axis.

Using these wild cards, the user can program a movement as follows.

```
?1 = 12345.1234
?2 = 50.34
```

Besides for programming movements, the wild cards can also be used to refer to the axes in the following G functions and instructions.

G functions.		Instructions.	
G14	G134	#MOVE ABS	#LINK
G45	G135	#MOVE ADD	#UNLINK
G74	G145	#MOVE INF	#PARK
G92	G158	#CAM ON	#UNPARK
G100	G170	#CAM OFF	#SERVO ON
G101	G171	#FOLLOW ON	#SERVO OFF
G112	G198	#FOLLOW OFF	
G130	G199	#TOOL AX	
G132			

# 1.

**CREATING A PROGRAM.**  
Programming of the axes.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 1.5 List of "G" functions.

The following tables show a list of "G" functions available at the CNC. The meaning of the "M", "D" and "V" fields of the table is the following:

- M· Modal function.
- D· Default function.
- V· Displayed function.

Next to each function, it indicates which chapter of this manual describes it; if no chapter is indicated, the function is described in another manual.

### ·M· Modal function.

A modal function, once programmed, remains active until an incompatible "G" code is programmed or an M02 or an M30 or until an EMERGENCY or a RESET is carried out or the CNC is turned off and back on.

Those cases indicated with "!", mean the function remains active even after an M02, M30 or a reset and after the CNC is powered off and back on.

### ·D· Default function.

It is the function that is activated by default; in other words, the function assumed by the CNC on power-up, after executing an M02 or M30 and after an EMERGENCY or a RESET.

Those cases indicated with "?" mean that the default quality of the function depends on the settings of the CNC machine parameters.

### ·V· Displayed function.

The function is displayed in automatic and jog modes next to the current machining conditions.

Function	M	D	V	Meaning	
G00	*	?	*	Rapid positioning.	8.1
G01	*	?	*	Linear interpolation.	8.2
G02	*		*	Clockwise circular (helical) interpolation.	8.3 / 8.6
G03	*		*	Counterclockwise circular (helical) interpolation.	8.3 / 8.6
G04			*	Dwell.	12.1
G05	*	?	*	Controlled corner rounding (modal).	11.3
G06			*	Arc center in absolute coordinates (not modal).	8.3.9
G07	*	?	*	Square corner (modal).	11.2
G08			*	Arc tangent to previous path.	8.4
G09			*	Arc defined by three points.	8.5
G10	*	*		Mirror image cancellation.	11.8
G11	*		*	Mirror image on X.	11.8
G12	*		*	Mirror image on Y.	11.8
G13	*		*	Mirror image on Z.	11.8
G14	*		*	Mirror image in the programmed directions.	11.8
G17	*	?	*	Main plane X-Y, and longitudinal axis Z.	4.2
G18	*	?	*	Main plane Z-X, and longitudinal axis Y.	4.2
G19	*		*	Main plane Y-Z, and longitudinal axis X.	4.2
G20	*		*	Main plane by two directions and longitudinal axis.	4.3
G30			*	Polar origin preset.	5.7
G31			*	Temporary polar origin shift to the center of arc.	8.3.8
G33	*		*	Electronic threading with constant pitch.	10.1
G34	*		*	Electronic threading with variable pitch.	10.2
G36			*	Automatic radius blend.	11.4
G37			*	Tangential entry.	11.6
G38			*	Tangential exit.	11.7
G39			*	Automatic chamfer blend.	11.5
G40	*	*		Cancellation of tool radius compensation.	13.1
G41	*		*	Left-hand tool radius compensation.	13.1
G42	*		*	Right-hand tool radius compensation.	13.1
G45				Turn tangential control on and off.	19.1
G50	*	?		Semi-rounded corner.	11.1
G53	*			Zero offset cancellation.	5.6
G54	!		*	Absolute zero offset 1.	5.5
G55	!		*	Absolute zero offset 2.	5.5
G56	!		*	Absolute zero offset 3.	5.5

1.

CREATING A PROGRAM.  
List of "G" functions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Function	M	D	V	Meaning	
G57	!		*	Absolute zero offset 4.	5.5
G58	!		*	Absolute zero offset 5.	5.5
G59	!		*	Absolute zero offset 6.	5.5
G60			*	Square corner (not modal).	11.2
G61			*	Controlled corner rounding (not modal).	11.3
G63	*		*	Rigid tapping.	10.3
G66			*	(·T· model). Pattern repeat canned cycle.	---
G68			*	(·T· model). Stock removal cycle along X axis.	---
G69			*	(·T· model). Stock removal canned cycle along Z axis.	---
G70	*	?	*	Programming in inches.	3.1
G71	*	?	*	Programming in millimeters.	3.1
G72			*	Scaling factor.	11.10
G73	*		*	Rotation of the coordinate system.	11.9
G74			*	Machine reference zero (home) search.	2.4
G80	*	*		(·M· model). Canned cycle cancellation.	---
G81	*		*	(·M· model). Drilling canned cycle.	---
G81			*	(·T· model). Turning canned cycle for straight sections.	---
G82	*		*	(·M· model). Drilling canned cycle with a variable peck.	---
G82			*	(·T· model). Facing canned cycle for straight sections.	---
G83	*		*	(·M· model). Deep-hole drilling canned cycle with constant peck.	---
G83			*	(·T· model). Drilling / tapping canned cycle.	---
G84	*		*	(·M· model). Tapping canned cycle.	---
G84			*	(·T· model). Turning canned cycle for curved sections.	---
G85	*		*	(·M· model). Reaming canned cycle.	---
G85			*	(·T· model). Facing canned cycle for curved sections.	---
G86	*		*	(·M· model). Boring canned cycle.	---
G86			*	(·T· model). Longitudinal threading canned cycle.	---
G87	*		*	(·M· model). Rectangular pocket canned cycle.	---
G87			*	(·T· model). Face threading canned cycle.	---
G88	*		*	(·M· model). Circular pocket canned cycle.	---
G88			*	(·T· model). Grooving canned cycle along the X axis.	---
G89			*	(·T· model). Z axis grooving canned cycle.	---
G90	*	?		Programming in absolute coordinates.	3.2
G91	*	?	*	Programming in incremental coordinates.	3.2
G92	!		*	Coordinate preset.	5.4
G93	*		*	Setting machining time in seconds.	6.2.1
G94	*	?		Feedrate in millimeters/minute (inches/minute).	6.2.1
G95	*	?	*	Feedrate in millimeters/revolution (inches/revolution).	6.2.1
G96	*		*	Constant surface speed.	7.2.2
G97	*	*		Constant turning speed.	7.2.2
G98	*	*		(·M· model). Withdrawal to the starting plane.	---
G99	*		*	(·M· model). Withdrawal to the reference plane at the end of the canned cycle.	---
G100			*	Probing until making contact.	---
G101	*			Include probe offset.	---
G102	*			Exclude probe offset.	---
G103			*	Probing until not making contact.	---
G104				Probe movement up to the programmed position.	---
G108	*	*		Feedrate blending at the beginning of the block.	6.2.2
G109			*	Feedrate blending at the end of the block.	6.2.2
G112	*			Changing of parameter range of an axis.	12.4
G120	!			Set lower linear limits of the work zone.	11.11.2
G121	!			Set upper linear limits of the work zone.	11.11.2
G122			*	Enable/disable the work zones.	11.11.4
G123	!			Set circular limits of the work zone.	11.11.2
G130	*		*	Percentage of acceleration to be applied per axis or spindle.	6.2.5
G131	*		*	Percentage of acceleration to be applied, global.	6.2.5
G132	*		*	Percentage of jerk to be applied per axis or spindle.	6.2.6
G133	*		*	Percentage of jerk to be applied, global.	6.2.6
G134	*		*	Percentage of Feed-Forward to be applied.	6.2.7
G135	*		*	Percentage of AC-Forward to be applied.	6.2.8
G136	*		*	Circular transition between blocks.	13.1.2
G137	*	*		Linear transition between blocks.	13.1.2
G138	*		*	Direct activation/cancellation of tool compensation.	13.1.2
G139	*	*		Indirect activation/cancellation of tool compensation.	13.1.2
G145				Freeze tangential control.	19.2
G151	*	*	*	Programming in diameters.	3.1

# 1.

**CREATING A PROGRAM.**  
List of "G" functions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

1.

**CREATING A PROGRAM.**  
List of "G" functions.

Function	M	D	V	Meaning	
G152	*			Programming in radius.	3.1
G157	*		*	Excluding axes in the zero offset.	5.5.3
G158	*		*	Incremental zero offset.	5.5.2
G159	!		*	Additional absolute zero offsets.	5.5
G160			*	(·M· model). Multiple machining in a straight line.	---
G160			*	(·T· model). Drilling / tapping canned cycle on the face of the part.	---
G161			*	(·M· model). Multiple machining in rectangular pattern.	---
G161			*	(·T· model). Drilling / tapping canned cycle on the side of the part.	---
G162			*	(·M· model). Multiple machining in a grid pattern.	---
G162			*	(·T· model). Slot milling canned cycle along the side of the part.	---
G163			*	(·M· model). Multiple machining in a circular pattern.	---
G163			*	(·T· model). Slot milling canned cycle along the face of the part.	---
G164			*	(·M· model). Multiple machining in an arc.	---
G165			*	(·M· model). Machining programmed with an arc-chord.	---
G170	*			Hirth axes OFF.	12.3
G171	*	*		Hirth axes ON.	12.3
G174	*			Set the machine coordinate.	5.2
G180			*	OEM subroutine execution.	15.5
G189					
G380			*	OEM subroutine execution.	15.5
G399					
G192	*		*	Turning speed limitation.	7.2.1
G193			*	Interpolating the feedrate.	6.2.2
G196	*		*	Constant surface speed (feedrate at the cutting point).	6.2.3
G197	*	*		Constant feedrate of the tool center.	6.2.3
G198				Setting of lower software travel limits.	12.2
G199				Setting of upper software travel limits.	12.2
G200				Exclusive manual intervention.	9.2
G201	*			Activate additive manual intervention.	9.1
G202	*	*		Cancel additive manual intervention.	9.1
G210	*		*	(·M· model). Bore milling canned cycle.	---
G211	*		*	(·M· model). Inside thread milling cycle.	---
G212	*		*	(·M· model). Outside thread milling cycle.	---
G233	*		*	Withdraw the axes after interrupting an electronic threading.	10.4
G261	*		*	Arc center in absolute coordinates (modal).	8.3.9
G262	*	*		Arc center referred to starting point.	8.3.9
G263	*		*	Arc radius programming.	8.3.2
G264	*		*	Cancel arc center correction.	8.3.11
G265	*	*		Activate arc center correction.	8.3.11
G266			*	Feedrate override at 100%.	6.2.4
G500			*	Generic user subroutines.	15.6
G599					



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 1.6 List of auxiliary (miscellaneous) M functions.

The following table shows a list of "M" functions available at the CNC. Next to each function, it indicates which chapter of this manual describes it; if no chapter is indicated, the function is described in another manual.

Function	Meaning	
M00	Program stop.	6.6.1
M01	Conditional program stop.	6.6.1
M02	End of program.	1.2.1
M03	Start the spindle clockwise.	7.3
M04	Start the spindle counterclockwise.	7.3
M05	Stop the spindle.	7.3
M06	Tool change.	6.6.1
M17	End of a global or local subroutine.	15.2
M19	Spindle orientation.	7.5
M29	End of a global or local subroutine.	15.2
M30	End of program.	1.2.1
M41	Selects gear ·1·.	7.4
M42	Selects gear ·2·.	7.4
M43	Selects gear ·3·.	7.4
M44	Selects gear ·4·.	7.4

# 1.

**CREATING A PROGRAM.**

List of auxiliary (miscellaneous) M functions.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 1.7 List of statements and instructions.

The following tables show a list of statements and instructions functions available at the CNC. Next to each of them, it indicates which chapter of this manual describes it; if no chapter is indicated, the function is described in another manual.

Instruction	Meaning	
\$GOTO	Block skip.	14.10
\$IF \$ELSEIF \$ELSE \$ENDIF	Conditional execution.	14.11
\$\$SWITCH \$CASE \$BREAK \$DEFAULT \$ENDSWITCH	Conditional execution.	14.12
\$FOR \$BREAK \$CONTINUE \$ENDFOR	Block repetition.	14.13
\$WHILE \$BREAK \$CONTINUE \$ENDWHILE	Conditional block repetition.	14.14
\$DO \$BREAK \$CONTINUE \$ENDDO	Conditional block repetition.	14.15

Instruction	Meaning	
L	Call to a global subroutine.	15.3.2
LL	Call to a local subroutine.	15.3.1
#ABORT	Abort the execution of the program and resume it in another block or program.	14.2
#ACS	Fixture coordinate system.	20.4
#ANGAX OFF	Turn angular transformation off.	18.1
#ANGAX ON	Turn angular transformation on.	18.1
#ANGAX SUSP	Freeze angular transformation.	18.2
#ASPLINE ENDTANG	Akima splines. Type of final tangent.	26.10
#ASPLINE MODE	Akima splines. Selection of tangent type.	26.10
#ASPLINE STARTTANG	Akima splines. Type of starting tangent.	26.10
#AXIS	Axis upon which the manual intervention is applied.	9.1
#CALL	Call to a global or local subroutine.	15.3.3
#CALL AX	Add a new axis to the configuration.	26.5
#CALL SP	Add a spindle to the configuration.	26.6
#CAM ON	Activate the electronic cam (real coordinates).	26.16
#CAM OFF	Cancel the electronic cam.	26.16
#CAX	Axis C. Activating the spindle as C axis.	17.1
#CD OFF	Cancel collision detection.	26.9
#CD ON	Activating collision detection.	26.9
#CLEAR	Channels. It clears the synchronism marks of the channel.	26.14
#CLOSE	Close a file.	25.3
#CONTJOG	Manual intervention. Feedrate in continuous jog.	9.3.1
#COMMENT BEGIN	Beginning of comment.	1.9
#COMMENT END	End of comment.	1.9
#CS	Machining coordinate system.	20.4
#CSROT ON	Activate tool orientation in the part coordinate system.	20.9.1
#CSROT OFF	Cancel tool orientation in the part coordinate system.	20.9.2
#CYL	"C" axis. Machining of the turning side of the part.	17.3
#DEF	Macros. Define Macros.	26.13
#DEFROT	How to manage the discontinuities in the orientation of rotary axes.	20.9.3
#DELETE	It initializes the global user variables.	1.10
#DFHOLD	Disable the feed-hold signal.	14.9
#DGWZ	It defines the graphic display area.	26.1
#DMC ON	Activating the DMC.	24.1
#DMC OFF	Deactivating the DMC.	24.2
#DSBLK	End of the single-block treatment.	14.7
#DSTOP	Disable the cycle stop signal.	14.8
#EFHOLD	Disable the feed-hold signal.	14.9
#ERROR	Display an error on the screen.	23.1

1.

CREATING A PROGRAM.

List of statements and instructions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Instruction	Meaning	
#ESBLK	Beginning of the single-block treatment.	14.7
#ESTOP	Enable the cycle stop signal.	14.8
#EXBLK	It executes a block in the indicated channel.	16.2
#EXEC	It executes a program in the indicated channel.	16.1
#FACE	"C" axis. Machining on the face of the part.	17.2
#FEEDND	Smooth the path and the feedrate.	12.5
#FLUSH	Interrupt block preparation.	14.6
#FOLLOW OFF	Independent axis. End the synchronization movement.	26.15
#FOLLOW ON	Independent axis. Begin the synchronization movement (real coordinates).	26.15
#FREE AX	Free an axis from the configuration.	26.5
#FREE SP	Free a spindle from the configuration.	26.6
#HSC OFF	It cancels the HSC mode.	21.6
#HSC ON	HSC mode. Optimizing the contouring error.	21.4
#HSC ON [FAST]	HSC mode. Optimizing the machining speed.	21.5
#INCJOG	Manual intervention. Feedrate in incremental jog.	9.3.2
#INIT MACROTAB	Macros. Initialize the table of macros.	26.13
#ISO	ISO generation.	26.2
#KIN ID	Select a kinematics.	20.3
#KINORG	Transform the current part zero considering the position of the table kinematics.	20.11
#LINK	Activate the electronic coupling (slaving) of axes.	26.3
#MASTER	Selecting the master spindle of the channel.	7.1.1
#MCALL	Modal call to a local or global subroutine initializing parameters.	15.3.5
#MCS	Program a movement referred to machine zero.	5.1
#MCS OFF	Cancel the machine coordinate system.	5.1
#MCS ON	Activate the machine coordinate system.	5.1
#MDOFF	Turning the subroutine into non-modal.	15.4
#MEET	Channels. It activates the mark in the indicated channel.	26.14
#MOVE	Independent axis. Positioning move.	26.15
#MPG	Manual intervention. Resolution of the handwheels.	9.3.3
#MSG	Display a message on the screen.	23.3
#OPEN	Open file for writing.	25.1
#PARK	Park an axis.	26.4
#PATH	Define the location of the global subroutines.	15.4
#PATHND	Smooth the path.	12.5
#PCALL	Call to a global or local subroutine initializing parameters.	15.3.4
#POLY	Polynomial interpolation.	26.11
#PROBE 1	(·M· model). Tool calibration (dimensions and wear).	*
#PROBE 1	(·T· model). Tool calibration.	*
#PROBE 2	(·M· model). Probe calibration.	*
#PROBE 2	(·T· model). Tabletop probe calibration.	*
#PROBE 3	(·M· model). Surfacing measuring.	*
#PROBE 3	(·T· model). Part measurement along the ordinate axis.	*
#PROBE 4	(·M· model). Outside corner measuring.	*
#PROBE 4	(·T· model). Part measurement along the abscissa axis.	*
#PROBE 5	(·M· model). Inside corner measuring.	*
#PROBE 6	(·M· model). Angle measurement on the abscissa axis.	*
#PROBE 7	(·M· model). Outside corner and angle measurement.	*
#PROBE 8	(·M· model). Hole measuring.	*
#PROBE 9	(·M· model). Circular boss measuring.	*
#PROBE 10	(·M· model). Rectangular part centering.	*
#PROBE 11	(·M· model). Circular part centering.	*
#PROBE 12	(·M· model). Tabletop probe calibration.	*
#RENAME AX	Rename the axes.	26.5
#RENAME SP	Rename the spindles.	26.6
#REPOS	Repositioning axes and spindles from an OEM subroutine.	15.8.1
#RET	End of a global or local subroutine.	15.2
#RETDSBLK	Execute subroutine as a single block.	15.3.7
#ROUNDPAR	Type of corner rounding.	11.3
#ROTATEMZ	Positioning a turret magazine.	6.4
#RPT	Block repetition.	14.4
#RTCP	RTCP transformation.	20.6
#SCALE	Scaling factor.	11.10
#SELECT ORI	Select onto which rotary axes of the kinematics the tool orientation is calculated for a given direction on the work piece (part).	20.9
#SELECT PROBE	Probe selection.	*
#SERVO ON	Activates the closed loop mode.	26.8

# 1.

## CREATING A PROGRAM.

List of statements and instructions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

1.

## CREATING A PROGRAM.

List of statements and instructions.

Instruction	Meaning	
#SERVO OFF	Activates the open loop mode.	26.8
#SET AX	Set axis configuration.	26.5
#SET OFFSET	Manual intervention. Manual path movement limits.	9.3.4
#SET SP	Set spindle configuration.	26.6
#SIGNAL	Channels. It activates the mark in its own channel.	26.14
#SLOPE	Acceleration control.	26.12
#SPLINE OFF	Akima splines. It cancels spline adaptation.	26.10
#SPLINE ON	Akima splines. It activates spline adaptation.	26.10
#SYNC	Spindle synchronization. Synchronization of the real coordinate.	26.7
#SYNC POS	Manual intervention. Synchronization of coordinates and additive manual offset.	9.3.5
#TANGCTRL OFF	Cancel tangential control.	19.1
#TANGCTRL ON	Activate tangential control.	19.1
#TANGCTRL SUSP	Freeze tangential control.	19.2
#TANGFEED RMIN	Minimum contouring radius for applying constant feedrate	6.2.3
#TCAM ON	Activate the electronic cam (theoretical coordinates).	26.16
#TFOLLOW ON	Independent axis. Begin the synchronization movement (theoretical coordinates).	26.15
#TIME	Dwell	12.1
#TLC	Correct the implicit tool length compensation of the program.	20.7
#TOOL AX	Longitudinal tool axis selection.	4.4
#TOOL ORI	Tool perpendicular to the inclined plane.	20.5
#TSYNC	Spindle synchronization. Synchronization of the theoretical coordinate.	26.7
#UNLINK	Cancel the electronic coupling (slaving) of axes.	26.3
#UNPARK	Unpark an axis	26.4
#UNSYNC	Spindle synchronization. Decouple the spindles.	26.7
#VIRTAX ON	Activate the virtual tool axis.	22.1
#VIRTAX OFF	Cancel the virtual tool axis.	22.2
#WAIT	Channels. It waits for a mark to be activated in the indicated channel.	26.14
#WAIT FOR	Wait for an event.	14.5
#WARNING	Display a warning on the screen.	23.2
#WARNINGSTOP	Display a warning on the screen and interrupt the program.	23.2
#WRITE	Writing in a file.	25.2

(\*) Refer to the probe manual.



## 1.8 Programming the block labels.

The labels allow the blocks to be identified. The programming of labels assists in the monitoring of the program and allows block skips and repetitions to be programmed. In this last case, it is recommended to program the labels alone in the block. The CNC has two types of labels, one is based on numbers and the other on names. Both types of labels can be programmed in the same block.

### Number labels.

Number labels are defined with the letter "N" followed by the block number (0-4294967295), where the order is irrelevant and the numbers need not be sequential. When the label is used as the destination of a block skip; the character ":" needs to be added after the number.

If the label is not used in block skips or repetitions (programmed without ":"), it may go in any block position, not necessarily at the beginning. If the label is not used in block skips or repetitions, it must go at the beginning of the block.

```
N10: X12 T1 D1  
X34 N10 S100 M3
```

### Name labels.

Name labels are programmed between angle brackets. The name of the label may be up to 14 characters long and may consist of uppercase and lowercase characters as well as numbers (no blank spaces are allowed). This type of label is not be defined at the beginning of the block.

```
[CYCLE] G81 I67
```

**1.**

**CREATING A PROGRAM.**  
Programming the block labels.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 1.9 Comment programming.

Any comment may be associated with the blocks. When executing the program, the CNC ignores this information.

The CNC offers various methods to include comments in the program.

### Programming comments in parenthesis "(" and ")".

The comment must go in parenthesis "(" and ")". Comments programmed this way need not go at the end of the block; it may go in the middle and there may be more than one comment in the same block.

```
N10 G90 X23.45 F100 (comment) S200 M3 (comment)
```

### Programming comments with the ";" character.

The information to be considered as comment must go after the ";" character. The comment may be programmed alone in the block or may be added at the end of a block.

```
N10 G90 X23.45 T1; comment
```

### Programming comments with the #COMMENT instruction.

The instructions #COMMENT BEGIN and #COMMENT END indicate the beginning and end of a comment. The blocks programmed between them are considered by the CNC as a single comment and are ignored when executing the program.

```
#COMMENT BEGIN
P1: Machining width.
P2: Machining length.
P3: Machining depth.
#COMMENT END
```

1.

**CREATING A PROGRAM.**  
Comment programming.

## 1.10 Variables and constants.

### Constants.

They are fixed values that cannot be modified by program; constants are numbers in decimal, binary and hexadecimal system and read-only tables and variables because their value cannot be changed within a program. Hexadecimal values are represented preceded by the \$ symbol.

Hexadecimal	Decimal	Binary
\$4A	74	0100 1010

### CNC variables.

The CNC has a number of internal variables that may be accessed from the user program, from the PLC or from the interface. All information regarding CNC variables can found in the manual on "CNC Variables".

### User variables.

The user can create his own variables. These are read-write variables and are evaluated during block preparation.

Variable.	Meaning.
(V.)P.name (V.)P.name[nb]	These variables maintain their value in local and global subroutines called upon from the program. The variables are deleted after executing M30 or a reset.
(V.)S.name (V.)S.name[nb]	These variables maintain their value between programs and after a reset. Variables are deleted when the CNC is turned off and they can also be deleted from the part-program using the #DELETE instruction.

Replace the suffix *name* with the name of the variable. Substitute the suffix *nb* for the number of array elements (first time) or element number within the array (subsequent times).

V.P.myvar V.S.myvar	Variable with the name of "myvar".
V.P.myvar[4] V.S.myvar[4]	Define the variable (first time it is used). <ul style="list-style-type: none"> <li>• Variable with the name of "myvar".</li> <li>• Four-element array variable.</li> </ul>
V.P.myvar[4]=100 V.S.myvar[4]=100	Once the variable is defined. <ul style="list-style-type: none"> <li>• Variable with the name of "myvar".</li> <li>• Assign the value 100 to the fourth element of the array.</li> </ul>

### Initialize the user variables.

Variables can be deleted from the part-program using the instruction #DELETE. This instruction must always go with some variable; it must not be programmed alone in the block.

```
#DELETE V.P.localvar1
#DELETE V.S.globalvar1 V.S.globalvar2
```



## 1.11 Arithmetic parameters.

Arithmetic parameters are general purpose variables that the user may utilize to create his/her own programs. The CNC has global, local and common arithmetic parameters. The range of available parameters of each type is defined in the machine parameters.

Arithmetic parameters are programmed with the "P" code followed by the parameter number. There are some tables for consulting the value of these parameters; refer to the operating manual to learn how to handle these tables.

The user may use the arithmetic parameters when editing its own programs. During execution, the CNC will replace these parameters with the values assigned to them at the time.

```
P0=0 P1=1 P2=20 P3=50 P4=3
P10=1500 P100=800 P101=30
...
GP0 XP0 YP0 SP10 MP4      ==>   G0 X0 Y0 S1500 M3
GP1 XP2 YP3 FP100         ==>   G1 X20 Y50 F800
MP101                     ==>   M30
```

### Local arithmetic parameters.

Local parameters can only be accessed from the program or subroutine where they have been programmed. There are seven groups of local parameters in each channel.

The maximum range of local parameters is P0 to P99, the typical range being P0 to P25.

When the parameters are used in the block calling a subroutine may also be referred to by the letters A-Z (except Ñ and Ç) so "A" is the same as P0 and "Z" the same as P25.

### Global arithmetic parameters.

Global parameters can be accessed from any program and subroutine called from a program. The value of these parameters is shared by the program and the subroutines. There is a group of global parameters in each channel.

The maximum range of global parameters is P100 to P9999, the typical range being P100 to P299.

### Common arithmetic parameters.

The common parameters may be accessed from any channel. The value of these parameters is shared by all the channels. Reading and writing these parameters interrupts block preparation.

The maximum range of common parameters is P10000 to P19999, the typical range being P10000 to P10999.

### Programming the arithmetic parameters.

In blocks programmed in ISO code, it is possible to define the values of all the fields "N", "G", "F", "S", "T", "D", "M", "H", "NR" and axis coordinates using parameters. Using indirect addressing, it is also possible to define the number of a parameter with another parameter; "P[P1]", "P[P2+3]".

In blocks having statements, the values of any expression may be defined with parameters.

1.

CREATING A PROGRAM.  
Arithmetic parameters.

## 1.12 Arithmetic and logic operators and functions.

An operator is a symbol that indicates the mathematical or logic operations to carry out. The CNC offers the following types of operators.

### Arithmetic operators.

To perform arithmetic operations.

Operator.	Operation.	Example.	Result.
+	Add.	$P1 = 3+4$	$P1=7$
-	Subtract. Change sign.	$P2 = 5-2$ $P2 = -[3+4]$	$P2=3$ $P2=-7$
*	Multiply.	$P3 = 2*3$	$P3=6$
/	Divide.	$P4 = 9/2$	$P4=4.5$
MOD	Module or remainder of a division.	$P5 = 5 \text{ MOD } 2$	$P5=1$
**	Exponential.	$P6 = 2**3$	$P6=8$

Operator.	Operation.	Example.	Result.
+=	Compounded addition.	$P1 += 3$	$P1=P1+3$
-=	Compounded subtraction.	$P2 -= 5$	$P2=P2-5$
*=	Compounded multiplication.	$P3 *= 2$	$P3=P3*2$
/=	Compounded division.	$P4 /= 9$	$P4=P4/9$

### Relational operators.

Used for doing comparisons.

Operator.	Operation.	Example.	Result.
==	Equal to.	$P1 == 4$	True of false.
!=	Different from, other than.	$P2 != 5$	
>=	Greater than or equal to.	$P3 >= 10$	
<=	Less than or equal to.	$P4 <= 7$	
>	Greater than.	$P5 > 5$	
<	Less than.	$P6 < 5$	

### Binary operators.

Used for doing binary comparisons between constants and/or arithmetic expressions. If the constant or the result of the arithmetic expression is a decimal number, the decimal portion will be ignored.

Operator.	Operation.	Example.	Result.
&	Binary AND.	$1010 \& 1100$	1000
	Binary OR.	$1010   1100$	1110
^	Exclusive OR (XOR).	$1010 \wedge 1100$	0110
INV[...]	Complement.	INV[0] INV[1]	1 0

### Logic operators.

Used for doing logic comparisons between conditions. Each condition should go between brackets, otherwise, an undesired comparison may be done due to the priority between operators.

Operator.	Operation.	Example.
*	Logic AND.	$\$IF [P11 == 1] * [P12 >= 5]$
+	Logic OR.	$\$IF [P21 != 0] + [P22 == 8]$

# 1.

CREATING A PROGRAM.

Arithmetic and logic operators and functions.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Boolean constants.**

Constant.	Operation.	Example.
TRUE	True.	\$IF V.S.VAR == TRUE
FALSE	Not true.	\$IF V.S.VAR == FALSE

**Trigonometric functions.**

Operator.	Operation.	Example.	Result.
SIN[...]	Sine.	P1 = SIN[30]	P1 = 0.5
COS[...]	Cosine.	P2 = COS[30]	P2 = 0.866
TAN[...]	Tangent.	P3 = TAN[30]	P3 = 0.5773
ASIN[...]	Arc-sine.	P4 = ASIN[1]	P4 = 90
ACOS[...]	Arc-cosine.	P5 = ACOS[1]	P5 = 0
ATAN[...]	Arc-tangent. (result between $\pm 90^\circ$ ).	P6 = ATAN[1]	P6 = 45
ARG[... , ...]	Arctangent y/x. (result between 0 and $360^\circ$ ).	P7=ARG[-1,1]	P7=315

In these type of functions the following must be borne in mind:

- In the "TAN" function, the argument cannot take the values  $...-90^\circ, 90^\circ, 270^\circ...$
- In the "ASIN" and "ACOS" functions, the argument must always be within  $\pm 1$ .

**Mathematical functions.**

Operator.	Operation.	Example.	Result.
ABS[...]	Absolute value.	P1 = ABS[-10]	P1 = 10
SQR[...]	Square function.	P2 = SQR[4]	P2 = 16
SQRT[...]	Square root.	P3 = SQRT[16]	P3 = 4
LOG[...]	Decimal logarithm.	P4 = LOG[100]	P4 = 2
LN[...]	Neperian logarithm.	P5 = LN[100]	P5 = 4.6051
EXP[...]	"e" function.	P6 = EXP[1]	P6 = 2.7182
DEXP[...]	Decimal exponent.	P6 = DEXP[2]	P7 = 100

In these type of functions the following must be borne in mind:

- In the "LN" and "LOG" functions, the argument must be grater than zero.
- In the "SQRT" function, the argument must be positive.

**Other functions.**

Operator.	Operation.	Example.	Result.
INT[...]	Returns the integer.	P1 = INT[4.92]	P1 = 4
FRACT[...]	Returns decimal portion.	P2 = FRACT[1.56]	P2 = 0.56
ROUND[...]	Rounds up or down to the nearest integer	P3 = ROUND[3.12] P4 = ROUND[4.89]	P3 = 3 P4 = 5
FUP[...]	Due to excess, round to an integer.	P5 = FUP[3.12] P6 = FUP[9]	P5 = 4 P6 = 9
EXIST[...]	It checks whether the selected variable or parameter exists or not	\$IF EXIST[P1] \$IF EXIST[P3] == TRUE \$IF EXIST[P3] == FALSE	

In the "EXIST" function, programming "\$IF EXIST[P1] == TRUE" is the same as programming "\$IF EXIST[P1]".

1.

CREATING A PROGRAM.

Arithmetic and logic operators and functions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 1.13 Arithmetic and logic expressions.

An expression is any valid combination of operators, constants, parameters and variables. Expressions may be used to program the numerical portion of any function, statement, etc.

The priorities of the operators and the way they can be associated determine how these expressions are calculated:

Priority from highest to lowest	To be associated
Functions, - (change sign)	from right to left.
** (exponent), MOD (remainder)	from left to right.
* (multiplication, logic AND), / (division)	from left to right.
+ (suma, OR lógico), - (resta)	from left to right.
Relational operators	from left to right.
& (AND), ^ (XOR)	from left to right.
(OR)	from left to right.

Brackets should be used in order to clarify the order in which the expression is to be evaluated. Using redundant or additional brackets will neither cause errors nor slow down the execution.

$$P3 = P4/P5 - P6 * P7 - P8/P9$$

$$P3 = [P4/P5] - [P6 * P7] - [P8/P9]$$

### Arithmetic expressions.

Their result is a numerical value. They consist of a combination of arithmetic and binary operators with constants, parameters and variables.

This type of expressions may also be used to assign values to parameters and variables:

P100 = P9   P101 = P[P7]   P102 = P[P8 + SIN[P8\*20]]  
 P103 = V.G.TOOL  
 V.G.FIXT[1].X=20   V.G.FIXT[1].Y=40   V.G.FIXT[1].Z=35

### Relational expressions.

Their result is a TRUE or a FALSE. They combine relational and logic operators with arithmetic expressions, constants, parameters and variables.

... [P8==12.6] ...

It compares if the value of P8 is equal to 12.6.

... ABS[SIN[P4]] > 0.8 ...

It compares if the absolute value of the sine of P4 is greater than 0.8.

... [[P8<=12] + [ABS[SIN[P4]] >=0.8] \* [V.G.TOOL==1]] ...



CREATING A PROGRAM.

Arithmetic and logic expressions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

1.

**CREATING A PROGRAM.**

Arithmetic and logic expressions.



# MACHINE OVERVIEW

# 2

## 2.1 Axis nomenclature

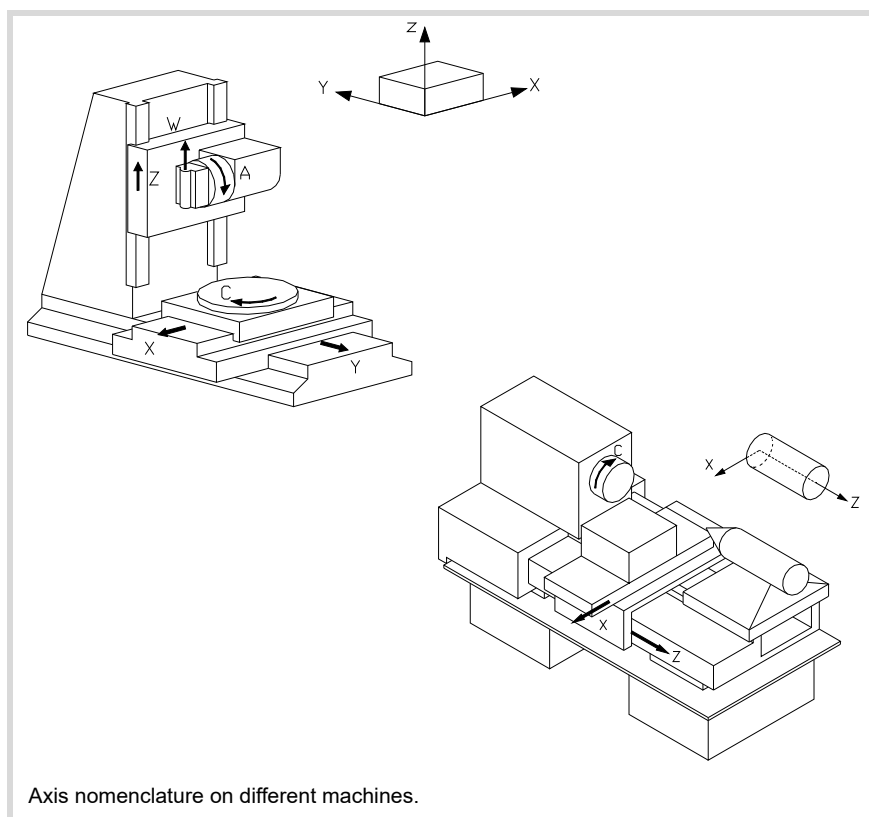
With this CNC, the manufacturer may select up to 28 axes (that must be properly defined as linear, rotary, etc. by setting machine parameters), without no limitation as how to program them and they may all be interpolated at the same time.

The DIN 66217 standard denomination for the axes is:

- X-Y-Z Main axes of the machine. The X-Y axes form the main work plane whereas the Z axis is parallel to the main axis of the machine and perpendicular to the XY plane.
- U-V-W Auxiliary axes, parallel to X-Y-Z respectively.
- A-B-C Rotary axes, on X-Y-Z respectively.
- E Extrusion axis for additive manufacturing machines or 3D printing.

However, the machine manufacturer may call the axes differently.

As an option, the name of the axes may be followed by a number between 1 and 9 (X1, X3, Y5, A8...).



Axis nomenclature on different machines.

**FAGOR** 

FAGOR AUTOMATION

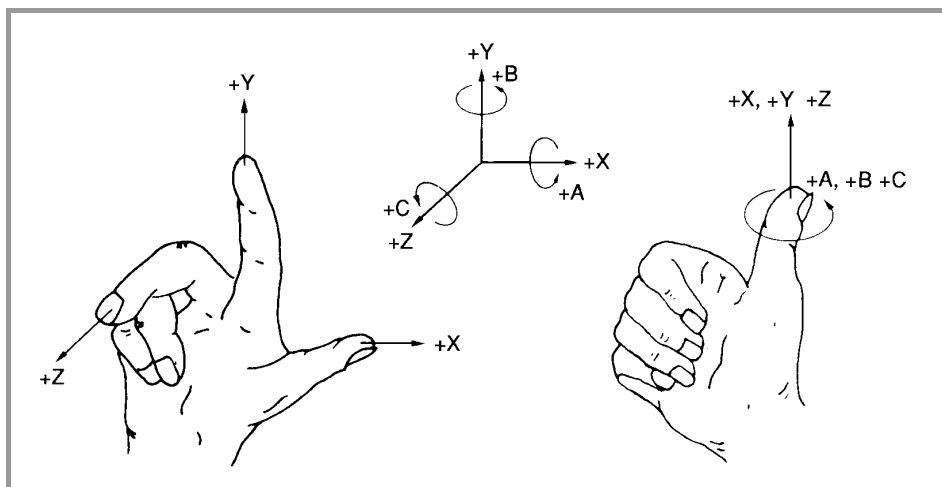
CNC 8070

(REF: 2008)

### Right-hand rule

The direction of the X-Y-Z axes can easily be remembered using the right-hand rule (see the drawing below).

On rotary axes, the positive turning direction is determined by the direction pointed by your fingers when holding the rotary axis with your hand while your thumb points in the positive direction of the linear axis.



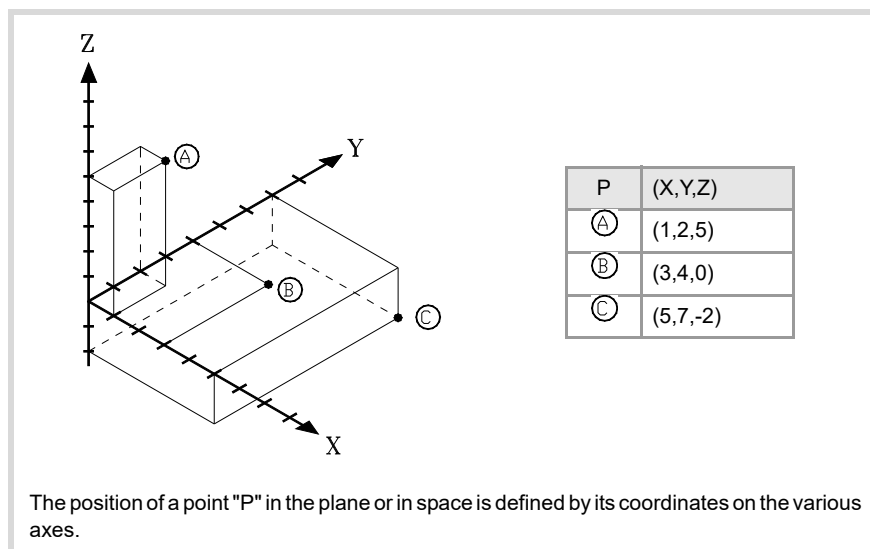
2.

MACHINE OVERVIEW  
Axis nomenclature

## 2.2 Coordinate system

Since one of the CNC's purposes is to control the movement and positioning of the axes, a coordinate system is required that permits defining the position of the various target (destination) points in the plane (2D) or in space (3D).

The main coordinate system is formed by the X-Y-Z axes. These axes are perpendicular to each other and they meet at the origin point used as reference for the various points.



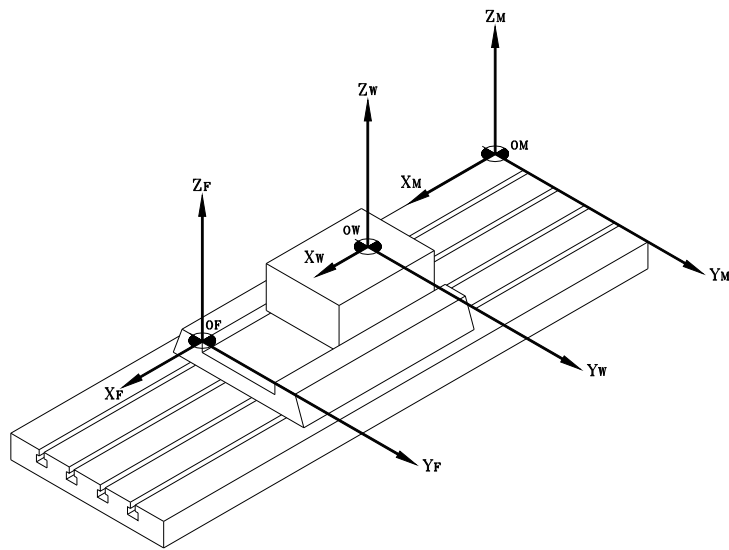
Other types of axes such as auxiliary and rotary axes may also be part of the coordinate system.

## 2.3 Reference systems

A machine may use the following reference systems.

- Machine reference system.  
It is the coordinate system of the machine and it is set by the manufacturer of the machine.
- Fixture reference system.  
It establishes a coordinate system associated with the fixtures being used. It is activated by program and may be set by the operator in any position of the machine.  
When the machine has several fixtures, each one may have its own reference system associated with it.
- Part reference system (datum point).  
It establishes a coordinate system associated with the part being machined. It is activated by program and may be set by the operator anywhere on the part.

**Example of the various coordinate systems on a milling machines.**



$X_M Y_M Z_M$  Machine reference system.  
 $X_F Y_F Z_F$  Fixture reference system.  
 $X_W Y_W Z_W$  Part reference system (datum point).

### 2.3.1 Origins of the reference systems

The position of the different reference systems is determined by their respective origin points.

**O<sub>M</sub>**  
**Machine zero.**

It is the origin point of the machine reference system, set by the machine manufacturer.

**O<sub>F</sub>**  
**Fixture zero**

It is the origin point of the fixture reference system being used. Its position is defined by the operator by using the "fixture offset" and is referred to machine zero.

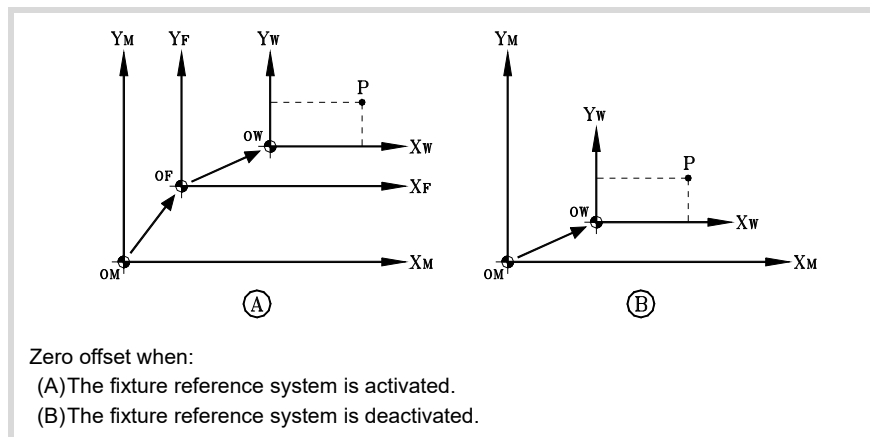
The "fixture offset" may be set by program or from the CNC's front panel, as described in the Operating Manual.

**O<sub>w</sub>**  
**Part zero**

It is the origin point of the reference system of the part (workpiece). Its position is set by the operator using the "zero offset" and is referred:

- To the fixture offset, if the fixture reference system is active. When changing the fixture reference system, the CNC updates the part zero position by referring to the new fixture zero point.
- To the machine zero point (home), if the fixture reference system is NOT active. When activating the fixture reference system, the CNC updates the part zero position by referring it to the fixture zero point.

The "zero offset" may be set from the program or from the CNC front panel as described in the Operating Manual.



# 2.

## 2.4 Home search

### 2.4.1 Definition of "Home search"

It is the operation used to synchronize the system. This operation must be carried out when the CNC loses the position of the origin point (e.g. by turning the machine off).

In order to perform the "Home search", the machine manufacturer has set particular points of the machine; the machine zero and the machine reference point.

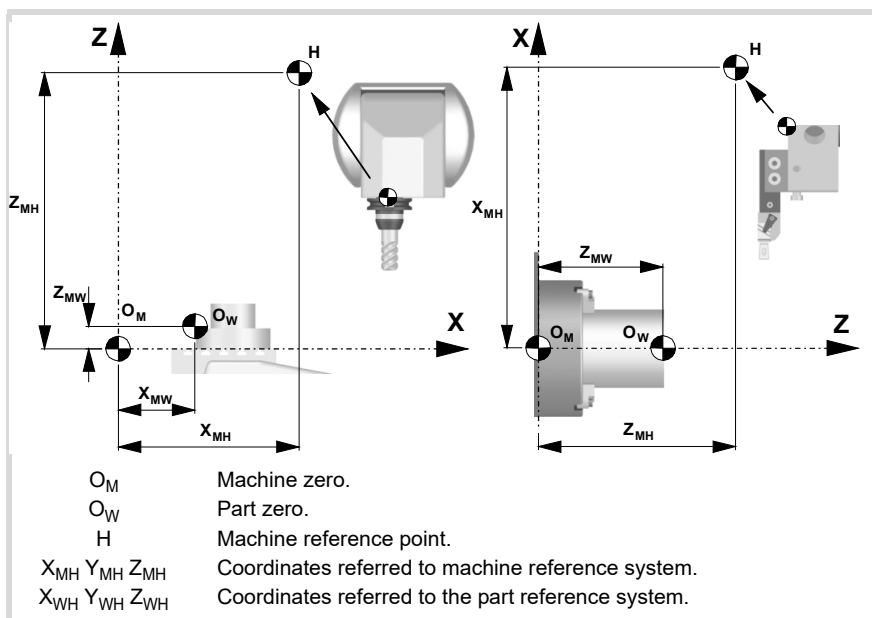
- Machine zero.

It is the origin point of the machine reference system.

- Machine reference point.

It is the physical point where the system is synchronized (except when the machine uses  $I_0$  distance-coded reference marks or absolute feedback). It may be located anywhere on the machine.

When "searching home", the axes move to the machine reference point and the CNC assumes the coordinate values assigned to that point by the machine manufacturer, referred to machine zero. When using  $I_0$  distance-coded reference marks or absolute feedback, the axes will only move the distance necessary to verify their position.



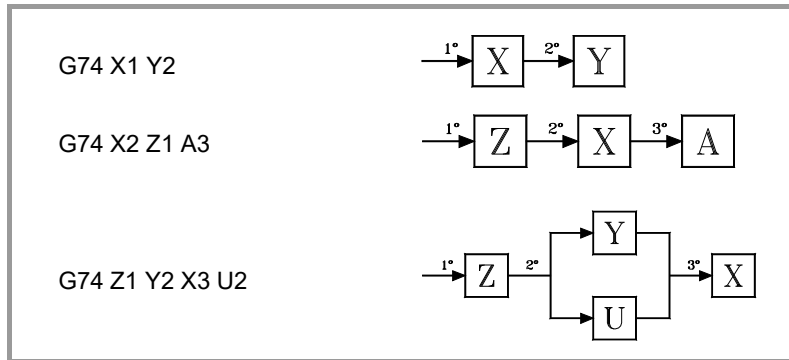
When programming a "Home search", neither the fixture offsets nor the zero offsets are canceled; therefore, the coordinates are displayed in the active reference system.

On the other hand, if "Home search" is carried out one axis at a time in JOG mode (not in MDI), the active offsets are canceled and the coordinates being displayed are referred to machine zero.

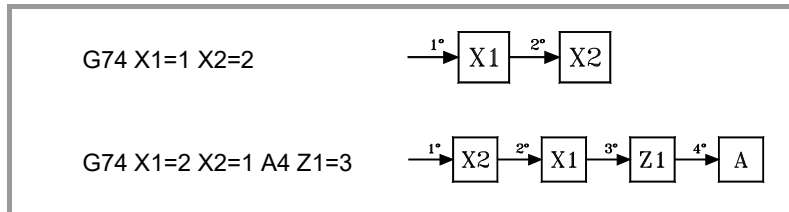
## 2.4.2 "Home search" programming

When programming a "Home search", the axes are homed sequentially in the order set by the operator. All the axes need not be included in the "Home search", only those being homed.

The "Home search" is programmed using the G74 function followed by the axes to be homed and the number indicating their homing order. If the same order number is assigned to several axes, those axes start homing at the same time and the CNC waits for all of them to end before homing the next one.



When having numbered axes, they may be defined together with the other ones by assigning them the order number as follows.



### Spindle home search

The spindle home search is always carried out together with the first axis regardless of the order in which it has been defined.

### Home search and loop status.

Axes usually work in closed loop, although rotary axes can also work in open loop so they can be controlled as if they were spindles.

The home search is carried out with the axes and spindles controlled in position; i.e. in closed position loop. The CNC will close the position loop automatically on all axes and spindles for which a home search has been programmed using function G74.

### Using an associated subroutine

If the machine manufacturer has associated a home-search subroutine to the G74 function, this function may be programmed alone in the block and the CNC will automatically execute the associated subroutine [G.M.P. "REFPSUB (G74)"].

When using a subroutine, the "Home search" is carried out exactly as described earlier.

2.

**MACHINE OVERVIEW**

Home search



FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)



## 3.1 Programming in millimeters (G71) or in inches (G70)

The displacements and feedrates of the axes may be defined in millimeters or in inches. The unit system may be selected by program using the following functions:

G70	Programming in inches.
G71	Programming in millimeters.

Both functions may be programmed anywhere in the program; they do not have to go alone in the block.

### Operation

After executing one of these functions, the CNC assumes that unit system for the following blocks. If none of these functions is programmed, the CNC uses the unit system set by machine manufacturer [G.M.P. "INCHES"].

When changing the unit system, the CNC converts the currently active feedrate into the new unit system.

```
...  
G01 G71 X100 Y100 F508      (Programming in millimeters.)  
                             (Feedrate: 508 mm/minute)  
...  
G70                          (It changes the units.)  
                             (Feedrate: 20 inches/minute)  
...
```

### Properties of the functions

The G70 and G71 functions are modal and are incompatible.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G70 or G71 as set by the machine manufacturer [G.M.P. "INCHES"].

## 3.2 Absolute (G90) or incremental (G91) coordinates.

The coordinates of the various points may be defined in absolute coordinates (referred to the active origin point) or incremental coordinates (referred to the current position). The type of coordinates may be selected by program using the following functions:

- G90            Programming in absolute coordinates.
- G91            Programming in incremental coordinates.

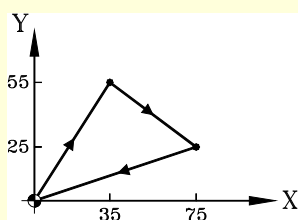
Both functions may be programmed anywhere in the program; they do not have to go alone in the block.

### Operation

After executing one of these functions, the CNC assumes that programming mode for the following blocks. If none of these functions is programmed, the CNC uses the work mode selected by machine manufacturer [G.M.P. "ISYSTEM"].

Depending on the active work mode (G90/G91), the coordinates of the points are defined as follows:

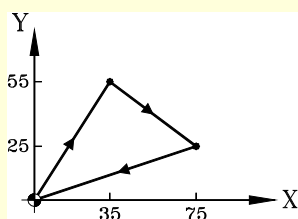
- When programming in absolute coordinates (G90), the coordinates of the point are referred to the current origin of the coordinate system, usually the part zero.



```
N10 G00 G71 G90 X0 Y0
N20 G01 X35 Y55 F450
N30 X75 Y25
N40 X0 Y0
N50 M30
```

Programming in absolute coordinates.

- When programming in incremental coordinates (G91), the coordinates of the point are referred to the current tool position. The preceding sign indicates the direction of the movement.



```
N10 G00 G71 G90 X0 Y0
N20 G01 G91 X35 Y55 F450
N30 X40 Y-30
N40 X-75 Y-25
N50 M30
```

Programming in incremental coordinates.

### Function properties

The G90 and G91 functions are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G90 or G91 as set by the machine manufacturer [G.M.P. "ISYSTEM"].

# 3.

COORDINATE SYSTEM  
Absolute (G90) or incremental (G91) coordinates.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 3.2.1 Rotary axes.

The CNC admits different ways to configure a rotary axis depending on how it is going to move. Hence, the CNC can have rotary axes with travel limits, for example between 0° and 180° (linearlike rotary axis); axes that always move in the same direction (unidirectional rotary axis); axes that choose the shortest path (positioning-only rotary axis).

All rotary axes must be programmed in degrees; therefore, they will not be affected by the mm-inch conversion. The number of revolutions the axis will turn when programming a distance greater than the module depends on the type of axis. The limits to display the position values (coordinates) also depend on the type of axis.

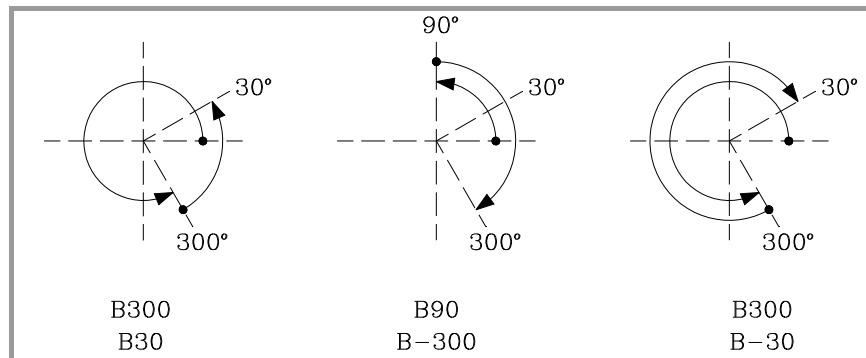
#### Linearlike rotary axis.

The axis behaves like a linear axis, but it is programmed in degrees. The CNC displays the position values between the travel limits.

#### Normal rotary axis.

This type of rotary axis can turn in both directions. The CNC displays the position values between the limits of the module.

G90 movements.	G91 movements.
The sign of the position value indicates the moving direction; the absolute position value indicates the target position.	Normal incremental movement. The sign of the position value indicates the moving direction; the absolute position value indicates the position increment.
Even if the programmed distance is greater than the module, the axis never turns more than one revolution.	If the programmed distance is greater than the module, the axis turns more than one revolution.



#### Unidirectional rotary axis.

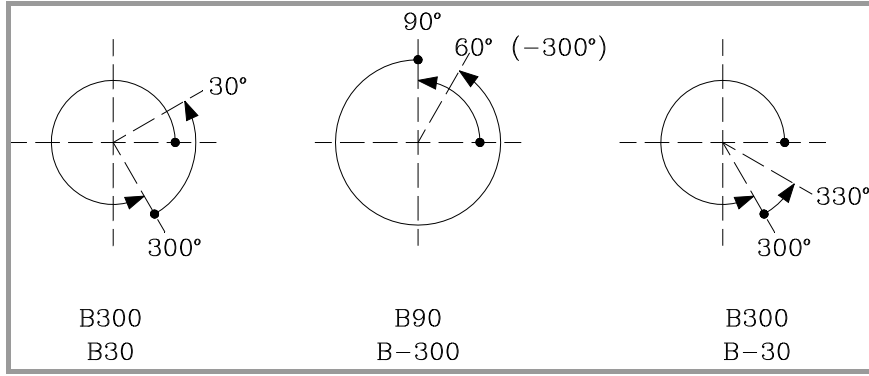
This type of rotary axis only moves in one direction, the one that has been preset for it. The CNC displays the position values between the limits of the module.

G90 movements.	G91 movements.
The axis moves in the preset direction up to the programmed position.	The axis only admits movements in the preset direction. The sign of the position value indicates the moving direction; the absolute position value indicates the position increment.
Even if the programmed distance is greater than the module, the axis never turns more than one revolution.	If the programmed distance is greater than the module, the axis turns more than one revolution.

3.

COORDINATE SYSTEM

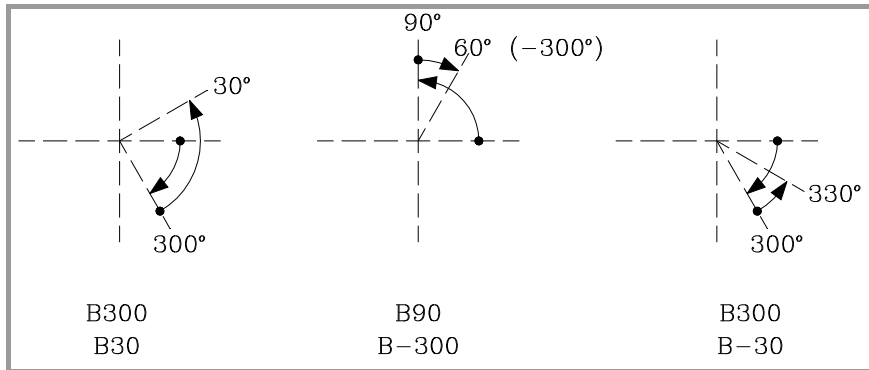
Absolute (G90) or incremental (G91) coordinates.



**Positioning-only rotary axis.**

This type of rotary axis can move in both directions; but in absolute movements, it only moves via the shortest path. The CNC displays the position values between the limits of the module.

G90 movements.	G91 movements.
The axis moves via the shortest path up to the programmed position.	Normal incremental movement. The sign of the position value indicates the moving direction; the absolute position value indicates the position increment.
Even if the programmed distance is greater than the module, the axis never turns more than one revolution.	If the programmed distance is greater than the module, the axis turns more than one revolution.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 3.3 Absolute and incremental coordinates in the same block (I).

The "I" command may be added to the programmed coordinate and it may be used to make it incremental. This command is non-modal and indicates that the coordinate is programmed incrementally, regardless of the rest of the block and of the G90/G91 function that is currently active. This way, it is possible to program absolute and incremental movements in the same block without having to use the G90/G91 functions. This kind of incremental programming is the same as G91 regardless of the scope of the application and of the result.

#### Programming.

This kind of incremental programming is only allowed when programming Cartesian or Polar coordinates. Add the "I" command after the numeric value of the coordinate to be programmed incrementally.

```
G01 X12.4 Y-0.2 Z10I
    X and Y axis movement in absolute coordinates.
    Incremental Z axis movement.

G02 X100 Y10I I20 J0
    The X coordinate of the end point is programmed in absolute (X100) and the Y
    coordinate in incremental (Y10I).

G01 R100I Q45
    Polar coordinates. Incremental radius programming.

G01 R150 Q15I
    Polar coordinates. Incremental angle programming.

G09 X35 Y20 I-15I J25
    The first point (X35 Y20) is in absolute coordinates. The X coordinate of the second
    point is programmed in incremental (I-15I) and the Y coordinate is absolute (J25).
```

#### Axis programming.

Regarding the axes, the CNC admits incremental programming when they represent coordinates (position values), blocks like G00, G01, G02, etc and G198, G199 (software limits). The incremental format is not allowed when the axes have a different meaning (G112, G74, G14, etc).

#### Axis programming using wild cards.

The CNC allows incremental programming in the wild-cards for the axes; for @1, @2, @3 and for all the ?n.

```
@1=12I @2=-34I @3=12.6I
?1=24I ?5=-23I
```

#### Parametric programming.

The CNC allows incremental programming the parameters are used as coordinates (position values).

```
XP1I
X-P10I
Z [P10+P20]I
Z2=P14I
```

#### Canned cycles.

In the canned cycles, incremental programming is only allowed in the prior movement; it is not allowed in their entry parameters.

```
X100I G81 I-25
```

3.

COORDINATE SYSTEM

Absolute and incremental coordinates in the same block (I).

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 3.4 Programming in radius (G152) or in diameters (G151).



The following functions are oriented to lathe type machines. Programming in diameters is only available on the axes allowed by the machine manufacturer (DIAMPROG=YES).

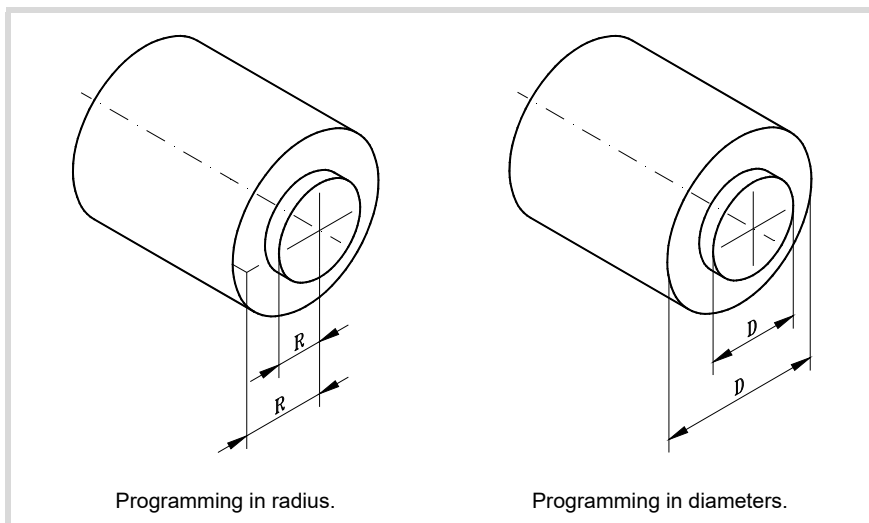
Programming in radius or diameters may be selected by program with these functions:

- |      |                           |
|------|---------------------------|
| G151 | Programming in diameters. |
| G152 | Programming in radius.    |

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

#### Operation

After executing one of these functions, the CNC assumes that programming mode for the following blocks.



When switching programming modes, the CNC changes the way it displays the coordinates of the corresponding axes.

#### Function properties

Functions G151 and G152 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G151 if machine parameter DIAMPROG of any of the axes is set to YES.

# 3.

COORDINATE SYSTEM

Programming in radius (G152) or in diameters (G151).

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 3.5 Coordinate programming

### 3.5.1 Cartesian coordinates

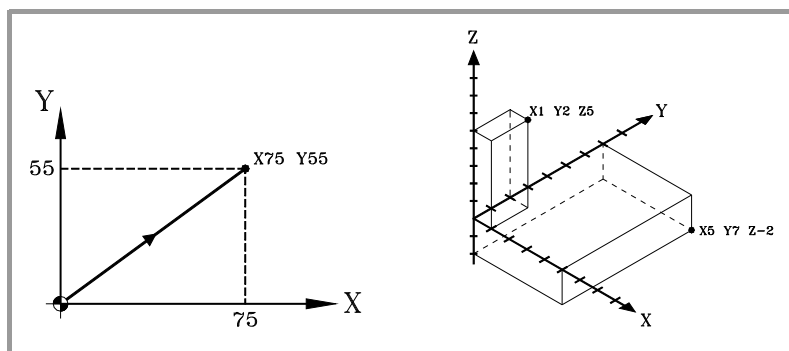
Coordinates are programmed according to a Cartesian coordinate system. This system consists of two axes in the plane and three or more in space.

#### Definition of position values

The position of a point in this system is given by its coordinates in the different axes. The coordinates are programmed in absolute or incremental coordinates and in millimeters or inches.

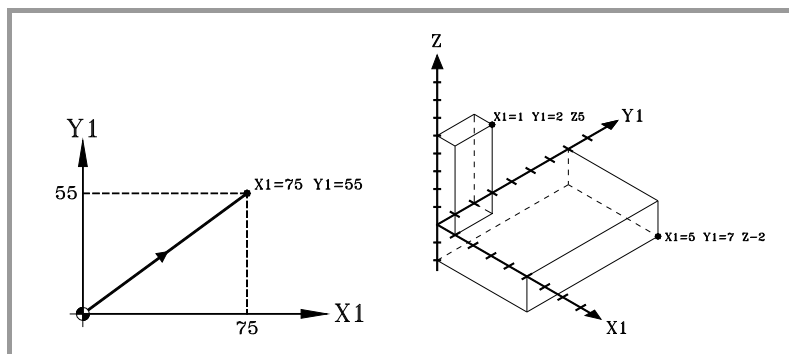
#### Standard axes (X...C)

The coordinates are programmed with the axis name followed by the coordinate value.



#### Numbered axes (X1...C9)

If the axis name is like X1, Y2... the "=" sign must be included between the axis name and the coordinate.



# 3.

### 3.5.2 Polar coordinates

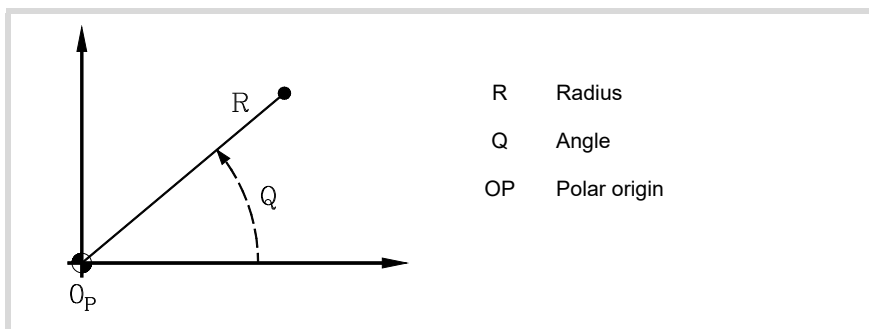
When having circular elements or angular dimensions, polar coordinates may be more convenient to express the coordinates of the various points in the plane.

This type of coordinates requires a reference point referred to as "polar origin" that will be the origin of the polar coordinate system.

#### Definition of position values

The position of the various points is given by defining the radius "R" and the angle "Q" as follows:

Radius	It will be the distance between the polar origin and the point.
Angle	It will be the one formed by the abscissa axis and the line joining the polar origin with the point.



The radius may be given in mm or in inches whereas the angle is given in degrees.

Both values may be given in either absolute (G90) or incremental (G91) coordinates.

- When working in G90, the "R" and "Q" values will be absolute. The value assigned to the radius must always be positive or zero.
- When working in G91, the "R" and "Q" values will be incremental. Although negative "R" values may be programmed, when programming in incremental coordinates, the resulting value assigned to the radius must always be positive or zero.

When programming a "Q" value greater than 360°, the module will be assumed after dividing it by 360. Thus, Q420 is the same as Q60 and Q-420 is the same as Q-60.

#### Polar origin preset

The "polar origin" may be selected from the program using function G30. If not selected, it assumes as "polar origin" the origin of the active reference system (part zero). See chapter ["5 Origin selection"](#).

The selected "polar origin" is modified in the following instances:

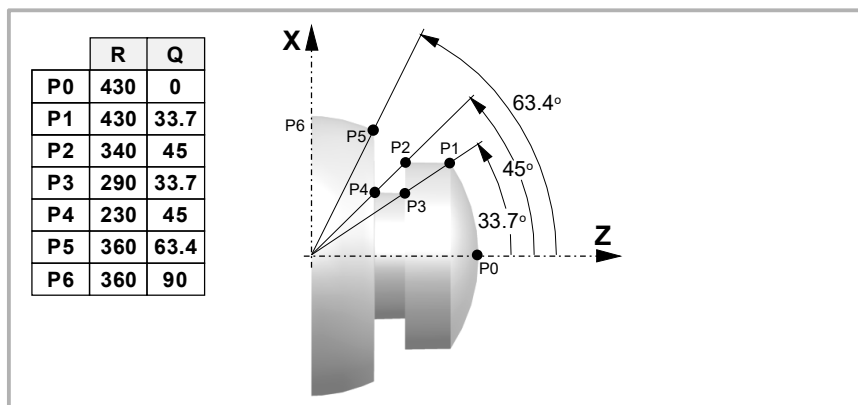
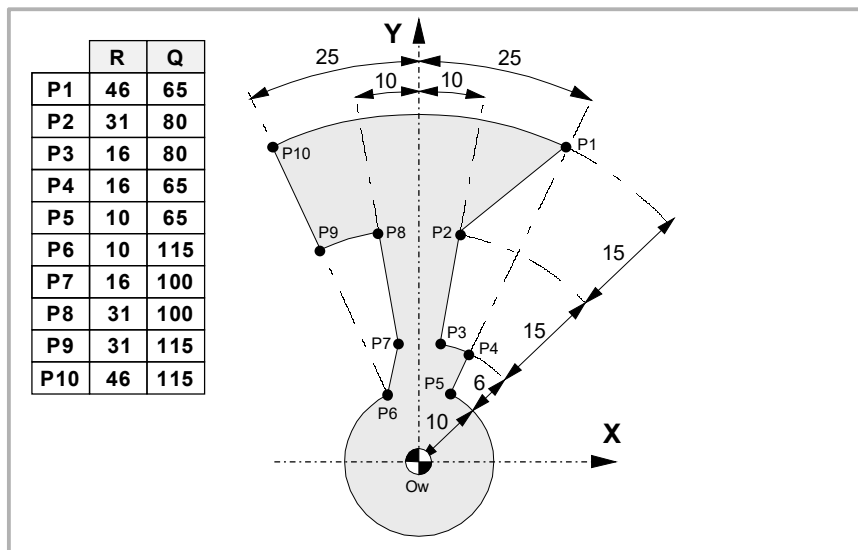
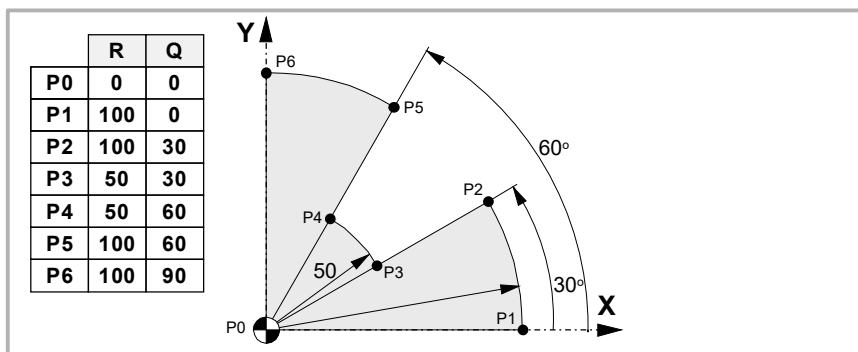
- When changing the work plane, the CNC assumes the part zero as the new "polar origin".
- On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes the part zero as the new polar origin.

3.

COORDINATE SYSTEM  
Coordinate programming



Examples. Point definition in polar coordinates.



3.

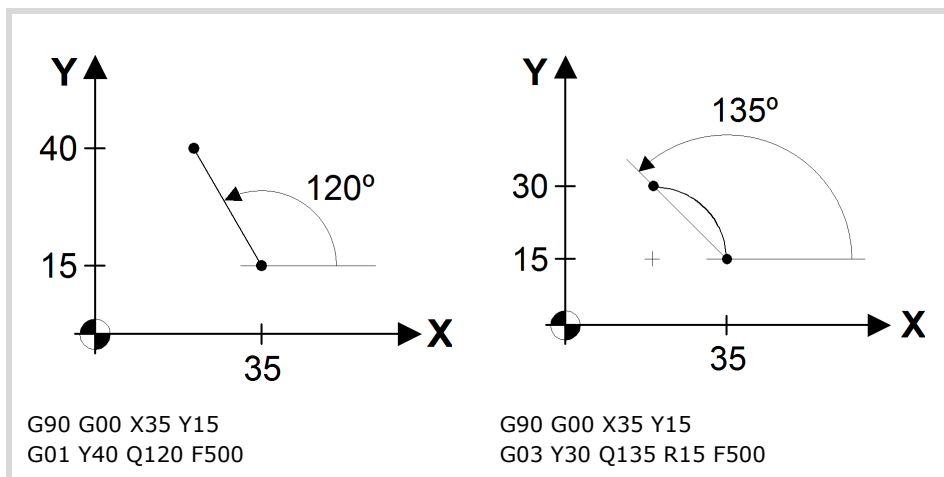
COORDINATE SYSTEM  
Coordinate programming

### 3.5.3 Angle and Cartesian coordinate.

In the main plane, a point may be defined using one of its Cartesian coordinates (X..Z) and the angle (Q) formed by the abscissa axis and the line joining the starting point and the final point. To represent a point in space, the rest of the coordinates may be programmed in Cartesian coordinates.

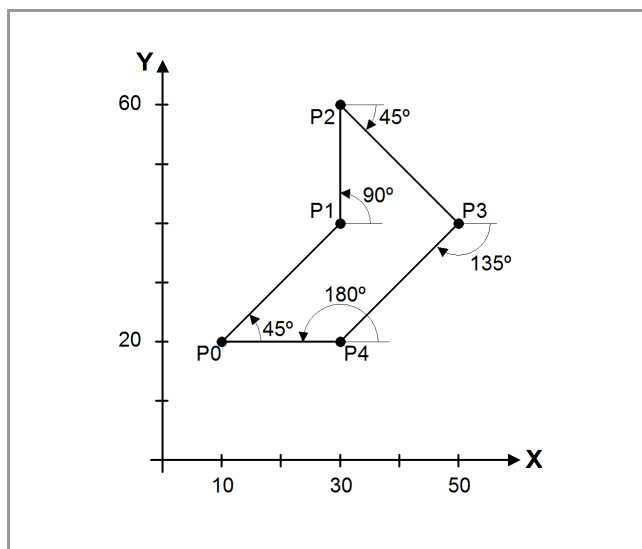
Both values, coordinate and angle, must always be programmed; otherwise, compatibility is maintained with Polar/Cartesian programming. This type of programming is valid for linear and circular interpolations.

- The coordinates may be absolute (G90) or increment (G91) and may be given in mm or inches.
- The angle will always be an absolute value (regardless of the active G90/G91 function) and it must be given in degrees.



Like in Polar programming, coordinate-angle programming is not possible while the MCS function is active.

#### Programming example (·M· model)



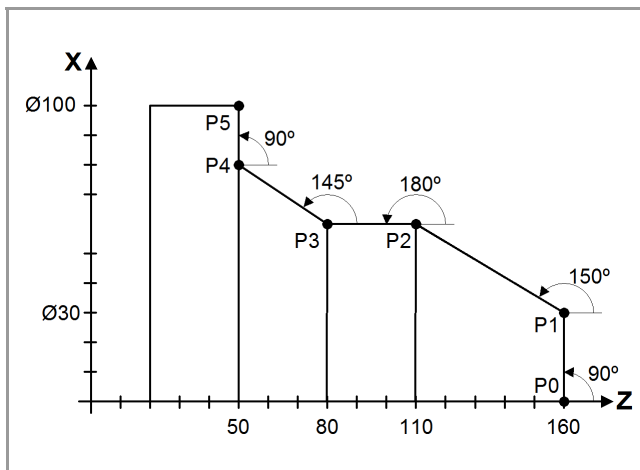
```
G00 G90 X0 Y20 ; Point P0
G01 X30 Q45 ; Point P1
G01 Y60 Q90 ; Point P2
G01 X50 Q-45 ; Point P3
G01 Y20 Q-135 ; Point P4
G01 X10 Q180 ; Point P0
```

(REF: 2008)

3.

COORDINATE SYSTEM  
Coordinate programming

### Programming example (-T- model)



```
G00 G90 X0 Z160 ; Point P0
G01 X30 Q90 ; Point P1
G01 Z110 Q150 ; Point P2
G01 Z80 Q180 ; Point P3
G01 Z50 Q145 ; Point P4
G01 X100 Q90 ; Point P5
```

# 3.

**COORDINATE SYSTEM**  
Coordinate programming

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

# 3.

## COORDINATE SYSTEM

Coordinate programming



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# WORK PLANES.

# 4

The work planes determine which axes define the work plane/trihedron and which axis corresponds to the longitudinal axis of the tool. Plane selection is required to execute operations like:

- Circular and helical interpolations.
- Corner chamfering and rounding.
- Tangential entries and exits.
- Machining canned cycles.
- Tool radius and length compensation.

These operations, except tool length compensation, can only be executed in the active work plane. Tool length compensation, on the other hand, can only be applied on the longitudinal axis.

## Commands for changing the work planes.

### Mill model or lathe model with "trihedron" type axis configuration.

Function.	Meaning.
G17	Main plane formed by the first axis (abscissa), second (ordinate) and third axis (perpendicular) of the channel.
G18	Main plane formed by the third axis (abscissa), first axis (ordinate) and second axis (perpendicular) of the channel.
G19	Main plane formed by the second axis (abscissa), third axis (ordinate) and first axis (perpendicular) of the channel.
G20	Select any work plane formed by the first three axes of the channel.

Instruction.	Meaning.
#TOOL AX	Select the longitudinal axis of the tool.

### Lathe model with "plane" type axis configuration.

Function.	Meaning.
G18	Main plane formed by the second axis (abscissa) and first axis (ordinate) of the channel.
G20	Select the longitudinal axis of the tool.

Instruction.	Meaning.
#TOOL AX	Select the longitudinal axis of the tool.



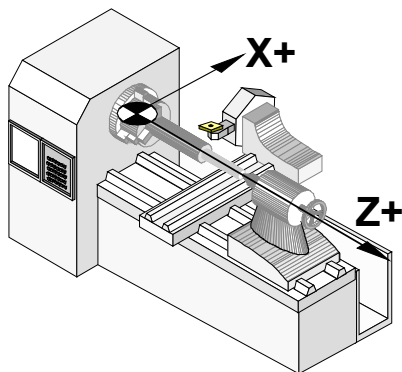
FAGOR AUTOMATION

CNC 8070

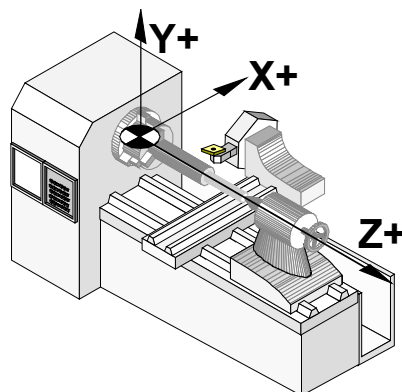
(REF: 2008)

## 4.1 About work planes on lathe and mill models.

The operation of the work planes depends on the geometric configuration of the axes. At a mill model, the geometric configuration of the axes is always of the "trihedron" type whereas at a lathe model, the geometric configuration of the axes may be either a "trihedron" type or a "plane" type (parameter GEOCONFIG).



Configuration of "plane" type axes.



Configuration of "Trihedron" type axes.

### Configuration of "Trihedron" type axes (lathe or mill model).

This configuration has three axes forming a trihedron Cartesian XYZ type. There may be more axes, besides those forming the trihedron; that may be part of the trihedron or be auxiliary axes, rotary axes, etc.

The order of the axes in the channel sets the main work planes, those selected with functions G17, G18 and G19. Function G20 may be used to form any work plane with the first three axes of the channel. The work plane by default is set by the manufacturer (parameter IPLANE), the usual plane being G17 at a mill model and G18 at a lathe model.

The CNC displays the ·G· functions associated with the work planes.

### Configuration of "plane" type axes (lathe model).

This configuration has two axes forming the usual work plane on a lathe. There may be more axes, but they cannot be part of the trihedron; there must be auxiliary, rotary, etc.

With this configuration, the work plane is always G18 and will be formed by the first two axes defined in the channel, the second axis as abscissa and the first axis as ordinate. The ·G· functions associated with the work planes have the following effects.

Function.	Meaning.
G17	It does not change planes and shows a warning about it.
G18	It has no effect (except when function G20 is active).
G19	It does not change planes and shows a warning about it.
G20	It is permitted if it does not change the main plane; i.e. it can only be used to change the longitudinal axis.

The CNC does not display the ·G· functions associated with the work planes because it is always the same plane.

# 4.

WORK PLANES.  
About work planes on lathe and mill models.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 4.2 Select the main new work planes.

### 4.2.1 Mill model or lathe model with "trihedron" type axis configuration.

The main planes may be selected by program using functions G17, G18 and G19 and are formed by two of the first three axes of the channel. The third axis corresponds to the axis perpendicular to the plane, which coincides with the longitudinal axis of the tool, the one on which tool length compensation is applied.

- G17 Main plane formed by the first axis (abscissa), second (ordinate) and third axis (perpendicular) of the channel.
- G18 Main plane formed by the third axis (abscissa), first axis (ordinate) and second axis (perpendicular) of the channel.
- G19 Main plane formed by the second axis (abscissa), third axis (ordinate) and first axis (perpendicular) of the channel.

The OEM, can use machine parameter LCOMPTYP to change the behavior of the longitudinal axis when changing planes so the CNC keeps the longitudinal axis that was active before changing planes.

Function G20 may select any plane with the first three axes of the channel. Function G20 and the instruction #TOOL AX can change the longitudinal axis of the tool.

### Programming.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

#### Programming format.

The programming format is:

G17  
G18  
G19

G17  
G18  
G19

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Functions G17, G18, G19 and G20 are modal and incompatible with each other. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G17 or G18 as set by the machine manufacturer (parameter "IPLANE").

# 4.

## WORK PLANES.

Select the main new work planes.

## 4.2.2 Lathe model with "plane" type axis configuration.

The work plane is always G18 and will be formed by the first two axes defined in the channel. Functions G17 and G19 have no meaning for the CNC.

**G18** Main plane formed by the second axis (abscissa) and first axis (ordinate) of the channel.

In the case of lathe tools, tool length compensation is applied on all the axes where a tool offset has been defined.

On milling tools, tool length compensation is applied on the second axis of the channel. If the X (first axis of the channel) and Z (second axis of the channel) axes have been defined, the work plane will be the ZX and Z will be the longitudinal axis. Function G20 and the instruction #TOOL AX can change the longitudinal axis of the tool.

### Programming.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

#### Programming format.

The programming format is:

G18

G18

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Functions G18 and G20 are modal and incompatible with each other. On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC assumes function G18.

# 4.

**WORK PLANES.**  
Select the main new work planes.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### 4.3 Select any work plane and longitudinal axis.

The meaning of function G20 depends on the type of configuration of the machines axes; "plane" type for lathe or "trihedron" type for lathe or mill.

- When the axis configuration is of trihedron type, function G20 allows defining any work plane formed by the first three axes of the channel. To build a plane with other axes, first include them in the main trihedron (instruction #SET AX).
- When the axis configuration is of plane type, the work plane is always G18 and function G20 allows changing the longitudinal axis of the tool.

#### Programming.

When programming this instruction, you must define the new abscissa and ordinate axes of the plane and the longitudinal axis of the tool. If the longitudinal axis coincides with one of the axes of the plane, you must also define which axis is perpendicular to the plane.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

G20 X~C{axistype} X~C{axistype} X~C{axistype} <X~C{axistype}>  
{axistype} Value that sets the location of the axis in the plane.

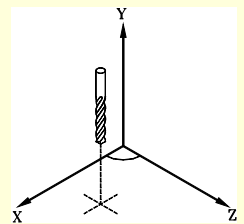
Values for setting the location of the axis in the plane.

The work plane is defined by selecting the abscissa and ordinate axes, the perpendicular axis and the longitudinal axis of the tool. It is selected by assigning one of the following values to the axes programmed with G20.

Value.	Type of axis within the work plane.
1	Abscissa axis.
2	Ordinate axis.
±3	Longitudinal axis of the tool. The sign indicates tool orientation.
4	Reserved.
5	Axis perpendicular to the work plane, only required when the longitudinal axis of the tool is the same as the abscissa or ordinate axis. Otherwise, the longitudinal axis of the tool will be the perpendicular axis.

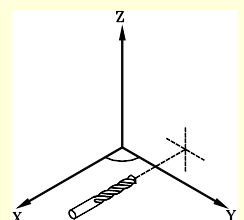
#### G20 X1 Z2 Y3

The X axis is the abscissa axis.  
The Z axis is the ordinate axis.  
The Y axis is the longitudinal axis of the tool and the axis perpendicular to the plane.



#### G20 X1 Y2 X3 Z5

The X axis is the abscissa axis and the longitudinal axis of the tool.  
The Y axis is the ordinate axis.  
The Z axis is the axis perpendicular to the plane.



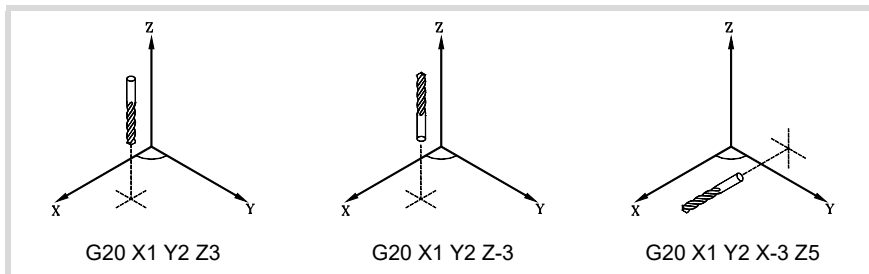
# 4.

WORK PLANES.  
Select any work plane and longitudinal axis.

**Select the longitudinal axis of the tool.**

When selecting the longitudinal axis with G20, tool orientation may be established according to the programmed sign.

- If the parameter to select the longitudinal axis is positive, the tool is positioned in the positive direction of the axis.
- If the parameter to select the longitudinal axis is negative, the tool is positioned in the negative direction of the axis.

**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

Function G20 is modal and incompatible with G17, G18 and G19. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G17 or G18 as set by the machine manufacturer (parameter "IPLANE").

## 4.

**WORK PLANES.**

Select any work plane and longitudinal axis.

## 4.4 Select the longitudinal axis of the tool.

The instruction #TOOL AX allows changing the longitudinal axis of the tool except on those for turning. This instruction allows to select any machine axis as the new longitudinal axis.

### Programming.

When programming this instruction, you must define the new axis and the orientation of the tool.

#### Programming format.

The programming format is the following; the list of arguments appears inside the curly brackets.

#TOOL AX [X~C{+|-}]

{+|-} Tool orientation.

#TOOL AX [Z+]

#TOOL AX [V2-]

#### Define the orientation of the tool.

Tool orientation is set as follows.

+ sign Positive tool orientation.

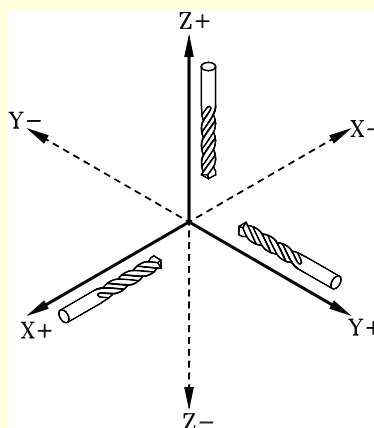
- sign Negative tool orientation.

##### Positive tool orientation.

#TOOL AX [X+]

#TOOL AX [Y+]

#TOOL AX [Z+]

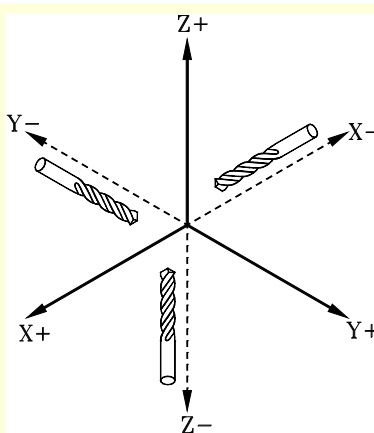


##### Negative tool orientation.

#TOOL AX [X-]

#TOOL AX [Y-]

#TOOL AX [Z-]



# 4.

## WORK PLANES.

Select the longitudinal axis of the tool.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 4.

## WORK PLANES.

Select the longitudinal axis of the tool.

# ORIGIN SELECTION

# 5

With this CNC, it is possible to program movements in the machine reference system or apply offsets in order to use reference systems referred to the fixtures or the part without having to change the coordinates of the different points of the part in the program.

There are three different offset types; fixture offset, zero offsets and PLC offsets. The CNC may have several of these offsets active at the same time, in that case, the system coordinate origin being used will be defined by the sum of the active offsets.

Type of offset.	Description.
Fixture offset.	Distance between the machine reference zero and the fixture's zero point. On machines using several fixtures, this offsets allows selecting the particular fixture to be used.
Zero offset.	Distance between the fixture's zero point and the part zero. If the fixture zero is not active (no fixture offset), the zero offset is measured from machine zero. The zero offset may be set by presetting a coordinate or a zero offset.
PLC offset.	Special offset handled by the PLC that is used to correct the deviations due to dilatations, etc. The PLC always applies this offset, even when programming with respect to machine zero.

## 5.1 Programming with respect to machine zero

Machine zero is the origin of the machine reference system. Movements referred to machine zero are programmed using the instructions #MCS and #MCS ON/OFF.

### Program a movement referred to machine zero.

This instruction may be added to any block containing a movement so it is executed in the machine reference system.

```
G00 X30 Y30
G92 X0 Y0           (Coordinate preset)
G01 X20 Y20
#MCS X30 Y30       (Movement referred to machine zero. Offsets canceled)
G01 X40 Y40       (Offsets restored)
G01 X60 Y60
M30
```

### Machine coordinate system.

The #MCS ON and #MCS OFF instructions activate and deactivate the machine reference system; therefore, the movements programmed between them are executed in the machine reference system. Both instructions must be programmed alone in the block.

```
G92 X0 Y0           (Coordinate preset)
G01 X50 Y50
#MCS ON             (Beginning of programming referred to machine zero)
G01 ...
G02 ...
G00 ...
#MCS OFF           (End of programming referred to machine zero. Offsets restored)
```

### Considerations for movements referred to machine zero.

#### Zero offsets and coordinate transformations

When executing a movement referred to machine zero, the CNC ignores the active offsets (except the PLC offset), the kinematics and cartesian transformations; therefore, the movement is carried out in the machine reference system. Once the movement has ended, the CNC restores the offsets, kinematics and cartesian transformations that were active.

The programmed movements do not admit polar coordinates, nor other kinds of transformations such as mirror image, coordinate (pattern) rotation or scaling factor. While the #MCS function is active, functions for setting a new origin such as G92, G54-G59, G158, G30, etc. are not admitted either.

#### Tool radius and length compensation

Tool radius and length compensation is also canceled during the movements referred to machine zero. The CNC assumes that the coordinates have been programmed with respect to the tool base, not to the tool tip.

5.

ORIGIN SELECTION

Programming with respect to machine zero



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### System units; millimeters or inches

When moving with respect to machine reference zero, the G70 or G71 units (inches/millimeters) selected by the user are ignored. It assumes the units predefined at the CNC (INCHES parameter); assumed by the CNC on power-up. These units are assumed for defining the coordinates, for the feedrate and for the speed.

# 5.

## ORIGIN SELECTION

Programming with respect to machine zero

## 5.2 Set the machine coordinate (G174).



Use this function with caution. Changing the machine coordinate can cause the axes to exceed the travel limits during the movement.

Function G174 may be used to set the machine coordinate of an axis or spindle; in other words, temporarily set a new machine zero. The new machine coordinate stays active until the axis or spindle is homed; then, the CNC restores the original machine reference zero (set in the machine parameters).

After executing function G174, the CNC assumes that the programmed coordinate defines the current position referred to machine reference zero (home). The zero offsets, movements with respect to machine zero, etc. will be referred to the coordinate programmed in G174.

### Programming the function.

Program function G174, and then the machine coordinate of a single axis or spindle. For gantry axes, program the machine coordinate of the master axis. With this function only the machine coordinate of an axis or spindle may be set; to set the machine coordinates of several, program one G174 for each one of them.

When setting the machine coordinate, the CNC ignores the G70/G71 units (inches/millimeters) selected by the user and uses the unit system pre-defined at the CNC (parameter INCHES). The CNC also ignores all the other options, radius/diameter, mirror image, scaling factor, etc.

### Programming format.

The programming format is:

G174 X..C

G174 S

X..C            Machine coordinate at the axes.

S                Machine coordinate at the spindles.

G174 X100

G174 S180

### Considerations and limitations.

- Function G174, by itself, does not cause any axis or spindle movement. After executing function G174, the CNC considers that the axis or spindle is homed and verifies that it is within the software travel limits.
- The CNC does not allow setting the machine coordinate on slaved axes, tandem or on axes that are part of the active kinematics or active transform. The CNC permits setting the machine coordinate for tandem axes. Before setting the new machine coordinate, the CNC checks that the axis or spindle is in position and it is not synchronized, if that's not the case, it issues an error message.



On Sercos axes, function G174 also resets the coordinate of the drive. Setting the machine coordinates on position-Sercos axes requires drive version V6.20 or newer.

- On gantry axes, the CNC applies the coordinate defined in G174 to both axes, master and slave.
- A G174 can be executed in a deactivated multi-axis group.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Function G174 is modal. This function is neither affected by functions M02 and M30 nor by a reset, by an emergency or by turning the CNC off. On power-up, the CNC assumes the machine coordinates that were active when the CNC was turned off.

# 5.

ORIGIN SELECTION  
Set the machine coordinate (G174).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### 5.3 Fixture offset

With fixture offsets, it is possible to select the fixture system to be used (when having more than one fixture). When applying a new fixture offset, the CNC assumes the point set by the new selected fixture as the new fixture zero.

#### Definition

In order to apply a fixture offset, it must have been previously set. To do that, the CNC has a table where the operator may define up to 10 different fixture offsets. The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- By program, assigning the corresponding value (of the "n" offset and of the "Xn" axis) to the "V.A.FIXT[n].Xn" variable.

#### Activation

Once the fixture offsets have been defined in the table, they may be activated via program by assigning to the "V.G.FIX" variable, the offset number to be applied.

Only one fixture offset may be active at a time; therefore, when applying a fixture offset, it will cancel the previous one. Assigning a value of "V.G.FIX=0" will cancel the active fixture offset.

Fixture offset value on milling machine.

	X	Y
V.G.FIX=1	30	50
V.G.FIX=2	120	50

```

N100 V.A.FIXT[1].X=30 V.A.FIXT[1].Y=50
N110 V.A.FIXT[2].X=120 V.A.FIXT[2].Y=50
...
N200 V.G.FIX=1          (It applies the first fixture offset)
N210 ...                (Programming at fixture 1)
N300 V.G.FIX=2          (It applies the first fixture offset)
N310 ...                (Programming at fixture 2)
N400 V.G.FIX=0          (Cancel fixture offset. No fixture system is active)
    
```

#### Considerations

A fixture offset, by itself, does not cause any axis movement.

#### Properties

On power-up, the CNC assumes the fixture offset that was active when the CNC was turned off. On the other hand, the fixture offset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

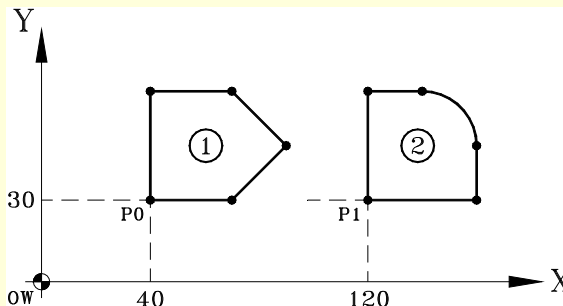
## 5.4 Coordinate preset (G92)

Coordinate presetting is done with function G92 and it may be applied onto any axis of the machine.

When presetting coordinates, the CNC interprets that the axis coordinates programmed after function G92 define the current position of the axes. The rest of the axes that have not been defined with G92 are not affected by the preset.

5.

ORIGIN SELECTION  
Coordinate preset (G92)



N100 G90 G01 X40 Y30	(Positioning at P0)
N110 G92 X0 Y0	(Presetting P0 as part zero)
...	(Machining of profile 1)
N200 G90 G01 X80 Y0	(Positioning at P1)
N210 G92 X0 Y0	(Presetting P1 as part zero)
...	(Machining of profile 2)
N300 G92 X120 Y30	(Recovering OW as part zero)

### Considerations

A coordinate preset, by itself, does not cause any axis movement.

When homing an axis in JOG mode, the preset for that axis is canceled.

### Function properties

G92 is modal, the preset values remain active until the preset is canceled (with another preset, a zero offset or with G53).

On power-up, the CNC assumes the coordinate preset that was active when the CNC was turned off. On the other hand, the coordinate preset is neither affected by functions M02 and M30 nor by RESETTING the CNC.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 5.5 Zero offsets (G54-G59/G159)

The zero offsets may be used to set the part zero at different positions of the machine. When applying a zero offset, the CNC assumes as the new part zero the point defined by the selected zero offset.

### Defining zero offsets.

In order to apply a zero offset, it must have been previously defined. To do that, the CNC has a table where the operator may define up to 99 different zero offsets. The table data may be defined manually (as described in the operating manual) or via program (using variables).

The OEM may have configured the zero offset table in one of the following ways (machine parameter FINEORG).

- Each zero offset has a single value. When executing function G159, the CNC assumes this value as the new zero offset.
- Each zero offset has a coarse (or absolute) value and a fine (or incremental) value. When executing function G159, the CNC assumes as new zero offset the sum of both parts.

### Activating a zero offset.

Once the zero offsets have been defined in the table, they may be activated via program by programming function G59 followed by the offset number to be activated.

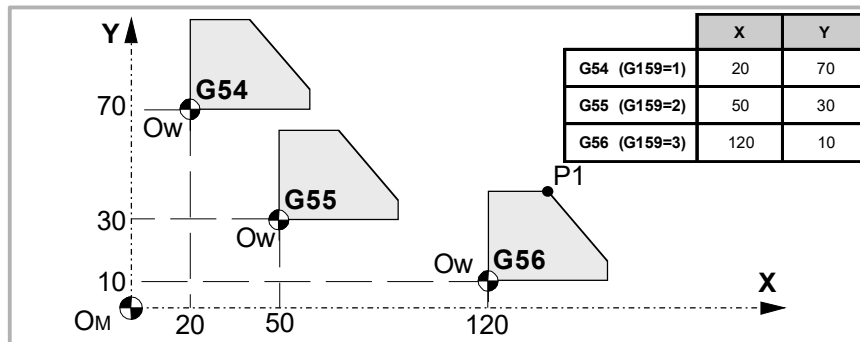
G159=2      The CNC applies the second zero offset.

G159=11     The CNC applies the 11th zero offset.

The first six zero offsets of the table can also be applied using functions G54 through G59; G54 for the first one (same as G159=1), G55 for the second one (same as G159=2) and so on.

G54          The CNC applies the first zero offset (G159=1).

G59          The CNC applies the sixth zero offset (G159=6).



```
N100 V.A. ORGT[1].X=20 V.A. ORGT[1].Y=70
N110 V.A. ORGT[2].X=50 V.A. ORGT[2].Y=30
N100 V.A. ORGT[3].X=120 V.A. ORGT[3].Y=10
```

...

```
N100 G54
      (It applies the first zero offset)
```

```
N200 G159=2
      (It applies the second zero offset)
```

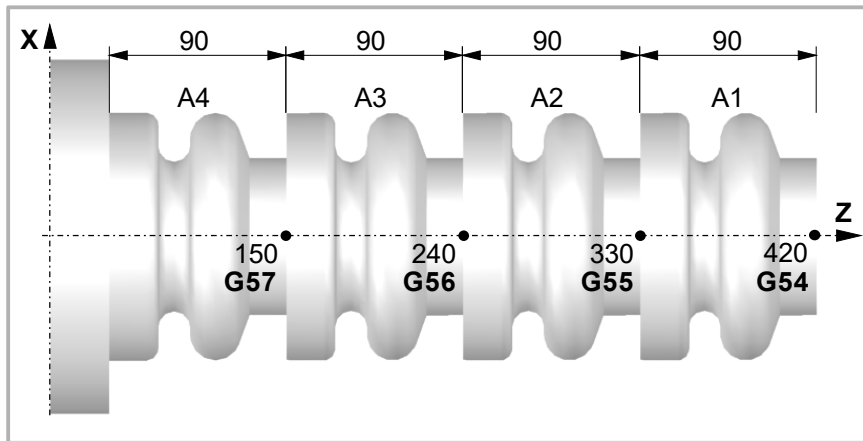
```
N300 G56 X20 Y30
      (It applies the third zero offset)
      (The axes move to point X20 Y30 (point P1) referred to the third origin)
```

5.

ORIGIN SELECTION  
Zero offsets (G54-G59/G159)

5.

**ORIGIN SELECTION**  
Zero offsets (G54-G59/G159)



	X	Z
G54 (G159=1)	0	420
G55 (G159=2)	0	330
G56 (G159=3)	0	240
G57 (G159=4)	0	150

```

N100 V.A. ORGT[1].X=0 V.A. ORGT[1].Z=420
N110 V.A. ORGT[2].X=0 V.A. ORGT[2].Z=330
N100 V.A. ORGT[3].X=0 V.A. ORGT[3].Z=240
N100 V.A. ORGT[4].X=0 V.A. ORGT[4].Z=150
N100 G54 (It applies the first absolute zero offset)
... (Machining of profile A1)
N200 G55 (It applies the second absolute zero offset)
... (Machining of profile A2)
N300 G56 (It applies the third absolute zero offset)
... (Machining of profile A3)
N200 G57 (It applies the fourth absolute zero offset)
... (Machining of profile A4)
    
```

Only one zero offset may be active at a time; therefore, when applying a zero offset, the previous one will be canceled. When programming G53, the zero offset currently active will be canceled.

The function corresponding to the selected zero offset may be programmed in any block of the program. When added to a block with path information, the zero offset will be applied before executing the programmed movement.

**Considerations**

A zero offset, by itself, does not cause any axis movement.

When homing an axis in JOG mode, the absolute zero offset for that axis is canceled.

**Properties of the functions**

Functions G54, G55, G56, G57, G58, G59 and G159 are modal and incompatible with each other and with G53 and G92.

On power-up, the CNC assumes the zero offset that was active when the CNC was turned off. On the other hand, the zero offset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

## 5.5.1 Variables for setting zero offsets

### Zero offset table (without fine setting of the absolute zero offset).

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

Variable.	R/W	Meaning.
(V.)[ch].A.ORG.xn	R	Value of the active zero offset (absolute G159 + incremental G158).
(V.)[ch].A.ADDORG.xn	R	Value of the active incremental zero offset (G158).
(V.)[ch].A.ORG[nb].xn	R/W	Offset set in the zero offset [nb].

### Zero offset table (with fine setting of the absolute zero offset).

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

Variable.	R/W	Meaning.
(V.)[ch].A.ORG.xn	R	Value of the active zero offset (coarse absolute G159 + fine absolute G159 + incremental G158).
(V.)[ch].A.ADDORG.xn	R	Value of the active incremental zero offset (G158).
(V.)[ch].A.COARSEORG.xn	R	Value of the active absolute zero offset (G159), coarse part.
(V.)[ch].A.FINEORG.xn	R	Value of the active absolute zero offset (G159), fine part.
(V.)[ch].A.ORG[nb].xn	R/W	Offset set in the zero offset [nb]; coarse part plus fine part. When writing this variable, the value is assigned to the coarse part deleting the fine part.
(V.)[ch].A.COARSEORGT[nb].xn	R/W	Offset set in the zero offset [nb]; coarse part.
(V.)[ch].A.FINEORGT[nb].xn	R/W	Offset set in the zero offset [nb]; fine part.

### Syntax of the variables.

- ch· Channel number.
- nb· Zero offset number.
- xn· Name, logic number or index of the axis.

V.A.ORG.Z	Z axis.
V.A.ADDORG.3	Axis with logic number ·3·.
V.[2].A.COARSEORG.3	Axis with index ·3· in the channel ·2·.
V.[2].A.FINEORG.3	Axis with index ·3· in the channel ·2·.
V.A.ORG[1].Z	Zero offset G54 (G159=1). Z axis.
V.A.ORG[1].Z	Zero offset G54 (G159=1). Z axis.
V.A.COARSEORGT[4].3	Zero offset G57 (G159=4). Axis with logic number ·3·.
V.[2].A.FINEORGT[9].3	Zero offset G159=9. Axis with index ·3· in the channel ·2·.

# 5.

ORIGIN SELECTION  
Zero offsets (G54-G59/G159)

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

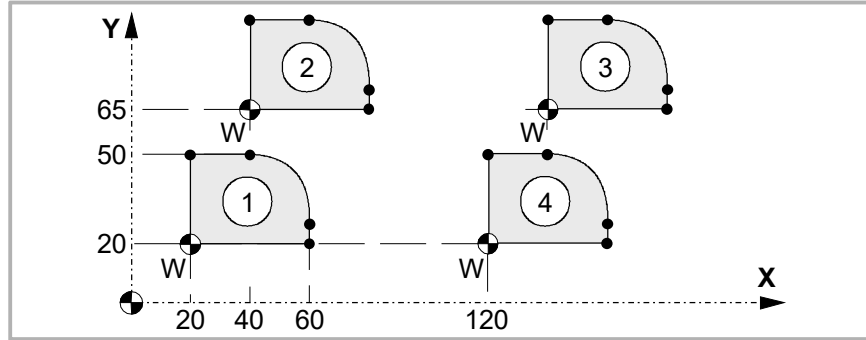
(REF: 2008)

### 5.5.2 Incremental zero offset (G158)

When applying an incremental zero offset, the CNC adds it to the absolute zero offset active at a time.

#### Programming

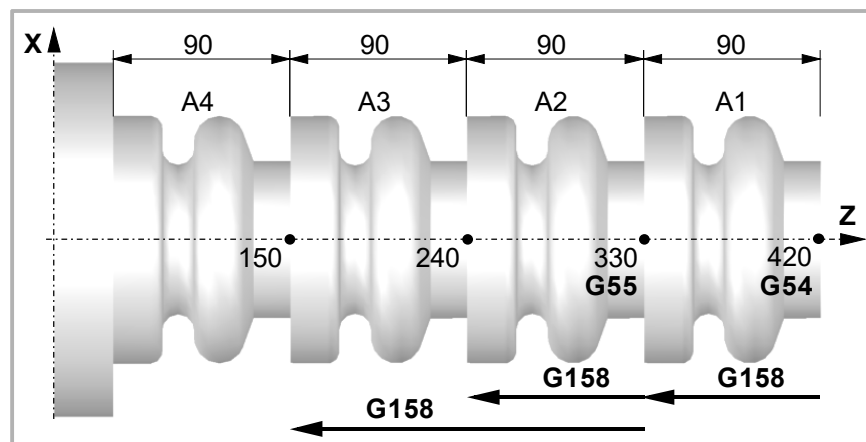
Incremental zero offset are defined by program using function G158 followed by the values of the zero offset to be applied on each axis. To cancel the incremental zero offset, program function G158 without axes in the block. To cancel the incremental zero offset only on particular axes, program a 0 (zero) incremental offset for each one of them.



	X	Y
G54 (G159=1)	20	20
G55 (G159=2)	120	20

```

N100 G54          (It applies the first zero offset)
...              (Machining of profile 1)
N200 G158 X20 Y45 (Apply incremental zero offset)
...              (Machining of profile 2)
N300 G55          (It applies the second zero offset. G158 stays active)
...              (Machining of profile 3)
N400 G158        (Cancel incremental zero offset. G55 stays active)
...              (Machining of profile 4)
    
```



	X	Z
G54 (G159=1)	0	420
G55 (G159=2)	0	330

```

N100 G54          (It applies the first absolute zero offset)
...              (Machining of profile A1)
N200 G158 Z-90   (Apply incremental zero offset)
    
```

5.

**ORIGIN SELECTION**  
 Zero offsets (G54-G59/G159)



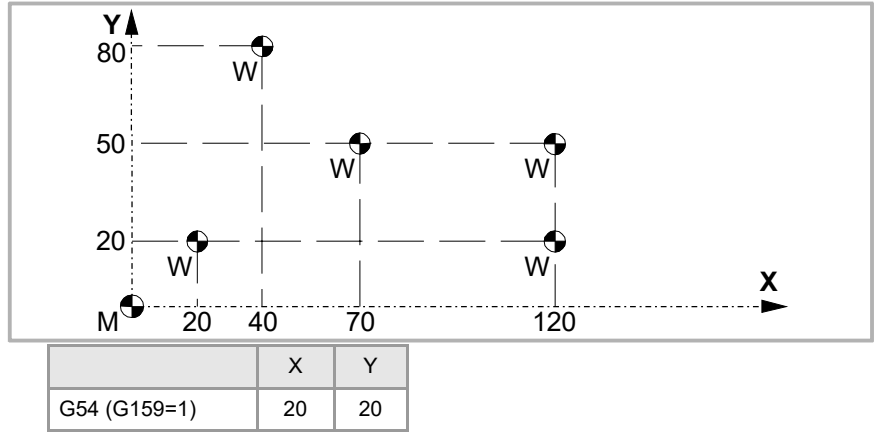
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

...	(Machining of profile A2)
N300 G55	(It applies the second absolute zero offset) (The incremental zero offset stays active)
...	(Machining of profile A3)
N200 G158 Z-180	(It applies the second incremental zero offset)
...	(Machining of profile A4)

Only one incremental zero may be active at a time for each axis; therefore, applying an incremental zero offset on an axis cancels the one that was active on that axis. The offsets on the rest of the axes are not affected.



N100 G54	(Apply absolute zero offset)
N200 G158 X20 Y60	(It applies the first incremental zero offset)
N300 G158 X50 Y30	(It applies the second incremental zero offset)
N400 G158 X100	(It applies the third incremental zero offset)
N500 G158 Y0	(It applies the fourth incremental zero offset)
N600 G158 X0	(Cancel incremental zero offset)

The incremental zero offset is not canceled after applying a new absolute zero offset (G54-G59 or G159).

**Considerations**

An incremental zero offset, by itself, does not cause any axis movement.  
When homing an axis in JOG mode, the incremental zero offset for that axis is canceled.

**Function properties**

Function G158 is modal.  
On power-up, the CNC assumes the incremental zero offset that was active when the CNC was turned off. On the other hand, the incremental zero offset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

**5.**

ORIGIN SELECTION  
Zero offsets (G54-G59/G159)

### 5.5.3 Excluding axes in the zero offset (G157)

Excluding axes allows to select on to which axes the next absolute zero offset will not be applied. After applying the zero offset, the programmed axis exclusion is canceled and it has to be programmed again in order to apply it again.

#### Activation

Axis exclusion must be programmed using function G157 followed by the axes and the value indicating whether that axis is excluded (<axis>=1) or not (<axis>=0).

The exclusion may also be activated by programming only the axes affected by the exclusion after function G157.

The exclusion and the zero offset may be programmed in the same block. In that case, the exclusion will be activated before applying the zero offset.

G55  
(It applies the second zero offset on all the axes)

G157 X Z  
(Activation of the exclusion on the X-Z axes)  
G57  
(It applies the fourth zero offset, except on the X-Z axes. Those axes keep the previous zero offset)  
...  
G159=8  
(It applies the eighth zero offset on all the axes)

G59 G157 Y  
(It applies the sixth zero offset, except on the Y axis. That axis keeps the previous zero offset)  
...  
G54  
(It applies the first zero offset on all the axes)

Excluding axes does not affect the active zero offsets. If an axis is excluded, when applying a new zero offset, the CNC maintains the one that was active for that axis.

#### Considerations

Excluding axes does not affect the coordinate preset or the incremental zero offsets which are always applied on to all the axes. Likewise, neither fixture offsets nor PLC offsets are affected.

#### Function properties

Function G157 is modal and it remains active until an absolute zero offset is applied.

On power-up or after an EMERGENCY, the CNC does not assume any axis exclusion.

5.

ORIGIN SELECTION  
Zero offsets (G54-G59/G159)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

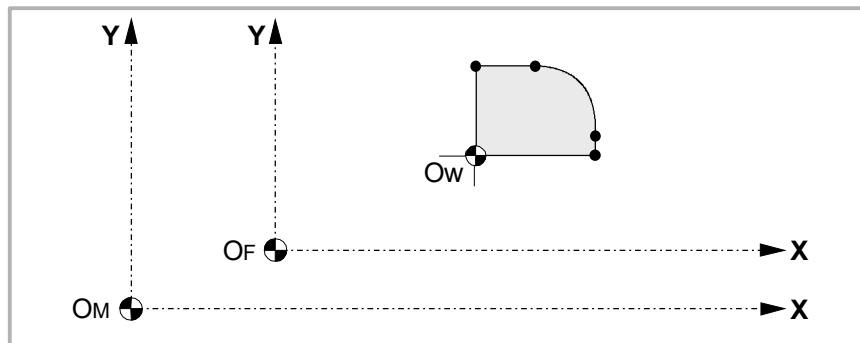


## 5.6 Zero offset cancellation (G53)

Executing function G53 cancels the active zero offset resulting either from a preset (G92) or from a zero offset, including the incremental offset and the defined axis exclusion. It also cancels the zero offset due to a probing operation.

Fixture offsets and PLC offsets are not affected by this function.

Contrary to the #MCS and #MCS ON/OFF instructions that always execute movements referred to machine zero, function G53 allows to execute movements referred to the fixture zero (if it is active).



N10 V.G.FIX=1	(Activate fixture offset. Program with respect to OF)
N20 G54	(Apply the zero offset. Program with respect to OW)
N30 #MCS X20 Y20	(Activate machine coordinate system. Program with respect to OM)
N40 G01 X60 Y0	(Program with respect to OW)
N50 G53	(Cancel zero offset G54. Program with respect to OF)

Function G53 may be programmed in any block of the program. When added to a block with path information, the offset or preset is canceled before executing the programmed movement.

### Considerations

Function G53, by itself, does not cause any axis movement.

### Function properties

Function G53 is modal and incompatible with function G92, zero offsets and probing.

# 5.

**ORIGIN SELECTION**  
Zero offset cancellation (G53)

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 5.7 Polar origin preset (G30)

Function G30 may be used to preset any point of the work plane as the new polar origin. If not selected, it assumes as polar origin the origin of the active reference system (part zero).

### Programming

The polar origin preset must be programmed alone in the block. The programming format is "G30 I J", where:

I, J They define the abscissa and ordinate of the new polar origin. They must be defined in absolute coordinates referred to part zero.

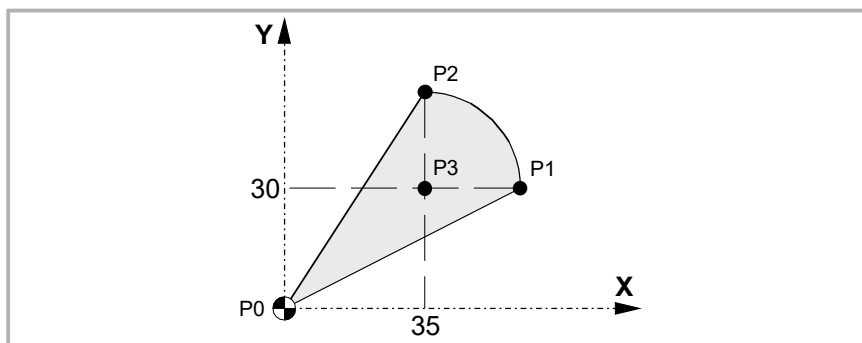
When programmed, both parameters must be programmed.

If not programmed, it will assume the current tool position as the polar origin.

Therefore, function G30 may be programmed as follows:

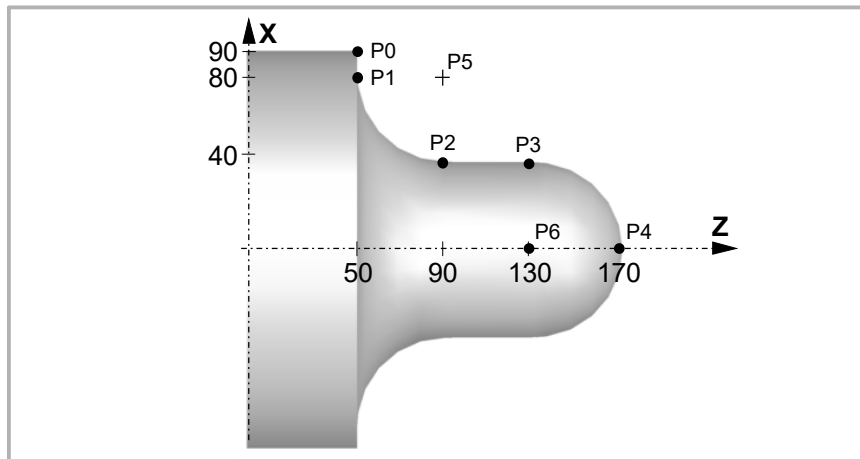
G30 I J It assumes as the new polar origin the point whose abscissa is "I" and ordinate "J" referred to part zero.

G30 The current tool position is assumed as the new polar origin.



Assuming that the starting point is X0 Y0, you get:

```
G30 I35 J30      (Preset P3 as the polar origin)
G90 G01 R25 Q0  (Point P1)
G03 Q90         (Point P2)
G01 X0 Y0      (Point P0)
M30
```



```

G18 G151 ; Main plane Z-X, and programming in diameters.
G90 X180 Z50 ; Point P0, programming in diameters.
G01 X160 ; Point P1, in a straight line (G01).
G30 I90 J160 ; Presets P5 as polar origin.
G03 Q270 ; Point P2, in arc (G03).
G01 Z130 ; Point P3, in a straight line (G01).
G30 I130 J0 ; Presets P6 as polar origin.
G02 Q0 ; Point P4, in arc (G02).
    
```

### Function properties

The G30 function is modal. The polar origin stays active until another value is preset or the work plane is changed. When changing the work plane, it assumes the part zero of that plane as the new polar origin.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes the currently selected part zero as the new polar origin.

# 5.

**ORIGIN SELECTION**  
Polar origin preset (G30)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 5.

## ORIGIN SELECTION

Polar origin preset (G30)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 6.1 Machining feedrate (F)

The machining feedrate may be selected by programmed using the "F" code which remains active until another value is programmed. The programming units depend on the active work mode (G93, G94 or G95) and the type of axis being moved (linear or rotary).

G94 - Feedrate in millimeters/minute (inches/minute).

G95 - Feedrate in millimeters/revolution (inches/revolution).

G93 - Machining time in seconds.

The programmed "F" is effective in linear (G01) or circular (G02, G03) interpolations. Movements in G00 (rapid traverse) are executed in rapid regardless of the programmed "F" value.

### **Movement without programmed feedrate.**

In principle, the CNC will show the corresponding error when programming a movement in G01/G02/G03 without having set a feedrate.

Optionally, the manufacturer may have configured the CNC to make the movements at the maximum machining feedrate, set by machine parameter MAXFEED.

### **Feedrate limitation.**

The manufacturer may have limited the maximum feedrate with machine parameter MAXFEED. When trying to exceed the maximum feedrate via part-program, via PLC or from the operator panel, the CNC limits the feedrate to the maximum value set without showing any error message or warning.

If this parameter is set to zero, the machining feedrate is not limited and the CNC assumes the one set for G00 as the maximum feedrate.

### **Variable to limit the feedrate via PLC.**

(V.) [n] . PLC . G00FEED is a variable that may be written from the PLC to set, at a particular moment and in real time, the maximum feedrate of the channel for any type of movement.

### **Feedrate regulation.**

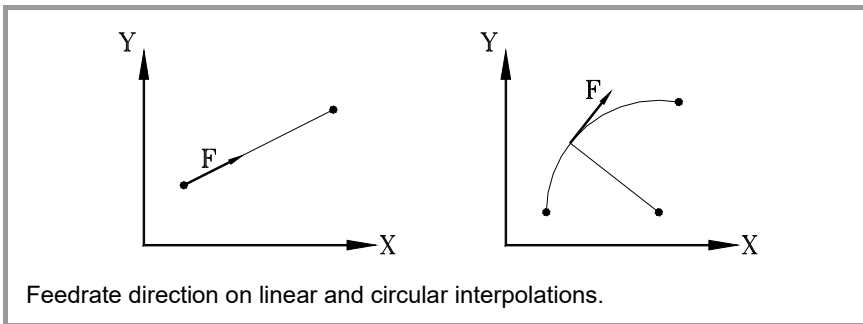
The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by program or by PLC. However, the maximum override is limited by the machine manufacturer [G.M.P. "MAXOVR"].

When making movements in G00 (rapid traverse), the feedrate override percentage will be fixed at 100% or it may be varied between 0% and 100% depending on how the machine manufacturer has set [G.M.P. "RAPIDOVR"].

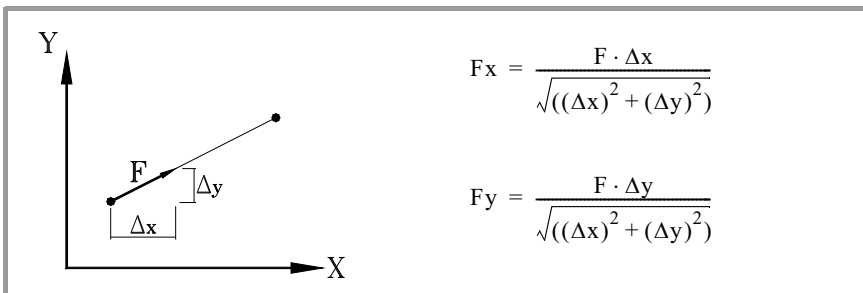
When carrying out threading operations, the feedrate percentage will be fixed at 100% of the programmed feedrate.

### Understanding how the CNC calculates the feedrate.

The feedrate is measured along the tool path, either along the straight line (linear interpolations) or along the tangent of the indicated arc (circular interpolations).



When only the main axes are involved in the interpolation, the relationship between the components of the feedrate on each axis and the programmed "F" is the same as between the displacement of each axis and the resulting programmed displacement.



When rotary axes are involved in the interpolations, the feedrate of these axes is calculated so the beginning and the end of their movement coincides with the beginning and the end of the main axes. If the feedrate calculated for the rotary axis is greater than the maximum allowed, the CNC will adapt the programmed "F" so the rotary axis can turn at its maximum speed.

# 6.

TECHNOLOGICAL FUNCTIONS  
Machining feedrate (F)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 6.2 Feedrate related functions

### 6.2.1 Feedrate programming units (G93/G94/G95)

The functions related to programming units permit selecting whether mm/minute (inches/minute) or mm/revolution (inches/rev.) are programmed or, instead, the time the axes will take to reach their target position.

#### Programming

The functions related to programming units are:

G94	Feedrate in millimeters/minute (inches/minute).
G95	Feedrate in millimeters/revolution (inches/revolution).
G93	Machining time in seconds.

These functions may be programmed anywhere in the program and they don't have to go alone in the block. If the moving axis is rotary, the programming units will be in degrees instead of millimeters or inches as follows:

	Linear axes	Rotary axes
G94	millimeters (inches)/minute	degrees/minute
G95	millimeters (inches)/revolution	degrees/revolution
G93	seconds	seconds

#### **G94** Feedrate in millimeters/minute (inches/minute).

After executing G94, the CNC interprets that the feedrates programmed with the "F" code are in millimeters/minute (inches/minute). If the moving axis is rotary, the CNC interprets that the programmed feedrate is in degrees/minute.

#### **G95** Feedrate in millimeters/revolution (inches/revolution)

After executing G95, the CNC interprets that the feedrates programmed with the "F" code are in mm/rev (inches/rev) of the master spindle of the channel. If the moving axis is rotary, the CNC interprets that the programmed feedrate is in degrees/revolution.

If the spindle does not have an encoder, the CNC will use the programmed theoretical rpm to calculate the feedrate. This function does not affect the movements in G00 which are always executed in millimeters/minute (inches/minute).

#### **G93** Machining time in seconds

After executing G93, the CNC interprets that the movements must be carried out in the time period (seconds) indicated by the "F" code.

This function does not affect the movements in G00 which are always executed in millimeters/minute (inches/minute).

#### Properties of the functions

Functions G93, G94 and G95 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G94 or G95 as set by the machine manufacturer [G.M.P. "IFEED"].

# 6.

## 6.2.2 Feedrate blend (G108/G109/G193)

With these functions, it is possible to blend the feedrate between consecutive blocks programmed with different feedrates.

### Programming

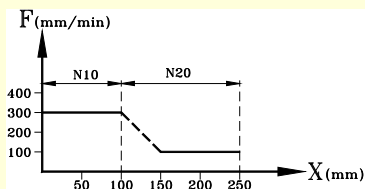
The functions related to feedrate blending are:

- G108 Feedrate blending at the beginning of the block.
- G109 Feedrate blending at the end of the block.
- G193 Interpolating the feedrate.

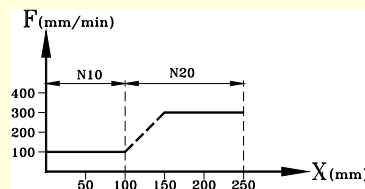
These functions may be programmed anywhere in the program and they don't have to go alone in the block.

#### G108 Feedrate blending at the beginning of the block

When G108 is active, the adaptation to the new feedrate (by accelerating or decelerating) takes place at the beginning of the next block and the current block ends at the programmed feedrate "F".



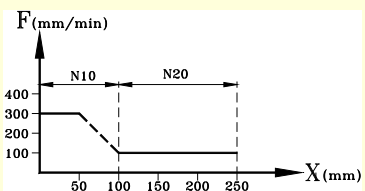
N10 G01 G108 X100 F300  
N20 X250 F100



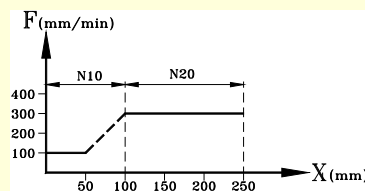
N10 G01 G108 X100 F100  
N20 X250 F300

#### G109 Feedrate blending at the end of the block

When programming G109 the adaptation to the new feedrate (by accelerating or decelerating) takes place at the end of the current block so the next block starts executing at its programmed feedrate "F".



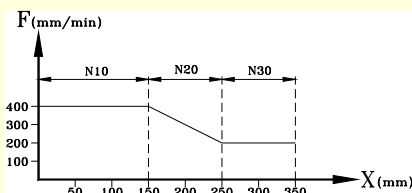
N10 G01 G109 X100 F300  
N20 X250 F100



N10 G01 G109 X100 F100  
N20 X250 F300

#### G193 Interpolating the feedrate

When programming G193, the adaptation to the new feedrate is interpolated linearly during the movement programmed in the block.



N10 G01 X150 F400  
N20 G193 X250 F200  
N30 X350

6.

TECHNOLOGICAL FUNCTIONS  
Feedrate related functions



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

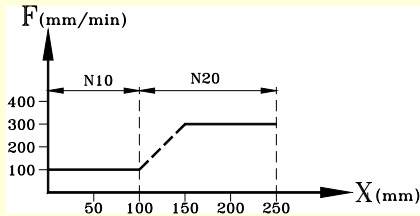


## Considerations

Adapting the feedrate (G108 and G109) is only available when the manufacturer has set the CNC to operate with either trapezoidal or square-sine (bell shaped) acceleration. Feedrate interpolation (G193) is only available when the manufacturer has set the CNC to operate with linear acceleration. The type of acceleration active at the CNC may be consulted in general machine parameter SLOPETYPE.

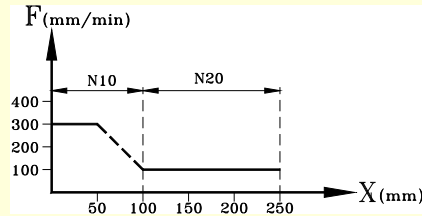
By default, the CNC applies the most restrictive feedrate adaptation in each situation without exceed the feedrate defined for each block. In other words, the CNC applies G108 to raise the feedrate and G109 to lower it.

Raise the feedrate, G108.



N10 G01 X100 F100  
N20 X250 F300

Lower the feedrate, G109.



N10 G01 X100 F300  
N20 X250 F100



*The CNC will not interpolate the feedrate for the mirror image with a reduced feedrate. In this situation, as the axis reaches F0 at the end of the block before G193 (mirror image point), it performs the next movement at the programmed feedrate along with to G193.*

N10 G0 X1100  
N20 G01 X1000 F120  
N30 G01 G193 X2000 F100 ; Mirror image of movement with a reduced feedrate.

## Properties of the functions

Functions G108, G109 and G193 are not modal.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC applies function G108 to accelerate and G109 to decelerate.

### 6.2.3 Constant feedrate mode (G197/G196)

With these functions, it is possible to choose whether the feedrate at the tool center is maintained constant while machining or the feedrate at the cutting edge so when working with tool radius compensation, the programmed "F" corresponds to the contact point between the part and the tool.

#### Programming

The functions related to the feedrate type are:

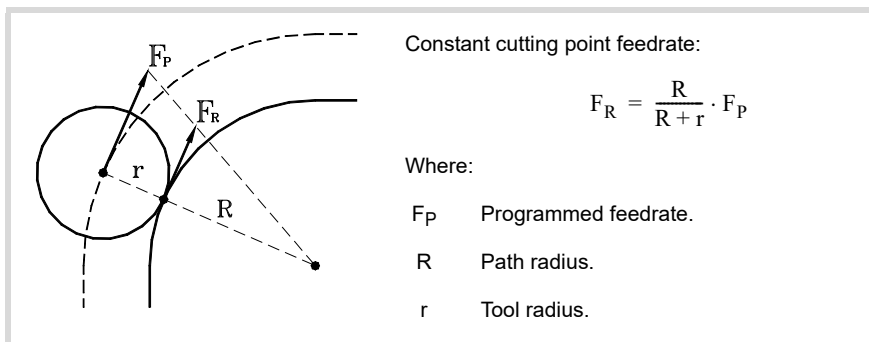
- G197      Constant tool center feedrate.
- G196      Constant cutting point feedrate.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

#### G197

##### Constant tool center feedrate

After executing G197, the CNC interprets that the programmed "F" corresponds to the tool center. This means that the feedrate at the cutting point increases on inside arcs and decreases on outside arcs.



#### G196

##### Constant cutting point feedrate

After executing G196, the CNC interprets that the programmed "F" corresponds to the contact point between the tool and the part. This results in a uniform part surface even on arcs.

##### Minimum radius for applying constant feedrate

Using the instruction "#TANGFEED RMIN [<radius>]" a minimum radius may be set so that constant tangential feedrate is only applied on those arcs whose radius is bigger than the minimum set. If it is not programmed or it is set to zero, the CNC will apply constant tangential feedrate on all the arcs.

The minimum radius is applied from the next motion block on and it keeps its value after executing G197.

#### Properties of the functions

Functions G197 and G196 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G197.

6.

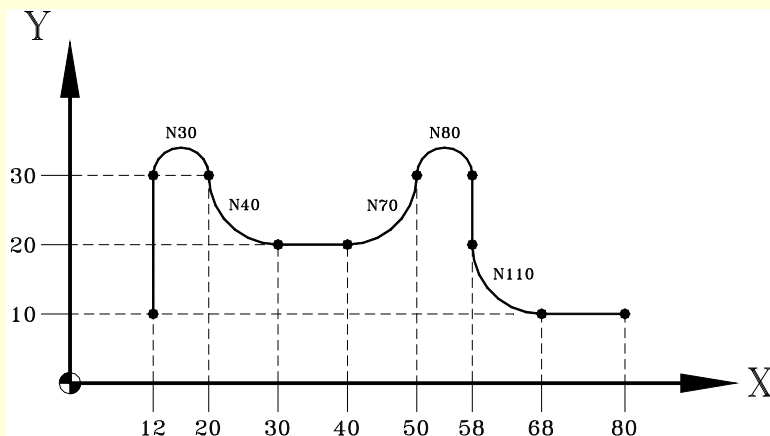
TECHNOLOGICAL FUNCTIONS  
Feedrate related functions



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



N10 G01 G196 G41 X12 Y10 F600	(Tool radius compensation and constant tangential feedrate)
N20 G01 X12 Y30	
N30 G02 X20 Y30 R4	(Constant tangential feedrate)
N40 G03 X30 Y20 R10	(Constant tangential feedrate)
N50 #TANGFEED RMIN [5]	(Minimum radius = 5)
N60 G01 X40 Y20	
N70 G03 X50 Y30 R10	(Constant tangential feedrate)
N80 G02 X58 Y30 R4	(No constant tangential feedrate. $R_{PROGRAMMED} < R_{MINIMUM}$ )
N90 G01 X58 Y20	
N100 #TANGFEED RMIN [15]	(Minimum radius = 15)
N110 G03 X68 Y10 R10	(No constant tangential feedrate. $R_{PROGRAMMED} < R_{MINIMUM}$ )
N120 G01 X80 Y10	
N130 G01 G40 X100	
N140 M30	

# 6.

## TECHNOLOGICAL FUNCTIONS Feedrate related functions

## 6.2.4 Cancellation of the % of feedrate override (G266)

### **G266** **Feedrate override at 100%**

This function sets the feedrate override at 100%, which can neither be changed by selector switch on the operator panel nor via PLC.

Function G266 only affects the block where it has been programmed, therefore, it only makes sense to add it to a block that defines a movement (motion block).

**6.****TECHNOLOGICAL FUNCTIONS**  
Feedrate related functions

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

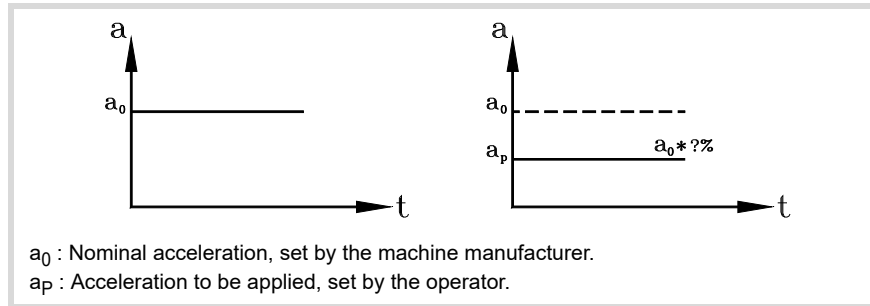
## 6.2.5 Acceleration control (G130/G131)

These functions allow to change the acceleration and deceleration of the axes and spindles.

### Programming

The functions related to acceleration control are:

- G130 Percentage of acceleration to be applied per axis or spindle.
- G131 Percentage of acceleration to be applied, global.



### G130 Percentage of acceleration to be applied per axis or spindle.

The percentage of acceleration to be applied per axis or spindle is set by G130 followed by the axes and spindles together with the percentage to be applied to each of them.

The acceleration values to be applied must be integers (not decimals).

```

...
G00 X0 Y0
G01 X100 Y100 F600
G130 X50 Y20          (Acceleration on the X axis = 50%)
                      (Acceleration on the Y axis = 20%)

G01 X0
G01 Y0
G131 100 X50 Y80     (Restore 100% of acceleration on all the axes)
                      (Movement to point X=50 Y=80)
...
    
```

### G131 Percentage of acceleration to be applied, global

The percentage of acceleration to be applied to all the axes and spindles is set by G131 followed by the new acceleration value to be applied.

The acceleration values to be applied must be integers (not decimals).

When added to a motion block, the new values will be assumed before executing the move.

### Considerations

The #SLOPE instruction determines the influence of the values defined with these values.

- In rapid positioning (G00)
- In the acceleration or deceleration stage.
- In the jerk of the acceleration or deceleration stages.

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

## Properties of the functions

Functions G130 and G131 are modal and incompatible with each other.

On power-up, after an M02, M30, EMERGENCY or a RESET, the CNC restores 100% of acceleration for all the axes and spindles.

6.

TECHNOLOGICAL FUNCTIONS

Feedrate related functions



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 6.2.6 Jerk control (G132/G133)

The jerk of axes and spindles may be modified with these functions.

### Programming

The functions associated with jerk control are:

G132	Percentage of jerk to be applied per axis or spindle.
G133	Percentage of jerk to be applied, global.

#### G132

##### Percentage of jerk to be applied per axis or spindle.

The percentage of jerk to be applied per axis or spindle is set by G132 followed by the axes and spindles together with the new jerk to be applied to each of them.

The jerk values to be applied must be integers (not decimals).

```
G00 X0 Y0
G01 X100 Y100 F600
G132 X20 Y50          (Jerk on the X axis = 20%)
                      (Jerk on the Y axis = 50%)

G01 X0
G01 Y0
G133 100 X50 Y80     (Restore 100% of jerk on all the axes. Movement to point X=50 Y=80)
```

#### G133

##### Percentage of jerk to be applied, global

The percentage of jerk to be applied to all the axes and spindles is set by G133 followed by the new jerk value to be applied.

The jerk values to be applied must be integers (not decimals).

When added to a motion block, the new jerk values will be assumed before executing the move.

### Considerations

The #SLOPE instruction determines whether the new percentages are to be applied or not on to rapid traverse movements (G00).

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

### Properties of the functions

Functions G132 and G133 are modal and incompatible with each other.

On power-up, after an M02, M30, EMERGENCY or a RESET, the CNC restores 100% of jerk for all the axes and spindles.

# 6.

## 6.2.7 Feed-Forward control (G134)

Feed-Forward control may be used to reduce the amount of following error (axis lag).

Feed-forward may be applied via machine parameters and via PLC as well as by program. The value defined by PLC will be the one with the highest priority and the one defined by the machine parameters will have the lowest priority.

### Programming

#### G134

#### Percentage of Feed-Forward to be applied

The Feed-Forward percentage is set by G134 followed by the axes and the new percentage of Feed-Forward to be applied to each axis.

The Feed-forward values to be applied may be defined with up to two decimals.

G134 X50.75 Y80 Z10	(Percentage of Feed-Forward to be applied:)
	(On the X axis = 50.75%)
	(On the Y axis = 80%)
	(On the Z axis = 10%)

### Considerations

The maximum Feed-Forward value to be applied is 120%.

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

The value defined with G134 prevails over those defined in the machine parameters, but not over the one defined by PLC.

### Properties of the functions

Function G134 is modal.

On power-up, after an M02 or M30, EMERGENCY or a RESET, the CNC restores the Feed-Forward set by the machine manufacturer for each axis.

### Variable to define the feed-forward from the PLC

The write variable (V.) A.PLCCFFGAIN.Xn may be used to set the feed-forward for each axis from the PLC. The value defined by this variable prevails over the ones defined by machine parameters or by program.

Setting this variable with a negative value cancels its effect (a zero value is also valid). This variable is initialized neither by a reset nor when validating the parameters.

# 6.



## 6.2.8 AC-Forward control (G135)

AC-Forward control may be used to improve system response in acceleration changes and reduce the amount of following error (axis lag) on the acceleration and deceleration stages.

AC-forward may be applied via machine parameters and via PLC as well as by program. The value defined by PLC will be the one with the highest priority and the one defined by the machine parameters will have the lowest priority.

### Programming

#### G135

##### Percentage of AC-Forward to be applied

The AC-Forward percentage is set by G135 followed by the axes and the new percentage of AC-Forward to be applied to each axis.

The AC-forward values to be applied may be defined with up to one decimal.

```
G135 X55.8 Y75 Z110      (Percentage of AC-Forward to be applied:)
                        (On the X axis = 55.8%)
                        (On the Y axis = 75%)
                        (On the Z axis = 110%)
```

### Considerations

The maximum AC-Forward value to be applied is 120%.

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

The value defined with G135 prevails over those defined in the machine parameters, but not over the one defined by PLC.

### Properties of the functions

The G135 function is modal.

On power-up, after an M02 or M30, EMERGENCY or a RESET, the CNC restores the AC-Forward set by the machine manufacturer for each axis.

### Variable to define the AC-forward from the PLC

The write variable (V.) `A.PLCACGAIN.Xn` may be used to set the AC-forward for each axis from the PLC. The value defined by this variable prevails over the ones defined by machine parameters or by program.

Setting this variable with a negative value cancels its effect (a zero value is also valid). This variable is initialized neither by a reset nor when validating the parameters.

# 6.

## 6.3 Spindle speed (S)

The spindle speed is selected by program using the spindle name followed by the desired speed. The speeds of all the spindles of the channel may be programmed in the same block. See chapter "7 *The spindle. Basic control.*".

```
S1000
S1=500
S1100 S1=2000 S4=2345
```

The programmed speed stays active until another value is programmed. On power-up, after executing an M02 or M30, and after an emergency or reset, the spindles assume a ·0· speed.

The speed may be programmed in rpm or in m/min (feet/min), depending on whether G197 or G196 is active. The default units are rpm.

### Spindle start and stop

Defining a speed does not imply starting the spindle. The startup is defined using the following auxiliary functions.

- M03 - Starts the spindle clockwise.
- M04 - Starts the spindle counterclockwise.
- M05 - Stops the spindle.

### Maximum speed

The maximum turning speed in each range (gear) is limited by the machine manufacturer. When programming a higher turning speed, the CNC limits its value to the maximum allowed by the active range (gear). The same thing occurs when trying to exceed the maximum limits using the "+" and "-" keys of the operator panel or doing it via PLC or by program.

### Speed override

The programmed "S" speed may be varied between 50% and 120% using the "+" and "-" of the operator panel or via PLC. However the maximum and minimum variation may be different depending on how the machine manufacturer has set [A.M.P. "MINOVR" y "MAXOVR"].

Likewise, the incremental step associated with the "+" and "-" keys of the Operator Panel to change the programmed spindle speed "S" will be 10; but this value may be different depending on the setting of axis machine parameter [P.P.E. "STEPOVR"].

During threading operations, the programmed speed cannot be overridden and it will be set at 100% of the programmed "S" speed.

6.

TECHNOLOGICAL FUNCTIONS  
Spindle speed (S)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 6.4 Tool number (T)

The "T" code identifies the tool to be selected. The tools may be in a magazine managed by the CNC or in a manual magazine (referred to as ground tools).

The programming format is T<0-4294967294> and it can be programmed using parameters and arithmetic expressions. In these cases, by default, the value calculated is rounded up to an integer. If the result is negative, the CNC will issue the pertinent error message.

### Definition

To select a tool, it must be previously defined. To do that, the CNC offers a table where the user may define the data for each tool. On the other hand, when having a magazine managed by the CNC, one must define the magazine position occupied by each tool. To do that, the CNC offers a table where the user may define the position of each tool. The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- Via program, using the associated variables (as described in the relevant chapter of this manual).

### Select a tool

The tool required for machining may be selected by program using the "T<n>" code where <n> is the tool number.

- On a lathe, the "T" code selects the tool in the tool holder.

#### Example on a lathe model.

```
N10 ...
N20 T1      (The CNC selects tool T1 in the turret)
N30 ...      (The CNC loads tool T1 into the spindle)
N40 ...
N50 T2      (The CNC selects tool T2 in the turret)
```

- On a milling machine, The "T" code only selects the tool. After selecting a tool, function M06 must be programmed to load it into the spindle. Loading/unloading is carried out depending on the subroutine associated with the M06 (if so defined by the machine manufacturer).

#### Example on a mill model.

```
N10 ...
N20 T1      (The CNC selects tool T1 in the tool magazine)
N30 M06     (The CNC loads tool T1 into the spindle)
N40 ...
N50 T2      (The CNC selects tool T2)
N60 ...
N70 ...
N80 ...
N90 M06     (The CNC loads tool T2 into the spindle)
N100 ...
N110 M30
```

# 6.

## Loading and unloading a tool in the magazine

To load the tools into the magazine, the magazine must be in load mode. To unload the tools from the magazine, the magazine must be in unload mode. The tools are loaded from ground to the magazine going through the spindle and are unloaded to ground going through the spindle.

The magazine's work mode is set by variable  $V.[n].TM.MZMODE$  where  $n$  is the channel number. Depending on the value of the variable, the manager will assume one of the following work modes.

Value	Meaning
0	Normal mode (by default and after Reset).
1	Magazine loading mode.
2	Magazine unloading mode.

Once the magazine is load or unload mode, the operation is carried out from the program using the code  $T_n$  where  $n$  is the tool number. Once the tools have been loaded or unloaded, the magazine must be set to normal mode (value of  $\cdot 0$ ).

```
V.[1].TM.MZMODE = 1
T1 M6
T2 M6
...
V.[1].TM.MZMODE = 0
```

### Loading a tool in a specific magazine position

Some tools, due to their characteristics (size, weight, etc.) must be placed in a specific magazine position.

The command  $POS_n$  defines the magazine position for the tool. It must always be programmed in the same block as  $T_n$ .

```
V.[1].TM.MZMODE = 1
T3 M6 POS24
    (Places tool 3 in magazine position 24)
...
V.[1].TM.MZMODE = 0
```

The magazine position can only be selected when the magazine is in load mode. Otherwise, it issues the relevant error message.

### Loading a tool in a system with several magazines

When using more than one tool magazine, one must indicate in which one of them the tool is to be loaded using the code  $MZ_n$ , where  $n$  indicates the magazine number. It must always be programmed in the same block as  $T_n$ .

```
T1 MZ1 M6
    (Place tool 1 in the first magazine)
T8 MZ2 POS17 M6
    (Place tool 8 in position 17 of the second magazine)
```

### Considerations. The tool and the M06 function.

The machine manufacturer may have associated a subroutine with the "T" code, that will be executed automatically when selecting the tool. If the M06 has been included in this subroutine, the tool will be loaded into the spindle when executing the "T" code.

6.

TECHNOLOGICAL FUNCTIONS  
Tool number (T)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Positioning a turret magazine.

The CNC allows positioning the turret in a particular position whether there is a tool in the indicated position or not. If the selected position contains a tool, the CNC assumes it as programmed tool; otherwise, the CNC assumes T0.

### Programming.

When programming this instruction, you must define the magazine number and the position to be selected on the turret. The new position of the turret may be defined either incrementally, defining the number of positions to rotate and the rotating direction, or absolutely, defining the position to be reached.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets

```
#ROTATEMZ{mz} P{pos}
```

```
#ROTATEMZ{mz} {±n}
```

{mz} Magazine number.

{pos} Absolute turret position.

{±n} Number of positions to rotate; the sign indicates the rotating direction, positive or negative. If only the sign is programmed, the turret rotates one position.

```
#ROTATEMZ1 P5
```

(Absolute position; select position 5).

```
#ROTATEMZ2 +3
```

(Incremental positioning; rotate the turret 3 positions in positive direction).

```
#ROTATEMZ1 -7
```

(Incremental positioning; rotate the turret 7 positions in negative direction).

```
#ROTATEMZ2 +
```

(Incremental positioning; rotate the turret 1 position in positive direction).

```
#ROTATEMZ1 -
```

(Incremental positioning; rotate the turret 1 position in negative direction).

# 6.

TECHNOLOGICAL FUNCTIONS

Tool number (T)

**FAGOR** 

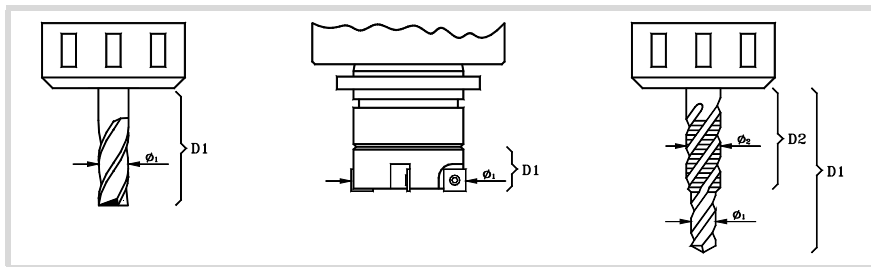
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 6.5 Tool offset number (D)

The tool offset contains the tool dimensions. Each tool may have several offsets associated with it in such a way that when using combined tools having parts with different dimensions, a different offset number will be used for each of those parts.



When activating a tool offset, the CNC assumes the tool dimensions defined for that offset, therefore when working with tool radius or length compensation, the CNC will apply those dimensions for compensating the path.

### Definition

To activate an offset, it must be previously defined. To do that, the CNC offers a portion of the tool table where the user may define several offsets. The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- Via program, using the associated variables (as described in the relevant chapter of this manual).

The offsets are only associated with the tool for which they have been defined. This means that when activating a tool offset, the offset of the corresponding active tool will be activated.

### Activation

Once the tool offsets have been defined in the table, they may be selected by program using the "D<n>" code where <n> is the number of the offset to be applied. The offset number may also be defined using a parameter or an arithmetic expression.

If no tool offset is programmed, the CNC assumes tool offset D1.

	N10 ...	
	N20 T7 D1	(Select tool T7 and tool offset D1)
	N30 M06	(Load tool T7 into the spindle)
	N40 F500 S1000 M03	
	N50 ...	(Operation 1)
	N60 D2	(Select tool offset D2 of T7)
	N70 F300 S800	
	N80 ...	(Operation 2)
	N90 ...	

Only one tool offset may be active at a time; therefore, activating a tool offset will cancel the previous one. Programming "D0" will cancel the active offset.

6.

TECHNOLOGICAL FUNCTIONS  
Tool offset number (D)



FAGOR AUTOMATION

CNC 8070

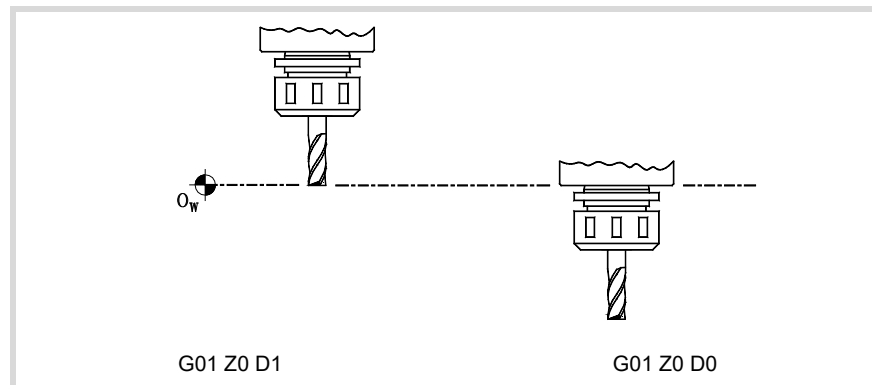
(REF: 2008)

N10 ...	
N20 T1 M06	(Select and load tool T1. Offset D1 is activated by default)
N30 F500 S1000 M03	
N40 ...	(Operation 1)
N50 T2	(Prepare tool T2)
N60 D2	(Select tool offset D2 for tool T1)
N70 F300 S800	
N80 ...	(Operation 2)
N90 M6	(loading tool T2 with its offset D1)
N100 F800 S1200 M03	
N110 ...	(Operation 3)
N120 ...	

### Considerations

Activating the tool offset also activates tool length compensation. This compensation is also activated after a tool change because it "D1" is assumed after the change (if another one has not been programmed).

Canceling the tool offset with "D0" also cancels tool length and radius compensation.



# 6.

**TECHNOLOGICAL FUNCTIONS**  
Tool offset number (D)

## 6.6 Auxiliary (miscellaneous) functions (M)

Auxiliary "M" functions are related to the overall CNC program execution and the control of the various devices of the machine such as spindle gear change, coolant, tool changes and so on.

### Programming

Up to 7 "M" functions may be programmed in a block. The programming format is M<0 - 65535>, and it can be programmed using parameters and arithmetic expressions. In these cases, by default, the value calculated is rounded up to an integer. If the result is negative, the CNC will issue the pertinent error message.

### Execution

Depending on how they have been set by the machine manufacturer ("M" function table):

- The "M" functions will be executed before or after the movement of the block where they were programmed.

When setting an "M" to be executed after the move of the block, depending on the active function G05 or G07:

- |     |  |
|-----|--|
| G05 | The "M" function is executed with the theoretical end of the movement (when the axes have not reached position). |
| G07 | The "M" function is executed with the real end of the movement (when the axes are already in position).          |

- The CNC will wait or not for the confirmation that the "M" function has been executed before resuming program execution. If it has to wait for confirmation, it will have to be received before or after executing the movement of the block where it has been programmed.
- The "M" functions that have not been set in the table will be executed before the movement of the block where they have been programmed and the CNC will wait for the "M-done" confirmation before executing the movement of the block.

Certain "M" functions have a particular internal meaning in the CNC. The section on ["6.6.1 List of "M" functions"](#) in this chapter shows a list of these functions with their internal meaning for the CNC.

### Associated subroutine

The "M" functions may have an associated subroutine that will be executed instead of the function.

If, within a subroutine associated with an "M" function, the same "M" function is programmed, this function will be executed, but not its associated subroutine.

6.

TECHNOLOGICAL FUNCTIONS  
Auxiliary (miscellaneous) functions (M)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 6.6.1 List of "M" functions

### Program Interruption (M00/M01)

#### M00

##### Program stop.

Function M00 interrupts the execution of the program. It does not stop the spindle or initialize the cutting conditions.

The [CYCLE START] key of the operator panel must be pressed again in order to resume program execution.

This function should be set in the "M" function table so it is executed at the end of the block where it is programmed.

#### M01

##### Conditional program stop.

When the external conditional stop switch is active (PLC signal "M01 STOP"), it interrupts program execution. It does not stop the spindle or initialize the cutting conditions.

The [CYCLE START] key of the operator panel must be pressed again in order to resume program execution.

This function should be set in the "M" function table so it is executed at the end of the block where it is programmed.

### Tool change (M06)

#### M06

##### Tool change.

The M06 function executes the tool change. The CNC will manage the tool change and update the table for the tool magazine.

This function should be set in the "M" table so it executes the subroutine corresponding to the tool changer installed on the machine.

6.

TECHNOLOGICAL FUNCTIONS  
Auxiliary (miscellaneous) functions (M)

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 6.7 Auxiliary functions (H)

Auxiliary "H" functions are used to send information out to the PLC. They differ from the "M" functions in that the "H" functions do not wait for confirmation that the function has been executed in order to go on executing the program.

### Programming

Up to 7 "H" functions may be programmed in the same block. The programming format is H<0 - 65535>, and it can be programmed using parameters and arithmetic expressions. In these cases, by default, the value calculated is rounded up to an integer. If the result is negative, the CNC will issue the pertinent error message.

### Execution

The auxiliary "H" functions are executed at the beginning of the block where they have been programmed.

**6.****TECHNOLOGICAL FUNCTIONS**  
Auxiliary functions (H)

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

The CNC can have up to four spindles distributed between the various channels of the system. A channel may have one, several or no spindles associated with it.

Each channel can only control its spindles; it is not possible to start up or stop the spindles of another channel directly. Indirectly, the CNC can control the spindles of another channel using the instruction #EXBLK.

## Multi-spindle channel

A multi-spindle channel is one that has two or more spindles. Via part-program or MDI, it is possible to indicate which spindle the commands are directed to; when not indicated, the commands are directed to the master spindle of the channel.

All the spindles of the channel may be running at the same time. Also, each of them may be in a different mode; turn in different directions, be in positioning mode, etc.

## Master spindle of the channel

The master spindle is the main spindle of the channel. In general, whenever a channel has a single spindle, it will be its master spindle. When a channel has several spindles, the CNC will choose the master spindle according to the established criteria. See "[7.1 The master spindle of the channel](#)" on page 124.

## 7.1 The master spindle of the channel

The master spindle is the main spindle of the channel. It is the spindle that receives the commands when no specific spindle is mentioned. In general, whenever a channel has a single spindle, it will be its master spindle.

### CNC criterion for selecting the master spindle after executing M02, M30 or after an emergency or a reset or restarting the CNC.

The selection of the master spindle in the channel depends on machine parameter MASTERSPDL. This parameter indicates whether the channel keeps the current master spindle or recovers its original master spindle after executing M02, M30 or after an emergency or reset or restarting the CNC.

MASTERSPDL	Meaning.
Temporary.	The channel recovers its original master spindle if it is free; otherwise, it selects as master the first spindle available in the original configuration.
Maintained.	The channel keeps the active master spindle.

When a channel does not keep its master spindle, on CNC power-up and after a reset, the channel assumes as master spindle the first spindle defined by the machine parameters of the channel (original master). If this spindle is parked or "handed out" to another channel, the channel assumes as master spindle the next spindle defined by the machine parameters and so on. If the channel does not have spindles of the original configuration (the one defined by the machine parameters) because they are parked or "handed out", it assumes as master spindle the first one of the current configuration that is not parked.

### Exchanging spindles between channels.

When spindles are being exchanged between channels, the behavior of this parameter also depends on parameter AXISEXCH that sets whether the channel change of a spindle is temporary or permanent. If the current master spindle of the channel is a spindle "loaned" by another channel and its permission to change channels is temporary (AXISEXCH = Temporary), the spindle returns to its original channel.

### Which one is the master spindle after executing M30?

When executing an M30, it follows the same criteria, but considering that after executing this function, the temporary spindle swaps are not undone; they are undone at the beginning of the next program. This means that the original master might not be available after executing an M30, but it will be at the beginning of the next program. In this situation, after an M30, the channel will temporarily assume a master spindle that will change at the beginning of the next program.

### Which one is the master spindle after modifying the configuration of the channel?.

If no master spindle is indicated, after parking or swapping spindles, it assumes one according to the following criteria. In general, whenever a channel has a single spindle, it will be its master spindle.

- If the whole system only has one spindle, it will be the master spindle of the current channel.
- If a spindle is added to a channel that does not have one, it will be the master spindle.
- If a channel releases its master spindle and it has only one spindle left, this one will be its new master spindle.
- If a channel having two spindles but no master spindle releases one of them, the remaining one will be its master spindle.
- At first, in a channel with several spindles, the master spindle will be the one configured by machine parameters.

7.

THE SPINDLE. BASIC CONTROL.  
The master spindle of the channel

- If two or more spindles remain in a channel and none of the previous rules can be applied, it applies the following criteria.

If any of the spindles is the original master, it is assumed as master spindle. If it is parked, it selects the next spindle from the original configuration (those defined by the machine parameters) and so on.

If no spindles from the original configuration are available in the channel, it assumes as master spindle the first one of its current configuration. If it is parked, it selects the next spindle and so on.

### **Which one is the master spindle after parking or unparking spindles?**

The same treatment is applied as when modifying the configuration of the channel.

7.

**THE SPINDLE. BASIC CONTROL.**

The master spindle of the channel

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 7.1.1 Manual selection of a master spindle

### Selecting a new master spindle

Whenever a channel has a single spindle, it will be its master spindle. When a channel has several spindles, the CNC will choose the master spindle according to the criteria described earlier. However, a different master spindle may be selected via MDI or via part-program using the #MASTER instruction.

#### Programming format.

```
#MASTER sp
```

sp                      Spindle name.

```
#MASTER S
#MASTER S2
```

### Canceling the master spindle

The master spindle may be selected at any time. If the master spindle changes channels, the channel will select a new master spindle according to the criterion described earlier.

On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC acts as defined by the OEM (parameter MASTERSPDL).

# 7.

**THE SPINDLE. BASIC CONTROL.**  
The master spindle of the channel



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 7.2 Spindle speed

The spindle speed is selected by program using the spindle name followed by the desired speed. The speeds of all the spindles of the channel may be programmed in the same block. It is not possible to program the speed of a spindle that is not in the channel.

The programmed speed stays active until another value is programmed. On power-up, after executing an M02 or M30, and after an emergency or reset, the spindles assume a ·0· speed.

### Programming format

The spindle may have any name in the S, S1...S9 range. There is no need to program the "=" sign for the "S" spindle.

Sn={vel}

S{vel}

Sn                    Spindle name.

S                     Spindle "S".

{vel}                Turning speed.

```
S1000
S1=500
S1100 S1=2000 S4=2345
```

The speed may be programmed in rpm or in m/min (feet/min), depending on whether G97 or G96 is active. The default units are rpm.

### Spindle start and stop

Defining a speed does not imply starting the spindle. The startup is defined using the following auxiliary functions. See ["7.3 Spindle start and stop"](#) on page 130.

M03 - Starts the spindle clockwise.

M04 - Starts the spindle counterclockwise.

M05 - Stops the spindle.

### Speed ranges (gears)

Each spindle may have up to 4 different ranges (gears). Each gear means a speed range for the CNC work in. The programmed speed must be within the active gear; otherwise, a gear change will be required. The CNC does not admit speeds higher than the one defined for the last gear.

The gear change may be automatic or manual. When the change is manual, the gear is selected with auxiliary functions M41 through M44. When the change is automatic, the CNC itself generates these functions according to the programmed speed. See ["7.4 Gear change."](#) on page 132.

7.

THE SPINDLE. BASIC CONTROL.  
Spindle speed

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 7.2.1 G192. Turning speed limitation

Function G192 limits the spindle turning speed in both work modes; G96 and G97. This function is especially useful when working at constant cutting speed while machining large parts or when doing spindle maintenance work.

If function G192 is not programmed, the turning speed is limited by machine parameter G00FEED of the gear.

### G192. Programming the turning speed limit

The turning speed limit is set by programming function G192 and then the maximum speed for each spindle. This function may be programmed while the spindle is running; in this case, the CNC will limit the speed to the new programmed value.

#### Programming format

The spindle may have any name in the S, S1...S9 range. There is no need to program the "=" sign for the "S" spindle.

G192 Sn={vel}

G192 S{vel}

{vel}                    Maximum spindle speed.

```
G192 S1000
G192 S1=500
```

The maximum turning speed is always set in RPM. It is possible to program using arithmetic parameters, variables or expressions.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Function G192 is modal.

The CNC cancels function G192 on power-up. After executing an M02 or M30, and after an emergency or a reset, the CNC maintains function G192.

7.

THE SPINDLE. BASIC CONTROL.

Spindle speed



## 7.2.2 Constant surface speed



*The following functions are oriented to lathe type machines. In order for Constant Surface Speed mode to be available, the machine manufacturer must have set one of the axis -face axis- (usually axis perpendicular to the shaft of the part).*

The functions related to spindle speed programming may be used to select either Constant Surface Speed mode or Constant turning speed mode. Constant Surface Speed is only available at the master spindle of the channel.

G96 - Constant surface speed.

G97 - Constant turning speed.

At constant surface speed, the CNC changes the spindle speed as the perpendicular axis moves in order to maintain the cutting speed constant between the tool and the part, thus optimizing the machining conditions. When working at constant surface speed, it is recommended to limit by program the maximum turning speed that the spindle can reach. See "7.2.1 G192. Turning speed limitation" on page 128.

### G96. Constant surface speed.

The G96 function only affects the master spindle of the channel.

After executing G96, the CNC interprets that the spindle speeds programmed for the master spindle of the channel are in meters/minute (feet/minute). This work mode is activated when programming a new speed while G96 is active.

This function may be programmed anywhere in the program and it doesn't have to go alone in the block. It is recommended to program the speed in the same block as the G96 function. The spindle gear (range) must be selected in the same block or in a previous one.

### G97. Constant turning speed

The G97 function affects all the spindles of the channel.

After executing G97, the CNC interprets that the spindle speeds programmed are in rpm and starts working at constant turning speed.

This function may be programmed anywhere in the program and it doesn't have to go alone in the block. It is recommended to program the speed in the same block as the G97 function; if not programmed, the CNC assumes as programmed speed the one the spindle is currently turning at. The gear may be selected at any time.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Functions G96 and G97 are modal and incompatible with each other. The CNC assumes function G97 on power-up or after an emergency. After executing an M02 or M30, and after a reset, the behavior of the function G96 depends on the parameter SPDLSTOP.

SPDLSTOP	Meaning.
Yes	After an M2, M30 or reset, the CNC cancels the function G96 (active G97).
No	After an M2, M30 or reset, the CNC keeps the function G96 active.

7.

THE SPINDLE. BASIC CONTROL.  
Spindle speed

**FAGOR**

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 7.3 Spindle start and stop

A speed must be set in order to start up a spindle. The spindle start-up and stop are defined using the following auxiliary functions.

M03 - Start the spindle clockwise.

M04 - Start the spindle counterclockwise.

M05 - Stop the spindle.

These functions are modal and incompatible with each other and with function M19.

### M03/M04. Start the spindle clockwise/counterclockwise.

Function M03 starts the spindle clockwise and function M04 counterclockwise. These functions should be set in the "M functions" table so they are executed at the end of the block where it is programmed.

These functions may be defined together with the programmed speed or in a separate block. If the block where they are programmed does not mention any spindle, they will be applied to the master spindle of the channel.

```
S1000 M3
  (The spindle "S" starts clockwise at 1000 rpm)
S1=500 M4
  (The spindle "S1" starts counterclockwise at 500 rpm)
M4
  (The master spindle starts counterclockwise)
```

If several spindles are programmed in a single block, functions M3 and M4 apply to all of them. To start the spindles in different directions, define next to each M function the spindle it is associated with, as follows.

M3.S / M4.S          M3 or M4 associated with the spindle S.

```
S1000 S2=456 M3
  (Spindle "S" turning at 1000 rpm and S2 at 456 rpm, both clockwise)
M3.S S1000 S2=456 M4.S2
  (The spindle "S" turns clockwise at 1000 rpm)
  (The spindle "S2" turns counterclockwise at 456 rpm)
```

### M05. Stop the spindle.

Function M05 stops the spindle.

To stop a spindle, define next to the M5 the spindle it is associated with, as follows. If it does not mention any spindle, it applies to the master spindle.

M5.S                  Function M5 associated with the spindle S.

```
S1000 S2=456 M5
  (Stops the master spindle)
M5.S M5.S2 S1=1000 M3.S1
  (Stops the spindles "S" and "S2")
  (Spindle "S1" turns clockwise)
```

### Turning direction preset in the tool table.

It is possible to set a preset turning direction for each tool. The value is defined in tool table.

When assigning a turning direction in the table, the CNC will verify, during execution, that the turning direction in the table is the same as the one programmed (M03/M04). If the two directions are not the same, the CNC will display the corresponding error message. The CNC verifies this every time an M03, M04 or M06 is programmed.

7.

THE SPINDLE. BASIC CONTROL.

Spindle start and stop



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### Knowing which is the preset turning direction.

The turning direction preset for each tool may be consulted in the tool table; the one for the active tool can also be consulted with a variable.

(V.)G.SPDLTURDIR

This variable returns the preset turning speed of the active tool. ·0· value if there is no preset turning direction, ·1· if it is M03, and ·2· if it is M04.

### Canceling the preset turning direction temporarily.

The preset turning speed of the active tool may be temporarily canceled from the part-program. This is done by setting the variable V.G.SPDLTURDIR = 0.

When changing a tool, this variable will take the corresponding value according to what has been set in the tool table.

7.

THE SPINDLE. BASIC CONTROL.

Spindle start and stop

**FAGOR** 

FAGOR AUTOMATION

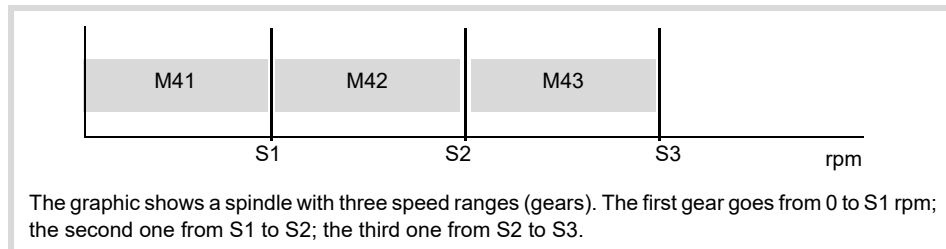
CNC 8070

(REF: 2008)

## 7.4 Gear change.

Each spindle may have up to 4 different ranges (gears). Each gear means a speed range for the CNC work in. The programmed speed must be within the active gear; otherwise, a gear change will be required.

The gear change may be automatic or manual. When the change is manual, the gear is selected with auxiliary functions M41 (gear 1) through M44 (gear 4). When the change is automatic, the CNC itself generates these functions according to the programmed speed.



The configuration of the speed ranges or gears (automatic or manual change, maximum speed for each gear, etc.) is defined by the machine manufacturer. See ["How to know the gear configuration of a spindle."](#) on page 133.

### Manual gear change

When the change is manual, the gear is selected with auxiliary functions M41 through M44.

M41 - Selects gear ·1·.

M42 - Selects gear ·2·.

M43 - Selects gear ·3·.

M44 - Selects gear ·4·.

These functions may be defined together with the programmed spindles or in a separate block. If the block where they are programmed does not mention any spindle, they will be applied to the master spindle of the channel.

```
S1000 M41
S1=500 M42
M44
```

If several spindles are programmed in a single block, the functions apply to all of them. To apply different gears to the spindles, define next to each M function the spindle it is associated with, as follows.

M41.S                    Function M41 associated with the spindle S.

```
S1000 S2=456 M41
(Gear 1 for spindle "S" and S2)
M41.S M42.S3
(Gear 1 for spindle "S")
(Gear 2 for spindle "S3")
```

### Influence of the reset, turning the CNC off and of the M30.

Gears (speed ranges) are modal. On power-up, the CNC assumes the gear defined by the machine manufacturer. After executing an M02 or M30 and after an emergency or a reset, the CNC maintains the active gear.

### Knowing which is the active gear.

The window for the M functions in the jog and automatic modes shows which is the active gear; if none is shown, it means that gear ·1· is the active one.

The active gear may be consulted using the following variable.

7.

THE SPINDLE. BASIC CONTROL. Gear change.

(V.)[n].G.MS[i]

Variable that can only be read from the PRG and PLC.

The variable indicates the status of the auxiliary Mi function. This variable returns a ·1· if it is active and a ·0· if not.

## Gear change on Sercos spindles.

When using Sercos spindles, functions M41-M44 also involve changing the drive's velocity set.

## How to know the gear configuration of a spindle.

Both the type of gear change (automatic or manual) and the maximum speed for each gear are defined by the machine manufacturer. The configuration may be checked directly in the machine parameter table or using the following variables.

### How to know whether the spindle has an automatic tool changer or not.

(V.)SP.AUTOGEAR.Sn

Variable that can only be read from the PRG and PLC.

This variable indicates whether the spindle Sn has an automatic gear change or not. This variable returns a ·1· if the gear changer is automatic and a ·0· if it is manual.

### Number of gears available

(V.)SP.NPARSETS.Sn

Variable that can only be read from the PRG and PLC.

This variable indicates the number of gears defined for the spindle Sn.

### Maximum speed for each gear

(V.)SP.G00FEED[g].Sn

Variable that can only be read from the PRG and PLC.

This variable indicates the maximum speed of the spindle Sn in the gear g.

### Gear active by default.

(V.)SP.DEFAULTSET.Sn

Variable that can only be read from the PRG and PLC.

This variable indicates which is the gear assumed by the CNC on power-up for the spindle Sn.

7.

THE SPINDLE. BASIC CONTROL.  
Gear change.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 7.5 Spindle orientation.



This work mode is only available on machines having a rotary encoder installed on the spindle.

The spindle orientation is defined with function M19. This function stops the spindle and it positions it at an angle defined by parameter "S". See *"How is positioning carried out"* on page 135.

After executing function M19, the spindle no longer works in speed mode and it switches to positioning mode. This mode stays active until the spindle is started up again in speed mode with M3/M4.

### Programming a spindle orientation

Every spindle positioning move requires an M19 and the positioning angle. If no angle is defined, the CNC orients the master spindle at 0°.

Even if function M19 is active, if an "S" value is defined without an M19, the CNC assumes it as the new turning speed for the next time the spindle is turned on in speed mode using functions M03/M04.

#### Programming format (1).

When executing function M19, the CNC interprets that the value entered with the "Sn" code indicates the angular position of the spindle. If several spindles are programmed in a single block, function M19 applies to all of them.

M19 S{pos}

S{pos}                      Spindle to be oriented and positioning angle.  
The angle is defined in degrees.

```
M19 S0
  (Positioning of spindle S at 0°)
M19 S2=120.78
  (Positioning of spindle S2 at 120.78°)
M19 S1=10 S2=34
  (Positioning of spindle S1 at 10° and S2 at 34°)
```

This angular position is programmed in degrees and it is always assumed in absolute coordinates, thus not being affected by functions G90/G91. To do the positioning, the CNC calculates the module (between 0 and 360°) of the programmed value.

#### Programming format (2). Positioning of the spindle at 0°

To orient the spindle to the ·0· position, it may also be programmed by defining, next to the M19, the spindle to be oriented. If no spindle is defined, the CNC assumes that the master spindle is the one to be oriented.

M19.S

S                              Spindle to be oriented at 0°.

```
M19.S4
  (Positioning of spindle S4 at 0°)
M19
  (Positioning of the master spindle at 0°)
```

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

M19 is modal and incompatible with M03, M04 and M05.

On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC sets the spindle in speed mode with function M05.

7.

THE SPINDLE. BASIC CONTROL.  
Spindle orientation.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## How is positioning carried out

When executing function M19, the CNC behaves as follows.

- 1 The CNC stops the spindle (if it was turning).
- 2 The spindle no longer works in speed mode and it switches to positioning mode.
- 3 If it is the first time the M19 is executed, the CNC homes the spindle (home search).
- 4 It positions the spindle at 0° or at the angular position defined by the "S" code (if it has been programmed). To do that, it will calculate the module (between 0 and 360°) of the programmed value and the spindle will reach that position.

```
N10 G97 S2500 M03
    (The spindle turns at 2500 RPM)
N20 M19 S50
    (Spindle in positioning mode. The spindle orients at 50°)
N30 M19 S150
    (Positioning at 150°)
N40 S1000
    (New spindle speed. The spindle stays in positioning mode)
N50 M19 S-100
    (Positioning at -100°)
N60 M03
    (Spindle controlled in speed. The spindle turns at 1000 RPM)
N70 M30
```

### It executes the M19 for the first time

When executing the M19 function for the first time, it homes the spindle. The M19 functions programmed afterwards only orient the spindle. To home the spindle again, use function G74.

7.

THE SPINDLE. BASIC CONTROL.  
Spindle orientation.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 7.5.1 The turning direction for spindle orientation

The turning direction for positioning may be set with function M19; if not defined, the CNC applies a turning direction by default. Each spindle may have a different turning direction by default.

### Turning direction by default

If no turning direction has been defined, the CNC behaves as follows. If when executing function M19, there was an M3 or M4 active, even if the speed is zero, this function will set the spindle orienting direction. If no M3 or M4 is active, the turning direction is set depending on machine parameter `SHORTESTWAY`.

- If it is a `SHORTESTWAY` spindle, it positions via the shortest way.
- If it is not a `SHORTESTWAY` type spindle, it positions in the same direction as the last spindle movement.

### Turning direction defined set by the user.

The positioning direction programmed next to the M19 is applied to all the spindles programmed in the block. If the turning direction has not been programmed, each spindle will turn in the direction set earlier; if none has been defined, it will assume the turning direction by default.

The programmed turning direction is maintained until another one is programmed.

#### Programming format (1). Turning direction for all the spindles programmed.

M19.POS S{pos}

M19.NEG S{pos}

POS                    Positioning in the positive direction

NEG                    Positioning in the negative direction

S{pos}                Spindle to be oriented and positioning angle.

M19.NEG S120 S1=50

(The negative direction is applied to spindle "S" and "S1")

M19.POS S120 S1=50

(The positive direction is applied to spindle "S" and "S1")

If no spindle is defined, the CNC orients the master spindle at 0° in the indicated direction.

When programming the orienting direction for a `SHORTESTWAY` type spindle, the programmed direction will be ignored.

#### Programming format (2). Turning direction for one spindle only.

Since several spindles may be programmed in a block, it is possible to apply the turning direction just to one of them. The rest of the spindles will turn in their active direction.

M19.POS.S S{pos} S{pos}

M19.NEG.S S{pos} S{pos}

POS.S                Spindle that orients in the positive direction.

NEG.S                Spindle that orients in the negative direction.

S{pos}                Spindle to be oriented and positioning angle.

M19.NEG.S1 S1=100 S34.75

(The negative direction is applied to spindle "S1")

7.

THE SPINDLE. BASIC CONTROL.  
Spindle orientation.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### How to know the type of spindle.

The type of spindle may be checked directly in the machine parameter table or using the following variables.

(V.)SP.SHORTESTWAY.Sn

Variable that can only be read from the PRG and PLC.

This variable indicates whether the spindle Sn orients via the shortest way. This variable returns a ·1· if so.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC cancels the turning direction set by the user.

7.

THE SPINDLE. BASIC CONTROL.

Spindle orientation.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 7.5.2 M19 function with an associated subroutine.

Function M19 may have a subroutine associated with them that the CNC executes instead of the function. If, within a subroutine associated with an "M" function, the same "M" function is programmed, The CNC will executed it; but not its associated subroutine.

Although the function may affect more than one spindle in the same block, the CNC only executes the subroutine once. The following behavior is applicable to all positioning movements programmed in the block.

When programming function M19 and a positioning (M19 S), the CNC executes the subroutine associated with the function and ignores the positioning movement. The CNC executes the positioning when executing the M19 function from the subroutine.

- If inside the subroutine, the M19 function is not accompanied by a positioning move (S), the CNC executes the positioning programmed in the calling block.
- If inside the subroutine, the M19 function is accompanied by a positioning move (S), the CNC executes this positioning.

The same criterion applies to the moving direction. If the turning direction is programmed with the M19 function that calls the subroutine, the turning direction is applied in the M19 programmed inside the subroutine if it another one has not been specified.

**7.**

**THE SPINDLE. BASIC CONTROL.**  
Spindle orientation.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 7.5.3 Positioning speed

It is possible to set the spindle positioning (orienting) speed; if it is not set, the CNC assumes the one set by machine parameter `REFEED1` as the positioning speed. Each spindle may have a different positioning speed.

#### Programming format.

The positioning speed is set as follows.

S.POS={vel}

S                      Spindle name.

{vel}                 Positioning speed.

```
M19 S.POS=120 S1.POS=50
      (Positioning of spindle "S" at 120 rpm and S1 at 50 rpm)
```

The positioning speed is given in rpm.

#### Knowing the active positioning speed.

The active positioning speed for the CNC may be consulted using the following variable.

(V.)SP.SPOS.Sn

Variable that can only be read from the PRG and PLC.

This variable indicates the active positioning speed for the spindle Sn.

7.

THE SPINDLE. BASIC CONTROL.  
Spindle orientation.

## 7.6 M functions with an associated subroutine.

M3, M4, M5, M19 and M41 to M44 may have a subroutine associated with them that the CNC executes instead of the function. Although a function may affect more than one spindle in the same block, the CNC only executes the subroutine once per block.

If, within a subroutine associated with an "M" function, the same "M" function is programmed, The CNC will executed it; but not its associated subroutine. When a spindle M function is programmed inside the subroutine, it will be applied to the spindles programmed in the subroutine block itself. If the subroutine block does not indicate which spindles the function is for, the CNC will assume that it is for the spindles programmed in the subroutine calling block.

The CNC relates the functions with the spindles according to the following criterion, whether it is in the calling block or inside the subroutine.

- If the M function is assigned to a spindle (for example, M3, S), the CNC only applies the function to the indicated spindle.
- If functions M3 and M4 are not assigned to any spindle, the CNC applies them to all the spindles whose speed has been programmed in the block and have not been assigned to another M function. If there are no spindles with programmed speed, the CNC applies them to the master spindle.
- If function M19 is not assigned to any spindle, the CNC applies them to all the spindles whose speed has been programmed in the block and have not been assigned to another M function.
- If functions M5 and M41 to M44 are not assigned to any spindle, the CNC applies them to the master spindle.

Inside the subroutine, the CNC applies this criterion to all the programmed spindle M functions, not only to the M functions that correspond to the calling block.

7.

**THE SPINDLE. BASIC CONTROL.**  
M functions with an associated subroutine.



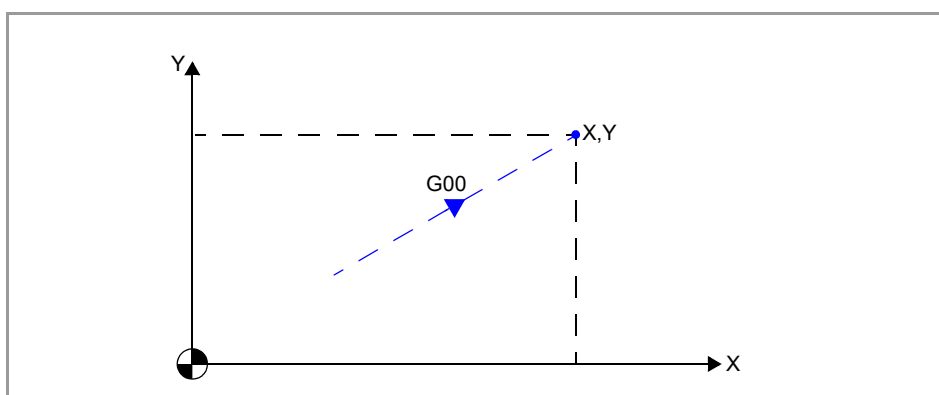
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 8.1 Rapid traverse (G00).

Function G00 executes a rapid positioning, according to a straight line and to the rapid traverse determined by the OEM, from the current position to the programmed point. Regardless of the number of axes involved, the resulting path is always a straight line. If auxiliary or rotary axes are programmed in the linear interpolation block, the CNC will calculate the feedrate for those axes so their movement begins and ends simultaneously with the main axes.



### Programming.

The G00 function can be modal or non-modal, depending on how it has been configured by the OEM (parameter G0MODAL).

- When function G00 is modal, after it has been programmed, it remains active until an incompatible function is programmed (G01, G02, G03, G33 or G63). The function G00 may be programmed alone in the block or it may be added to a movement block.
- If the G00 function is non-modal, it must be programmed for each rapid traverse block; if it is not programmed, the CNC uses G01.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G00 <X..C{position}>

X..C{position}      Optional. End point of the movement.  
Units: Millimeters, inches or degrees.

G00  
(Activate function G00 without moving).  
G00 X50.87 Y38.45  
(Movement in cartesian coordinates).  
G00 R50.23 Q45  
(Movement in polar coordinates).

### End point of the movement.

- For cartesian coordinates, defining the coordinates of the end point (X..C) on the various axes. All the axes need not be programmed, only the ones to move.

- For polar coordinates, define the radius (R) and the angle (Q) of the end point relative to the polar origin. The "R" radius will be the distance between the polar origin and the point. The "Q" angle will be formed by the abscissa axis and the line joining the polar origin with the point. If the angle or the radius is not programmed, it keeps the value programmed for the last move.

## Considerations.

### Feedrate behavior.

- A movement in G00 temporarily cancels the programmed "F" feedrate and the CNC executes the movement based according to the rapid traverse specified by the OEM (parameter G00FEED). The CNC recovers the "F" feedrate when programming a movement function G01, G02, G03, etc.
- When several axes are involved, the resulting feedrate is calculated so at least one of the axis moves at its maximum speed.
- When defining an "F" value and G00 in the same block, the CNC will store the value assigned to "F" and it will apply it next time a G01, G02 or G03 type function is programmed.

### Feedrate override.

The override percentage is set at 100% or it may be varied between 0% and 100% using the switch on the operator panel, depending on how it was set by the OEM (parameter RAPIDOVR).

### Canned cycles.

Within the range of influence of a canned cycle or modal subroute (#MCALL), the last programmed G will be maintain active, G0 or G1, meaning G0 remains as modal.

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Function G00 may be programmed as G0.

The G00 function can be modal or non-modal, depending on how it has been configured by the OEM (parameter G0MODAL). Function G00 is modal and incompatible with G01, G02, G03, G33 and G63. For the following program block, if there is no programmed movement function in a non-modal G00 function (G0, G1, G2, G3, G33 or G63), the CNC uses G1.

On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G00 or G01 as set by the OEM (parameter IMOVE). If the CNC assumes the function G00, and this function is defined as non-modal (parameter G0MODAL), after programming G1, G2 or G3, the CNC assumes G1 as a modal function.

8.

**PATH CONTROL:**  
Rapid traverse (G00).



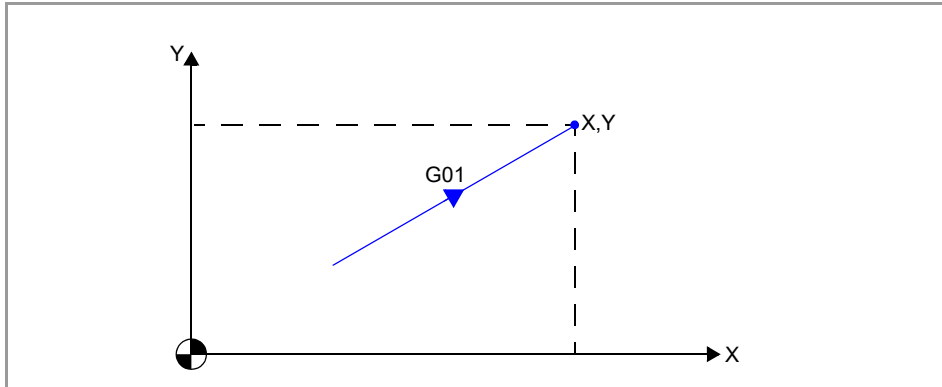
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 8.2 Linear interpolation (G01).

Function G01 activates the linear movement, according to the active "F" feedrate, for the following programmed movements. If auxiliary or rotary axes are programmed in the linear interpolation block, the CNC will calculate the feedrate for those axes so their movement begins and ends simultaneously with the main axes.



**PATH CONTROL:**  
Linear interpolation (G01).

### Programming.

The function G01 may be programmed alone in the block or it may be added to a movement block. The function G01 is modal; after it has been programmed, it remains active until an incompatible function is programmed (G00, G02, G03, G33 or G63).

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G01 <X..C{position}> <F{feedrate}>

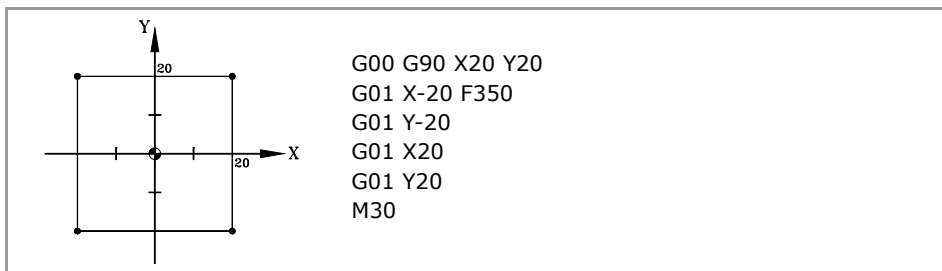
X..C{position}      Optional. End point of the movement.  
Units: Millimeters, inches or degrees.

F{feedrate}      Optional. Feedrate.  
Units: Units are dependent on the active function.  
- If G93, seconds.  
- If G94, Millimeters/minute, inches/minute or degrees/minute.  
- If G95, Millimeters/minute, inches/minute or degrees/turn.

G01  
(Activate function G01 without moving).  
G01 X600 Y400 F150  
(Movement in cartesian coordinates, using feedrate programming).  
G01 R600 Q20 F200  
(Movement in polar coordinates, using feedrate programming).

### End point of the movement.

- For cartesian coordinates, defining the coordinates of the end point (X..C) on the various axes. All the axes need not be programmed, only the ones to move.

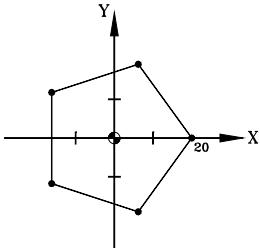


FAGOR AUTOMATION

CNC 8070

(REF: 2008)

- For polar coordinates, define the radius (R) and the angle (Q) of the end point relative to the polar origin. The "R" radius will be the distance between the polar origin and the point. The "Q" angle will be formed by the abscissa axis and the line joining the polar origin with the point. If the angle or the radius is not programmed, it keeps the value programmed for the last move.



```
G00 G90 X20 Y0
G01 R20 Q72 F350
G01 Q144
G01 Q216
G01 Q288
G01 Q360
M30
```

**Feedrate.**

The programmed feedrate "F" stays active until a new value is programmed, thus not being necessary to program it in every block.

**Considerations for the feedrate.**

- When several axes are involved, the CNC calculates the feedrate for each axis so the resulting path is executed at the programmed feedrate "F" .
- The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by the program or by the PLC. However, the maximum override is limited by the OEM (parameter MAXOVR).
- The behavior of the auxiliary axes is determined by general machine parameter FEEDND.

Parameter. FEEDND	Meaning.
Yes	The programmed feedrate will be the result of composing the movements onto all the axes of the channel. (main and auxiliary). No axis will surpass the programmed feedrate.
No	If a movement has been programmed on any of the main axes, the programmed feedrate will be the result of composing the movement only onto these axes. The rest of the axes move at their corresponding feedrate to end the movement of them all at the same time. Auxiliary axes can exceed the programmed feedrate, but they may not exceed their maximum working feed rate (parameter MAXFEED). If an axis were to exceed the MAXFEED, the programmed feedrate of the main axes would be limited by the CNC.  If none of the main axes are programmed, the programmed feedrate will be reached on the axis moving the farthest so they can all reach their destination at the same time.

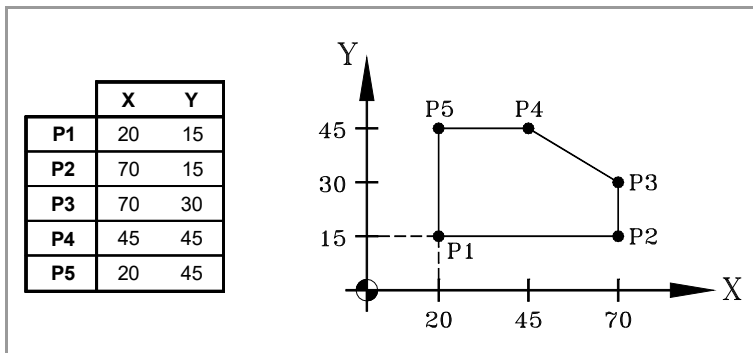
**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

- Function G01 may also be programmed as G1.
- Function G01 is modal and incompatible with G00, G02, G03, G33 and G63.
- On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G00 or G01 as set by the OEM (parameter IMOVE). If the CNC assumes the function G00, and this function is defined as non-modal (parameter G0MODAL), after programming G1, G2 or G3, the CNC assumes G1 as a modal function.

8.  
**PATH CONTROL.**  
 Linear interpolation (G01).



**Programming example (M model).  
Absolute and incremental cartesian coordinates.**



**Absolute coordinates.**

```
N10 G00 G90 X20 Y15
N20 G01 X70 Y15 F450
N30 Y30
N40 X45 Y45
N50 X20
N60 Y15
N70 G00 X0 Y0
N80 M30
```

**Incremental coordinates.**

```
N10 G00 G90 X20 Y15
N20 G01 G91 X50 Y0 F450
N30 Y15
N40 X-25 Y15
N50 X-25
N60 Y-30
N70 G00 G90 X0 Y0
N80 M30
```

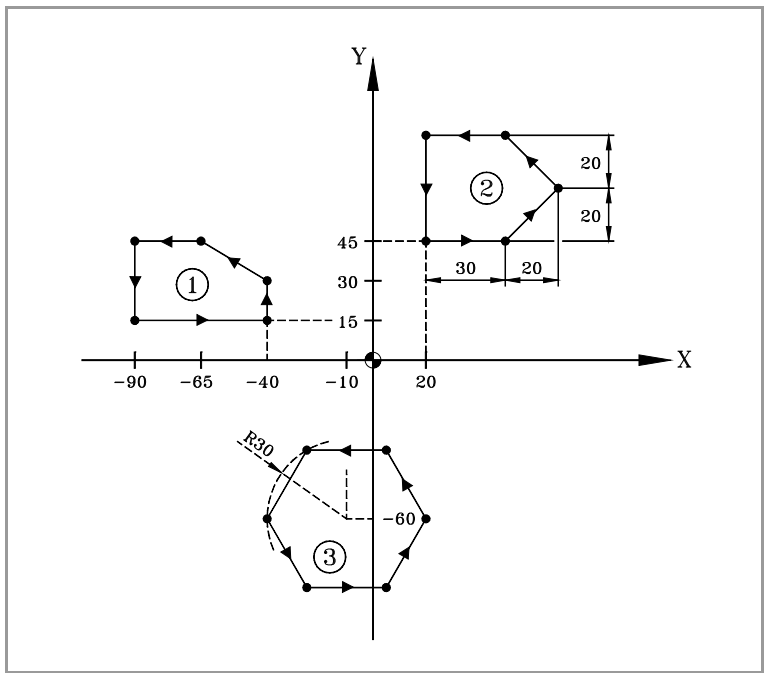


**PATH CONTROL:**  
Linear interpolation (G01).

**Programming example (M model).  
Cartesian and Polar coordinates.**

**8.**

**PATH CONTROL:**  
Linear interpolation (G01).



```

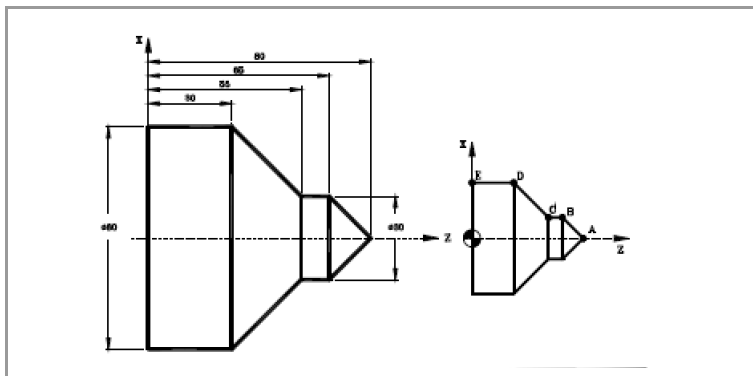
N10 T1 D1
N20 M06
N30 G71 G90 F450 S1500 M03(Initial conditions)
N40 G00 G90 X-40 Y15 Z10(Approaching profile 1)
N50 G01 Z-5
N60 X-40 Y30(Machining of profile 1)
N70 X-65 Y45
N80 X-90
N90 Y15
N100 X-40(End of profile 1)
N110 Z10
N120 G00 X20 Y45 F300 S1200(Approaching profile 2)
N130 G92 X0 Y0(Preselect part zero)
N140 G01 Z-5
N150 G91 X30(Machining of profile 2)
N160 X20 Y20
N170 X-20 Y20
N180 X-30
N190 Y-40(End of profile 2)
N200 G90 Z10
N210 G92 X20 Y45(Recuperate part zero)
N220 G30 I-10 J-60(Preselect polar origin)
N230 G00 R30 Q60 F350 S1200(Approaching profile 3)
N240 G01 Z-5
N250 Q120(Machining of profile 3)
N260 Q180
N270 Q240
N280 Q300
N290 Q360
N300 Q60(End of profile 3)
N310 Z10
N320 G00 X0 Y0
N330 M30
    
```



CNC 8070

(REF: 2008)

**Programming example (T model).  
Programming in radius.**



**Absolute coordinates.**

```
G90 G95 G96 F0.15 S180 T2 D1 M4 M41
G0 X50 Z100
G1 X0 Z80; Point A
G1 X15 Z65; A-B section
Z55; B-C section
X40 Z30; C-D section
Z0; D-E section
G0 X50 Z100
M30
```

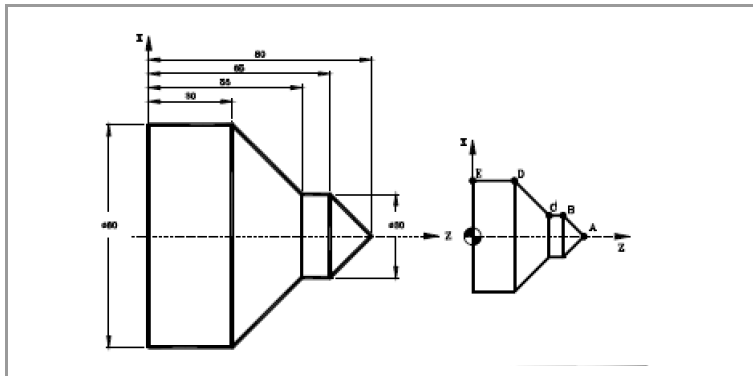
**Incremental coordinates.**

```
G90 G95 G96 F0.15 S180 T2 D1 M4 M41
G0 X50 Z100
G1 X0 Z80; Point A
G1 G91 X15 Z-15; A-B section
Z-10; B-C section
X25 Z-25; C-D section
Z-30; D-E section
G0 G90 X50 Z100
M30
```



**PATH CONTROL:**  
Linear interpolation (G01).

**Programming example (T model).  
Programming in diameters.**



**Absolute coordinates.**

```
G90 G95 G96 F0.15 S180 T2 D1 M4 M41
G0 X100 Z100
G1 X0 Z80; Point A
G1 X30 Z65; A-B section
Z55; B-C section
X80 Z30; C-D section
Z0; D-E section
G0 X100 Z100
M30
```

**Incremental coordinates.**

```
G90 G95 G96 F0.15 S180 T2 D1 M4 M41
G0 X100 Z100
G1 X0 Z80; Point A
G1 G91 X30 Z-15; A-B section
Z-10; B-C section
X50 Z-25; C-D section
Z-30; D-E section
G0 G90 X100 Z100
M30
```

8.

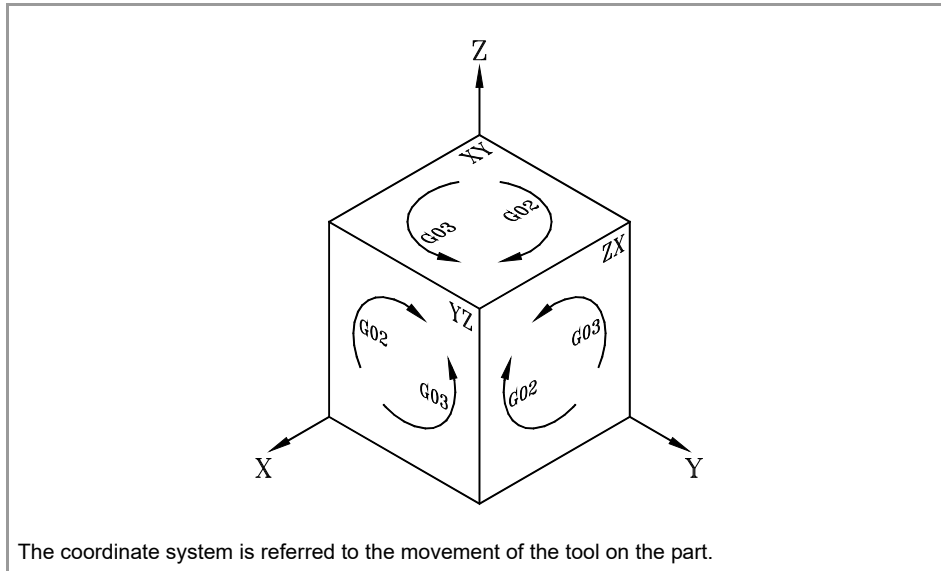
**PATH CONTROL:**  
Linear interpolation (G01).

### 8.3 Circular interpolation (G02/G03).

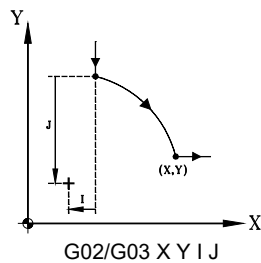
Movements programmed for G02 and G03 are executed along a circular tool path at the programmed feedrate "F" from the current position to the indicated target point. A circular interpolation can only be executed in the active plane.

- G02 Clockwise circular interpolation.
- G03 Counterclockwise circular interpolation.

Clockwise (G02) and counterclockwise (G03) moving directions have been established according to the following coordinate system.

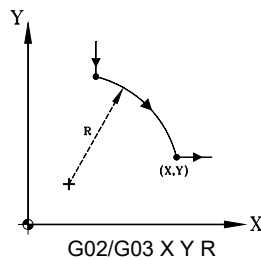


#### Programming.



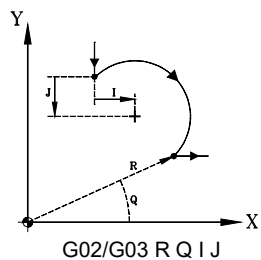
#### Cartesian coordinates (Arc center programming).

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and those of its center (referred to the starting point of the arc) according to the axes of the active work plane.



#### Cartesian coordinates (arc radius programming).

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and its radius.



#### Polar coordinates.

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point (radius and angle) and those of its center (relative to the starting point of the arc) according to the axes of the active work plane.

# 8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Considerations for the feedrate.

- The programmed feedrate "F" stays active until a new value is programmed, thus not being necessary to program it in every block.
- The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by the program or by the PLC. However, the maximum override is limited by the OEM (parameter MAXOVR).

# 8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Functions G02 and G03 may also be programmed as G2 and G3.
- Functions G02 and G03 are modal and incompatible with each other and with G00, G01, G33 and G63. Function G74 (Home search) also cancels functions G02 and G03.
- On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G00 or G01 as set by the OEM (parameter IMOVE). If the CNC assumes the function G00, and this function is defined as non-modal (parameter G0MODAL), after programming G1, G2 or G3, the CNC assumes G1 as a modal function.



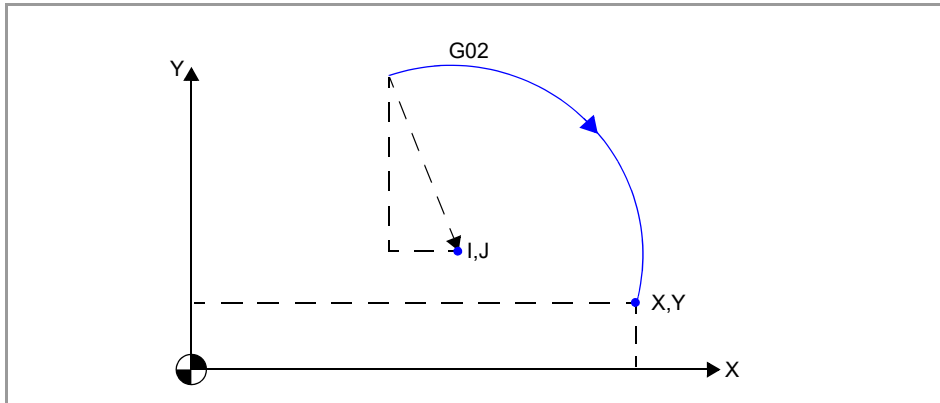
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 8.3.1 Cartesian coordinates (Arc center programming).

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and those of its center (referred to the starting point of the arc) according to the axes of the active work plane.



#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 <X..C{end\_point}> <I..K{center}>

- X..C{end\_point}                      Optional. End point of the arc.  
Units: Millimeters, inches or degrees.
- I..K{center}                          Optional. Arc center referred to starting point.  
Units: mm or inches.

G02 X50 Y0 I28 J13

#### End point of the arc.

The end point is defined with its coordinates along the axes of the active work plane and may be given in either absolute (G90) or incremental (G91) coordinates. If they are not programmed or are the same as the starting point, a full circle will be executed.

#### Arc center referred to starting point.

The coordinates of the center are measured according to the starting point. The arc center must be defined in cartesian coordinates by the letters "I", "J" or "K" depending on the active plane. When one of the center coordinates is zero, it does not have to be programmed. These coordinates are not affected by functions G90 and G91.

Plane.	Programming the center.
G17 G18 G19	Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively. G17 (XY plane) G02/G03 X... Y... I... J... G18 (ZX plane) G02/G03 X... Z... I... K... G19 (YZ plane) G02/G03 R... Z... J... K...
G20	Letters "I", "J" and "K" are associated with the abscissa, ordinate and perpendicular axes of the defined plane.
#FACE [X, C, Z] #CYL [Z, C, X, R]	The active trihedron is formed by the axes defined in the activation instruction of the C axis. The "I", "J", and "K" centers are associated with the axes in the same order that they were defined when activating the C axis.



PATH CONTROL:  
Circular interpolation (G02/G03).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Programming examples.

8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).

<p>XY plane (G17)</p> <p>XY</p> <p>Y 55 15 + 20 60 X</p> <p>...</p> <p>G02 X60 Y15 I0 J-40</p> <p>...</p>	<p>XY plane (G17)</p> <p>XY</p> <p>Y 50 30 + 30 50 X</p> <p>N10 G17 G71 G94 N20 G01 X30 Y30 F400 N30 G03 X30 Y30 I20 J20 N40 M30</p>	<p>YZ plane (G19)</p> <p>YZ</p> <p>Z 55 40 25 + + 35 55 75 Y</p> <p>N10 G19 G71 G94 N20 G00 Y55 Z0 N30 G01 Y55 Z25 F400 N40 G03 Z55 J20 K15 N50 Z25 J-20 K-15 N60 M30</p>
---	--	---

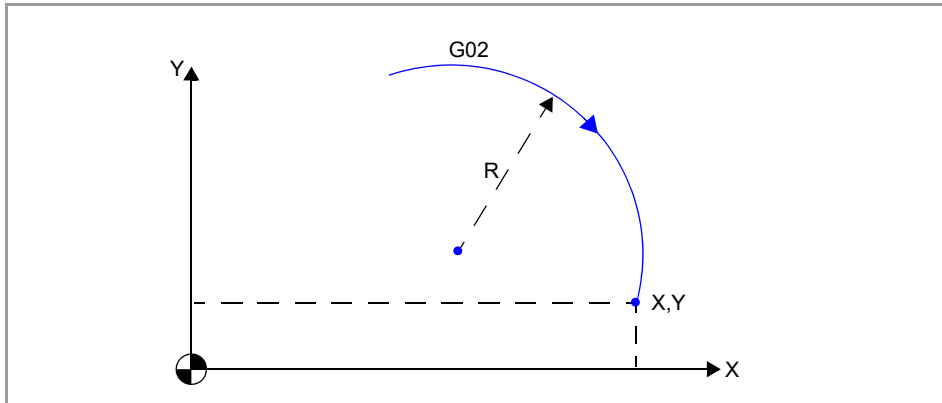


### 8.3.2 Cartesian coordinates (arc radius programming).



When programming an arc using the radius, it is not possible to program full circles because there are infinite solutions.

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and its radius.



#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 X..C{end\_point} <R{radius}>

X..C{end_point}	End point of the arc. Units: Millimeters, inches or degrees.
R{radius}	Optional. Radius of the arc. Units: mm or inches.

```
G02 X50 Y0 R25
G02 X50 Y0 R-25
```

#### End point of the arc.

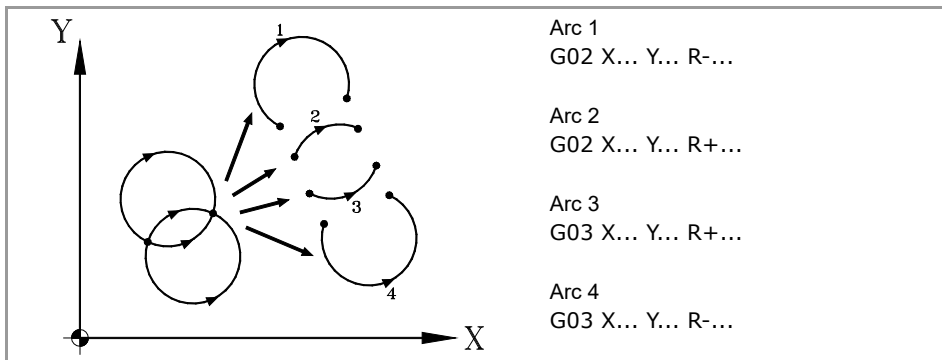
The end point is defined with its coordinates along the axes of the active work plane and may be given in either absolute (G90) or incremental (G91) coordinates.

The programming format depends on the active work plane.

G17 (XY plane)	G02/G03 X... Y... R...
G18 (ZX plane)	G02/G03 X... Z... R...
G19 (YZ plane)	G02/G03 R... Z... R...

#### Arc radius.

The arc radius is defined using the letter "R". If the arc is smaller than 180°, the radius will be programmed with a positive sign and with a negative sign if it is greater than 180°. This way and depending on the selected circular interpolation (G02 or G03), the desired arc will be defined. The radius value stays active until a new value is assigned or an arc is programmed using the center coordinates or a movement is programmed in polar coordinates.



PATH CONTROL:  
Circular interpolation (G02/G03).

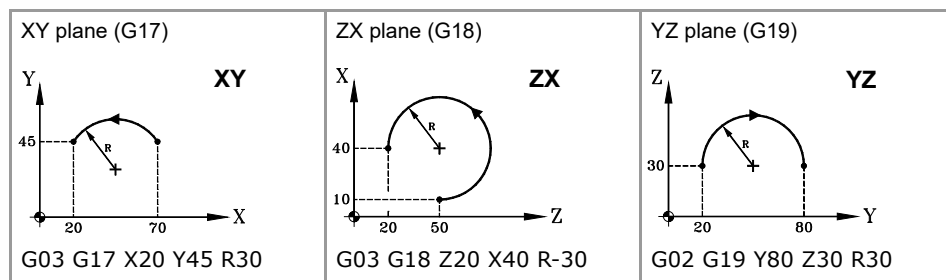


FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Programming examples.



8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).

### 8.3.3 Cartesian coordinates (arc radius pre-programming) (G263).

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point. The arc radius is programmed in an earlier block, using function G263 or the command "R1".

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

```
G263={radius}
R1={radius}
G02/G03 X..C{end_point}
X..C{end_point}           End point of the arc.
                           Units: Millimeters, inches or degrees.
{radius}                   Optional. Radius of the arc.
                           Units: mm or inches.
```

```
G263=25
G02 X50 Y0
R1=-33
G03 X88.32 Y12.34
```

#### Arc radius.

The radius value is programmed in the same block or in one before the definition of the circular interpolation. Both ways of defining the radius (G263 or R1) are equivalent. The CNC keeps the radius value until a circular interpolation is programmed by defining the center coordinates or a movement is programmed in polar coordinates.

The following examples will make semicircles with a radius of 50.

```
N10 G01 G90 X0 Y0 F500
N20 G263=50
N30 G02 X100
;-----
N10 G01 G90 X0 Y0
N20 G02 G263=50
N30 X100
;-----
N10 G01 G90 X0 Y0 F450
N20 G01 R1=50
N30 G02 X100
;-----
N10 G01 G90 X0 Y0
N20 G02 R1=50
N30 X100
```

#### Programming examples.

<pre>G01 G90 G94 X30 Y20 F350 G263=25 G02 X60 G263=-25 G03 X30 M30</pre>	<pre>G17 G71 G94 G00 X55 Y0 G01 X55 Y25 F400 G263=-25 G03 Y55 Y25 M30</pre>	<pre>G17 G71 G94 G01 X30 Y20 F400 G03 Y60 R1=30 G02 X75 G03 Y20 G02 X30 M30</pre>
--	---	---

8.

PATH CONTROL:  
Circular interpolation (G02/G03).

**FAGOR**

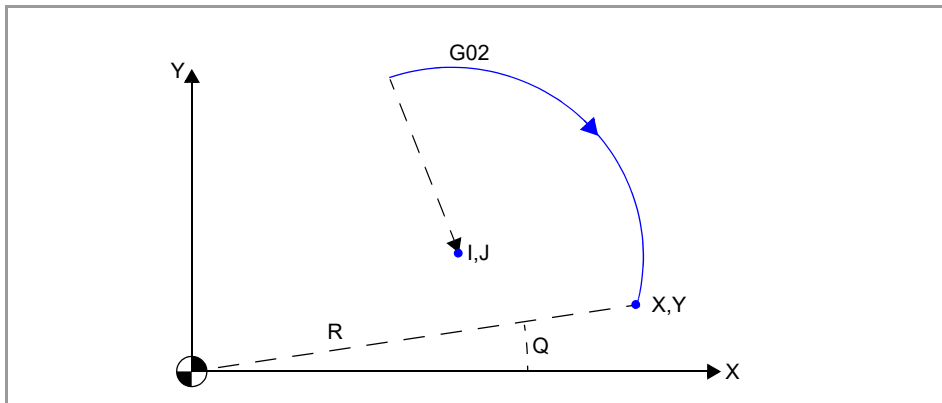
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 8.3.4 Polar coordinates.

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point (radius and angle) and those of its center (relative to the starting point of the arc) according to the axes of the active work plane.



#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 <R{final\_radius}> <Q{final\_angle}> <I..K{center}>

R{final\_radius}                      Optional. Radius of the end point of the arc.  
Units: Millimeters, inches or degrees.

Q{final\_angle}                      Optional. Angle of the end point of the arc.  
Units: Millimeters, inches or degrees.

I..K{center}                      Optional. Arc center referred to starting point.  
Units: mm or inches.

```
G02 R50 Q25 I28 J13
```

#### End point of the arc.

The end point is defined by its polar coordinates, radius (R) and the angle (Q) of the end point relative to the polar origin. The "R" radius will be the distance between the polar origin and the point. The "Q" angle will be formed by the abscissa axis and the line joining the polar origin with the point. If the angle or the radius is not programmed, it keeps the value programmed for the last move.

If the angle or the radius is not programmed, it keeps the value programmed for the last move. The radius and the angle may be defined both in absolute (G90) and incremental coordinates (G91). When programming the angle in G91, it is incremented with respect to the polar origin of the previous point; if programmed in G90, it indicates the angle formed with the horizontal going through the polar origin.

Programming a 360° angle in G91 means programming a whole circle. Programming a 360° angle in G90 means programming an arc where the target point forms a 360° angle with the horizontal going through the polar origin.

#### Arc center referred to starting point.

The coordinates of the center are measured according to the starting point. The arc center must be defined in cartesian coordinates by the letters "I", "J" or "K" depending on the active plane. When one of the center coordinates is zero, it does not have to be programmed; if neither of them are programmed, it will assume the polar origin as the arc center. These coordinates are not affected by functions G90 and G91.

8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).

**FAGOR** 

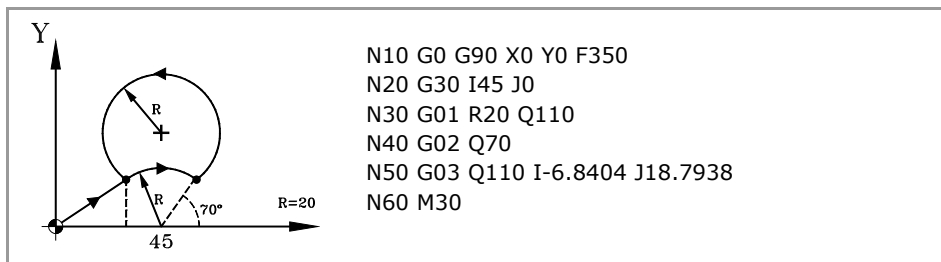
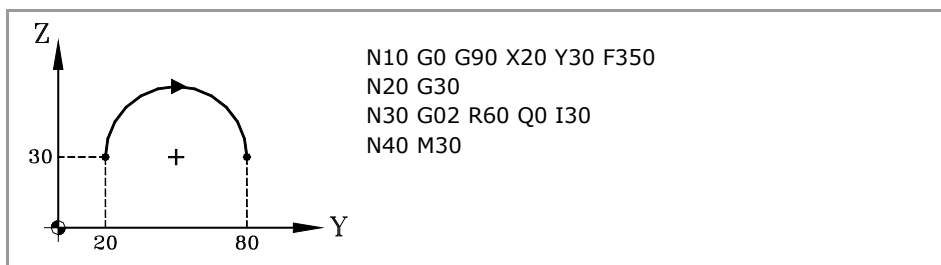
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

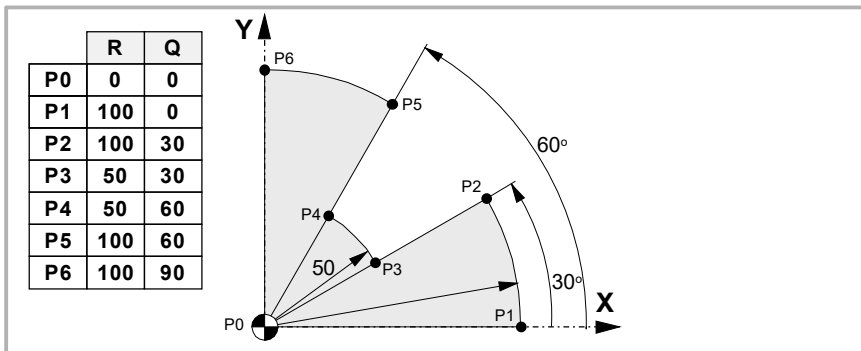
Plane.	Programming the center.
G17 G18 G19	Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively. G17 (XY plane) G02/G03 R... Q... I... J... G18 (ZX plane) G02/G03 R... Q... I... K... G19 (YZ plane) G02/G03 R... Q... J... K...
G20	Letters "I", "J" and "K" are associated with the abscissa, ordinate and perpendicular axes of the defined plane.
#FACE [X, C, Z] #CYL [Z, C, X, R]	The active trihedron is formed by the axes defined in the activation instruction of the C axis. The "I", "J", and "K" centers are associated with the axes in the same order that they were defined when activating the C axis.

### Programming examples.



**PATH CONTROL:**  
Circular interpolation (G02/G03).

8.3.5 Programming example (M model). Polar coordinates.

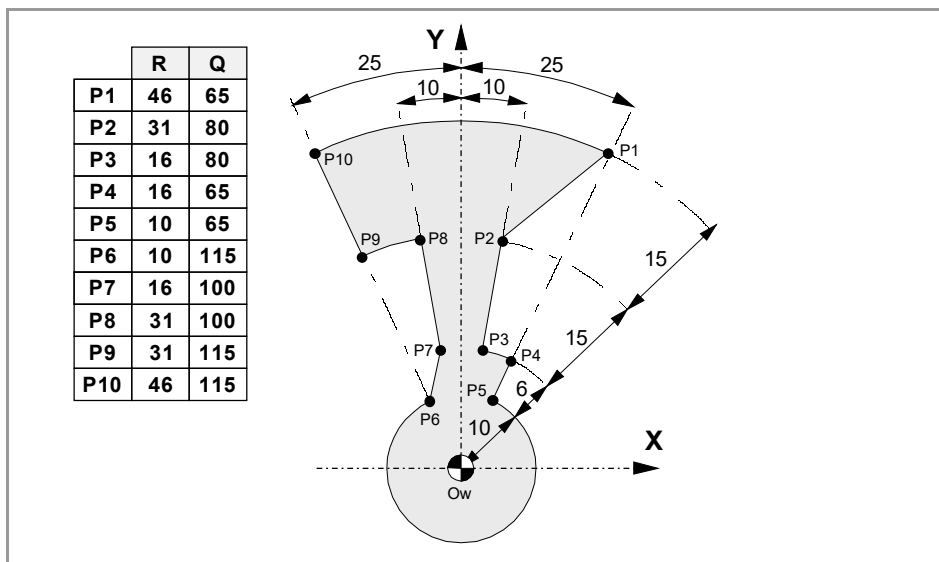


Absolute coordinates.	Incremental coordinates.	
G00 G90 X0 Y0 F350	G00 G90 X0 Y0 F350	; Point P0.
G01 R100 Q0	G91 G01 R100 Q0	; Point P1. Straight line.
G03 Q30	G03 Q30	; Point P2. Counterclockwise arc.
G01 R50 Q30	G01 R-50	; Point P3. Straight line.
G03 Q60	G03 Q30	; Point P2. Counterclockwise arc.
G01 R100 Q60	G01 R50	; Point P5. Straight line.
G03 Q90	G03 Q30	; Point P6. Counterclockwise arc.
G01 R0 Q90	G01 R-100	; Point P0, in a straight line.
M30	M30	

8.

**PATH CONTROL:**  
 Circular interpolation (G02/G03).

### 8.3.6 Programming example (M model). Polar coordinates.



#### Absolute coordinates.

G90 R46 Q65 F350; Point P1.  
 G01 R31 Q80; Point P2. Straight line.  
 G01 R16; Point P3. Straight line.  
 G02 Q65; Point P4. Clockwise arc.  
 G01 R10; Point P5. Straight line.  
 G02 Q115; Point P6. Clockwise arc.  
 G01 R16 Q100; Point P7. Straight line.  
 G01 R31; Point P8. Straight line.  
 G03 Q115; Point P9. Counterclockwise arc.  
 G01 R46; Point P10. Straight line.  
 G02 Q65; Point P1. Clockwise arc.  
 M30

#### Incremental coordinates.

G90 R46 Q65 F350; Point P1.  
 G91 G01 R-15 Q15; Point P2. Straight line.  
 G01 R-15; Point P3. Straight line.  
 G02 Q-15; Point P4. Clockwise arc.  
 G01 R-6; Point P5. Straight line.  
 G02 Q-310; Point P6. Clockwise arc.  
 G01 R6 Q-15; Point P7. Straight line.  
 G01 R15; Point P8. Straight line.  
 G03 Q15; Point P9. Counterclockwise arc.  
 G01 R15; Point P10. Straight line.  
 G02 Q-50; Point P1. Clockwise arc.  
 M30



**PATH CONTROL:**  
 Circular interpolation (G02/G03).



FAGOR AUTOMATION

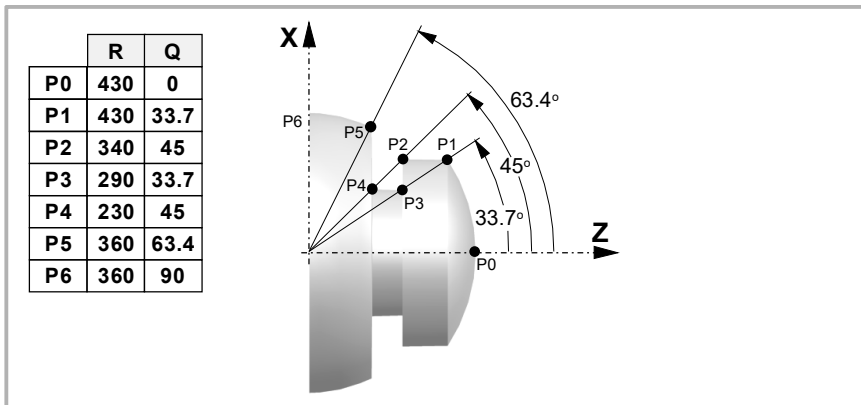
CNC 8070

(REF: 2008)

8.3.7 Programming example (T model). Programming examples.

8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).



Absolute coordinates.	Incremental coordinates.
G18	G18 ; Z-X plane,
G152	G152 ; Programming in radius.
G90 R430 Q0 F350	G90 R430 Q0 F350 ; Point P0.
G03 Q33.7	G91 G03 Q33.7 ; Point P1. Counterclockwise arc.
G01 R340 Q45	G01 R-90 Q11.3 ; Point P2. Straight line.
G01 R290 Q33.7	G01 R-50 Q-11.3 ; Point P3. Straight line.
G01 R230 Q45	G01 R-60 Q11.3 ; Point P4. Straight line.
G01 R360 Q63.4	G01 R130 Q18.4 ; Point P5. Straight line.
G03 Q90	G03 Q26.6 ; Point P6. Counterclockwise arc.
M30	M30



### 8.3.8 Polar coordinates. Temporary Polar origin shift to the center of arc (G31).

Function G31 shifts temporarily the polar origin to the center of the programmed arc. This function only acts in the block that contains it; once the block has been executed, it restores the previous polar.

#### Programming.

This function is added to the programmed circular interpolation G2/G3. Function G31 does not allow for the programming of the polar radius; it may only program the angle and one or both center coordinates.

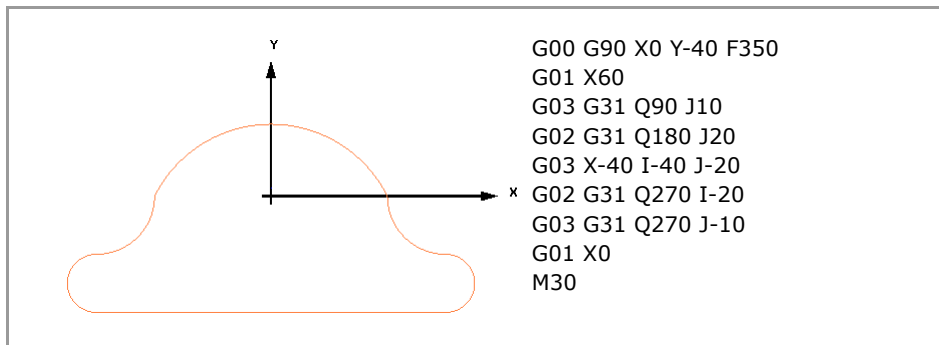
#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 G31 Q{end\_angle} <I..K{center}>

Q{final_angle}	Optional. Angle of the end point of the arc. Units: Millimeters, inches or degrees.
I..K{center}	Optional. Arc center referred to starting point. Units: mm or inches.

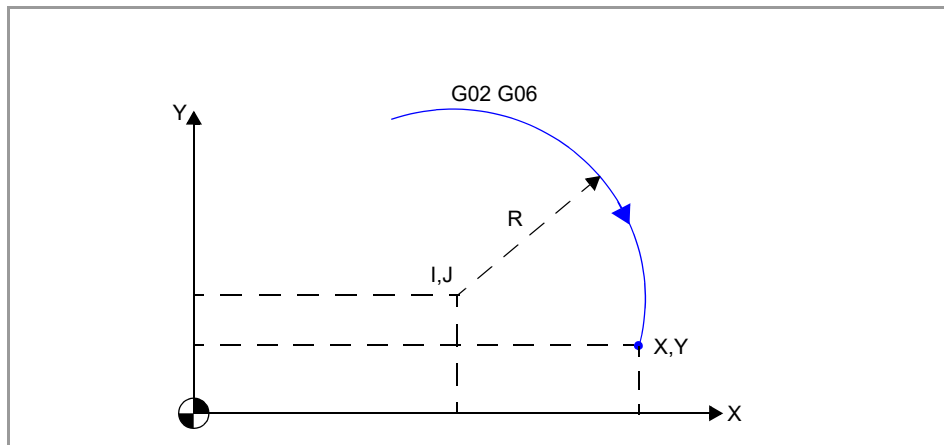
G02 G31 Q25 I28 J13



**PATH CONTROL:**  
 Circular interpolation (G02/G03).

### 8.3.9 Cartesian coordinates. Arc center in absolute coordinates (no-modal) (G06).

Function G06 indicates that the absolute coordinates are defined by the center of the arc, relative to the origin of the active reference system (part zero, polar origin, etc.).



#### Programming.

Add function G06 to a block where a circular interpolation has been defined. Function G06 is not modal, it only acts in the block where it is programmed.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 G06 <X..C{end\_point}> <I..K{center}>

- |                 |   |
|-----------------|---|
| X..C{end_point} | Optional. End point of the arc.<br>Units: Millimeters, inches or degrees. |
| I..K{center}    | Optional. Arc center in absolute coordinates.<br>Units: mm or inches.     |

```
G02 G06 X50 Y0 I38 J5
```

The example shows 2 different ways to define an arc using absolute center coordinates.

```
G90 G06 G02 X50 Y10 I20 J30
;-----
G91 G06 G02 X0 Y-40 I20 J30
```

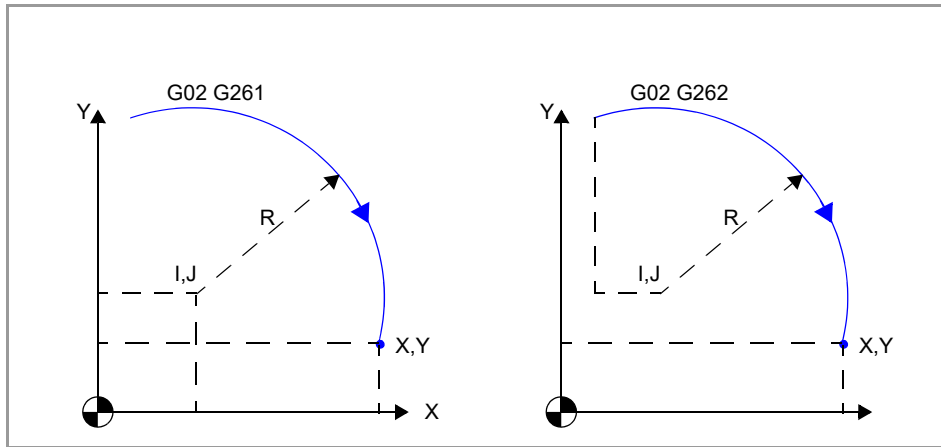
#### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G06 may also be programmed as G6.
- Function G06 is not modal.

(REF: 2008)

### 8.3.10 Cartesian coordinates. Arc center in absolute coordinates (modal) (G261/G262).

Function G261 indicates that the absolute coordinates are defined by the center of the arc, relative to the origin of the active reference system (part zero, polar origin, etc.). Function G262 cancels function G261 and the arc center is then defined according to the starting point of the arc.



#### Programming. Arc center in absolute coordinates (G261).

The function G261 may be programmed alone in the block or it may be added to a movement block. The function G261 is modal; after it has been programmed, it remains active until an incompatible function is programmed (G262).

##### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 G261 <X..C{end\_point}> <I..K{center}>

X..C{end\_point}                      Optional. End point of the arc.  
Units: Millimeters, inches or degrees.

I..K{center}                          Optional. Arc center in absolute coordinates.  
Units: mm or inches.

```
G02 G261 X50 Y0 I38 J5
G261
G02 X50 Y0 I38 J5
```

#### Programming. Arc center referred to starting point (G262).

The function G262 may be programmed alone in the block or it may be added to a movement block. The function G262 is modal; after it has been programmed, it remains active until an incompatible function is programmed (G261).

##### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G02/G03 G261 <X..C{end\_point}> <I..K{center}>

X..C{end\_point}                      Optional. End point of the arc.  
Units: Millimeters, inches or degrees.

I..K{center}                          Optional. Arc center in absolute coordinates.  
Units: mm or inches.

```
G02 G261 X50 Y0 I38 J5
G261
G02 X50 Y0 I38 J5
```



**PATH CONTROL:**  
Circular interpolation (G02/G03).



FAGOR AUTOMATION

CNC 8070

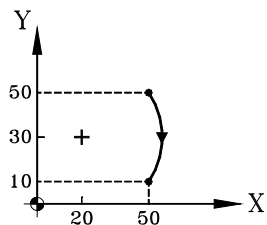
(REF: 2008)

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Functions G261 and G262 are modal and incompatible with each other.
- On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC assumes function G262.

### Programming example.

The example shows 2 different ways to define an arc using absolute center coordinates.



```
G261
G90 G02 X50 Y10 I20 J30
;-----
G261
G91 G06 G02 X0 Y-40 I20 J30
```

8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).

### 8.3.11 Arc correction (G264/G265).

In order to execute the programmed arc, the CNC calculates the initial and end radii, which must be the same. When this is not the case, the CNC attempts to execute the arc by correcting its center along the tool path. The tolerance allowed for the difference between both radii or for locating the corrected arc center is set by the OEM (parameters CIRINERR and CIRINFACT). Arc center correction may be turned on and off using the following functions:

- G264 Cancel arc correction.
- G265 Activate arc correction.

#### Programming. Activate arc correction (G265).

The function G265 may be programmed alone in the block or it may be added to a movement block. This function is modal; after it has been programmed, it remains active until an incompatible function is programmed (G264).

##### Programming format.

The programming format is:

G265

```
G02 G265 X50 Y0 I38 J5
G265
G02 X50 Y0 I38 J5
```

##### Arc correction using G265.

If the initial and final arc radii are not the same, the CNC tries to calculate a new center within the set tolerance so as to be able to execute an arc between the programmed points as close as possible to the defined arc. To calculate whether the margin of error is within the tolerance, the CNC takes into account the absolute error (difference between radii) and the relative error (% of radius). If any of these values is within the tolerance set by the OEM, the CNC corrects the center position.

If the CNC cannot find the center within those limits, it will issue the pertinent error message.

#### Programming. Cancel arc correction (G264).

The function G264 may be programmed alone in the block or it may be added to a movement block. This function is modal; after it has been programmed, it remains active until an incompatible function is programmed (G265).

##### Programming format.

The programming format is:

G264

```
G02 G264 X50 Y0 I38 J5
G264
G02 X50 Y0 I38 J5
```

##### Arc correction using G264.

When the difference between the initial and final radii is within the allowed tolerance, the CNC executes the arc with the radius calculated using the initial point, maintaining the position of the center.

If the difference between both radii exceeds the allowed tolerance, the relevant error message will be issued.

8.

**PATH CONTROL:**  
Circular interpolation (G02/G03).

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Functions G264 and G265 are modal and incompatible with each other.
- On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC assumes function G265.

8.

**PATH CONTROL.**

Circular interpolation (G02/G03).



FAGOR AUTOMATION

CNC 8070

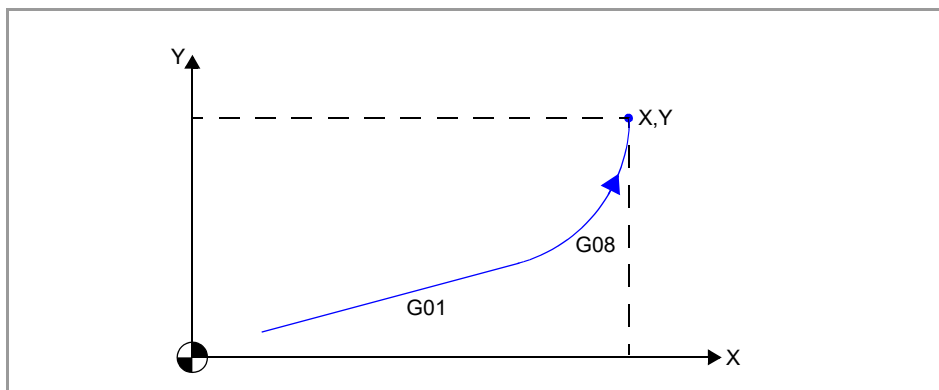
(REF: 2008)

## 8.4 Arc tangent to previous path (G08).

Function G08 may program a circular tool path tangent to the previous path without having to program the center coordinates (I, J or K). The previous path may be either linear or circular.



*Function G08 cannot be used to program full circles because there are infinite solutions.*



### Programming.

Program, along with function G08, the coordinates of the end point of the arc.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G08 X..C{end\_point}

X..C{end\_point}

End point of the arc.

Units: Millimeters, inches or degrees.

G08 G17 X50.87 Y38.45

Movement in cartesian coordinates.

G08 R20.23 Q45

Movement in polar coordinates.

### Coordinates of the end point of the arc

The end point may defined in cartesian or polar coordinates, both absolute and incremental.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G08 may also be programmed as G8.
- Function G08 is not modal, so it should always be programmed if you wish to execute an arc tangential to the previous path. After executing the function, the CNC restores the G00, G01, G02 or G03 function that was active before.

# 8.

**PATH CONTROL:**  
Arc tangent to previous path (G08).

**FAGOR** 

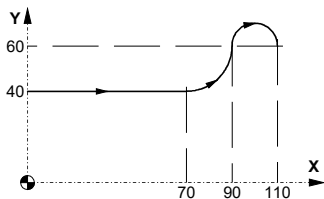
FAGOR AUTOMATION

CNC 8070

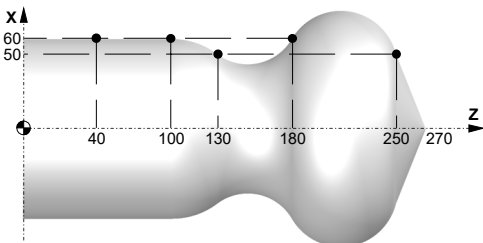
(REF: 2008)

**Programming examples.**

If you wish to program a straight line, then an arc tangential to the line and finally an arc tangential to the previous one.



```
G90 G01 X70
G08 X90 Y60
G08 X110
```



```
G18 ; ZX plane
G152 ; Programming in radius.
G90 G01 X0 Z270
X50 Z250
G08 X60 Z180
G08 X50 Z130
G08 X60 Z100
G01 X60 Z40
M30
```

8.

**PATH CONTROL:**  
Arc tangential to previous path (G08).

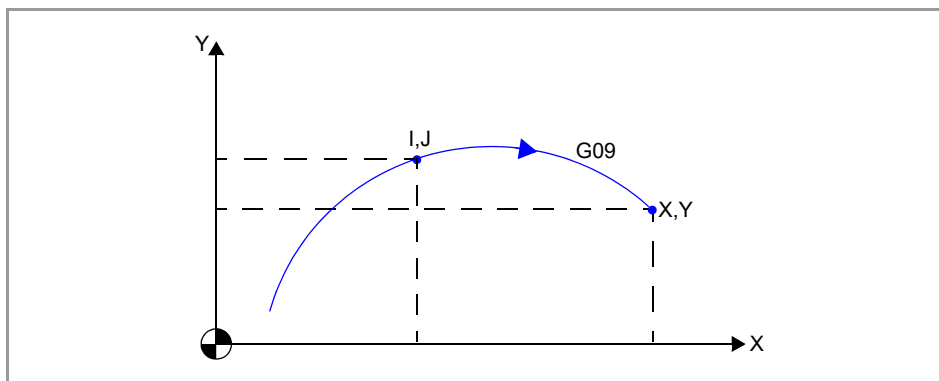


## 8.5 Arc defined by three points (G09).

Function G09 allows a circular tool path (arc) to be defined, by programming the end point and an intermediate point; this means, instead of programming the coordinates of the center, it programs any intermediate point. The starting point of the arc is the starting point of the movement.



Function G09 may not be used to programmed a full circle because all three points must be different.



### Programming.

Program, along with function G09, the end point and an intermediate point of the arc. When G09 is programmed it is not necessary to program the direction of movement (G02 or G03).

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G09 X..C{end\_point} I..K{intermediate\_point}

X..C{end\_point}                      End point of the arc.  
Units: Millimeters, inches or degrees.

I..K{intermediate\_point}          Intermediate point of the arc.  
Units: Millimeters, inches or degrees.

G09 G17 X50.87 Y38.45 I28.34 J34.58

Movement in cartesian coordinates.

G09 R20.23 Q45 I8 J-13.7

Movement in polar coordinates.

### Coordinates of the end point of the arc

The end point may defined in cartesian or polar coordinates, both absolute and incremental.

### Coordinates of the intermediate point of the arc.

The intermediate point must be defined in cartesian coordinates, using the letters "I", "J" or "K", depending on the active plane. These coordinates are affected by functions G90 and G91.

G17 G18 G19      Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively.

G20                Letters "I" and "J" are associated with the abscissa and ordinate axes of the defined plane.

# 8.

**PATH CONTROL:**

Arc defined by three points (G09).

**FAGOR**

FAGOR AUTOMATION

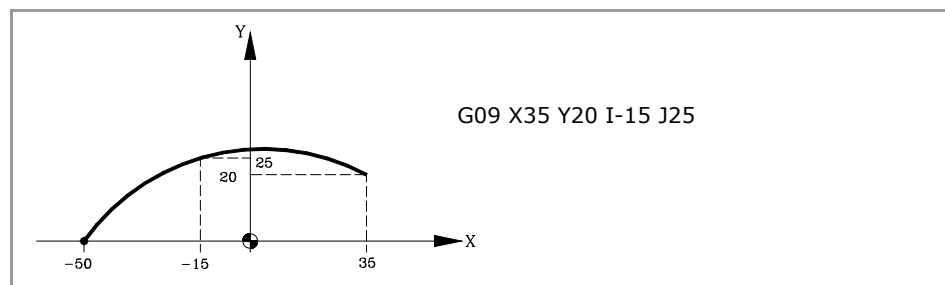
**CNC 8070**

(REF: 2008)

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G09 may also be programmed as G9.
- Function G09 is not modal, so it should always be programmed if you wish to execute an arc defined by three points. After executing the function, the CNC restores the G00, G01, G02 or G03 function that was active before.

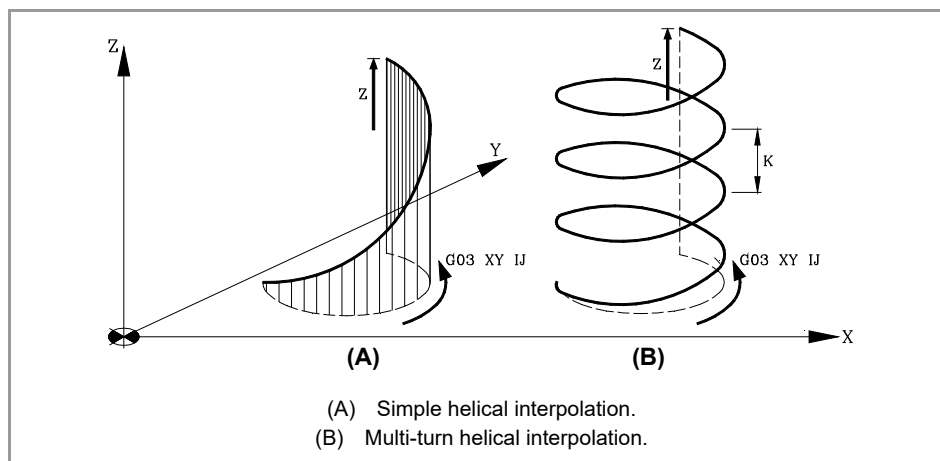
### Programming example.

**8.****PATH CONTROL.**

Arc defined by three points (G09).

## 8.6 Helical interpolation (G02/G03).

Helical interpolation consists of a circular interpolation in the work plane and a linear movement of the rest of the axes programmed. If the helical interpolation is supposed to make more than one turn, the helical pitch must be defined.



### Programming.

Program along with the circular interpolation using the functions G02, G03, G08 or G09 and then the linear movement for the other axes. If the helical interpolation is supposed to make more than one turn, the helical pitch must be defined.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets. The format for the circular interpolation is explained in the corresponding chapters.

G02/G03 (circular interpolation) X..C{linear\_movement} <I..K{pitch}>

G08 (circular interpolation) X..C{linear\_movement} <I..K{pitch}>

G09 (circular interpolation) X..C{linear\_movement} <I..K{pitch}>

X..C{linear\_movement}      Linear movement of the helical interpolation for one or several axes.  
Units: mm or inches.

I..K{pitch}                      Helical pitch.  
Units: mm or inches.

```
G01 G90 X-50 Y0 Z0
G02 G17 I50 J0 Z100 K37
G01 G90 X-50 Y0 Z0
G02 G17 X50 Y0 R50 Z110 K25
G01 G90 X-50 Y0 Z0
G02 R50 Q90 I50 J0 Z-90 K17
G01 G90 X-50 Y50 Z0
G01 Y0
G08 X50 Y0 Z58.45 K10.25
G01 G90 X-50 Y50 Z0
G01 Y0
G08 R50 Q65 Z69.45 K15.25
G01 G90 X-50 Y0 Z0
G09 G17 X65 Y-12.9 I32 J56.78 Z-88 K12
G01 G90 X-50 Y0 Z0
G09 G17 R45 Q-33 I32 J56.78 Z88 K11
```

### End point on the work plane.

For helical interpolation in various turns, it is not necessary to define the coordinates for the end point in the work plane. This point will be calculated by the CNC depending on the height and pitch of the helix.



**PATH CONTROL:**  
Helical interpolation (G02/G03).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Helical pitch.**

The helical pitch is defined using the letter "I", "J" or "K" associated with the axis perpendicular to the work plane. The pitch will not be affected by functions G90 and G91.

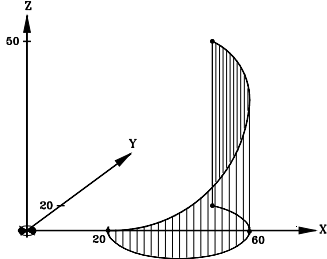
Plane.	Programming the center.
G17 G18 G19	The pitch is defined with the letter "K" (G17), "J" (G18) or "I" (G19).
G20	The pitch is defined with the letter "K".

**8.**

**PATH CONTROL:**  
Helical interpolation (G02/G03).

**Programming example.  
Helical interpolation.**

Various ways of defining a helical interpolation, where the starting point is X20 Y0 Z0.

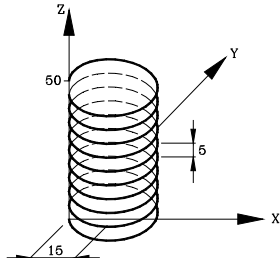


```

G03 X40 Y20 I20 J0 Z50
; -----
G03 X40 Y20 R-20 Z50
; -----
G03 R44.7213 Q26.565 I20 J0 Z50
; -----
G09 X40 Y20 I60 J0 Z50
    
```

**Programming example.  
Multi-turn helical interpolation.**

Various ways of defining a helical interpolation with several turns, where the starting point is X0 Y0 Z0.



```

G03 X0 Y0 I15 J0 Z50 K5
; -----
G03 R0 Q0 I15 J0 Z50 K5
    
```

Manual intervention makes it possible to activate the JOG mode by program; in other words, the axes may be jogged even while executing a program. The movement may be made using handwheels or the jog keys (incremental or continuous jog). The functions related to manual intervention are:

- G200 Exclusive manual intervention.
- G201 Activation of additive manual intervention.
- G202 Cancellation of additive manual intervention.

The difference between exclusive and additive interventions is that the exclusive one (G200) interrupts the execution of the program to activate the jog mode whereas the additive one (G201) lets you jog an axis while executing the programmed movements.

## **Feedrate behavior.**

The feedrate of the jogging movements during manual intervention is independent from the active "F" and may be defined by the operator using instructions in high-level language; a different feedrate may be set for each work mode (incremental or continuous jog). If not defined, the movements are carried out at the feedrate set by the OEM.

The feedrate may be varied between 0% and 200% using the selector switch on the CNC's operator panel which affects the same way the programmed "F" and the feedrate of manual intervention.



## 9.2 Exclusive manual intervention (G200).

With exclusive manual intervention, the axes may be jogged using handwheels or jog keys (continuous or incremental) by interrupting the execution of the program. This function can be applied on any axis of the machine; however, it cannot be applied on the spindle, although this can work in positioning mode.

To cancel manual intervention and resume program execution, press the [CYCLE START] key.

### Programming.

To activate the additive manual intervention, the function G200 must be programmed in the same block, followed by the #AXIS statement with the applicable axes. Programming G200 alone applies the manual intervention on all the axes.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
G200
G200 #AXIS[{axis}, .. , {axis}]
{axis}           Axis name.
                  Units: -.
```

```
G200 #AXIS [X, Z]
  (Interrupt the execution and activate additive manual intervention on XZ
  axes)
G200
  (Interrupt the execution and activate additive manual intervention on all
  axes)
```

### Considerations

If a manual intervention is executed before a circular interpolation and one of the axes involved in the circular interpolation is jogged, it could issue an error message indicating that a circle has been programmed wrong or it may execute a circle other than the one programmed.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Function G200 is non-modal and is incompatible with functions G201 and G202. On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC assumes function G202.

9.

**TOOL PATH CONTROL: MANUAL INTERVENTION.**  
Exclusive manual intervention (G200).

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 9.3 Jogging feedrate.

### 9.3.1 Feedrate in continuous jog (#CONTJOG).

This statement allows the feedrate to be configured in continuous jog mode for the specified axis. These values may be defined before or after activating manual intervention and stay active until the end of the program or a reset.

#### Programming.

Program the #CONTJOG statement, followed by the feedrate and the axis.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

#CONTJOG [{feed}] {axis}

{feed}                    Axis feedrate.  
Units: Millimeters/minute, inches/minute or degrees/minute.

{axis}                    Axis name.  
Units: -.

```
#CONTJOG [400] X
  (Manual intervention; feedrate in continuous jog for X)
#CONTJOG [600] Y
  (Manual intervention; feedrate in continuous jog for Y)
```

9.

TOOL PATH CONTROL. MANUAL INTERVENTION.

Jogging feedrate.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### 9.3.2 Feedrate in incremental jog (#INCJOG).

This instruction configures the indicated incremental movement and axis feedrate for each incremental jog position of the selector switch. These values may be defined before or after activating manual intervention and stay active until the end of the program or a reset.

#### Programming.

Program the #INCJOG statement, followed by the increment and feedrate at each jog position for the axis.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#INCJOG [[{increment_1},{feed_1}] [{increment_10},{feed_10}] ... [...]] <axis>
{feed_1}           Feedrate position, 1 to 10000 for the incremental jog switch.
..                Units: Millimeters/minute, inches/minute or degrees/minute.
{feed_10000}
{increment_1}      Increment position, 1 to 10000 for the incremental jog switch.
..                Units: Millimeters, inches or degrees.
{increment_10000}
{axis}            Axis name.
                  Units: -.
```

```
#INCJOG [[0.1,100][0.5,200][1,300][5,400][10,500]] X
(The movement and feedrate of the X axis for each incremental jog switch
position is as follows)
(Position 1 on the switch; 0.1 mm to 100 mm/min)
(Position 10 on the switch; 0.5 mm to 200 mm/min)
(Position 100 on the switch; 1 mm to 300 mm/min)
(Position 1000 on the switch; 5 mm to 400 mm/min)
(Position 10000 on the switch; 10 mm to 500 mm/min)
```

9.

TOOL PATH CONTROL: MANUAL INTERVENTION.

Jogging feedrate.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 9.3.3 Feedrate in incremental jog (#MPG).

This instruction allows for the configuration, for each position on the handwheel, of the resolution of the handwheel on the specified axis. These values may be defined before or after activating manual intervention and stay active until the end of the program or a reset.

#### Programming.

Program the #INCJOG statement, followed by the resolution at each jog position for the axis.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#MPG [{resolution_1},{resolution_10},{resolution_100}] {axis}
{resolution_1}      Resolution position, 1 to 100 for the handwheel switch.
..                  Units: Millimeters/pulse, inches/pulse or degrees/pulse.
{resolution_100}
{axis}              Axis name.
                    Units: -.
```

```
#MPG [0.1, 1, 10] X
(The resolution at each handwheel switch position is as follows)
(Position 1 on the switch; 0.1 mm/turn)
(Position 10 on the switch; 1.0 mm/turn)
(Position 100 on the switch; 10 mm/turn)
```



*This instruction sets the distance per handwheel pulse in a time period equal to the CNC's cycle time. If the feedrate required for this movement exceeds the maximum set by the machine manufacturer, the feedrate will be limited to this value and the axis moving distance will be less than what has been programmed in the instruction.*

*Example: If a 5 mm move is programmed and the cycle time is 4 msec, the resulting feedrate is 1250 mm/sec. If the maximum feedrate is limited to 1000 mm/s, the actual distance moved will be 4 mm.*

# 9.

TOOL PATH CONTROL: MANUAL INTERVENTION.  
Jogging feedrate.

### 9.3.4 Manual path movement limits (#SET OFFSET).

This statement allows for the configuration of the movement limits made under additive manual intervention. These limits are ignored when executing the movements by program. The limits may be defined after activating manual intervention and stay active until it is deactivated.

#### Programming.

Program the #SET OFFSET statement, followed by the lower and upper movement limits for the axis.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#SET OFFSET [{lower_limit},{upper_limit}] {axis}
{lower_limit}      Upper and lower movement limit.
{upper_limit}      Units: Millimeters, inches or degrees.
{axis}             Axis name.
                  Units: -.
```

```
#SET OFFSET [-20,35] Y
(Movement limit less than 20 mm and greater than 35 mm on the Y axis)
```

#### Upper and lower movement limit.

The limits are referred to the axis position. The lower limit must be less than or equal to zero and the upper limit must be zero or greater than zero.

9.

TOOL PATH CONTROL: MANUAL INTERVENTION.

Jogging feedrate.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 9.3.5 Synchronization of coordinates and additive manual offset (#SYNC POS).

This instruction synchronizes the preparation coordinate with the execution one and assumes the additive manual offset.

#### Programming.

Program the instruction #SYNC POS alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

#SYNC POS

```
#SYNC POS
```

**9.**

**TOOL PATH CONTROL. MANUAL INTERVENTION.**  
Jogging feedrate.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 9.4 Variables.

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W). Reading these variables interrupts block preparation.

Variable.	PRG	Meaning.
(V.)[ch].A.MANOF.xn	R	Distance moved in manual mode or tool inspection mode. Units: Millimeters, inches or degrees.
(V.)[ch].A.ADDMANOF.xn	R	Distance moved with G200 or G201. The value of this variable is maintained during the execution of the program even when canceling manual intervention. Units: Millimeters, inches or degrees.

### Syntax of the variables.

·ch· Channel number.

·xn· Name, logic number or index of the axis.

V.A.ADDMANOF.Z	Z axis.
V.A.ADDMANOF.4	Axis with logic number ·4·.
V.[2].A.ADDMANOF.1	Axis with index ·1· in the channel ·2·.

9.

TOOL PATH CONTROL: MANUAL INTERVENTION.  
Variables.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 9.

## TOOL PATH CONTROL. MANUAL INTERVENTION. Variables.



CNC 8070

(REF: 2008)

## 10.1 Electronic threading with constant pitch (G33)



*For electronic threading, the machine must have a rotary encoder installed on the spindle.*

The electronic threading executes the programmed thread in a single pass. In the electronic threading, the CNC does NOT interpolate the movement of the axes with the spindle.

Although, this type of threads are carried out along an axis, the CNC permits interpolating several axes. Besides, with electronic threading, it is possible to make threads with several entries (starts) and to blend threads.

Electronic threading may be carried out with any spindle, but when not using the master spindle, the spindle being used must be synchronized with it. The spindles may be synchronized via program using the instruction #SYNC or #TSYNC.

### Programming.

Program G33 followed by the coordinates of the end point of the thread and the thread pitch. Optionally, the entry angle may be defined which allows multi-entry (starts) thread.

### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

G33 X..Z{pos} I/J/K{pitch} <Q1={angle}>

X..Z{pos}            Coordinates of the end point.  
                          Units: millimeters/inches.

I/J/K{pitch}        Thread pitch.  
                          Units: millimeters/inches.

Q1={angle}         Optional. Spindle angular position for the starting point of the thread. If not programmed, the function assumes a value of 0.  
                          Units:  $\pm 359.9999$  degrees.

```
G33 Z-50 K3 Q1=0
      (3 mm pitch thread)
G33 Z-40 K1 Q1=30
G33 Z-80 K1 Q1=210
      (Two-start thread, at 30° and 210°)
```

### Coordinates of the end point.

Although, this type of threads are carried out along an axis, the CNC permits interpolating several axes. The coordinates of the end point may be defined either in Cartesian or Polar coordinates as well as in absolute and incremental coordinates.

**Thread pitch.**

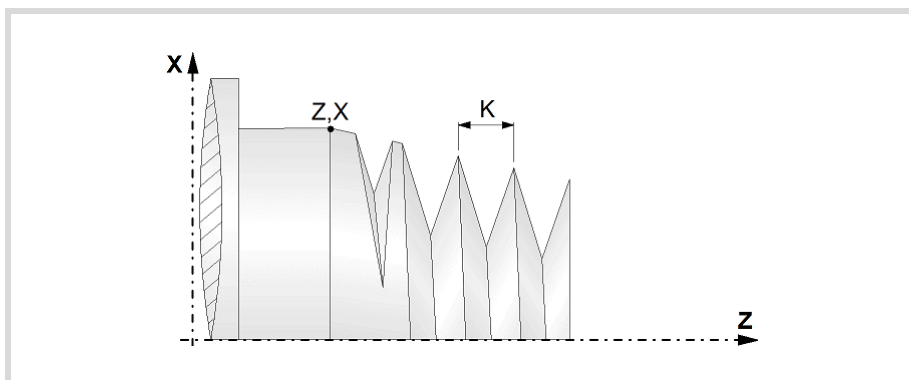
- The pitch is defined by the letters "I", "J" or "K" depending on the active plane.
  - G17 G18 G19 Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively.
  - G20 Letters "I", "J" and "K" are associated with the abscissa, ordinate and perpendicular axes of the defined plane.

**Example:**

Electronic threading on the Z axis and in different planes (configuration of XYZ axes in the channel).

G17 (XY plane) G33 Z40 K2	G18 (ZX plane) G33 Z40 K2	G19 (YZ plane) G33 Z40 K2
G20 Z1 Y2 X3 G33 Z40 I2	G20 Y1 Z2 X3 G33 Z40 J2	G20 Y1 Z3 X2 G33 Z40 K2

- When interpolating several axes in the electronic threading, the pitch is not defined on the path, it is defined on one of the axes.



- If the pitch of the thread is not programmed, the CNC acts as follows.
  - If no G33 or G34 has been programmed earlier, the CNC will issue an error message.
  - If a G33 has been programmed earlier, the pitch will be the pitch of the last G33 programmed.
  - If there is no G33, but a G34 has been programmed earlier, the pitch will be the final pitch of the last G34 programmed.

**Spindle angular position.**

Spindle angular position (within  $\pm 359.9999^\circ$ ) for the starting point of the thread. With this parameter, it is possible to make multi-entry threads. Programming it is optional; if not programmed, the function executes the thread at  $0^\circ$  (same as programming Q1=0).

**Considerations about execution.**

**Interrupt execution ([STOP] key or PLC mark \_FEEDHOL).**

The behavior of the CNC when interrupting a threading ([STOP] key or PLC mark \_FEEDHOL) depends on function G233. See "10.4 Withdraw the axes after interrupting an electronic threading (G233)." on page 195.

- If G233 is active, when interrupting the thread, the axes withdraw the distance programmed in that function. If when interrupting the threading the pass is near the end, the CNC ignores G233 and stops the axes at the end of the pass.
- If G233 is not active, when interrupting the threading the axes stop at the end of the pass.

**Spindle home search.**

If the spindle has not been homed (referenced), the first G33 will be done automatically when using the master spindle. If the spindle is the master and it has not been homed, it will issue a warning.

10.

ELECTRONIC THREADING AND RIGID TAPPING.  
Electronic threading with constant pitch (G33)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



**Feedrate behavior.**

The threading feedrate depends on the programmed spindle speed and thread pitch (Feedrate = Spindle speed x Pitch). The electronic threading is carried out at 100% of the calculated feedrate and these values cannot be modified from the CNC's operator panel or via PLC.

**Behavior of the spindle speed and of the spindle speed override**

If the OEM allows it, (parameter THREADOVR), the user can modify the speed override from the operator panel and, in that case, the CNC will adapt the feedrate automatically respecting the thread pitch. In order to be able to modify the override, the feed forward active on the axes involved in threading must be higher than 90%.

If more than one G33 have been programmed for the same thread, all the threading operations must start at the same speed; otherwise, the entry point (start) to the thread will not be the same in all the threads. The CNC permits changing the spindle override during the thread cutting pass.

If more than one G33 have been programmed for a multi-start (multi-entry) thread, all the threading operations must start at the same speed; otherwise, the angle between the starts (entry points) to the thread will not be the same as the one programmed. The CNC permits changing the spindle override during the thread cutting pass.

**Considerations about thread blending.**

When working in round corner (G05), the CNC lets blend different threads seamlessly in a single part. When thread blending, the CNC only takes into consideration the angular position of the spindle (Q1) in the first thread, after activating G33 or G34. The CNC ignores parameter Q1 until this function is canceled and activated back and it is synchronized when going through that angle.

**Blending a thread of fixed pitch (G33) with a thread of variable pitch (G34).**

The starting pitch of the variable thread (G34) must be the same as the pitch of the fixed-pitch thread (G33). The pitch increment of the variable thread will be half the increment ("K1"/2) in the first turn and a full increment "K1" in the following turns.

```
G33 Z-40 K2.5
G34 Z-80 K2.5 K1=1
```

**Blending a thread of variable pitch (G34) with a thread of fixed pitch (G33).**

This combination is used to end a variable-pitch thread (G34) with a portion of the thread that maintains the final pitch of the previous thread. In this case, the pitch is not programmed in the fixed-pitch thread G33 and the CNC will use the last pitch of the previous thread.

```
G34 Z-50 K2 K1=3
G33 Z-100
```

**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

Function G33 is modal and incompatible with G00, G01, G02, G03, G34, G63 and G100. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G00 or G01 as set by the machine manufacturer (parameter IMOVE).

**10.**

**ELECTRONIC THREADING AND RIGID TAPPING.**  
Electronic threading with constant pitch (G33)

**FAGOR** 

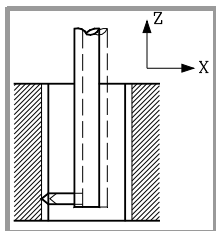
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 10.1.1 Programming examples (-M- model)

## Single-entry electronic threading



To make the following electronic thread in a single pass.

Position: X30 Y30 Z0  
 Depth: 30mm  
 Pitch: 1.5mm

```
S100 M03
G01 G90 X30 Y30 Z0
G33 Z-30 K1.5
M19 S0                (Spindle orientation)
G91 G00 X3            (Tool withdrawal)
G90 Z10              (Withdrawal and exit the hole)
```

Since a spindle speed of 100 rpm and a pitch of 1.5 mm have been programmed, the resulting feedrate will be 150 mm/min (pitch feedrate).

## Multi-entry electronic threading

To make a thread similar to the previous one but with three entries (starts), the first one being at 20°.

```
S100 M03
G01 G90 X30 Y30 Z0
G33 Z-30 K1.5 Q1=20    (First thread)
M19 S0
G91 G00 X3
G90 Z10
S100 M03
G33 Z-30 K1.5 Q1=140  (Second thread)
M19 S0
G91 G00 X3
G90 Z10
S100 M03
G33 Z-30 K1.5 Q1=260  (Third thread)
M19 S0
G91 G00 X3
G90 Z10
S100 M03
M30
```

10.

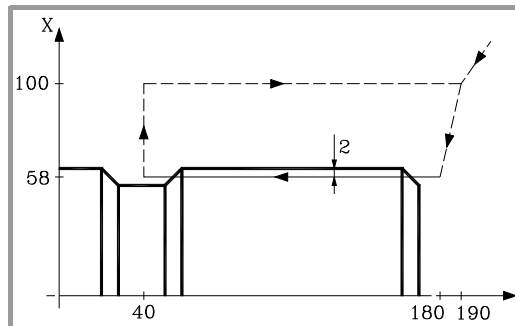
ELECTRONIC THREADING AND RIGID TAPPING.  
 Electronic threading with constant pitch (G33)

## 10.1.2 Programming examples (·T· model)

Example programming the X axis in radius.

### Longitudinal electronic threading

To make a cylindrical thread in a single pass, 2 mm deep and with a 5 mm pitch.

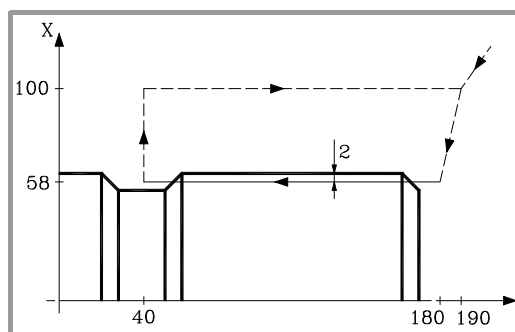


```
S100 M03
G00 G90 X200 Z190
X116 Z180
G33 Z40 K5
G00 X200
Z190
```

Since a spindle speed of 100 rpm and a pitch of 5 mm have been programmed, the resulting feedrate will be 500 mm/min (pitch feedrate).

### Multi-entry longitudinal electronic threading

To make a thread similar to the previous one, but with two entries shifted 180° from each other.



```
S100 M03
G00 G90 X200 Z190
X116 Z180
G33 Z40 K5 Q1=0
G00 X200
Z190
X116 Z180
G33 Z40 K5 Q1=180
G00 X200
Z190
```

# 10.

**ELECTRONIC THREADING AND RIGID TAPPING.**

Electronic threading with constant pitch (G33)

**FAGOR** 

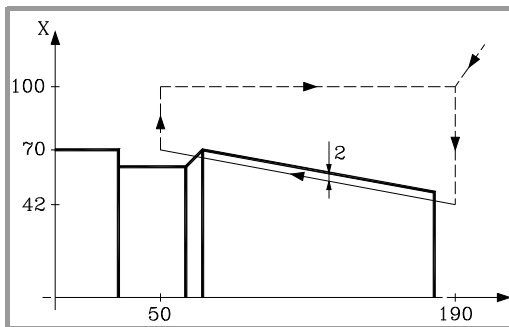
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## Electronic taper threading

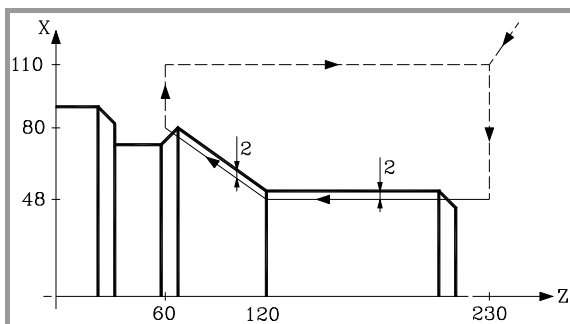
To make a taper thread in a single pass, 2 mm deep and with a 5 mm pitch.



```
S100 M03
G00 G90 X200 Z190
X84
G33 X140 Z50 K5
G00 X200
Z190
```

## Thread blending (joining)

We would like to blend a longitudinal thread with a taper thread with a depth of 2mm and a pitch of 5 mm.



```
S100 M03
G00 G90 G05 X220 Z230
X96
G33 Z120 X96 K5
G33 Z60 X160 K5
G00 X220
Z230
```

10.

ELECTRONIC THREADING AND RIGID TAPPING.  
Electronic threading with constant pitch (G33)

## 10.2 Electronic threading with variable pitch (G34)



*For electronic threading, the machine must have a rotary encoder installed on the spindle.*

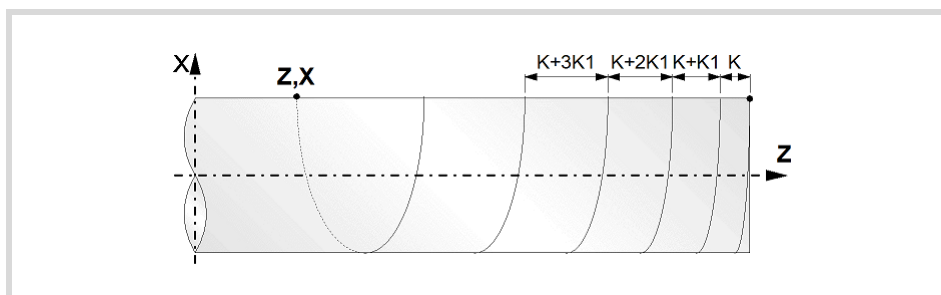
The electronic threading executes the programmed thread in a single pass. In the electronic threading, the CNC does NOT interpolate the movement of the axes with the spindle.

Although, this type of threads are carried out along an axis, the CNC permits interpolating several axes. Besides, with electronic threading, it is possible to make threads with several entries (starts) and to blend threads.

Electronic threading may be carried out with any spindle, but when not using the master spindle, the spindle being used must be synchronized with it. The spindles may be synchronized via program using the instruction #SYNC or #TSYNC.

### Programming.

Program function G34 followed by the coordinates of the end point of the thread and the starting thread pitch and the pitch increment or decrement of the thread pitch. Optionally, the entry angle may be defined which allows multi-entry (starts) thread.



### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

G34 X..Z{pos} I/J/K{pitch} K1={pitchvar} <Q1={angle}>

X..Z{pos}            Coordinates of the end point.  
Units: millimeters/inches.

I/J/K{pitch}        Optional. Starting thread pitch.  
Units: millimeters/inches.

Q1={angle}         Optional. Spindle angular position for the starting point of the thread. If not programmed, the function assumes a value of 0.  
Units: ±359.9999 degrees.

K1={pitchvar}     Optional. Increment (K1>0) or decrement (K1<0) of the thread pitch per spindle turn.  
Units: millimeters/inches.

G34 Z-50 K3 K1=2 Q1=0  
(Thread pitch 3 mm and 2 mm increment per turn)

G34 Z-40 K1 K1=1.5 Q1=30

G34 Z-80 K1 K1=1.5 Q1=210  
(Two-start thread, at 30° and 210°)

### Coordinates of the end point.

Although, this type of threads are carried out along an axis, the CNC permits interpolating several axes. The coordinates of the end point may be defined either in Cartesian or Polar coordinates as well as in absolute and incremental coordinates.

10.

**ELECTRONIC THREADING AND RIGID TAPPING.**  
Electronic threading with variable pitch (G34)

**FAGOR**

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Starting thread pitch.**

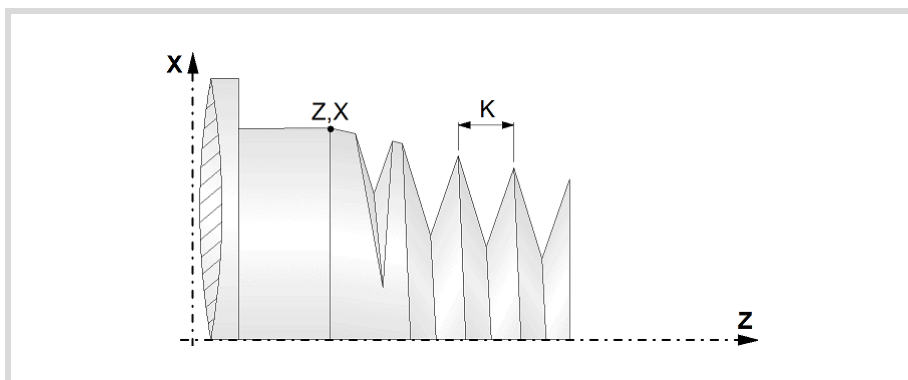
- The pitch is defined by the letters "I", "J" or "K" depending on the active plane.
  - G17 G18 G19 Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively.
  - G20 Letters "I", "J" and "K" are associated with the abscissa, ordinate and perpendicular axes of the defined plane.

**Example:**

Electronic threading on the Z axis and in different planes (configuration of XYZ axes in the channel).

G17 (XY plane) G34 Z40 K2 K1=1	G18 (ZX plane) G34 Z40 K2 K1=1	G19 (YZ plane) G34 Z40 K2 K1=1
G20 Z1 Y2 X3 G34 Z40 I2 K1=1	G20 Y1 Z2 X3 G34 Z40 J2 K1=1	G20 Y1 Z3 X2 G34 Z40 K2 K1=1

- When interpolating several axes in the electronic threading, the pitch is defined on one of the axes; not on the path.



- If the starting pitch of the thread is not programmed, the CNC acts as follows.
  - 1 If no G33 or G34 has been programmed earlier, the CNC will issue an error message.
  - 2 If a G33 has been programmed earlier, the starting pitch of G34 will be the pitch of the last G33 programmed.
  - 3 If there is no G33, but a G34 has been programmed earlier, the starting pitch of G34 will be the final pitch of the last G34 programmed.

**Spindle angular position.**

Spindle angular position (within ±359.9999°) for the starting point of the thread. With this parameter, it is possible to make multi-entry threads. Programming it is optional; if not programmed, the function executes the thread at 0° (same as programming Q1=0).

**Increment (K1>0) or decrement (K1<0) of the thread pitch per spindle turn.**

The function executes a thread with an I/J/K pitch in the first turn, I/J/K+K1 in the second one, I/J/K+2\*K1 in the third one and so on. Parameter K1 may be positive (pitch increment) or negative (pitch decrement) with the following limitations.

- If K1 is positive, it cannot be greater than or equal to twice the starting pitch.
- If K1 is positive, when incrementing the pitch while machining, none of the threading axes can exceed its maximum feedrate (parameter MAXFEED).
- If K1 is negative, the pitch while machining cannot reach zero or be negative, otherwise, the CNC will issue the corresponding error message.

The pitch increment depending on the starting pitch, final pitch and distance may be calculated as follows.

$$K1 = ( (\text{final pitch})^2 - (\text{starting pitch})^2 ) / 2 * (\text{distance})$$

10.

ELECTRONIC THREADING AND RIGID TAPPING.  
Electronic threading with variable pitch (G34)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Considerations about execution.

### Beginning of threading.

If threading begins in square (sharp) corner, the pitch increase in the first turn will be half the increment ("K1"/2) and it will be a full increment "K1" in the following turns.

### Interrupt execution ([STOP] key or PLC mark \_FEEDHOL).

The behavior of the CNC when interrupting a threading ([STOP] key or PLC mark \_FEEDHOL) depends on function G233. See "[10.4 Withdraw the axes after interrupting an electronic threading \(G233\).](#)" on page 195.

- If G233 is active, when interrupting the thread, the axes withdraw the distance programmed in that function. If when interrupting the threading the pass is near the end, the CNC ignores G233 and stops the axes at the end of the pass.
- If G233 is not active, when interrupting the threading the axes stop at the end of the pass.

### Spindle home search.

If the spindle has not been homed (referenced), the first G34 will be done automatically when using the master spindle. If the spindle is the master and it has not been homed, it will issue a warning.

### Feedrate behavior.

The threading feedrate depends on the programmed spindle speed and thread pitch (Feedrate = Spindle speed x Pitch). The electronic threading is carried out at 100% of the calculated feedrate and these values cannot be modified from the CNC's operator panel or via PLC.

### Behavior of the spindle speed and of the spindle speed override

If the OEM allows it, (parameter THREADOVR), the user can modify the speed override from the operator panel and, in that case, the CNC will adapt the feedrate automatically respecting the thread pitch. In order to be able to modify the override, the feed forward active on the axes involved in threading must be higher than 90%.

If more than one G34 have been programmed for the same thread, all the threading operations must start at the same speed; otherwise, the entry point (start) to the thread will not be the same in all the threads. The CNC permits changing the spindle override during the thread cutting pass.

If more than one G34 have been programmed for a multi-start (multi-entry) thread, all the threading operations must start at the same speed; otherwise, the angle between the starts (entry points) to the thread will not be the same as the one programmed. The CNC permits changing the spindle override during the thread cutting pass.

## Considerations about thread blending.

When working in round corner (G05), the CNC lets blend different threads seamlessly in a single part. When thread blending, the CNC only takes into consideration the angular position of the spindle (Q1) in the first thread, after activating G33 or G34. The CNC ignores parameter Q1 until this function is canceled and activated back and it is synchronized when going through that angle.

### Blending a thread of fixed pitch (G33) with a thread of variable pitch (G34).

The starting pitch of the variable thread (G34) must be the same as the pitch of the fixed-pitch thread (G33). The pitch increment of the variable thread will be half the increment ("K1"/2) in the first turn and a full increment "K1" in the following turns.

```
G33 Z-40 K2.5
G34 Z-80 K2.5 K1=1
```

# 10.

**ELECTRONIC THREADING AND RIGID TAPPING.**  
Electronic threading with variable pitch (G34)

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

**Blending a thread of variable pitch (G34) with a thread of fixed pitch (G33).**

This combination is used to end a variable-pitch thread (G34) with a portion of the thread that maintains the final pitch of the previous thread. In this case, the pitch is not programmed in the fixed-pitch thread G33 and the CNC will use the last pitch of the previous thread.

```
G34 Z-50 K2 K1=3
G33 Z-100
```

**Blending two variable-pitch threads (G34).**

The starting pitch of the second thread must be the same as the final pitch of the first one. In this case, the pitch is not programmed in the second thread and the CNC will use the last pitch of the previous thread. The pitch increment of the variable thread will be half the increment ("K1"/2) in the first turn and a full increment "K1" in the following turns.

```
G34 Z-50 K2 K1=3
G34 Z-100 K1=-2
```

**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

Function G34 is modal and incompatible with G00, G01, G02, G03, G33, G63 and G100. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G00 or G01 as set by the machine manufacturer (parameter IMOVE).

**10.**

**ELECTRONIC THREADING AND RIGID TAPPING.**  
Electronic threading with variable pitch (G34)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 10.3 Rigid tapping (G63)



*For rigid tapping, the machine must have a rotary encoder installed on the spindle.*

When rigid tapping, the CNC interpolates the movement of the longitudinal axis with the spindle.

### Programming

To define a rigid tapping, program function G63 and then the coordinates of the end point of the thread which may be defined in Cartesian or polar coordinates. The thread pitch will be calculated by the CNC depending on the active feedrate "F" and spindle speed "S" (Pitch = Feedrate / Spindle speed).

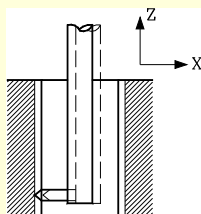
Function G63 starts the spindle in the direction indicated by the programmed speed "S" ignoring the active M3, M4, M5 or M19 functions. A negative turning speed can only be programmed if function G63 is active.

```
...
G94 F300
G01 G90 X30 Y30 Z50
G63 Z20 S200
...
```

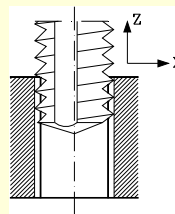
The thread pitch will be:  $\frac{F}{S} = \frac{300}{200} = 1,5\text{mm}$

Since G63 does not withdraw the tool automatically after the tap, an inverted tap must be programmed in order to withdraw the tool by inverting the turning direction of the spindle (by changing the sign of the "S" speed). If the thread is made with a cutter tip, the tool may be also be withdrawn by orienting the spindle (M19) and separating the tool tip away from the thread.

To make a 4 mm pitch thread in X30 Y30 Z0 in a single pass with a depth of 30mm.



```
G94 F400
G01 G90 X30 Y30 Z0
G63 Z-30 S100
M19 S0
G91 G01 X3
G90 Z10
```



```
G94 F400
G01 G90 X30 Y30 Z0
G63 Z-30 S-100
G01 Z10
```

### Multiple-entry threads

With this type of threading, it is possible to make threads with several entry points. The positioning for each entry must be defined before each threading operation.

```
...
G90 G01 X0 Y0 Z0 F150
M19 S0 (First entry at 0°)
G63 Z-50 S150 (Tapping)
G63 Z0 S-150 (Withdrawal)
M19 S120 (Second entry at 120°)
G63 Z-50 S150
```

# 10.

```
G63 Z0 S-150
M19 S240                (Third entry at 240°)
G63 Z-50 S150
G63 Z0 S-150
...
3-entry thread, 50mm deep and 1mm pitch.
```

# 10.

## Considerations for the execution

### Spindle speed behavior

Threading is carried out at the speed defined with function G63. If no particular threading speed is defined, threading will be executed at the speed active at the time. If a speed is programmed with function G63, it will be the active spindle speed at the end of the threading operation.

The spindle turning direction is determined by the sign of the programmed "S" speed ignoring the active M3, M4, M5 or M19 functions. Programming any of these functions will cancel G63.

### Feedrate behavior

While rigid tapping, the feedrate may be varied between 0% and 200% using the feedrate override switch on the CNC's operator panel or via PLC. The CNC will adapt the spindle speed in order to keep the interpolation between the axis and the spindle.

### Rigid tapping and tool inspection mode

When interrupting the rigid tapping and accessing the tool inspection mode, it is possible to jog the axes (only in jog mode) that are involved in threading. When moving the axis, the interpolated spindle will also move; the spindle used to make the thread. If several axes are involved in the rigid tapping, when moving one of the axes all the axes involved in the thread will also move with it.

This way, the axis may be moved into or out of the thread as often as desired until the repositioning softkey is pressed. The axes move at the programmed F except when an axis or spindle exceeds its maximum feedrate allowed (parameter `MAXMANFEED`), in which case, the feedrate will be limited to that value.

The spindle jogging keys are disabled during tool inspection. It is only possible to get out of the thread by jogging one of the axes involved in rigid tapping. Functions M3, M4, M5 and M19 cannot be programmed at the spindle either; these functions are ignored.

During repositioning, selecting one of the axes of the thread on the softkey menu will move all the axes and the spindle involved in the thread.

## Properties of the functions

Function G63 is modal and incompatible with G00, G01, G02, G03 and G33.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G00 or G01 as set by the machine manufacturer [G.M.P. "IMOVE"].

## 10.4 Withdraw the axes after interrupting an electronic threading (G233).



*This feature must be enabled by the OEM in the machine parameters (parameter RETRACTTHREAD); otherwise, when interrupting the execution while threading (using the [STOP] key or the PLC mark \_FEEDHOL) the axes always stop at the end of the pass.*

Function G233 may be used to program the safety distance that the axes will withdraw when interrupting a threading operation (G33/G34) either with the [STOP] key or from the PLC (mark \_FEEDHOL). In the threading canned cycles (G86/G87 of the -T- model) this function is ignored because it is implicit when programming the cycle.

This feature is particularly interesting in very long threads where it may be necessary to interrupt the threading operation either because the tool broke or because the machining is not right; for example due to vibrations on the work piece.

### Programming. Defining and canceling the safety distance for thread exit.

- To define a safety distance, program function G233 and then the distance on each axis.
- To cancel the safety distance on an axis, define a safety distance of zero on the axis.
- To cancel the function, program it alone in the block or define a zero distance on all axes where it is active. In either case, function G233 disappears from the history.

#### Programming format. Defining a safety distance.

The programming format is the following; the list of arguments appears between curly brackets.

G233 X..Z{distance}

X..Z{distance} Thread exit distance on the axis perpendicular to the threading operation.  
Units: millimeters/inches.

G233 X5

(The tool separates 5 mm from the thread on the X axis)

#### Programming format. Canceling the safety distance on an axis.

The programming format is:

G233 X0..Z0

G233 X0

(cancel the safety distance on an axis)

#### Programming format. Deactivate the function.

The programming format is:

G233

G233 X0..Z0

G233

G233 X0 Z0

Function G233 alone in the block also indicates the point where execution will resume after pressing [START].

#### Thread exit distance on the axis perpendicular to the threading operation.

The safety distance is only defined on the axis perpendicular to the threading operation; it is ignored on the rest of the axes. On outside longitudinal threads will be a positive distance and on inside ones it will be a negative distance.

On taper threading, the threading axis will be the one on which the pitch has been defined.

# 10.

**ELECTRONIC THREADING AND RIGID TAPPING.**

Withdraw the axes after interrupting an electronic threading (G233).

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## Programming. Define the block to resume the block to resume execution after pressing [START].

To resume execution, press the [START] key; execution resumes at the next block where function G233 is programmed alone.

### Operation.

The option to withdraw the axes when interrupting a threading operation depends on the configuration of the machine (parameter RETRACTTHREAD).

RETRACTTHREAD	Meaning.
ON	The CNC behavior depends on function G233.
OFF	The CNC ignores function G233 and stops the axes at the end of the threading operation.

When the feature is enabled (parameter RETRACTTHREAD), the CNC behaves as follows when interrupting a threading operation ([STOP] key or PLC mark \_FEEDHOL).

- If G233 is active, the axis perpendicular to threading withdraws the programmed distance from the part. The threading axis withdraws the necessary distance as not to damage the thread while maintaining the pitch.
- If G233 is active, and when interrupting the threading the pass is near the end, the CNC ignores G233 and stops the axes at the end of the pass.
- If G233 is not active, the axes stop at the end of the pass.

### Feedrate of the axes.

The axis perpendicular to threading withdraws from the part at the feedrate set in parameter MAXFEED of the active set. The threading axis maintains the pitch.

### Considerations and limitations.

#### G233 with several G33/G34 functions in a row.

Function G233 sets the thread exit distance for all G33/G34 threads programmed after it. If there are several G33/G34 functions in a row and a different thread exit is desired for each one of them, the corresponding G233 function must be programmed before each one of them.

#### Thread blending.

If there are several threads in a row (thread blending), function G233 considers all of them completed.

#### Threading canned cycles, ISO and conversational (-T- model).

Function G233 is only applied in electronic threading G33/G34; it is ignored in threading canned cycles, both ISO and conversational, because it is implicit in the cycles, programmed as thread exit.

- In canned cycles where a thread exit has been programmed, the withdrawal distance for the axis perpendicular to the threading is calculated automatically and corresponds to the value of that thread exit of each pass.
- In cycles where the thread exit has not been programmed, the behavior depends on parameter RETRACTTHREAD.

RETRACTTHREAD	Meaning.
ON	The tool withdraws to the safety coordinate, perpendicularly to the threading axis (just the same as when having a thread exit).
OFF	The axes stop at the end of the pass.

Once the tool has withdrawn the programmed distance, it returns to the starting point of the cycle. The machine waits for the CYCLE START command to repeat the aborted pass.

10.

ELECTRONIC THREADING AND RIGID TAPPING.  
Withdraw the axes after interrupting an electronic threading (G233).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Threading canned cycles, ISO and conversational (-T- model).**

The option to withdraw the axes when interrupting a threading operation depends on the configuration of the machine (parameter RETRACTTHREAD).

RETRACTTHREAD	Meaning.
ON	The CNC interrupts the threading and withdraws the axes.
OFF	The CNC stops the axes at the end of the pass.

The CNC ignores function G233 in threading canned cycles, both ISO and conversational, because it is implicit in the cycles, programmed as thread exit. When the feature is enabled (parameter RETRACTTHREAD), the CNC behaves as follows when interrupting a threading operation ([STOP] key or PLC mark \_FEEDHOL).

- In canned cycles where a thread exit has been programmed, the withdrawal distance for the axis perpendicular to the threading is calculated automatically and corresponds to the value of that thread exit of each pass.
- In cycles where the thread exit has not been programmed, the axes withdraw to the safety coordinate, perpendicularly to the threading axis (just the same as when having a thread exit).

Once the tool has withdrawn the programmed distance, it returns to the starting point of the cycle. The machine waits for the CYCLE START command to repeat the aborted pass.

**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

Function G233 is modal. On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC cancels this function.

**10.****ELECTRONIC THREADING AND RIGID TAPPING.**

Withdraw the axes after interrupting an electronic threading (G233).

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 10.4.1 Variables related to G233.

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

Variable.	PRG	Meaning.
V.[ch].G.RETREJ	R	The user has interrupted a thread and the CNC has withdrawn the axes from the thread. (0 = The CNC has resumed execution, or M30 or reset) (1 = The axes have reached the programmed distance)

#### Syntax of the variables.

```
V.G.RETREJ
```

### 10.4.2 Programming example.

Threading example with G33 and G233 where a thread exit has been programmed so that when interrupting execution, the axes stop at the starting point and repeat the threading operation.

```
N10 G90 G18 S500 M3
N20: G0 X20
N30 Z5
N50 X10
N60 G233 X5
      (Withdrawal from the thread)
N70 G33 Z30 K5
      (Threading block that may be interrupted with [STOP])
N80 G33 Z50 X15 K5
      (Thread exit block)
N90 G233
N100 $IF V.G.RETREJ == 0 $GOTO N120
N110 $GOTO N20
N120 ...
```

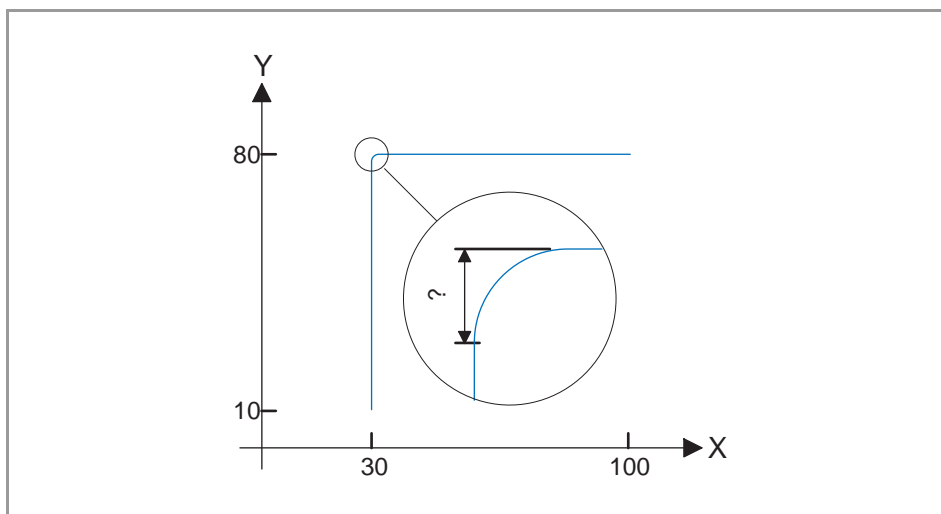
If a [STOP] occurs in block N70, the CNC interrupts the threading and withdraws the axes according to block N60. After withdrawing the axes, the CNC considers blocks N70 and N80 completed and goes on to execute block N90.

# 10.

ELECTRONIC THREADING AND RIGID TAPPING.  
Withdraw the axes after interrupting an electronic threading (G233).

## 11.1 Semi-rounded corner (G50).

The G50 function activates the semi-rounded edge machining. When working in semi-rounded corner, the CNC starts executing the next movement once the theoretical interpolation of the current move is completed without waiting for the axes to be in position. The distance from the programmed position to the position where the next move begins depends on the programmed feedrate. This function gives rounded edges.



### Programming.

The function G50 may be programmed alone in the block or it may be added to a movement block.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G50 <X..C{position}>

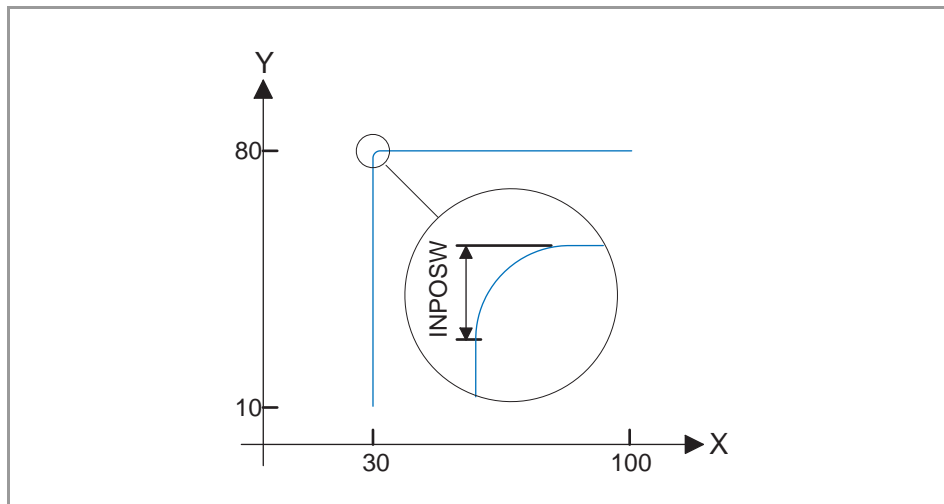
G50 (semi-rounded)  
G01 G91 Y70 F500

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G50 is modal and incompatible with G05, G07, G60, G61 and the HSC mode.
- On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G05, G07 or G50 as set by the OEM (parameter ICORNER).

## 11.2 Square corner (G07/G60).

Functions G07 (modal) and G60 (non-modal) activate the square edge machining. On a square edge, the CNC begins executing the next movement when it finishes the current interpolation of the current movement and when the axes are in position (parameter INPOSW). The theoretical and real profiles are the same, resulting in square edges.



# 11.

**GEOMETRIC ASSISTANCE:**  
Square corner (G07/G60).

### Programming.

The function G07 may be programmed alone in the block or it may be added to a movement block. Function G60 only affects the block where it has been programmed, therefore, it can only be added to a block that defines a movement (motion block).

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G07 <X..C{position}>

G60 X..C{position}>

X..C{position}      Optional. Position of the axes.  
Units: Millimeters, inches or degrees.

```
G07 Square edge (modal)
G01 X70
G01 G91 G60 Y70 F500 Square edge (non-modal)
```

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G07 is modal and incompatible with G05, G50, G60, G61 and the HSC mode. The G60 function is non-modal; after it is executed, the CNC restores the function G05, G07, G50 or HSC that was previously active.
- On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G05, G07 or G50 as set by the OEM (parameter ICORNER).



FAGOR AUTOMATION

CNC 8070

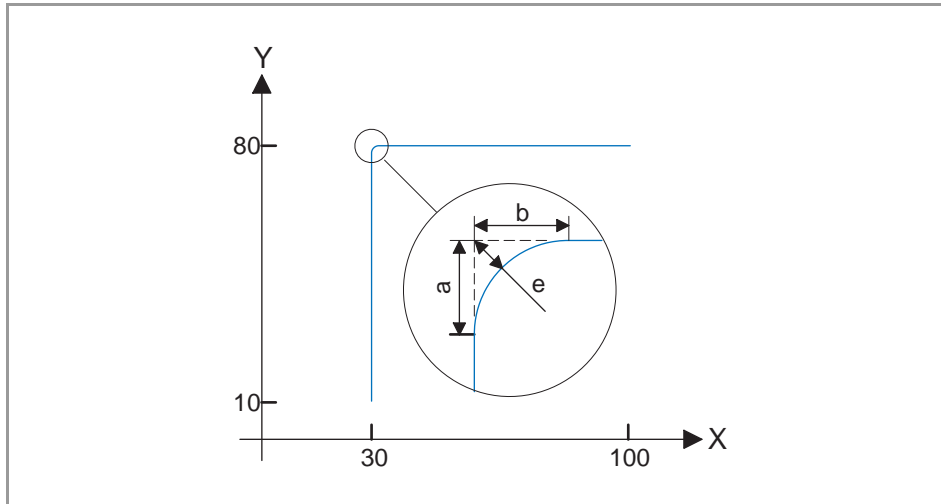
(REF: 2008)



### 11.3 Controlled corner rounding (G05/G61).

Functions G05, (modal) and G61 (non-modal) activate round edge machining. The CNC can configure five different types of edge rounding (instruction #ROUNDPAR). This operation may be applied to any corner, regardless of whether it is defined between straight and/or circular paths.

The corner is machined along a curved path, not with arcs. The shape of the curve depends on the type of corner rounding selected and on the dynamic conditions (feedrate and acceleration) of the axes involved.



11.

GEOMETRIC ASSISTANCE:  
Controlled corner rounding (G05/G61).

#### Programming.

The function G05 may be programmed alone in the block or it may be added to a movement block. Function G61 only affects the block where it has been programmed, therefore, it can only be added to a block that defines a movement (motion block).

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G05 <X..C{position}>

G61 X..C{position}>

X..C{position}      Optional. Position of the axes.  
Units: Millimeters, inches or degrees.

G05 Round edge (modal)  
G01 X70  
G01 G91 G61 Y70 F500 Round edge (non-modal)

#### Considerations.

The type of corner rounding is selected with the "#ROUNDPAR" instruction and stays active until a different one is selected.

#### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G05 is modal and incompatible with G07, G50, G60, G61 and the HSC mode. The G61 function is non-modal; after it is executed, the CNC restores the function G05, G07, G50 or HSC that was previously active.
- On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes function G05, G07 or G50 as set by the OEM (parameter ICORNER).

FAGOR 

FAGOR AUTOMATION

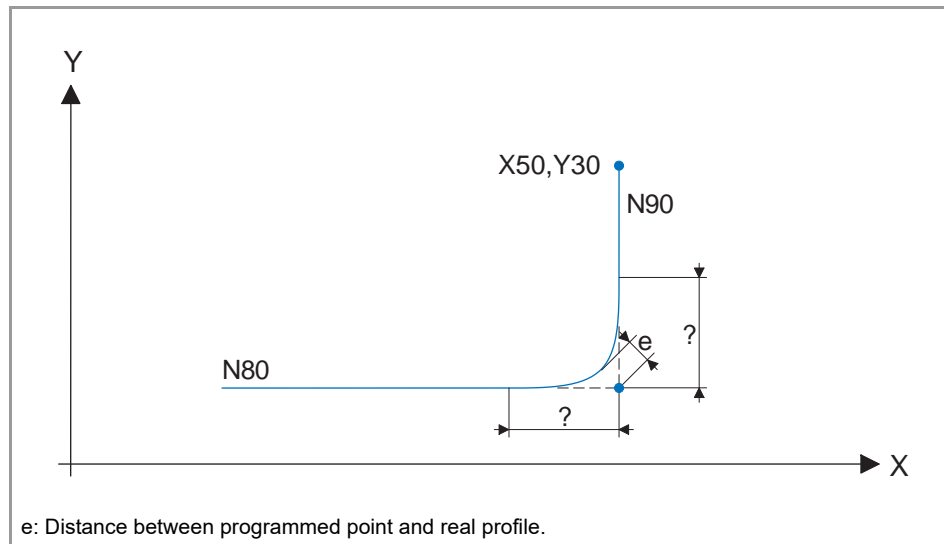
CNC 8070

(REF: 2008)

### 11.3.1 Edge rounding. #ROUNDPAR [1].

This sets the maximum deviation allowed between the programmed point and the profile resulting from the edge rounding. The distances from the programmed point to the points where the corner rounding begins and ends are calculated automatically and they cannot be greater than half the path programmed in the block. Both distances will be the same, except when one of them is limited to half the programmed path.

The corner is rounded by giving priority to the machining dynamic conditions (feedrate and acceleration). The CNC executes the machining operation closest to the programmed point without exceeding the programmed deviation and that does not require decreasing the programmed feedrate "F".



#### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

#ROUNDPAR [1,{error}]

{error}                      Distance between programmed point and real profile.  
Units: Millimeters, inches or degrees.

```
#ROUNDPAR [1,3]
G05 Round edge (modal)
N80 G01 G91 X50 F850
N90 G01 Y30
```

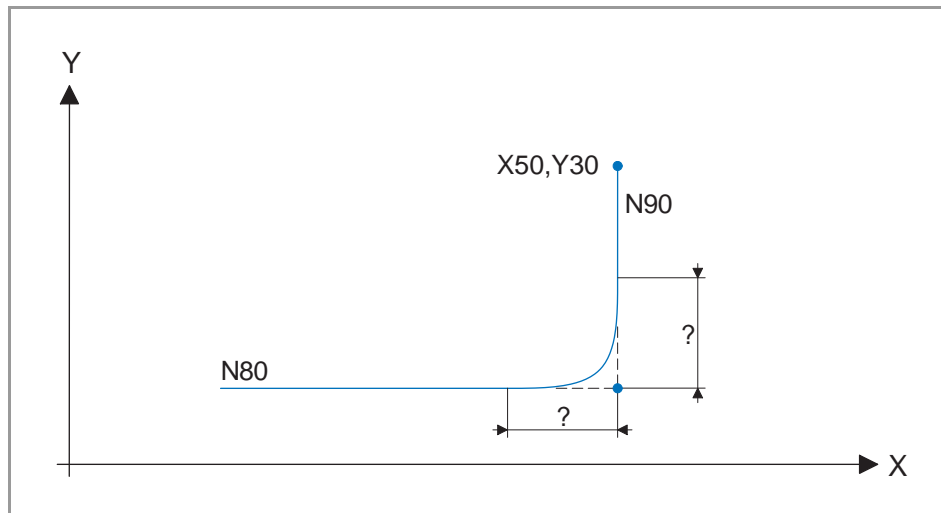
# 11.

**GEOMETRIC ASSISTANCE:**  
Controlled corner rounding (G05/G61).

### 11.3.2 Edge rounding. #ROUNDPAR [2].

This sets the percentage of the active feedrate "F" to be used to carry out the edge rounding. The distances from the programmed point to the points where the corner rounding begins and ends are calculated automatically and they cannot be greater than half the path programmed in the block. Both distances will be the same, except when one of them is limited to half the programmed path.

The CNC executes the edge rounding closest to the programmed point and that may be machined at the set feedrate percentage.



#### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

#ROUNDPAR [2, {feedrate}]

{feedrate}                      Percentage of feedrate "F" for corner rounding.  
Units: %.

```
#ROUNDPAR [2,40]
G05 Round edge (modal)
N80 G01 G91 X50 F850
N90 G01 Y30
```

# 11.

**GEOMETRIC ASSISTANCE:**  
Controlled corner rounding (G05/G61).

**FAGOR** 

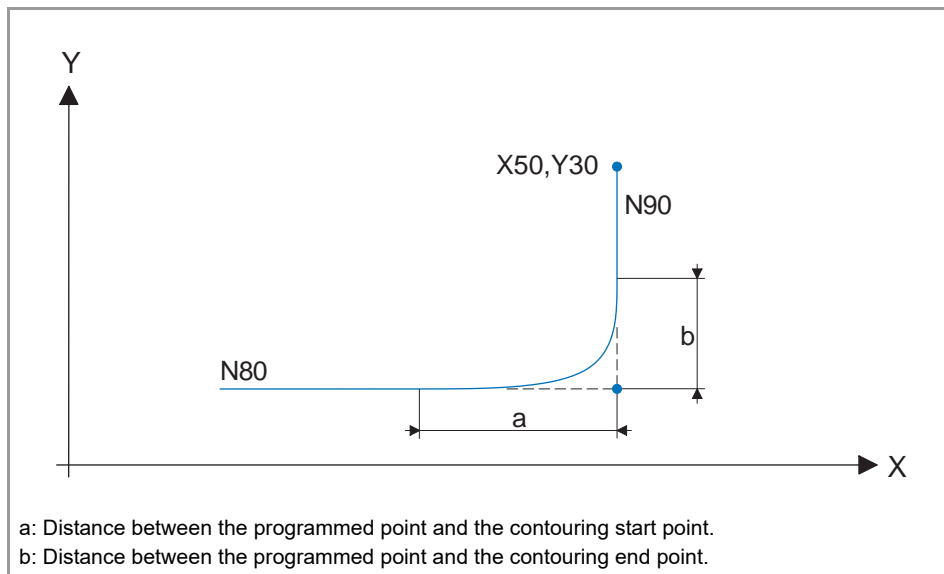
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 11.3.3 Edge rounding. #ROUNDPAR [3].

This defines the distance from the programmed point to the points where edge rounding begins and ends.



#### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

```
#ROUNDPAR [3,{start_distance},{end_distance}]
```

{start\_distance} Distance between the programmed point and the contouring start point.  
Units: Millimeters, inches or degrees.

{end\_distance} Distance between the programmed point and the contouring end point.  
Units: Millimeters, inches or degrees.

```
#ROUNDPAR [3,10,3]
G05 Round edge (modal)
N80 G01 G91 X50 F850
N90 G01 Y30
```

#### Considerations.

Depending on the programmed distances, there may be a deviation in the programmed profile and the resulting one.

# 11.

**GEOMETRIC ASSISTANCE:**  
Controlled corner rounding (G05/G61).



FAGOR AUTOMATION

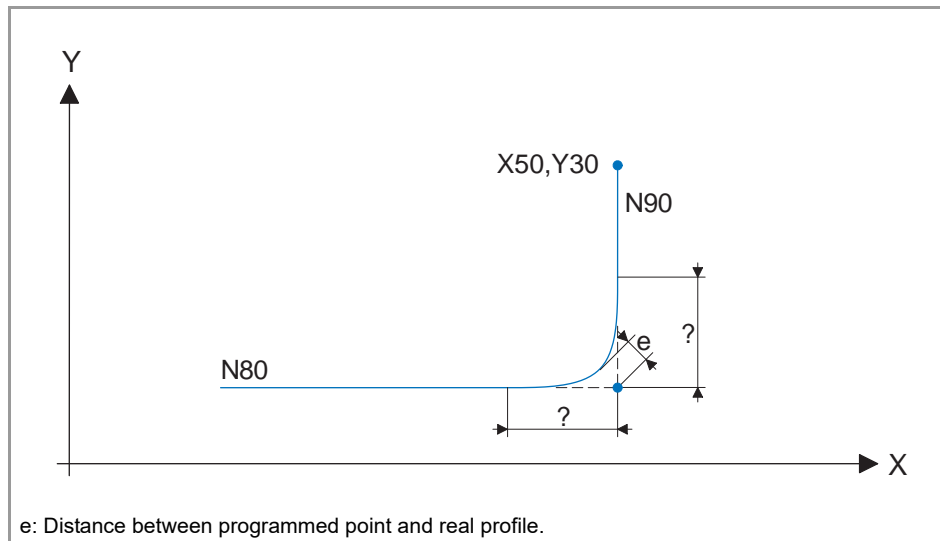
CNC 8070

(REF: 2008)

### 11.3.4 Edge rounding. #ROUNDPAR [4].

This sets the maximum deviation allowed between the programmed point and the profile resulting from the edge rounding. The distances from the programmed point to the points where the corner rounding begins and ends are calculated automatically and they cannot be greater than half the path programmed in the block. Both distances will be the same, except when one of them is limited to half the programmed path.

The corner is rounded by giving priority to the machining geometrical conditions. The programmed machining operation is executed by the CNC, by decreasing the programmed feedrate "F" if necessary.



#### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

```
#ROUNDPAR [4,{error}]
```

{error}                      Distance between programmed point and real profile.  
Units: Millimeters, inches or degrees.

```
#ROUNDPAR [4,3]
G05 Round edge (modal)
N80 G01 G91 X50 F850
N90 G01 Y30
```

# 11.

**GEOMETRIC ASSISTANCE:**  
Controlled corner rounding (G05/G61).

**FAGOR** 

FAGOR AUTOMATION

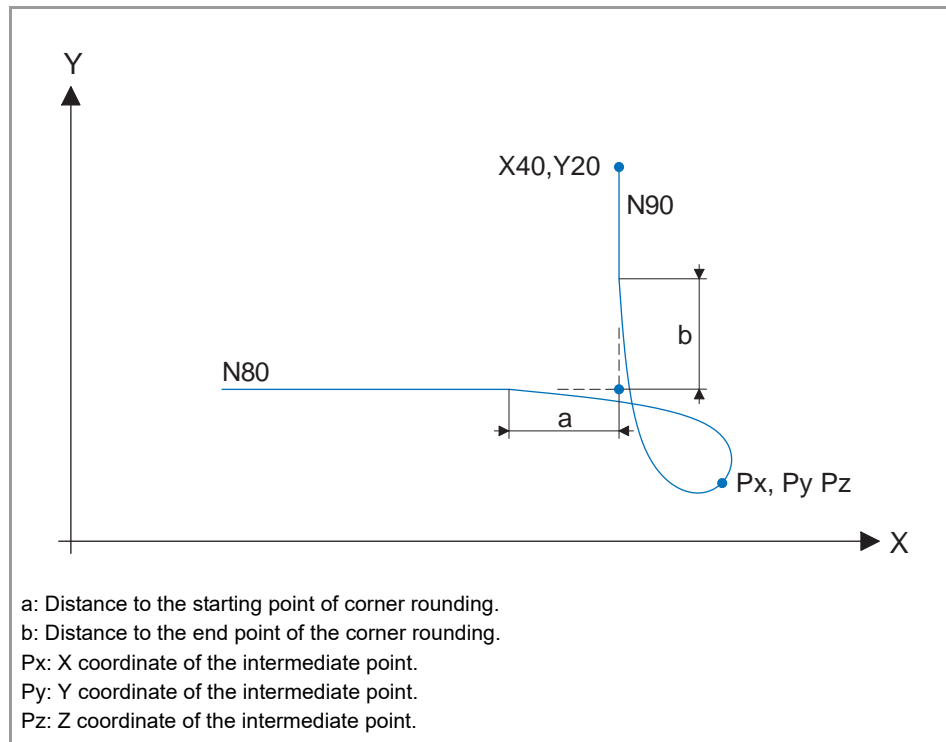
**CNC 8070**

(REF: 2008)

### 11.3.5 Edge rounding. #ROUNDPAR [5].

This defines the distance from the programmed point to the points where edge rounding begins and ends, and the coordinates of an intermediate point. The shape of the curve depends on the position of the intermediate point and on the distance from the programmed point to the starting and ending points of the edge rounding (see examples).

This executes a square edge and is aimed at laser, water jet, machines, etc. that use it to avoid “burning” the edge, thus not being recommended for a milling machine.



#### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

#ROUNDPAR [5,{start\_distance},{end\_distance},{point\_x},{point\_y},{point\_z}]

{start\_distance} Distance between the programmed point and the contouring start point.  
 Units: Millimeters, inches or degrees.

{end\_distance} Distance between the programmed point and the contouring end point.  
 Units: Millimeters, inches or degrees.

{axis\_1} Coordinates of the intermediate point on the abscissa axis.  
 Units: Millimeters, inches or degrees.

{axis\_2} Coordinates of the intermediate point on the ordinate axis.  
 Units: Millimeters, inches or degrees.

{axis\_3} Coordinates of the intermediate point on the axis perpendicular to the plane.  
 Units: Millimeters, inches or degrees.

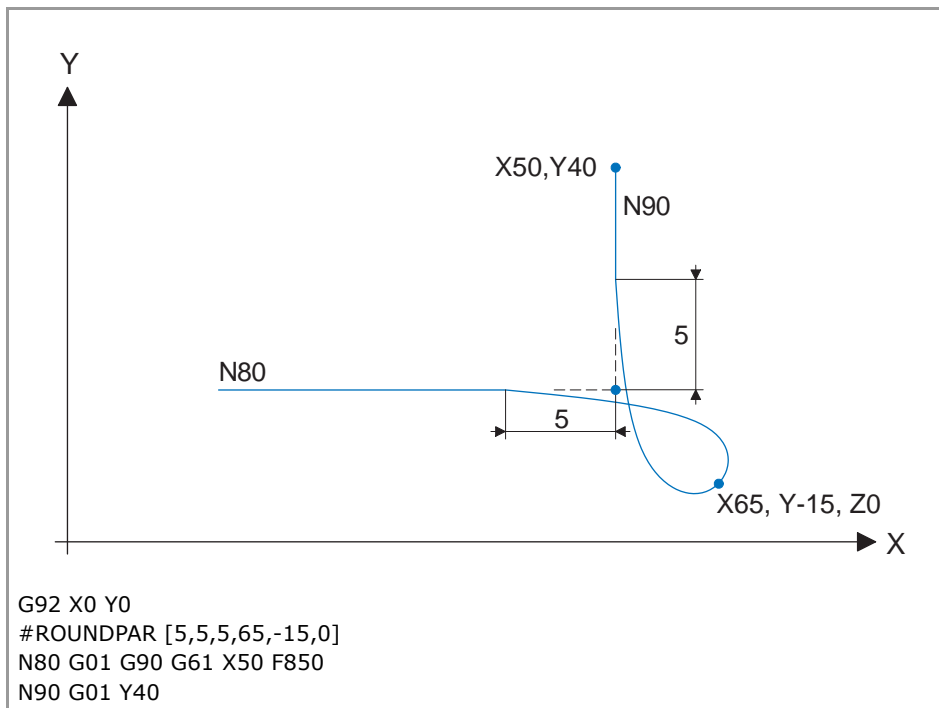
```
#ROUNDPAR [5,7,4,55,-15,0]
G05 Round edge (modal)
N80 G01 G91 X40 F850
N90 G01 Y20
```

# 11.

GEOMETRIC ASSISTANCE:  
 Controlled corner rounding (G05/G61).

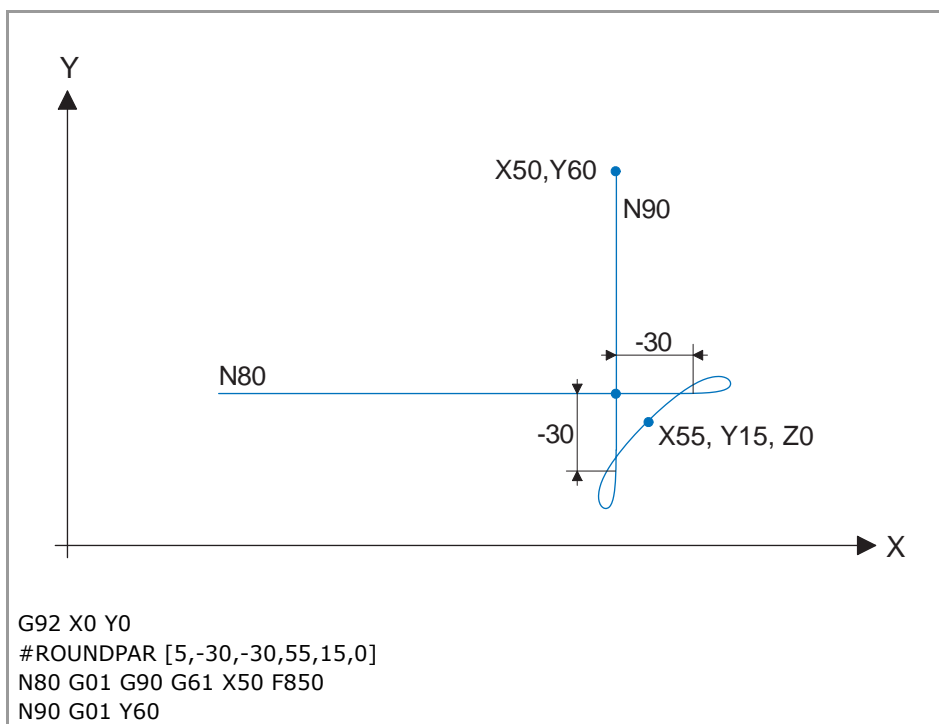
### Example 1.

Positive start and end distances.



### Example 2.

Negative and longer start and end distances (in absolute value) than the distance from the programmed point to the intermediate point on each axis (about 4 times).



# 11.

**GEOMETRIC ASSISTANCE:**  
Controlled corner rounding (G05/G61).

**FAGOR** 

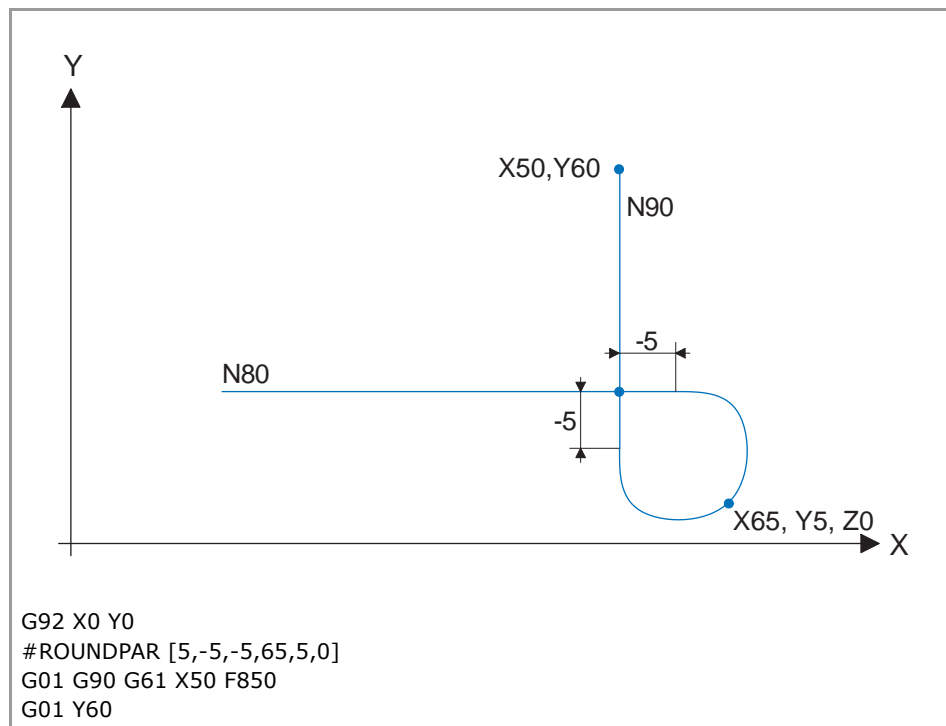
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### Example 3.

Negative and shorter start and end distances (in absolute value) than the distance from the programmed point to the intermediate point on each axi.



11.

**GEOMETRIC ASSISTANCE:**  
Controlled corner rounding (G05/G61).



FAGOR AUTOMATION

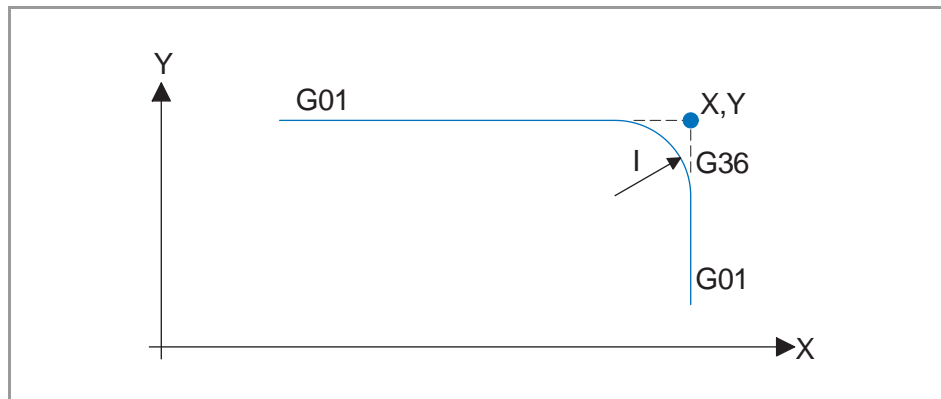
CNC 8070

(REF: 2008)



## 11.4 Corner rounding, radius blend, (G36).

Function G36 rounds a edge with the defined radius, without the need to calculate the center or the start and end points of the arc.



### Programming.

Program function G36 with its parameters alone in the block and between the two tool paths defining the edge to be rounded. These paths may be linear and/or circular.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G36 <I{radius}> <F{feedrate}>

I{radius}	Optional. Rounding radius. Units: mm or inches.
F{feedrate}	Optional. Feedrate. Units: Units are dependent on the active function. - If G93, seconds. - If G94, millimeters/minute or inches/minute. - If G95, millimeters/minute or inches/minute.

```
G01 G94 X10 Y50
G36 I5 F150 (Rounding; radius=5)
G01 X50 Y50
G36 (Rounding; radius=5)
G01 X50 Y10
```

### Rounding radius.

The "I" value of the rounding radius remains active until another value is programmed, therefore, it won't be necessary to program it in successive rounding operations with the same radius. The "I" value of the rounding radius is also used by functions G37, G38 and G39. This means that the rounding radius set in G36 will be the new value of the tangential entry radius (G37), tangential exit radius (G38) or chamfer size (G39) when programming one of these functions or vice versa.

```
N10 G01 X10 Y10 F600
N20 G01 X10 Y50
N30 G36 I5 (Rounding; radius=5)
N40 G01 X50 Y50
N50 G36 (Rounding; radius=5)
N60 G01 X50 Y10
N70 G39 (Chamfer; size=5)
N80 G01 X90 Y10
N90 G39 I10 (Chamfer; size=10)
N100 G01 X90 Y50
N110 G36 (Rounding; radius=10)
N120 G01 X70 Y50
N130 M30
```

# 11.

**GEOMETRIC ASSISTANCE:**  
Corner rounding, radius blend, (G36).

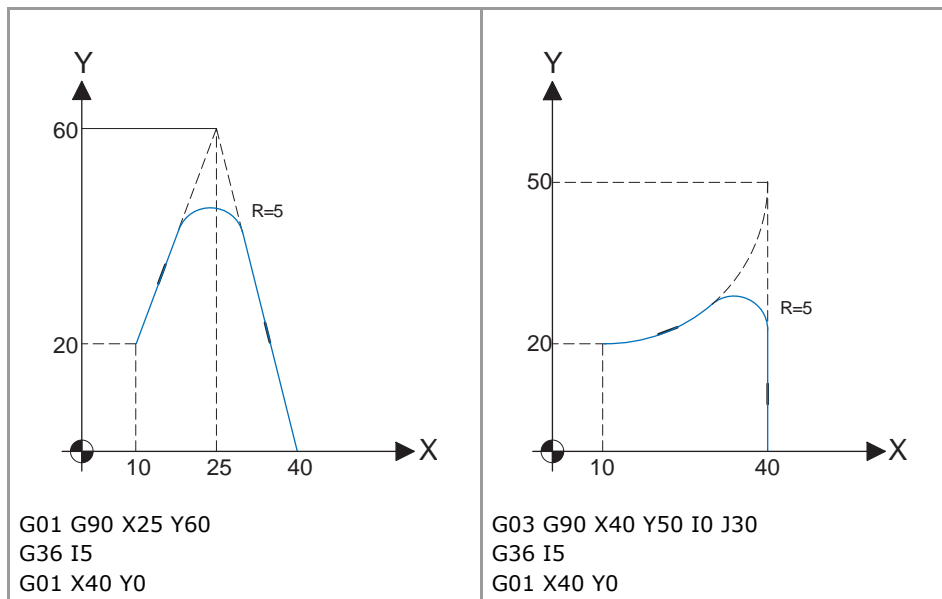
**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## Programming examples.



11.

**GEOMETRIC ASSISTANCE:**  
Corner rounding, radius blend, (G36).

## Considerations.

## Machining feedrate.

The programmed rounding feedrate depends on the type of movement programmed afterwards:

- If the next movement is in G00, the rounding will be carried out in G00.
- If the next movement is in G01, G02 or G03, the rounding will be carried out at the feedrate programmed in rounding definition block. If no feedrate has been programmed, the rounding will be carried out at the active feedrate.

```

N10 G01 G94 X10 Y10 F600
N20 G01 X10 Y50
N30 G36 I5 (Rounding; G00)
N40 G00 X50 Y50
N50 G36 (Rounding; F=600mm/min)
N60 G01 X50 Y10
N70 G36 F300 (Rounding; F=300mm/min)
N80 G01 X90 Y10 F600
N90 M30

```

## Plane change.

When defining a plane change between the two paths that define a rounding, it is carried out in the plane where the second path is defined.

```

N10 G01 G17 X10 Y10 Z-10 F600
N20 X10 Y50 Z0 (X-Y plane)
N30 G36 I10
N40 G18 (Z-X plane; the rounding is carried out in this plane)
N50 X10 Z30
N60 M30

```

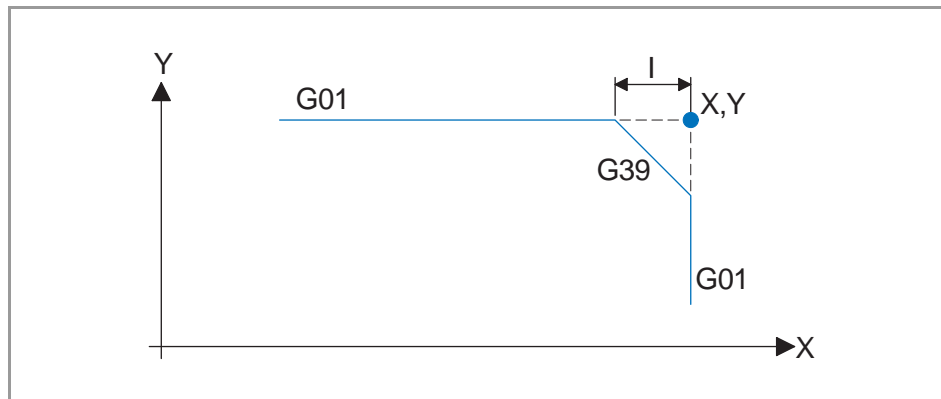
## Function properties.

Function G36 is not modal, therefore, it must be programmed every time a corner is to be rounded.

(REF: 2008)

## 11.5 Corner chamfering, (G39).

The defined size chamfering G39 function without having to calculate the intersection points.



### Programming.

Program function G39 with its parameters alone in the block and between the two tool paths defining the edge to be chamfered. These paths may be linear and/or circular.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G36 <I{size}> <F{feedrate}>

I{size}	Optional. Chamfer size. Units: mm or inches.
F{feedrate}	Optional. Feedrate. Units: Units are dependent on the active function. - If G93, seconds. - If G94, millimeters/minute or inches/minute. - If G95, millimeters/minute or inches/minute.

```
G01 G94 X10 Y50
G39 I5 F150 (Chamfer; size=5)
G01 X50 Y50
G39 (Chamfer; size=5)
G01 X50 Y10
```

### Chamfer size.

The "I" value of the chamfer size remains active until another value is programmed, therefore, it won't be necessary to program it in successive chamfering operations of the same size.

The "I" value of the chamfer size is also used by functions G36, G37 y G38. This means that the chamfer size set in G39 will be the new value of the tangential entry radius (G37), tangential exit radius (G38) or rounding radius (G36) when programming one of these functions or vice versa.

```
N10 G01 X10 Y10 F600
N20 G01 X10 Y50
N30 G36 I5 (Rounding; radius=5)
N40 G01 X50 Y50
N50 G36 (Rounding; radius=5)
N60 G01 X50 Y10
N70 G39 (Chamfer; size=5)
N80 G01 X90 Y10
N90 G39 I10 (Chamfer; size=10)
N100 G01 X90 Y50
N110 G36 (Rounding; radius=10)
N120 G01 X70 Y50
N130 M30
```

# 11.

**GEOMETRIC ASSISTANCE:**  
Corner chamfering, (G39).

**FAGOR** 

FAGOR AUTOMATION

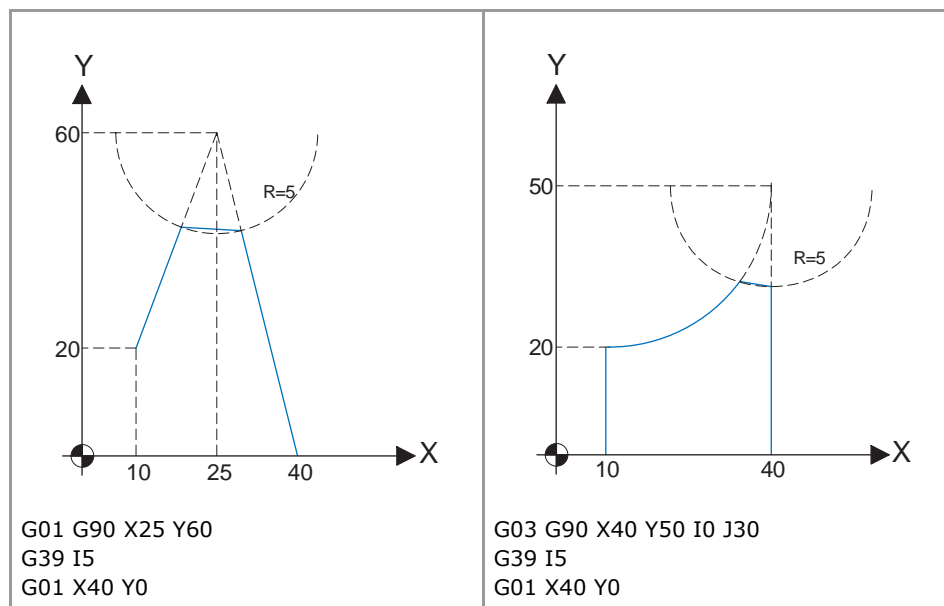
**CNC 8070**

(REF: 2008)

## Programming examples.

11.

**GEOMETRIC ASSISTANCE:**  
Corner chamfering, (G39).



## Considerations.

## Machining feedrate.

The programmed chamfering feedrate depends on the type of movement programmed afterwards:

- If the next movement is in G00, the chamfer will be carried out in G00.
- If the next movement is in G01, G02 or G03, the chamfer will be carried out at the feedrate programmed in chamfer definition block. If no feedrate has been programmed, the chamfer will be carried out at the active feedrate.

```

N10 G01 G94 X10 Y10 F600
N20 G01 X10 Y50
N30 G39 I5 (Corner chamfering in G00)
N40 G00 X50 Y50
N50 G39 (Chamfer; F=600mm/min)
N60 G01 X50 Y10
N70 G39 F300 (Chamfer; F=300mm/min)
N80 G01 X90 Y10 F600
N90 M30
  
```

## Plane change.

When defining a plane change between the two paths that define a chamfer, it is carried out in the plane where the second path is defined.

```

N10 G01 G17 X10 Y10 Z-10 F600
N20 X10 Y50 Z0 (X-Y plane)
N30 G39 I10
N40 G18 (Z-X plane; the chamfer is carried out in this plane)
N50 X10 Z30
N60 M30
  
```

## Function properties.

Function G39 is not modal, therefore, it must be programmed every time a corner is to be chamfered.

(REF: 2008)

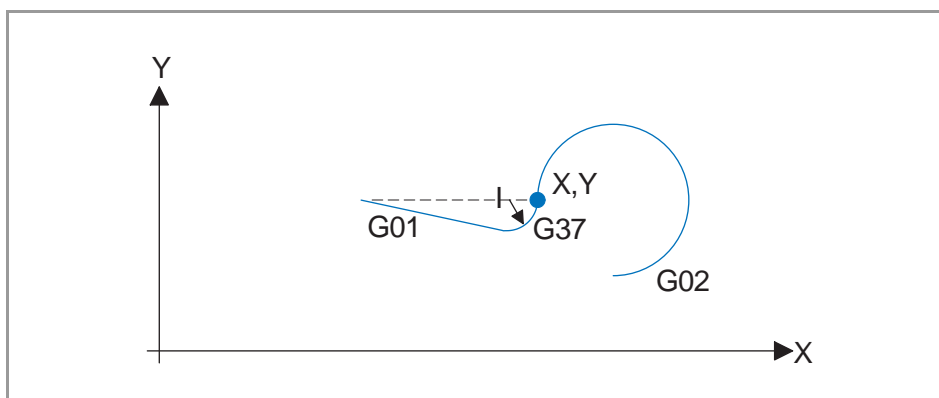


FAGOR AUTOMATION

CNC 8070

## 11.6 Tangential entry (G37).

Function G37 may be used to begin machining with a tangential entry of the tool without having to calculate the intersection points.



### Programming.

Program function G37 with its parameters alone in the block after the tool path to be modified. This tool path must be straight (G00 or G01).

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G37 <I{radius}>

I{radius}                      Optional. Radius of the tangential entry.  
Units: mm or inches.

```
G01 G90 X40 Y50 F800
G37 I10
G02 X70 Y20 I30 J0
```

### Radius of the tangential entry.

The "I" value of the tangential entry radius remains active until another value is programmed, therefore, it won't be necessary to program it in successive tangential entries with the same radius.

The "I" value of the entry radius is also used by functions G36, G38 and G39. This means that the entry radius set in G37 will be the new value of the tangential exit radius (G38), rounding radius (G36) or chamfer size (G39) when programming one of these functions or vice versa.

### Considerations.

The linear path before the tangential entry must have a length equal to or greater than twice the entry radius. Likewise, the radius must be positive and when working with tool radius compensation, it must be greater than the tool radius.

### Function properties.

Function G37 is not modal, therefore, it must be programmed every time a tangential entry is to be carried out.

# 11.

**GEOMETRIC ASSISTANCE:**  
Tangential entry (G37).

**FAGOR** 

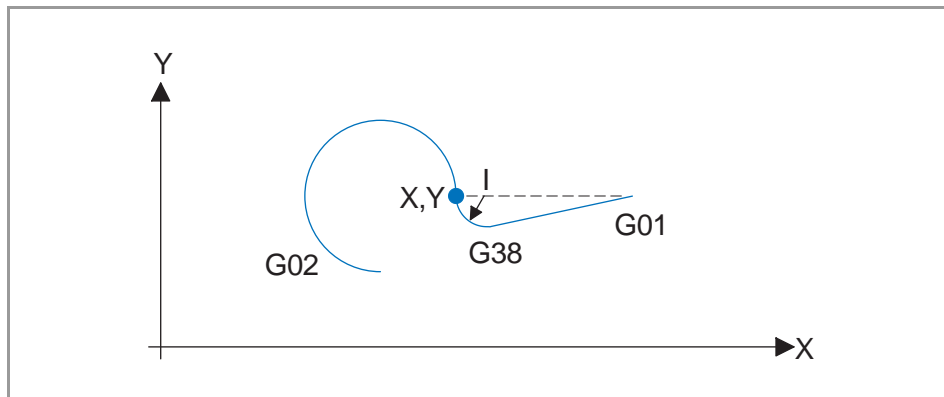
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 11.7 Tangential exit (G38).

Function G38 may be used to end machining with a tangential exit of the tool without having to calculate the intersection points.



### Programming.

Program function G38 with its parameters alone in the block before the tool path to be modified. This tool path must be straight (G00 or G01).

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G38 <I{radius}>

I{radius}                      Optional. Tangential exit radius.  
Units: mm or inches.

```
G02 X60 Y40 I20 J0 F800
G38 I10
G01 X100
```

### Tangential exit radius.

The "I" value of the tangential exit radius remains active until another value is programmed, therefore, it won't be necessary to program it in successive tangential exits with the same radius.

The "I" value of the exit radius is also used by functions G36, G37 and G39. This means that the exit radius set in G38 will be the new value of the tangential entry radius (G37), rounding radius (G36) or chamfer size (G39) when programming one of these functions or vice versa.

### Considerations.

The linear path after the tangential exit must have a length equal to or greater than twice the exit radius. Likewise, the radius must be positive and when working with tool radius compensation, it must be greater than the tool radius.

### Function properties.

Function G38 is not modal, therefore, it must be programmed every time a tangential exit is to be carried out.

# 11.

**GEOMETRIC ASSISTANCE:**  
Tangential exit (G38).



FAGOR AUTOMATION

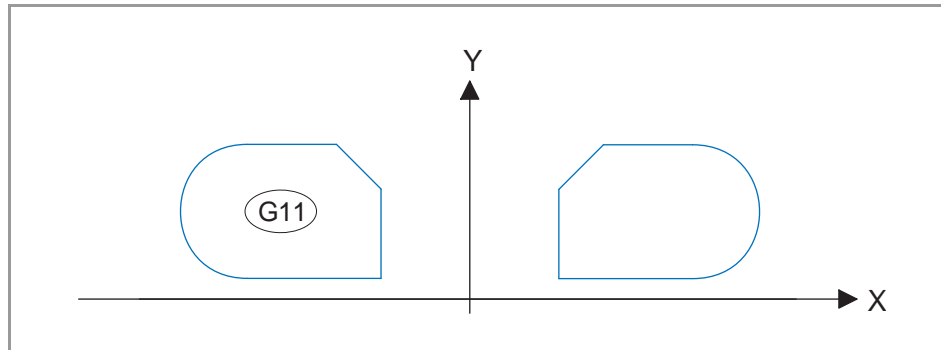
CNC 8070

(REF: 2008)

## 11.8 Mirror image (G10, G11, G12, G13, G14).

### 11.8.1 Activating the mirror image (G11, G12, G13, G14).

Functions G11 to G13 activate the mirror image on the main axes. The G14 function activates or deactivates the mirror image on any axis. Mirror image may be used to repeat the programmed machining operation in a symmetrical position with respect one or more axes. When using with mirror image, the movements of the axes where mirror image is applied are executed with the opposite sign.



G11	Mirror image on X.
G12	Mirror image on Y.
G13	Mirror image on Z.
G14	Mirror image in the programmed directions.

### Programming. G11, G12, G13.

Functions G11 to G13 can be combined inside one block and added to a motion block. If added to a tool path defining block, the mirror image will be activated before the movement. These functions do not cancel each other, thus being possible to keep mirror image active on several axes at the same time.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G11 G12 G13 <X..C{position}>

X..C{position}      Optional. Position of the axes.  
Units: Millimeters, inches or degrees.

G11 G12 G13

### Programming. G14.

Programming the G14 function with its parameters alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G14 X..C{value}

X..C{value}      Position of the axes.  
value=1: Cancels the mirror image to the axis  
value=-1: Activates the mirror image to the axis

G14 X-1 V-1 (Mirror image on the X and V axes)

G14 X1 (Mirror image cancellation on the X axis; the one on the V axis remains active)

G14 V1 (Mirror image cancellation on the V axis)

# 11.

**GEOMETRIC ASSISTANCE:**  
Mirror image (G10, G11, G12, G13, G14).

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### Considerations.

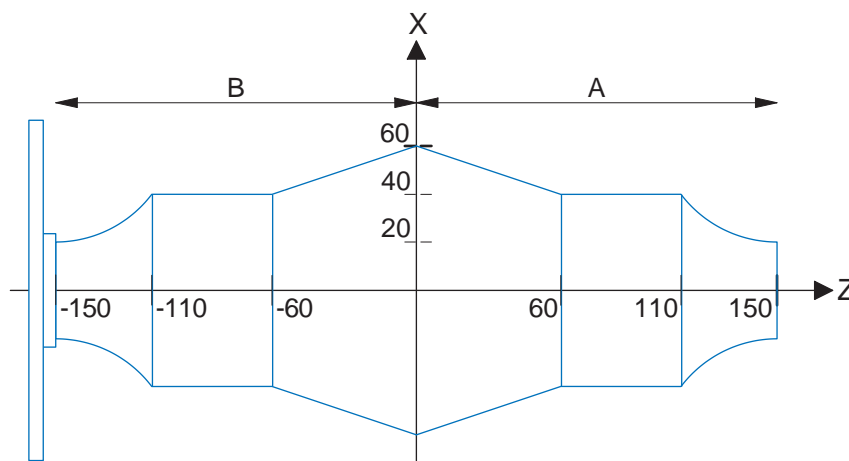
When machining a profile with a mirror image, the machining direction is opposite to that of the programmed profile. If this profile has been defined with tool radius compensation, when activating the mirror image, the CNC will change the type of compensation (G41 or G42) to obtain the programmed profile.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Functions G10, G11, G12, G13 and G14 are modal. Once mirror image is active on an axis, it stays active until canceled with G10 or G14.
- Functions G10 and G14 are incompatible with each other as well as with G11, G12 and G13.
- On power-up and after an emergency, the CNC cancels mirror image (it assumes function G10). The behavior of the mirror image function after executing an M02 or M30 and after a reset depends on the OEM setting (MIRRORCANCEL parameter).

MIRRORCANCEL	Meaning.
Yes	Functions M02, M30 and reset cancel the mirror image function.
No	Functions M02, M30 and reset do not affect the mirror image function.

### Programming example (T model).



%L PROFILE (Subroutine that defines the "A" zone of the part)

```
G90 G00 X40 Z150
G02 X80 Z110 R60
G01 Z60
G01 X124 Z-6
M29 (End of subroutine)
```

%PROGRAM (Main program)

```
G18 G151 (Main plane ZX and programming in diameters)
V.A.ORG[1].Z=160 (Definition of the first zero offset, G54)
G54 (Application of the zero offset)
LL PROFILE (Call to a subroutine; machining of zone "A")
G0 Z-150 (Movement to avoid colliding with the part)
G13 (Mirror image in Z)
LL PROFILE (Call to a subroutine; machining of zone "B")
G0 Z-200 (Return to starting point)
G10 (Cancel mirror image on all the axes)
M30
```

11.

GEOMETRIC ASSISTANCE:

Mirror image (G10, G11, G12, G13, G14).



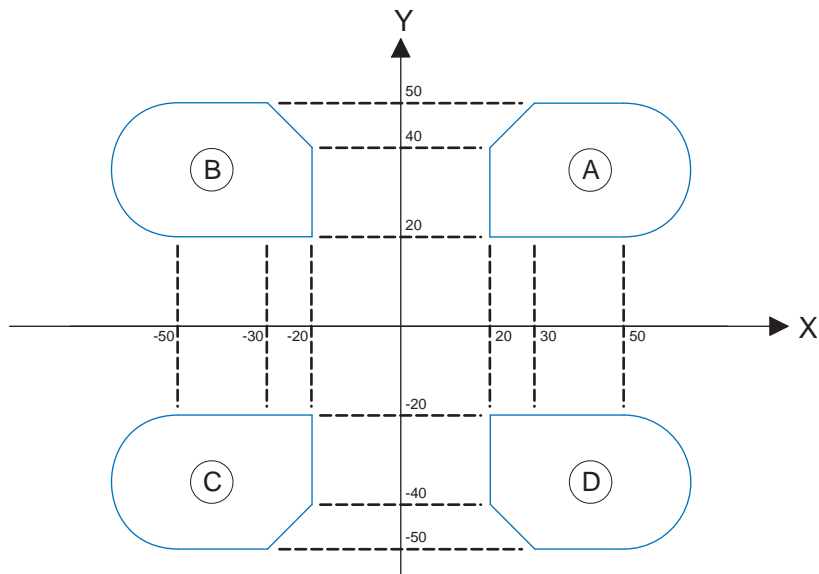
FAGOR AUTOMATION

CNC 8070

(REF: 2008)



**Programming example (M model).**



%L PROFILE (Subroutine with the profile)

```
N10 G00 X10 Y10
N20 G01 Z0 F400
N30 G01 X20 Y20 F850
N40 X50
N50 G03 X50 Y50 R15
N60 G01 X30
N70 X20 Y40
N80 Y20
N90 X10 Y10
N100 Z10 F400
M29 (End of subroutine)
```

%PROGRAM (Main program)

```
N10 G0 X0 Y0 Z10
N20 LL PROFILE (Call to a subroutine; machining of section "A")
N30 G11 (Mirror image in X)
N40 LL PROFILE (Call to a subroutine; machining of section "B")
N50 G12 (Mirror image on X and Y)
N60 LL PROFILE (Call to a subroutine; machining of section "C")
N70 G14 X1 (Mirror image cancellation on the X axis)
N80 LL PROFILE (Call to a subroutine; machining of section "D")
N90 G10 (Cancels the mirror image on all axes)
N100 G00 X0 Y0 Z50
M30
```

**11.**

**GEOMETRIC ASSISTANCE:**  
Mirror image (G10, G11, G12, G13, G14).



FAGOR AUTOMATION

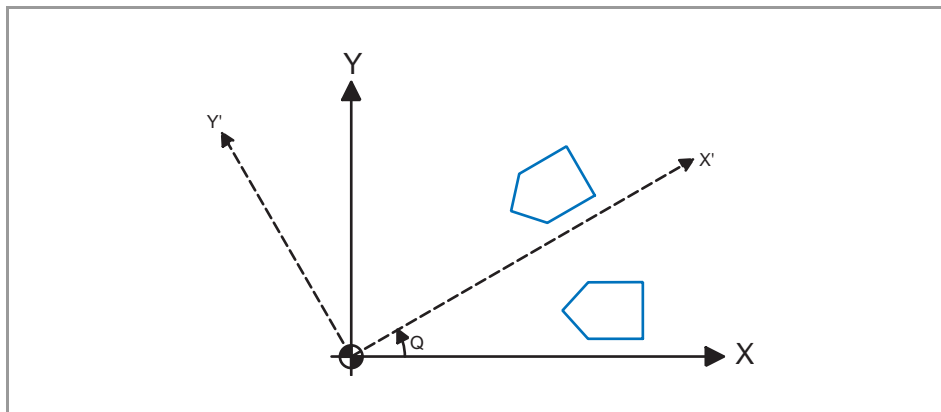
**CNC 8070**

(REF: 2008)



## 11.9 Pattern rotation (G73).

Function G73 may be used to rotate the coordinate system taking as rotation center the active reference system (part zero) or the programmed rotation center.



### Programming. Rotate the coordinates (pattern).

Programming the G73 function with its parameters alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G73 Q{angle} <I{abscissa} J{ordinate}>

Q{angle}	Rotation angle. Units: Degrees
I{abscissa}	Optional. Abscissa of the rotation center. Units: Millimeters, inches or degrees.
J{ordinate}	Optional. Ordinate of the rotation center. Units: Millimeters, inches or degrees.

G73 Q45  
(Rotate 90 degrees with the center at part zero)

G73 Q90 I20 J30  
(Rotate 90° with the center in I20 J30, when referring to the part zero.)

#### Rotation angle.

Function G73 is incremental; i.e. the various "Q" values programmed are added up.

#### Abscissa and ordinate of the rotational center.

The parameters I and J define the abscissa and ordinate of the rotation center, when referring to the part zero. This programming is optional and, if programmed, both parameters must be programmed. If not programmed, the part zero will be assume as the rotation center.

# 11.

**GEOMETRIC ASSISTANCE:**  
Pattern rotation (G73).

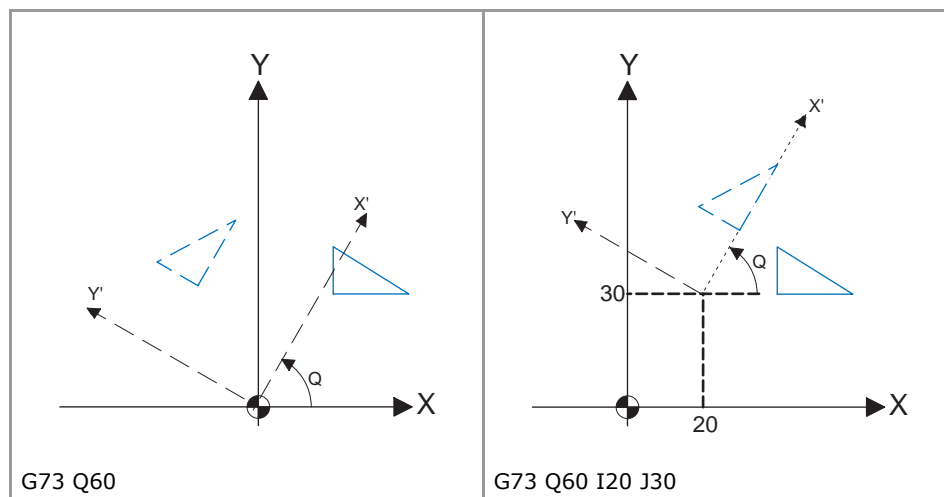
**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 11.

**GEOMETRIC ASSISTANCE:**  
 Pattern rotation (G73).

**Programming. Cancels coordinate rotation.**

Program the G73 function alone in the block (without parameters).

**Programming format.**

The programming format is:

G73

G73 (Cancels coordinate rotation)

**Considerations.**

The "I" and "J" values are affected by the active mirror images. If any mirror image function is active, the CNC applies first the mirror image and then the coordinate system rotation.

**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

- Function G73 is modal. The coordinate rotation stays active until it is canceled by function G73 or until the work plane is changed.
- On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC cancels the active coordinate system rotation.

**Programming example.**

```

%L PROFILE (Subroutine with the profile)
G01 X21 Y0 F300
G02 G31 Q0 I5 J0
G03 G31 Q0 I5 J0
G03 G31 Q180 I-10 J0
M29 (End of subroutine)

%PROGRAM (Main program)
$FOR P0=1, 8, 1 (Repeats the profile and the pattern rotation 8 times)
NLL PROFILE (Call to a subroutine; machining of section "D")
G73 Q45 (Coordinate rotation)
$ENDFOR
M30
    
```

**Summary of the variables.**

The following variables may be accessed from (PRG) the part-program and from the MDI/MDA mode, PLC and (INT) from an external application. For each variable, the table indicates whether the access is reading (R) or writing (W). PLC access to the variables, for both reading and writing, will be synchronous. Access to the variables from the part program returns the value for the block preparation (it does not stop the preparation), except when indicated otherwise.

Variables.	PRG	PLC	INT
<b>(V.)[ch].G.ROTPF</b> Position of the rotation center referred to part zero (abscissa). Units: Millimeters, inches or degrees.	R	R	R
<b>(V.)[ch].G.ROTPTS</b> Position of the rotation center referred to part zero (ordinate). Units: Millimeters, inches or degrees.	R	R	R
<b>(V.)[ch].G.ORGROT</b> Rotation angle of the coordinate system. Units: Degrees	R	R	R

**Syntax of the variables.**

·ch· Channel number.

V.[2].G.ROTPF	Channel ·2·.
V.[2].G.ROTPTS	Channel ·2·.
V.[2].G.ORGROT	Channel ·2·.

**11.**

**GEOMETRIC ASSISTANCE:**  
Pattern rotation (G73).



FAGOR AUTOMATION

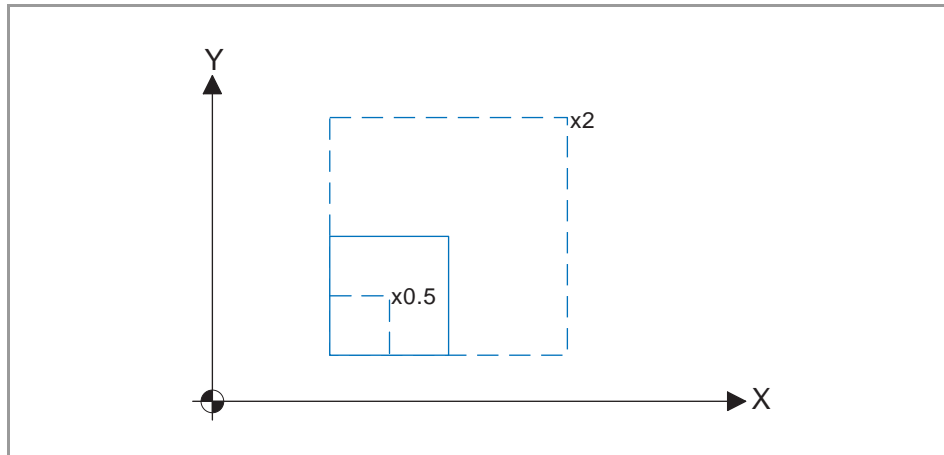
CNC 8070

(REF: 2008)

## 11.10 Scaling factor (G72#SCALE).

### 11.10.1 General scaling factor (G72#SCALE).

The G72 or #SCALE functions can enlarge or reduce the programmed geometry using a factor that is applied to all the axes of the channel. After activating the general scaling factor, all the programmed coordinates will be multiplied by this factor until it is canceled or a new scaling factor is programmed. In this way, by using the same program, families of parts that are similar in shape but different in size can be put together.



G72            General or per axis scaling factor.

#SCALE        General scaling factor.

The general scaling factor may be activated using the commands G72 or #SCALE. Both commands can be used indistinctively and affect the same scaling factor; i.e. the scaling factor programmed with G72 modifies the one programmed with #SCALE and vice versa.

### Programming. G72.

Programming the G72 function with its parameters alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G72 S{factor}

S{factor}            Scaling factor.  
Units: -.

G72 S2    (General Scaling factor of 2)

G72 S0.5    (General Scaling factor of 0.5)

#### Scaling factor.

Parameter "S" that sets the scaling factor must be programmed after function G72. If programmed before, it will be interpreted as spindle speed.

### Programming. #SCALE.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

#SCALE [{factor}]

{factor}            Scaling factor.  
Units: -.

# 11.

**GEOMETRIC ASSISTANCE:**  
Scaling factor (G72#SCALE).

```
#SCALE 2 (General Scaling factor of 2)
#SCALE 0.5 (General Scaling factor of 0.5)
```

## Programming. Cancel the scaling factor.

The general scaling factor is canceled using the same commands G72 or #SCALE, setting a scaling factor of  $\cdot 0\cdot$  or  $\cdot 1\cdot$ . When using function G72, the scaling factor is also canceled by programming it alone in the block.

### Programming format.

The programming format is:

```
G72
G72 S{factor}
S{factor}           Scaling factor (0/1).
                    Units: -.
```

```
G72
G72 S1
G72 S0
#SCALE [0]
#SCALE [1]
```

### Considerations.

- Activating the machine coordinate system (#MCS ON) cancels the scaling factor temporarily until the machine coordinate system is canceled (#MCS OFF). While the machine coordinate system is active, the scaling factor can neither be activated nor modified.
- The scaling factor has no effect when jogging the axis.
- The general scaling factor does not cancel the scaling factor per axis; the CNC applies both.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G72 is modal. The general scaling factor stays active until canceled with another general scaling factor.
- On power-up, after executing an M02 or M30 and after an emergency or reset, the CNC cancels the scaling factor.

# 11.

**GEOMETRIC ASSISTANCE:**  
Scaling factor (G72#SCALE).

**FAGOR** 

FAGOR AUTOMATION

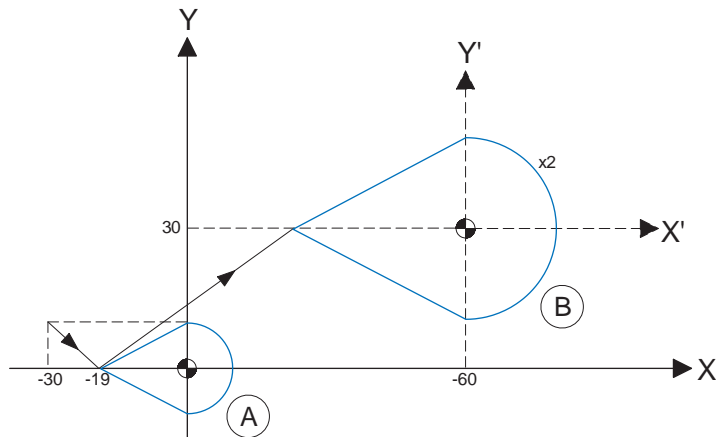
**CNC 8070**

(REF: 2008)

Programming example (M model).

11.

**GEOMETRIC ASSISTANCE:**  
Scaling factor (G72#SCALE).



```
%L PROFILE (Subroutine with the profile)
G90 X-19 Y0
G01 X0 Y10 F150
G02 X0 Y-10 I0 J-10
G01 X-19 Y0
M29 (End of subroutine)

%PROGRAM
G00 X-30 Y10
#CALL PROFILE (Call to a subroutine; machining of section "A")
G92 X-79 Y-30 (Coordinate preset)
#SCALE 2 (General Scaling factor of 2)
#CALL PROFILE (Call to a subroutine; machining of section "B")
#SCALE [1] (Cancel the scaling factor)
M30
```



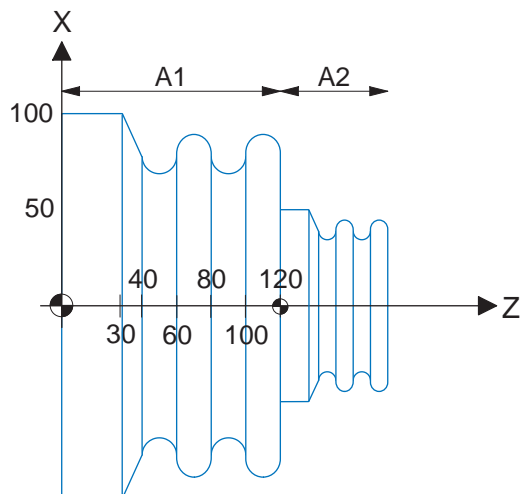
FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## Programming example (T model).



```
%L PROFILE (Subroutine that defines the "A1" zone of the part)
G90 G01 X200 Z0
G01 X200 Z30 F150
G01 X160 Z40
G03 X160 Z60 R10
G02 X160 Z80 R10
G03 X160 Z100 R10
G02 X160 Z120 R10
M29 (End of subroutine)
```

```
%PROGRAM (Main program)
G18 G151 (Main plane ZX and programming in diameters)
G00 X206 Z0 (Approach)
LL PROFILE (Call to a subroutine; machining of zone "A1")
G92 Z0 (Coordinate preset)
G72 S0.5 (Application of the scaling factor)
LL PROFILE (Call to a subroutine; machining of zone "A2")
G72 S1 (Cancellation of the scaling factor)
G01 X0
G0 X250 Z200 (Return to starting point)
G53 (Cancellation of coordinate preset)
M30
```

# 11.

**GEOMETRIC ASSISTANCE:**  
Scaling factor (G72#SCALE).

**FAGOR** 

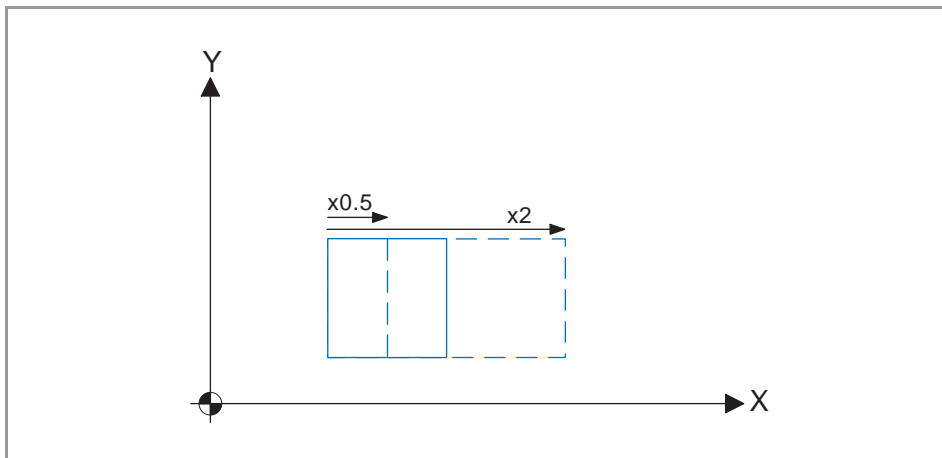
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 11.10.2 Scaling factor per axis (G72).

The G72 function can enlarge or reduce the programmed geometry, using a factor that may be different on each axis. After activating the scaling factor on an axis, all the programmed coordinates will be multiplied by this factor until it is canceled or a new scaling factor is programmed. In this way, by using the same program, families of parts that are similar in shape but different in size can be put together.



# 11.

**GEOMETRIC ASSISTANCE:**  
Scaling factor (G72#SCALE).

#### Programming.

Programming the G72 function with its parameters alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G72 X..C{scaling} <X..C{scaling}>

X..C{scaling}      Scaling factor. Programming a scaling factor of ·0· or ·1· cancels the active scaling factor.

```
G72 X2
  (Scaling factor 2 for the X axis)
G72 X2.5 Y0.8
  (Scaling factor 2.5 for the X axis and 0.8 for the Y axis)
G72 X1 Y2
  (Cancel the scaling factor for the X axis and apply a scaling factor of 2 for the Y axis)
```

#### Programming. Cancel the scaling factor.

The scaling factor per axis is cancelled with a G72 function, by setting a scaling value of 0 or 1. When programming the G72 function alone, the scaling factor is cancelled on all the axes.

#### Programming format.

The programming format is:

```
G72
G72 S{factor}
S{factor}              Scaling factor (0/1).
                       Units: -.
```

```
G72
G72 S1
G72 S0
```



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Considerations.

- Activating the machine coordinate system (#MCS ON) cancels the scaling factor temporarily until the machine coordinate system is canceled (#MCS OFF). While the machine coordinate system is active, the scaling factor can neither be activated nor modified.
- The scaling factor has no effect when jogging the axis.
- The scaling factor per axis does not cancel the general scaling; the CNC applies both.

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G72 is modal. The scaling factor per axis stays active until canceled by another scaling factor per axis.
- On power-up, after executing an M02 or M30 and after an emergency or reset, the CNC cancels the scaling factor.

### 11.10.3 Summary of the variables.

The following variables may be accessed from (PRG) the part-program and from the MDI/MDA mode, PLC and (INT) from an external application. For each variable, the table indicates whether the access is reading (R) or writing (W). PLC access to the variables, for both reading and writing, will be synchronous. Access to the variables from the part program returns the value for the block preparation (it does not stop the preparation), except when indicated otherwise.

Variables.	PRG	PLC	INT
<b>(V.)[ch].G.SCALE</b> Active general scaling factor. Units: -.	R	R	R

#### Syntax of the variables.

·ch· Channel number.

V.[2].G.SCALE                      Channel ·2·.

11.

GEOMETRIC ASSISTANCE:  
Scaling factor (G72#SCALE).

## 11.11 Work zones (G120/G121/G122/G123).

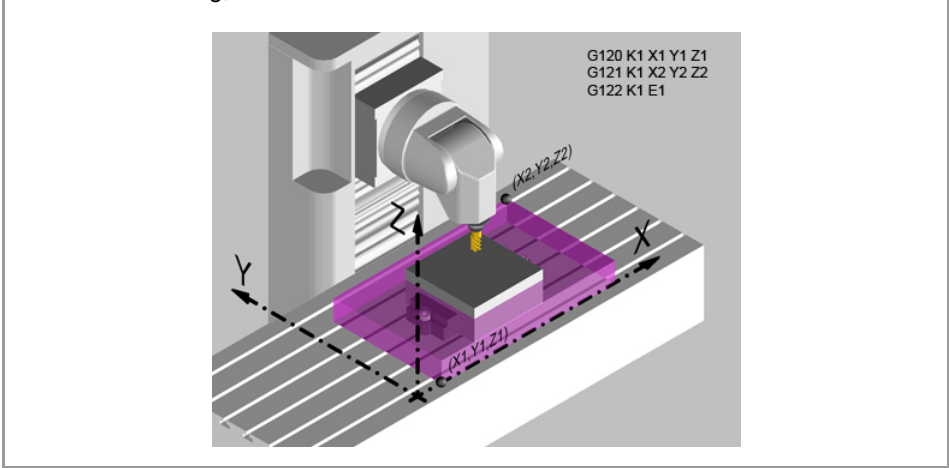
Functions G120 to G123 can define, enable and disable up to five work zones. The work zones define a restricted area for tool movement, either forbidding it to exit the programmed zone (no-exit zone) or forbidding it to enter it (no-entry zone). The CNC lets set five of these work zones that may be active at the same time.

Basically, a work zone is defined by programming the lower limit coordinate and the upper limit coordinate in one or several axes of the channel. It is also possible to combine a circular area in two of the axes with lower limits and upper limits in other axes of the channel.

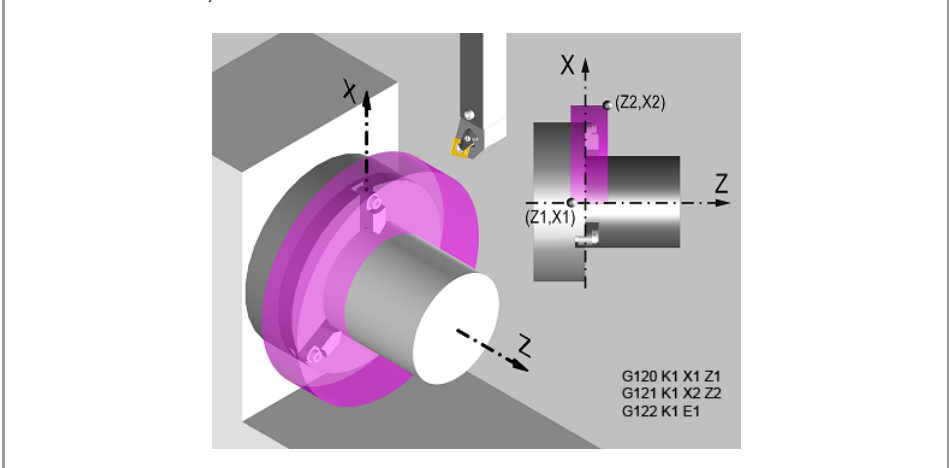
11.

**GEOMETRIC ASSISTANCE:**  
 Work zones (G120/G121/G122/G123).

Work zone in milling, defined in three linear axes.



Work zone in lathe, defined in two linear axes.



During the movement of the axes, either in jog or automatic mode, the CNC monitors the theoretical position values (coordinates) to check whether the tool gets into a no-entry zone or gets out of a no-exit zone. If so, the CNC stops the movement of the axes and issues the corresponding error message.

During the movement, the CNC can either monitor the tool tip or the tool base or both. This vigilance (monitoring) works with and without tool radius and length compensation. When the CNC monitors the tool tip, it takes into account its dimensions.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 11.11.1 CNC behavior when there are active work zones.

#### Some general points to consider.

- On power-up, the CNC will not monitor the zones whose limits are defined by axes that have non-absolute feedback and have not been homed.
- The CNC will not monitor the work zones during home search.
- The CNC takes into account the tool dimensions in the axes of the main trihedron. If there is an active kinematics, the CNC will take into account the direction of the tool.
- The CNC also applies the work zones to the axes that work as DRO, monitoring in this case the limit in the direction of the real position increment for no-exit zones.
- The CNC monitors the machine coordinates of the axes of the channel; i.e. it takes into account the programmed movements and those coming from the independent interpolator of the manual intervention as well as of the PLCOFFSET.

#### Safety distance.

- The work zone limits have a safety distance set in machine parameters (parameter ZONELIMITTOL) or from the variables. The CNC stops the axis when it reaches the safety distance of the zone; i.e. if the safety distance is 0.1 mm, the programmed coordinate can be, at the most, 0.1 mm before the limit.

#### Multi-channel system.

- When an axis changes channels, the CNC clears the limits of the axis in those zones.
- A channel axis cannot be changed while a zone that that axis belongs to is active.

#### Movements in automatic mode.

- Before starting the execution of a block, the CNC checks whether the end coordinates are inside a forbidden zone or the path crosses a forbidden zone.. If so, the CNC stops the movement of the axes and issues the corresponding error message. This check at the beginning of the block will also be done in the simulation modes.
- If manual intervention on an axis is enabled during the execution, from then on the CNC only checks the real position for the zones with limits on that axis. During block preparation, the CNC does not check the position for the zones with limits set on that axis.

#### Movements in jog mode (continuous, incremental jog or handwheels).

- When an axis reaches the limit of a zone, it stops and the CNC issues the corresponding warning message.
- The axis stops at the most restrictive limit of all the work zones in the moving direction respecting the safety distance (parameter ZONELIMITTOL). The CNC will look for the most restrictive limits from all the no-exit zones. From all the no-entry zones, the CNC will only consider the ones relevant to the position of the moving axis. The no-entry zone is considered relevant if the rest of the axes defined in the zone are in it, but not the moving axis.
- For no-exit zones, the CNC only checks the limit in the moving direction, thus allowing the axis to return to the valid zone.

# 11.

GEOMETRIC ASSISTANCE:

Work zones (G120/G121/G122/G123).

**FAGOR** 

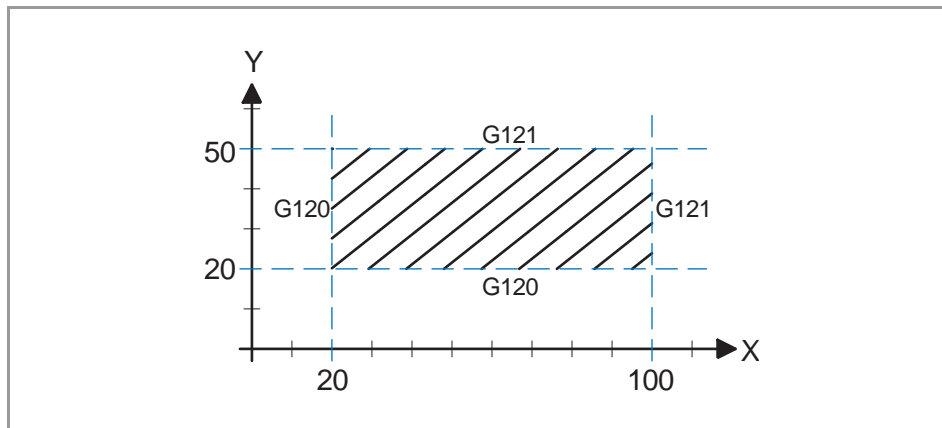
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 11.11.2 Defining the linear limits of the work zone (G120/G121).

Functions G120 and G121 can define the linear limits of the work zone. A work zone is defined by programming the lower limit coordinate and the upper limit coordinate in one or several axes of the channel. A work zone may be limited in all the axes of the channel. The limits of the work zones are defined in machine coordinates.



G120 Set lower linear limits of the work zone.

G121 Set upper linear limits of the work zone.

#### Programming.

Programming the G120 (lower limits) or G121 (upper limits) functions with their parameters alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G120 K{zone} X..C{limit}

G121 K{zone} X..C{limit}

K{zone} Zone number (between 1 and 5).  
Units: -.

X..C{limit} Lower limit (G120) or the upper limit (G121) of the zone, in machine coordinates.  
Units: Millimeters, inches or degrees.

```
G120 K1 X20 Y20
```

(Set the lower limits of zone 1 in the X Y axes)

```
G121 K1 X100 Y50
```

(Set the upper limits of zone 1 in the X Y axes)

#### Name of the axis and zone limit.

The limits of the work zone may be defined in all the axes of the channel, in machine coordinates. Both limits of a zone (lower and upper) may be positive or negative; but the lower limits must be smaller than the upper ones.

The work zone limits on the cross axis of a lathe type machine are always defined in radius, regardless of the setting of parameter DIAMPROG and the active function G151/G152.

#### Considerations.

- Setting the limits of a zone cancels the limits set before in that zone. The circular limits cancel the linear or circular limits set before in the 2 axes involved. The linear limits (G120 or G121) on an axis cancel the existing linear limits on that axis or the existing circular limits on that axis or on the other axis that defined the circular zone.
- It is possible to combine in a zone circular limits on 2 axes with linear limits on other two axes.

# 11.

GEOMETRIC ASSISTANCE:

Work zones (G120/G121/G122/G123).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

- The changes programmed in the limits or in the status of the zones interrupt block preparations.
- For no-entry zones, when repositioning the axes after a tool inspection, the user must decide the right repositioning order (sequence) of the axes so as not to invade the zone. In any case, during repositioning, the CNC will issue an error message before entering the forbidden zone.

### **Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

- Functions G120 and G121 are modal.
- On power-up, after executing an M02 or M30, and after a reset, the CNC keeps the set limits.

11.

**GEOMETRIC ASSISTANCE:**

Work zones (G120/G121/G122/G123).

**FAGOR** 

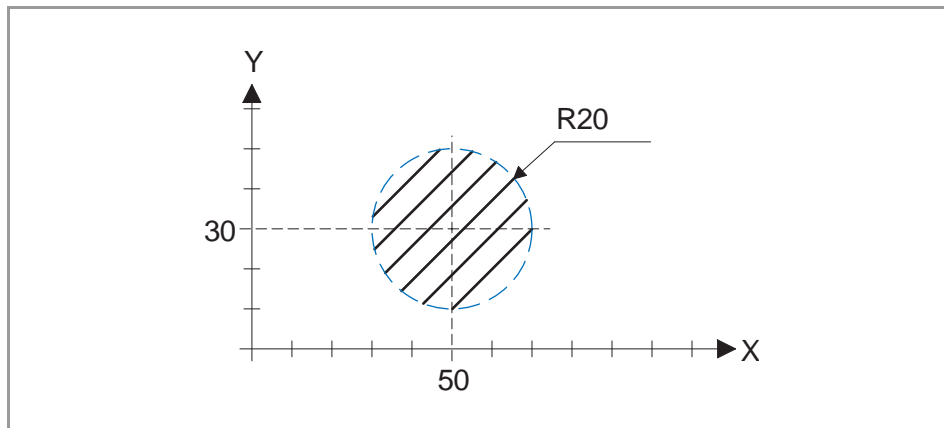
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 11.11.3 Set circular limits of the work zone (G123).

The G123 function defines the circular work zone. It is also possible to combine a circular area in two of the axes with lower limits and upper limits in other axes of the channel. A work zone may be limited in all the axes of the channel. The limits of the work zones are defined in machine coordinates.



# 11.

**GEOMETRIC ASSISTANCE:**  
 Work zones (G120/G121/G122/G123).

#### Programming.

Programming the G123 function with its parameters alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G123 K{zone} X..C{center} X..C{center} R{radius}

K{zone}                    Zone number (between 1 and 5).  
 Units: -.

X..C{center}            Center coordinates in the two axes that define the circle, in machine coordinates.  
 Units: Millimeters, inches or degrees.

R{radius}                Work zone radius.  
 Units: mm or inches.

```
G123 K2 X50 Y30 R20
  (Set a circular zone with a 20 radius in the X Y plane)
```

#### Name of the axis and zone limit.

The zone limit may be defined in any two axes of the channel, in machine coordinates.

#### Considerations.

- Setting the limits of a zone cancels the limits set before in that zone. The circular limits cancel the linear or circular limits set before in the 2 axes involved. The linear limits (G120 or G121) on an axis cancel the existing linear limits on that axis or the existing circular limits on that axis or on the other axis that defined the circular zone.
- It is possible to combine in a zone circular limits on 2 axes with linear limits on other two axes.
- The changes programmed in the limits or in the status of the zones interrupt block preparations.
- For no-entry zones, when repositioning the axes after a tool inspection, the user must decide the right repositioning order (sequence) of the axes so as not to invade the zone. In any case, during repositioning, the CNC will issue an error message before entering the forbidden zone.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G123 is modal.
- On power-up, after executing an M02 or M30, and after a reset, the CNC keeps the set limits.

11.

**GEOMETRIC ASSISTANCE:**  
Work zones (G120/G121/G122/G123).

**FAGOR** 

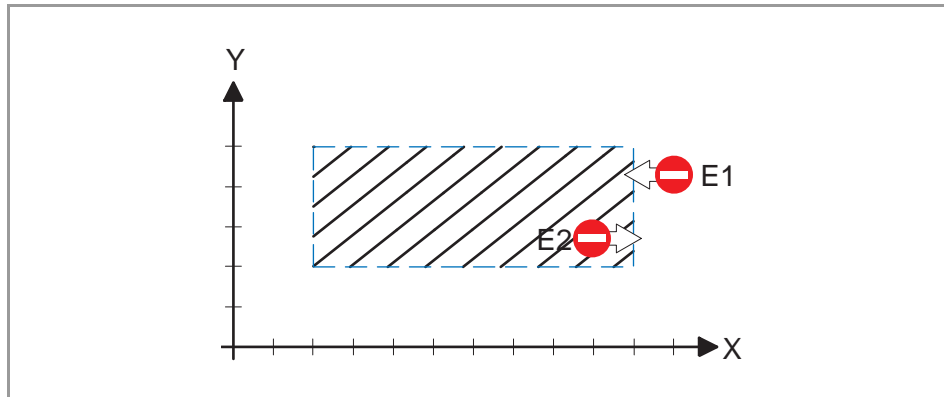
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 11.11.4 Enable/disable the work zones (G122).

The G122 function enables the work zones, as either no-entry zones or no-exit zones, or disables them. When a zone is enabled, the CNC, by default, watches the tool tip, but, it also offer the option monitor the tool base or both (tip and base). All zones may be enabled at the same time.



#### Programming.

Programming the G122 function with its parameters alone in the block.

#### Programming format.



*The command "E" must always be placed after the function G122; otherwise the CNC will interpret it as the name of an axis.*

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

G122 K{zone} E{action} <I{monitoring}>

K{zone} Zone number (between 1 and 5).  
Units: -.

E{action} Disable the zone or enable it as a no-entry or no-exit zone.  
E0: Disable the zone.  
E1: Enable the work zone as no-entry.  
E2: Enable the work zone as no-exit.  
Units: -.

I{vigilance} Optional (by default I0). Tool to monitor.  
I0: Monitor the tool tip.  
I1: Monitor the tool base.  
I2: Monitor the tool tip and the tool base.  
Units: -.

```
G122 K1 E1
      (Enable zone 1 as no-entry zone)
      (Monitor the tool tip)
```

```
G122 K2 E2 I2
      (Enable zone 2 as no-exit zone)
      (Monitor the tool tip and the tool base)
```

#### Monitor the tool tip or the tool base.

The CNC can monitor the tool tip and/or the tool base. When the CNC monitors the tool tip, it takes into account its dimensions. The vigilance (monitoring) works with and without tool radius and length compensation.

(REF: 2008)

# 11.

GEOMETRIC ASSISTANCE:  
Work zones (G120/G121/G122/G123).

**FAGOR**

FAGOR AUTOMATION

CNC 8070

## Considerations.

### CNC behavior when an axis invades the forbidden zone.

When one or several axes get into a no-entry zone or get out of a no-exit zone, the CNC interrupts the execution and issues the corresponding error message. To move the tool to a permitted zone, access the JOG mode and move the axes that overran the limits. These axes can only move in the direction that places them within the limits.

### Cancel the zone limits via PLC. PLC mark LIM(axis)OFF.

If the PLC mark LIM(axis)OFF of an axis is active, the CNC ignores the zone limits set for that axis (besides the software limits). This makes it easier to take the tool back to the permitted zone if it has invaded a forbidden zone.

### Activate several zones simultaneously.

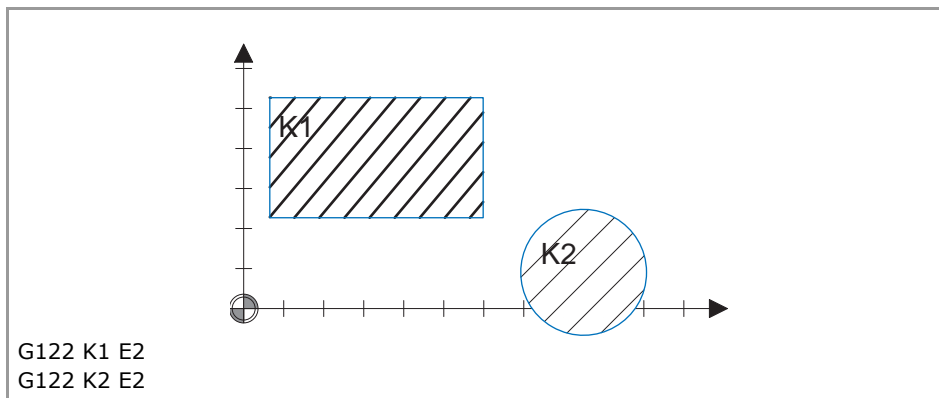
When activating several zones at the same time (that either overlap or they don't) on one or several axes, the CNC follows these criteria:

- If there are several no-exit zones active, it considers it an error to try to move the tool to a point located outside all of them.
- If there are several no-entry zones active, it considers it an error to try to move the tool to a point located inside any of them.
- If there are several no-entry and no-exit zones active, it considers it an error to try to move the tool to a point located inside any of the no-entry zones or outside all of the no-exit zones.
- If there are 2 circular or rectangular no-exit zones one inside the other, the CNC will only take into account the outside one.

## Programming examples.

### Example 1.

To allow movement only in the shaded zones, combine 2 no-exit zones, a rectangular one and a circular one.

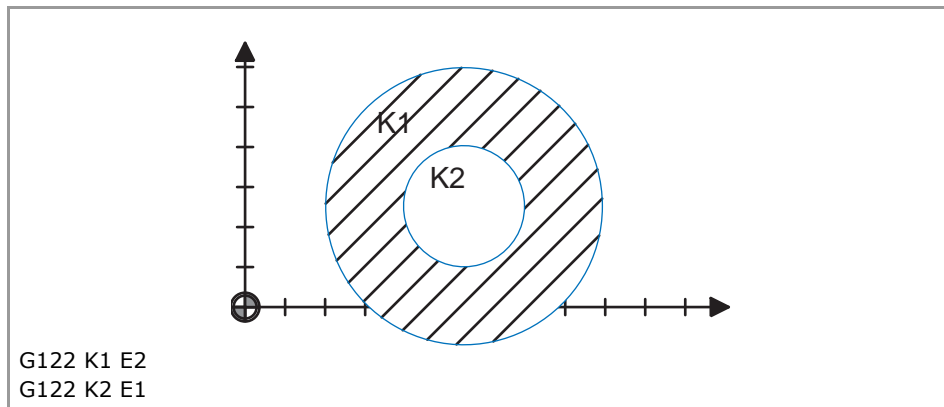


# 11.

**GEOMETRIC ASSISTANCE:**  
 Work zones (G120/G121/G122/G123).

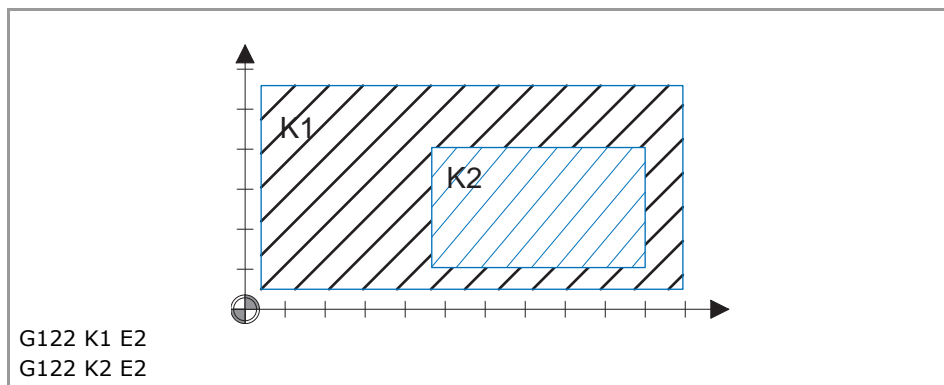
### Example 2.

To allow movement only in the shaded zone, combine 2 zones, one into the other; the outside no-exit zone and the inside no-entry zone.



### Example 3.

If there are 2 circular or rectangular no-exit zones one inside the other, the CNC will only take into account the outside one. The entire shaded area is the permitted zone.



### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

- Function G122 is modal.
- On power-up, after executing an M02 or M30, and after a reset, the CNC keeps the status of the work zones right before the shut-down (enabled/disabled).

### 11.11.5 Summary of the variables.

The following variables may be accessed from (PRG) the part-program and from the MDI/MDA mode, PLC and (INT) from an external application. For each variable, the table indicates whether the access is reading (R) or writing (W). PLC access to the variables, for both reading and writing, will be synchronous. Access to the variables from the part program returns the value for the block preparation (it does not stop the preparation), except when indicated otherwise.

Meaning.	PRG	PLC	INT
<b>V.[ch].MPA.ZONELIMITTOL.xn</b> Safety distance (set in parameter ZONELIMITTOL) applied by the CNC to the axis with respect to the limit of the work zone. Units: Millimeters, inches or degrees.	R*	R	R
<b>V.[ch].G.ZONEST[n]</b> Status of work zone [n]. (0=Disabled zone). (1=Work zone enabled as no-entry). (2=Work zone enabled as no-exit). Units: -.	R	R/W	---
<b>V.[ch].G.ZONETOOLWATCH[n]</b> Monitor the tool tip or tool base in work zone [n]. (0=Monitor the tool tip). (1=Monitor the tool base). (2=Monitor the tool tip and the tool base). Units: -.	R	R/W	---
<b>V.[ch].G.ZONEWARN[n]</b> Some axis has reached the limit of work zone [n]. (0=No). (1=Yes). Units: -.	R*	R	R
<b>V.[ch].A.ZONELIMITTOL.xn</b> Safety distance of the limits of the work zones. Units: Millimeters, inches or degrees.	R/W	R/W	---
<b>V.[ch].A.ZONELOWLIM[n].xn</b> Lower limit of zone [n]. Units: Millimeters, inches or degrees.	R	R/W	---
<b>V.[ch].A.ZONEUPLIM[n].xn</b> Upper limit of zone [n]. Units: Millimeters, inches or degrees.	R	R/W	---
<b>V.[ch].G.ZONECIR1[n]</b> Zone [n] center coordinate (circular zone) along the abscissa axis. Units: Millimeters, inches or degrees.	R	R/W	---
<b>V.[ch].G.ZONECIR2[n]</b> Zone [n] center coordinate (circular zone) along the ordinate axis. Units: Millimeters, inches or degrees.	R	R/W	---
<b>V.[ch].G.ZONER[n]</b> Radius of zone [n] (circular zone). Units: Millimeters, inches or degrees.	R	R/W	---
<b>V.[ch].G.ZONECIRAX1[n]</b> Logic axis corresponding to the first coordinate of the center of zone [n]. Units: -.	R	R/W	---
<b>V.[ch].G.ZONECIRAX2[n]</b> Logic axis corresponding to the second coordinate of the center of zone [n]. Units: -.	R	R/W	---

(\*) The CNC evaluates the variable during execution (it stops the block preparation).

#### Syntax of the variables.

- ch· Channel number.
- n· Zone number.

# 11.

GEOMETRIC ASSISTANCE:

Work zones (G120/G121/G122/G123).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

·xn· Name, logic number or index of the axis.

V.[2].G.ZONEST[1]	Channel ·2·. Zone 1.
V.A.ZONEUPLIM[1].Z	Z axis. Zone 1.
V.A.ZONEUPLIM[1].4	Axis with logic number ·4·. Zone 1.
V.[2].A.ZONEUPLIM[1].1	Axis with index ·1· in the channel ·2·. Zone 1.

# 11.

## GEOMETRIC ASSISTANCE:

Work zones (G120/G121/G122/G123).

# ADDITIONAL PREPARATORY FUNCTIONS

# 12

## 12.1 Dwell (G04 / #TIME).

The F04 function and the #TIME instruction may be used to interrupt the execution of the program for the specified period of time. Both commands are equivalent and either one may be used.

### Programming (1). G04.

Program function G04 and then the dwell time.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets. The K command may be left out when the dwell is programmed with a constant.

G04 K{time}

G04 {time}

K{time}                    Dwell.  
                                  Units: Seconds.

{time}                    Dwell (programmed with a constant).  
                                  Units: Seconds.

```
G04 K0.5
      (0.5 second dwell)
```

```
G04 8.5
      (8.5 second dwell)
```

```
P1=3
G04 KP1
      (3 second dwell)
```

```
P1=3
G04 K[P1+7]
      (10 second dwell)
```

### Programming (2). #TIME.

The dwell must be programmed with programming this instruction.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets. If the time is programmed with a constant or a parameter, the brackets [] may be left out .

#TIME [{time}]

#TIME {time}

{time}                    Dwell.  
                                  Units: Seconds.

## 12.

ADDITIONAL PREPARATORY FUNCTIONS  
Dwell (G04 / #TIME).

```
#TIME [5]
#TIME 5
    (5 second dwell)

P1=2
#TIME [P1]
#TIME P1
    (2 second dwell)

P1=2
#TIME [P1+3]
    (5 second dwell)
```

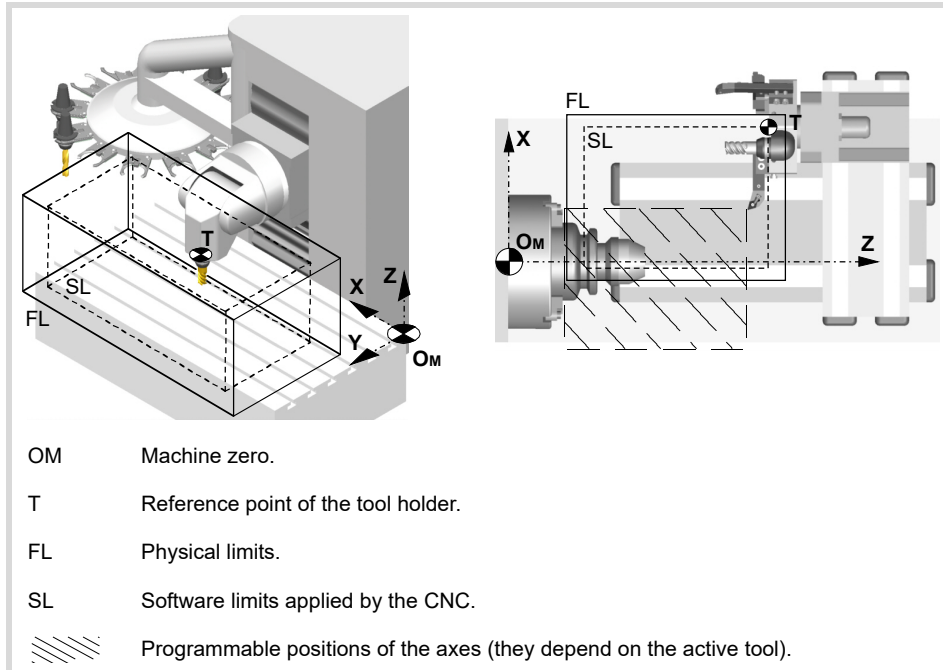
**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.**

Function G04 is not modal, therefore, it must be programmed every time a dwell is desired. Function G04 may also be programmed as G4.



## 12.2 Software limits.

The CNC lets set the software limits on linear axes and on linearlike rotary axes. The software limits set the travel limits for the axes to prevent the carriages from reaching the mechanical hard stops. The carriages reach the hard stops when the reference point of the tool holder is placed at the physical limits.



12.

ADDITIONAL PREPARATORY FUNCTIONS  
Software limits.

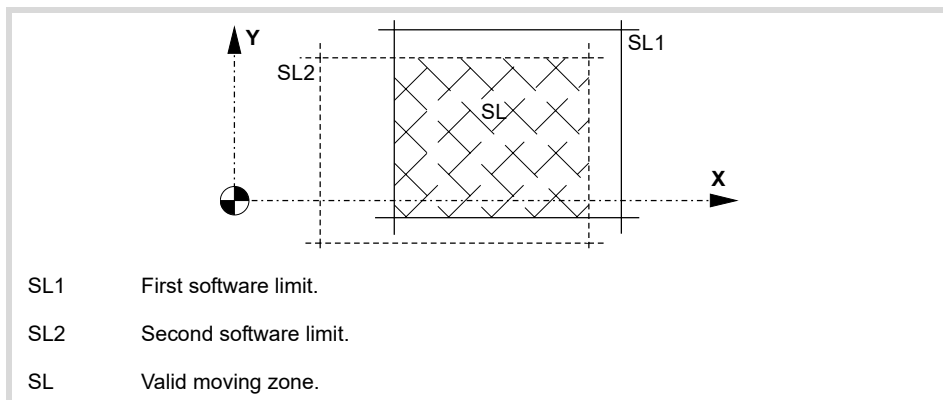
### CNC behavior when an axis reaches the software limits.

In automatic mode, when programming a position where the reference point of the tool holder is beyond the software limits, the CNC interrupts the execution and shows the pertinent error message. The programmable positions of the axes depend on the dimensions of each tool.

In jog mode, when an axis reaches the software limits, the CNC interrupts the execution and shows the pertinent error message. To take the axis to the work zone, access the JOG mode and move the axis that overran the travel limit. The axis can only move in the direction that places it within the limits.

### Software limits applied by the CNC (first and second limits).

Each axis can have two active software limits, called first and second limit. Since each software limit is defined by an upper and lower limit, each axis can have a total of two upper limits and two lower limits. Of the each limit pair, (lower and upper), the CNC applies the most restrictive one regardless of whether they belong to the first or to the second limit.



**FAGOR**

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 12.2.1 Define the first software limit (G198/G199).

The CNC lets set the software limits on linear axes and on linearlike rotary axes. The first software limits of the axes are pre-defined in the machine parameters (parameters LIMIT+ / LIMIT-). These limits may be changed by program using the following functions and variables.

G198 Set lower software travel limits (first limit).

G199 Set upper software travel limits (first limit).

The CNC also offers the following variables equivalent to functions G198/G199. See "[12.2.2 Define the first software limit via variables.](#)" on page 244.

V.A.NEGLIMIT.xn Set lower software travel limits (first limit). Variable equivalent to G198.

V.A.POSLIMIT.xn Set upper software travel limits (first limit). Variable equivalent to G199.

### Programming.

Program one of the functions G198/G199, and then, the axes and their new software limits. These functions may be used to program several axes.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

G198 X..C{soft\_limit}

G199 X..C{soft\_limit}

X..C{soft\_limit} Name of the axis and software limit.

Units: millimeters/inches.

```
G198 X-1000 Y-1000
      (New lower limits X=-1000 Y=-1000)
G199 X1000 Y1000
      (New upper limits X=1000 Y=1000)
```

#### Name of the axis and software limit.

Both limits of an axis (lower and upper) may be positive or negative; but the lower limits must be smaller than the upper ones. Otherwise, the might not move in any direction.

When setting both upper and lower limits of an axis to ·0·, the CNC cancels the first software limit of that axis and applies the second one (if it has been defined). To restore the first limit, it must be programmed again.

### Considerations.

#### Absolute (G90) or incremental (G91) programming.

Depending on the active work mode G90 or G91, the position of the new limits will be defined in absolute coordinates (G90) in the machine reference system or in incremental coordinates (G91) referred to the current active limits.

```
G90
G198 X-800
      (New lower limit X=-800)
G199 X500
      (New upper limit X=500)
.
G91
G198 X-700
      (New incremental lower limit X=-1500)
```

# 12.

ADDITIONAL PREPARATORY FUNCTIONS  
Software limits.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### Axes out of position.

If after setting the new limits, an axis positions beyond them, it will be possible to move that axis towards the work zone (between those limits).

### Programming on a lathe (radius/diameter).

The software limits are always defined in radius, regardless of the setting of parameter DIAMPROG and the active function G151/G152.

### Restore the software limits set in the machine parameters.

The software limits set in the machine parameters may be restored by program using their variables.

```
G198 X[V.MPA.NEGLIMIT.X] Y[V.MPA.NEGLIMIT.Y]
G199 X[V.MPA.POSLIMIT.X] Y[V.MPA.POSLIMIT.Y]
      (The CNC restores the limits set in the machine parameters)
```

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

On power-up or after validating the axis machine parameters the CNC assumes the software limits set by the machine parameters. After an M02 or M30 and after an emergency or a reset, the CNC maintains the software limits set by G198 and G199 or their equivalent variables.

12.

ADDITIONAL PREPARATORY FUNCTIONS  
Software limits.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 12.2.2 Define the first software limit via variables.

The first software limits can also be defined by the variables equivalent to G198/G199. The functions and the variables modify the same software limits; therefore, either ones may be used.

V.A.NEGLIMIT.xn	Set lower software travel limits (first limit). Variable equivalent to G198.
V.A.POSLIMIT.xn	Set upper software travel limits (first limit). Variable equivalent to G199.

On power-up, these variables assume the values of the machine parameters (LIMIT+ / LIMIT-).

### Programming of software limits.

Programming is equivalent to G198/G199. Both limits of an axis (lower and upper) may be positive or negative; but the lower limits must be smaller than the upper ones. When setting both upper and lower limits of an axis to  $\cdot 0 \cdot$ , the CNC cancels the first software limit of that axis and applies the second one (if it has been defined).

### Considerations.

#### Absolute (G90) or incremental (G91) programming.

Unlike functions G198/G199, the limits defined with variables do not depend on functions G90/G91, they are always in absolute coordinates and in the machine reference system.

#### Axes out of position.

If after setting the new limits, an axis positions beyond them, it will be possible to move that axis towards the work zone (between those limits).

#### Programming on a lathe (radius/diameter).

The software limits are always defined in radius, regardless of the setting of parameter DIAMPROG and the active function G151/G152.

### Influence of the reset, turning the CNC off and of the M30.

After an M02 or M30 and after an emergency or a reset, the CNC maintains the software limits set by these variables.

# 12.

ADDITIONAL PREPARATORY FUNCTIONS  
Software limits.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 12.2.3 Define the second software limit via variables.

The second software limits can only be defined by variables.

V.A.RTNEGLIMIT.xn     Set lower software travel limits (second limit).

V.A.RTPOSLIMIT.xn     Set upper software travel limits (second limit).

On power-up, these variables assume the values of the first software limits. As long as these variables are not defined with their own value, they copy the values of the first software limits.

#### Programming of software limits.

Both limits of an axis (lower and upper) may be positive or negative; but the lower limits must be smaller than the upper ones. When setting both upper and lower limits of an axis to  $-0$ , the CNC cancels the second software limit of that axis.

#### Considerations.

##### Absolute (G90) or incremental (G91) programming.

The limits defined with variables do not depend on functions G90/G91, they are always in absolute coordinates and in the machine reference system.

##### Axes out of position.

If after setting the new limits, an axis positions beyond them, it will be possible to move that axis towards the work zone (between those limits).

##### Programming on a lathe (radius/diameter).

The software limits are always defined in radius, regardless of the setting of parameter DIAMPROG and the active function G151/G152.

##### Influence of the reset, turning the CNC off and of the M30.

After an M02 or M30 and after an emergency or a reset, the CNC maintains the software limits set by these variables.

# 12.

ADDITIONAL PREPARATORY FUNCTIONS  
Software limits.

### 12.2.4 Variables associated with the software limits.

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

Variable.	R/W	Meaning.
V.[ch].MPA.NEGLIMIT.xn	R	Lower software limit (first limit) set in the machine parameters.
V.[ch].MPA.POSLIMIT.xn	R	Upper software limit (first limit) set in the machine parameters.
V.[ch].A.NEGLIMIT.xn	R/W	Lower software limit (first limit). Equivalent to G198.
V.[ch].A.POSLIMIT.xn	R/W	Upper software limit (first limit). Equivalent to G199.
V.[ch].A.RTNEGLIMIT.xn	R/W	Lower software limit (second limit).
V.[ch].A.RTPOSLIMIT.xn	R/W	Upper software limit (second limit).
V.[ch].G.SOFTLIMIT	R	Software limit reached on some axis. (0=No 1=Yes)

#### Syntax of the variables.

- ch· Channel number.
- xn· Name, logic number or index of the axis.

V.A.POSLIMIT.Z	Z axis.
V.A.POSLIMIT.4	Axis with logic number ·4·.
V.[2].A.POSLIMIT.1	Axis with index ·1· in the channel ·2·.
V.[2].G.SOFTLIMIT	Channel ·2·.

12.

ADDITIONAL PREPARATORY FUNCTIONS  
 Software limits.

## 12.3 Turn Hirth axis on and off (G170/G171).

A Hirth axis is the one that can only be positioned at particular positions, multiple of its pitch (parameter HPITCH). When a Hirth axis is not active, it behaves like a normal rotary or linear axis and can reach any position. Hirth axes may be canceled and activated by program using the following functions.

G170 Hirth axes OFF.  
G171 Hirth axes ON.

### Programming. Activate a Hirth axis.

Program function G171, and then, the axes to be activated as Hirth and the order in which they will be activated. Several axes may be programmed with this function.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

G171 X..C{n\_order}

X..C{n\_order} Name of the axis and order number.

```
G171 B1 C2
(Activate B first and then C, as Hirth axis)
```

### Programming. Cancel a Hirth axis.

Program function G170, and then the Hirth axes to be deactivated and the order in which they will be deactivated. Several axes may be programmed with this function.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

G170 X..C{n\_order}

X..C{n\_order} Name of the axis and order number.

```
G170 B2 C1
(Deactivate the C axis first and then B)
```

### Considerations.

- If when activating a Hirth axis, it is located in the wrong position, the CNC will issue a warning so the operator can turn it to a correct position.
- A Hirth axis must always be positioned at positions multiple of its pitch. For these positioning movements, the CNC takes into account the active zero offset (preset or zero offset).
- Both linear and rotary axes may be Hirth. Only those axes defined as Hirth axes by the OEM (parameter HIRTH) may be activated as Hirth axes.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Functions G170 and G171 are modal and incompatible with each other. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC activates all the Hirth axes.

# 12.

ADDITIONAL PREPARATORY FUNCTIONS

Turn Hirth axis on and off (G170/G171).

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 12.4 Set and gear change.

### 12.4.1 Change parameter set of an axis (G112)

The CNC may have up to 4 sets of parameters for each axis defined by the OEM in the machine parameter table. The parameter set may be selected by program using function G112. This function does not carry out any physical change on the machine (gear change), it only assumes the parameters of the selected set. When using Sercos axes, function G112 also involves changing the drive's velocity gear.

#### Programming.

Program G112 and then the axes and the new parameter set to be activated for each one of them. Several axes may be programmed with this function.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

G112 X..C{set}

X..C{set}          Name of the axis and parameter set (between 1 and 4).

G112 X2 Y3

(The CNC selects the second set of parameters for the X axis, and the third one for the Y axis)

#### Change of the drive's parameter set.

The CNC only allows the set of the spindle parameters to be modified when working as a C axis. In this case, a change to the set is programmed using the name of the axis, not the name of the spindle.

#CAX[S,C]

G112 C2

(It selects the second set of parameters for the C axis)

#### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Function G112 is modal. After validating the machine parameters, every time a program is executed from the automatic mode, on power-up, after executing an M02 or M30, after an emergency or a reset, the CNC acts as follows depending on how the OEM has set it (parameter DEFAULTSET).

DEFAULTSET	Meaning.
0	The CNC maintains the parameter set.
1..4	Set number assumed by the CNC.

# 12.

ADDITIONAL PREPARATORY FUNCTIONS

Set and gear change.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 12.4.2 Change the gear and set of a Sercos drive using variables.

The following variables may be used to change the gear and set of a Sercos drive for axes and for spindles. This variable does not affect the parameter set of the CNC.

(V.)[ch].A.SETGE.xn      Select the set and the gear at the Sercos drive.

(V.)[ch].A.SETGE.sn

(V.)[ch].SP.SETGE.sn

### Programming.

The drive can have 8 work ranges or gears identified from 0 through 7 (drive parameter GP6) and 8 sets of parameters (drive parameter GP4) identified from 0 through 7.

The 4 least significant bits indicate the work range and the 4 most significant bits indicate the set of parameters. If the value of any 4-bit set is '0', the CNC does not change the active gear or set at the drive. Example of some values of the variable.

Value.	Meaning.
\$21	First range or gear (gear '0'). Second set of parameters (set '1').
\$40	The drive maintains the active gear. Fourth set of parameters (set '3').
\$07	Seventh range or gear (gear '6'). The drive maintains the active parameter set.

### Considerations.

Only one change can be taking place at a time. If other gear or set changes are programmed during this process, even if they are at different drives, the CNC only keeps the one programmed last and ignores the rest of the intermediate changes.

# 12.

ADDITIONAL PREPARATORY FUNCTIONS  
Set and gear change.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 12.4.3 Variables related to set and gear change.

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

Variable.	R/W	Meaning.
(V.)[ch].A.ACTIVSET.xn (V.)[ch].A.ACTIVSET.sn (V.)[ch].SP.ACTIVSET.sn	R	Parameter set active at the axis or spindle. The variable returns the value of execution or preparation as follows. If the spindle belongs to the channel requesting the variable, it returns the preparation value; if the axis or spindle belongs to a different channel, the variable returns the execution value and interrupts block preparation.
(V.)[ch].A.SETGE.xn (V.)[ch].A.SETGE.sn (V.)[ch].SP.SETGE.sn	R/W	Select the set and the gear at the Sercos drive. The 4 least significant bits indicate the work range and the 4 most significant bits indicate the set of parameters. If the value of any 4-bit set is ·0·, the CNC does not change the active gear or set at the drive.

#### Syntax of the variables.

- ch· Channel number.
- xn· Name, logic number or index of the axis.
- sn· Name, logic number or index of the spindle.

V.A.ACTIVSET.Z	Z axis.
V.A.ACTIVSET.S	Spindle S.
V.SP.ACTIVSET.S	Spindle S.
V.SP.ACTIVSET	Master spindle.
V.A.ACTIVSET.4	Axis or spindle with logic number ·4·.
V.[2].A.ACTIVSET.1	Axis with index ·1· in the channel ·2·.
V.SP.ACTIVSET.2	Spindle with index ·2· in the system.
V.[2].SP.ACTIVSET.1	Spindle with index ·1· in the channel ·2·.

# 12.

## ADDITIONAL PREPARATORY FUNCTIONS

Set and gear change.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 12.5 Smooth the path and the feedrate.

By default, the CNC calculates the space and the feedrate on the main three axes and the rest of the axes follow their corresponding feedrate. This way, on a machine that moves more than three axes and has the kinematics and RTCP active, the tool tip moves at the programmed feedrate. However, when these machining operations show discontinued movements on the main axes, this process can generate irregularities on the velocity profile and, therefore, the resulting movement might not always be continuous.

To solve both situations, the CNC offers several instructions that make the movement a lot more continuous, thus improving part finish quality and reduce machining time considerably. These instructions are incompatible with each other.

#PATHND Smooth the path.  
#FEEDND Smooth the path and the feedrate.

### 12.5.1 Smooth the path (#PATHND).

This instruction being active (#PATHND ON), the CNC calculates the space between all the axes and obtains a smoother movement. If this instruction is not active (#PATHND OFF), the CNC calculates the space on the three main axes.

In either case, the CNC applies the programmed feedrate to the three main axes; the rest of the axes move at their corresponding feedrate to end the movement of them all at the same time.

#### Programming. Activate path smoothing.

Program the instruction alone in the block.

##### Programming format.

The programming format is:

#PATHND ON

```
#PATHND ON
```

#### Programming. Cancel path smoothing.

Program the instruction alone in the block.

##### Programming format.

The programming format is:

#PATHND OFF

```
#PATHND OFF
```

#### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Instructions #PATHND and #FEEDND are incompatible with each other. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes the behavior set by the machine manufacturer (parameter FEEDND).

# 12.

ADDITIONAL PREPARATORY FUNCTIONS

Smooth the path and the feedrate.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 12.5.2 Smooth the path and the feedrate (#FEEDND).

This instruction being active (#FEEDND ON), the CNC considers all the axes when calculating the space. The programmed feedrate will be the result of composing the movements onto all the axes of the channel. The CNC applies the programmed feedrate to all the axes.

If the instruction is not active (#FEEDND OFF) the programmed feedrate will be the result of composing the movement only onto the main axes. The rest of the axes move at their corresponding feedrate to end the movement of them all at the same time.

### Programming. Activate path and feedrate smoothing.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

```
#FEEDND ON
```

```
#FEEDND ON
```

### Programming. Cancel path and feedrate smoothing.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

```
#FEEDND OFF
```

```
#FEEDND OFF
```

### Considerations.

- The CNC only limits the programmed feedrate if an axis exceeds its maximum feedrate (parameter MAXFEED).
- If none of the main axes are programmed, the programmed feedrate will be reached on the axis moving the farthest so they can all reach their destination at the same time.

### Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Instructions #PATHND and #FEEDND are incompatible with each other. On power-up, after an M02 or M30 and after an emergency or a reset, the CNC assumes the behavior set by the machine manufacturer (parameter FEEDND).

# 12.

ADDITIONAL PREPARATORY FUNCTIONS  
Smooth the path and the feedrate.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Tool compensation allows programming the contour to be machined based on part dimensions and without taking into account the dimensions of the tool that will be used later on. This way, there is no need to calculate and redefine the tool path depending on the radius and length of each tool.

## Types of compensation

### Tool radius compensation (milling).

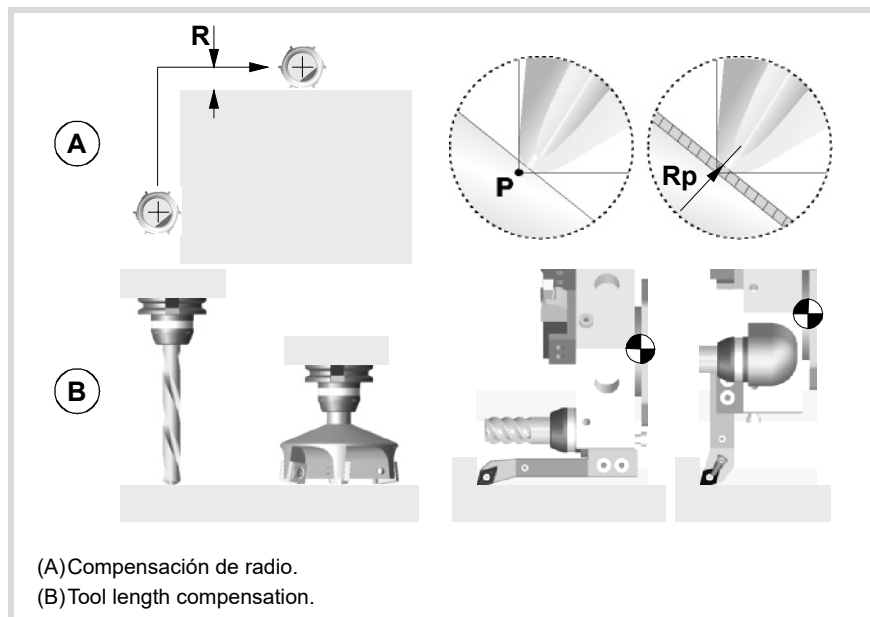
When working with tool radius compensation, the tool center follows the programmed path at a distance equal to the tool radius. Thus obtaining the right dimensions of the programmed part.

### Tool radius compensation (lathe).

The CNC assumes as theoretical tool tip (P) the result of the sides used when calibrating the tool. Without tool radius compensation, the theoretical tool tip (P) travels the programmed path leaving machining ridges in incline and curved sections. With tool radius compensation, it takes into account the tool tip radius and the location code (shape) or type of tool obtaining the dimensions of the programmed part.

### Tool length compensation.

When working with tool length compensation, the CNC compensates for the length difference between the different programmed tools.



## Compensation values

The compensation values applied in each case is calculated from the tool dimensions.

- In tool radius compensation, the applied value is the sum of the radius and radius wear of the selected tool.
- In tool length compensation, the applied value is the sum of the length and length wear of the selected tool.

The tool "T" and the tool offset "D", containing the tool dimensions, may be selected anywhere in the program, even while tool compensation is active. If no tool offset is selected, the CNC assumes tool offset "D1".

# 13.

TOOL COMPENSATION



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 13.1 Tool radius compensation

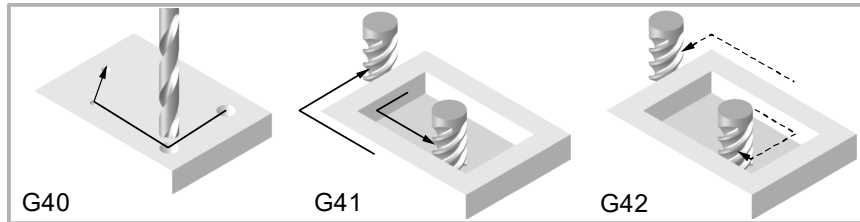
Radius compensation is applied in the active work plane, previously selected using functions G17 (XY plane), G18 (ZX plane), G19 (YZ plane) or G20 (user defined plane).

### Programming

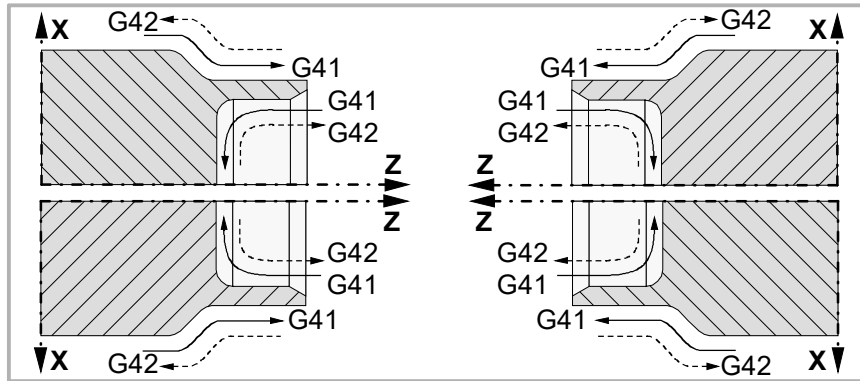
The functions for selecting tool radius compensation are:

- G41 Left-hand tool radius compensation.
- G42 Right-hand tool radius compensation.
- G40 Cancellation of tool radius compensation.

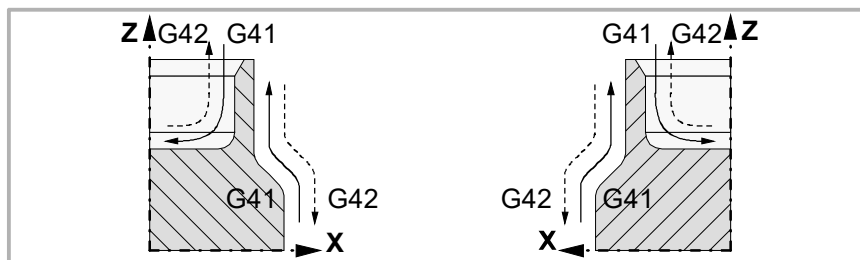
Tool radius compensation on a milling machine.



Tool radius compensation on a horizontal lathe.



Tool radius compensation on a vertical lathe.



Depending on the type of compensation selected (G41/G42), the CNC will place the tool will be placed to the left or to the right of the programmed path along the machining direction and will apply the compensation value. If no tool compensation is selected (G40), on a milling machine the CNC will place the tool center right on the programmed tool path; on a lathe, the CNC will place the theoretical tool tip on the programmed path.

When the radius compensation is active, the CNC analyzes the blocks to be executed beforehand, in order to detect any compensation errors related to steps, null arcs, etc. If any are detected, the blocks in which they originate will not be executed and a warning will appear on screen to warn the user that the programmed profile has been modified. A warning will come up for every profile correction made.

### Properties of the functions

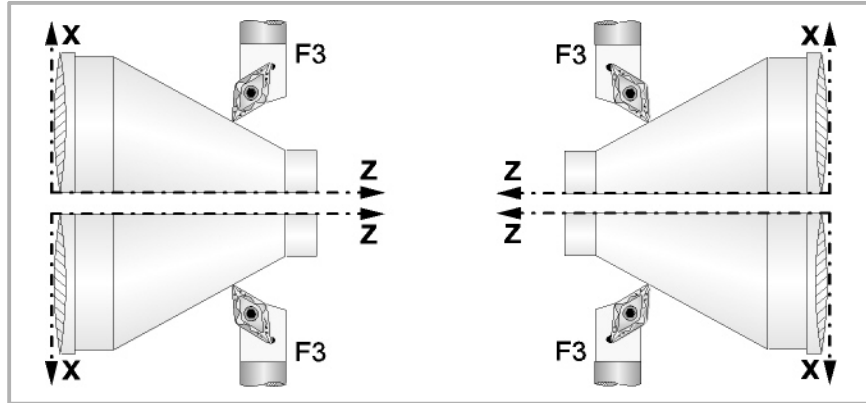
Functions G40, G41 and G42 are modal and incompatible with each other. On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G40.

### 13.1.1 Location code (shape or type) of the turning tools

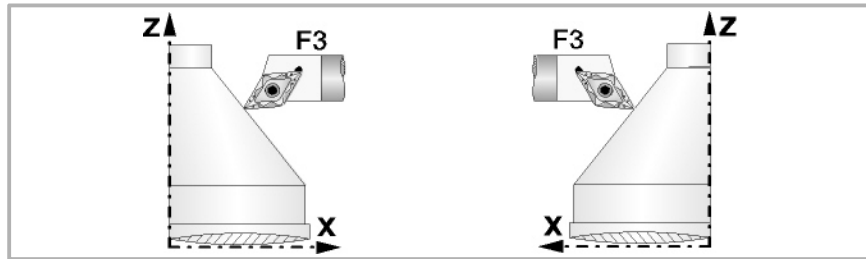
The location code indicates the type of tool and the sides used to calibrate it. The location code depends on the position of the tools and on the orientation of the machine axes.

The next example shows the location code F3 on different machines. Observe how the CNC keeps the relative position of the tool with respect to the axes.

Location code F3 on a horizontal lathe.



Location code F3 on a vertical lathe.



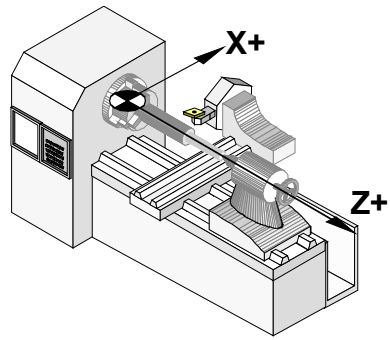
Here are the location codes available on most common horizontal lathes.

# 13.

## TOOL COMPENSATION

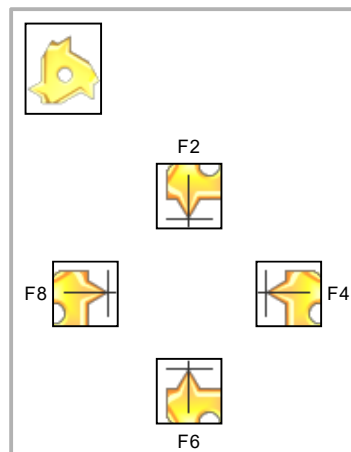
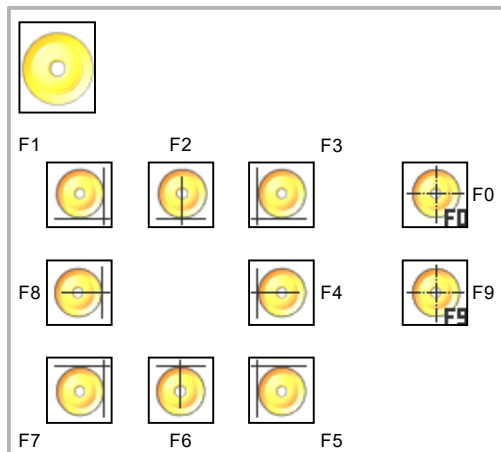
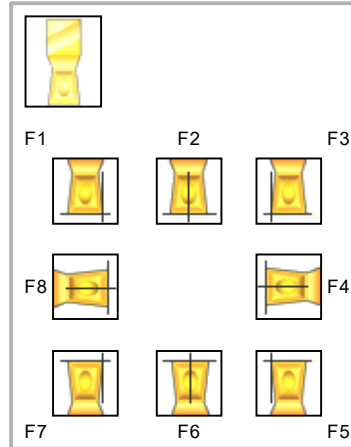
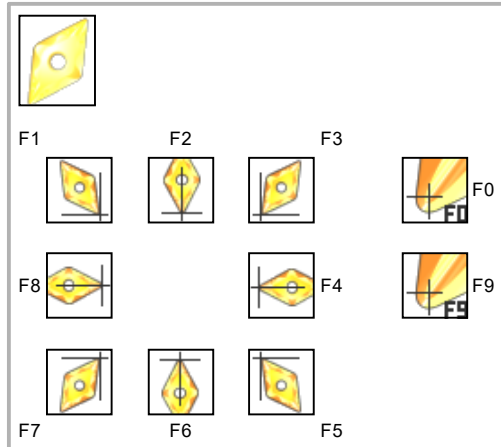
Tool radius compensation





# 13.

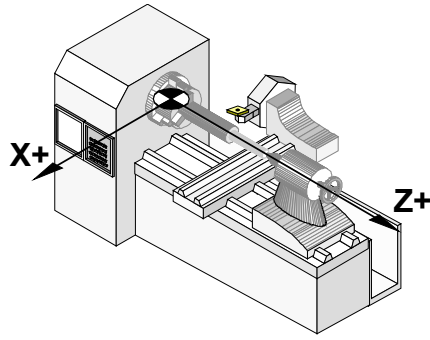
**TOOL COMPENSATION**  
Tool radius compensation



FAGOR AUTOMATION

CNC 8070

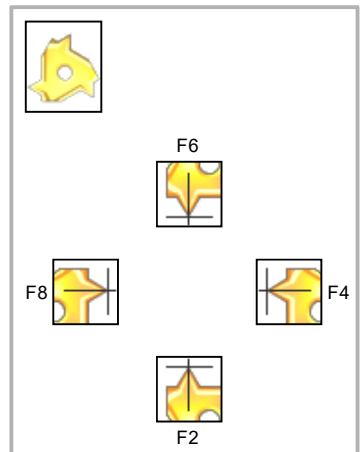
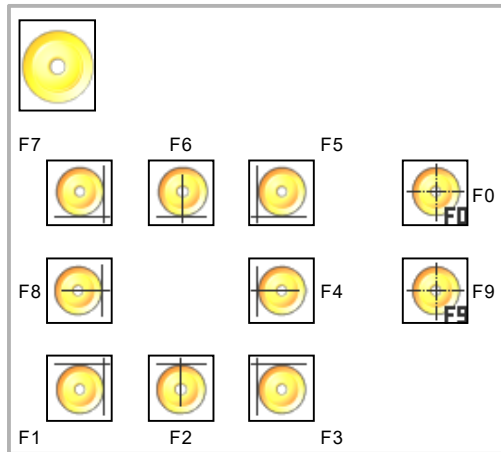
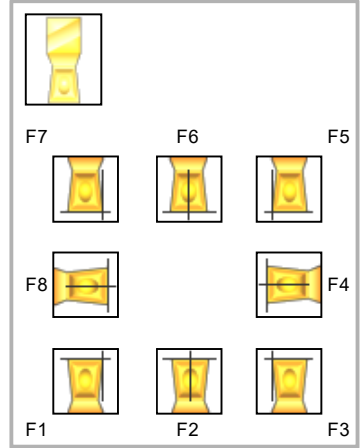
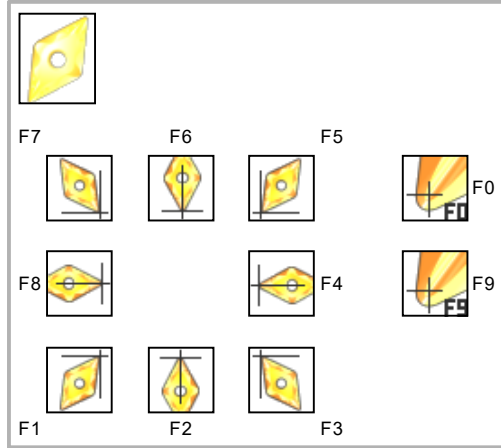
(REF: 2008)



# 13.

## TOOL COMPENSATION

Tool radius compensation



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 13.1.2 Functions associates with radius compensation

The functions associated with tool compensation may be programmed anywhere in the program, even while tool radius compensation is active.

### SELECTING THE TYPE OF TRANSITION BETWEEN BLOCKS

The transition between blocks determines how the compensated paths are joined together.

#### Programming

The type of transition may be selected from the program by means of the following functions:

- |      |                                     |
|------|-------------------------------------|
| G136 | Circular transition between blocks. |
| G137 | Linear transition between blocks.   |

#### G136

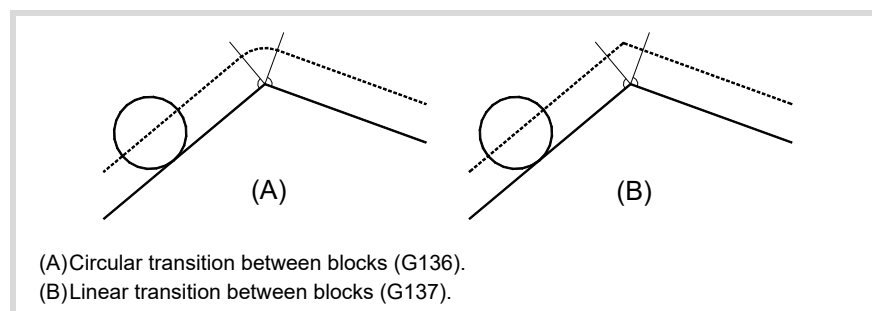
##### Circular transition between blocks.

Being function G136 active, the CNC joins the compensated paths using circular paths.

#### G137

##### Linear transition between blocks.

Being function G137 active, the CNC joins the compensated paths using linear paths.



#### Remarks

Later sections of this chapter offer graphic descriptions of how different paths are joined, depending on the type of transition selected (G136/G137).

#### Properties of the functions

Functions G136 and G137 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G136 or G137 depending on the value of machine parameter IRCOMP.

# 13.

**TOOL COMPENSATION**  
Tool radius compensation

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## HOW TOOL RADIUS IS ACTIVATED AND CANCELED

The functions associated with the strategy for activating and canceling establish how tool radius compensation starts and ends.

### Programming

The type of strategy may be selected from the program by means of the following functions:

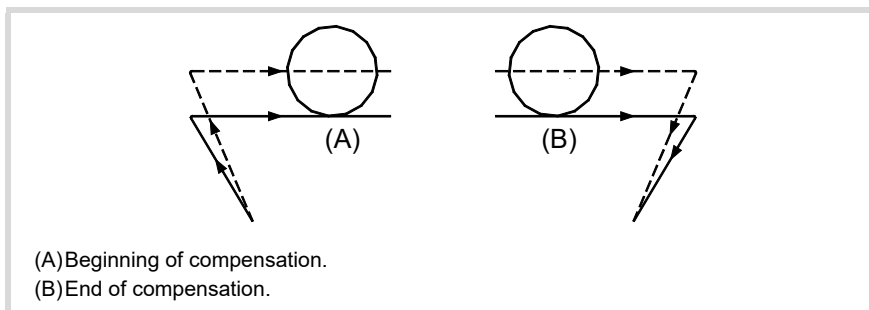
- |      |  |
|------|--|
| G138 | Direct activation/cancellation of tool compensation.   |
| G139 | Indirect activation/cancellation of tool compensation. |

#### G138

##### Direct activation/cancellation of tool compensation.

When compensation is turned on, the tool moves directly to the perpendicular of the next path (without contouring the corner).

When compensation is turned off, the tool moves directly to the programmed end point (without counting the corner).

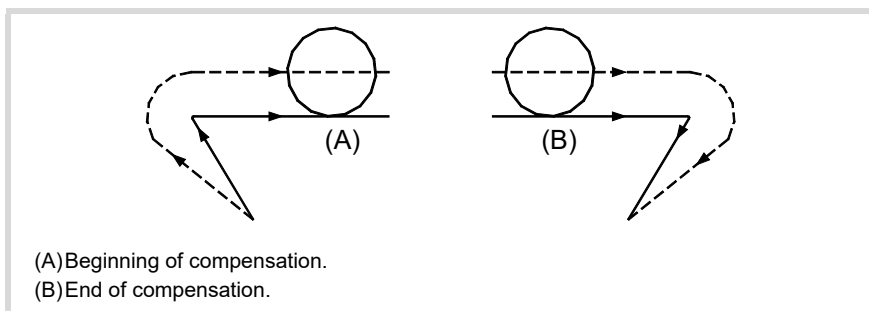


#### G139

##### Indirect activation/cancellation of tool compensation.

When compensation is turned on, the tool moves to the perpendicular of the next path contouring the corner.

When compensation is turned off, the tool moves to the end point contouring the corner.



The way the tool goes around the corner depends on the type of transition selected (G136/G37).

### Remarks

Later sections of this chapter offer a graphic description of how tool radius compensation begins and ends depending on the selected type of compensation ON/OFF (G138/G139).

### Properties of the functions

Functions G138 and G139 are modal and incompatible with each other.

# 13.

**TOOL COMPENSATION**  
Tool radius compensation



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G139.

# 13.

**TOOL COMPENSATION**  
Tool radius compensation

**FAGOR** 

FAGOR AUTOMATION

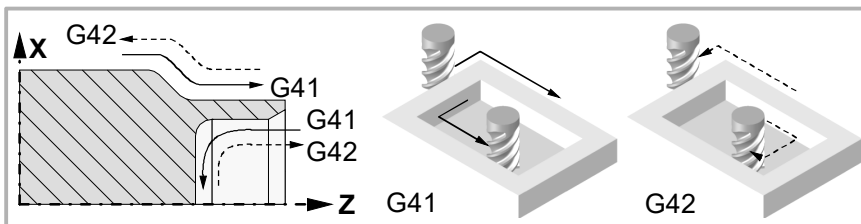
**CNC 8070**

(REF: 2008)

### 13.1.3 Beginning of tool radius compensation

Tool radius compensation is selected with these functions:

- G41 Left-hand tool radius compensation.
- G42 Right-hand tool radius compensation.



After executing one of these functions, radius compensation will be active for the next movement in the work plane, that must be a linear movement.

The way radius compensation will begin depends on how it is activated G138/G139 and on the type of transition G136/G137 selected:

- G139/G136  
The tool moves to the perpendicular of the next path contouring the corner along a circular path.
- G139/G137  
The tool moves to the perpendicular of the next path contouring the corner along linear paths.
- G138  
The tool moves directly to the perpendicular of the next path. Regardless of the type of transition (G136/G137) programmed.

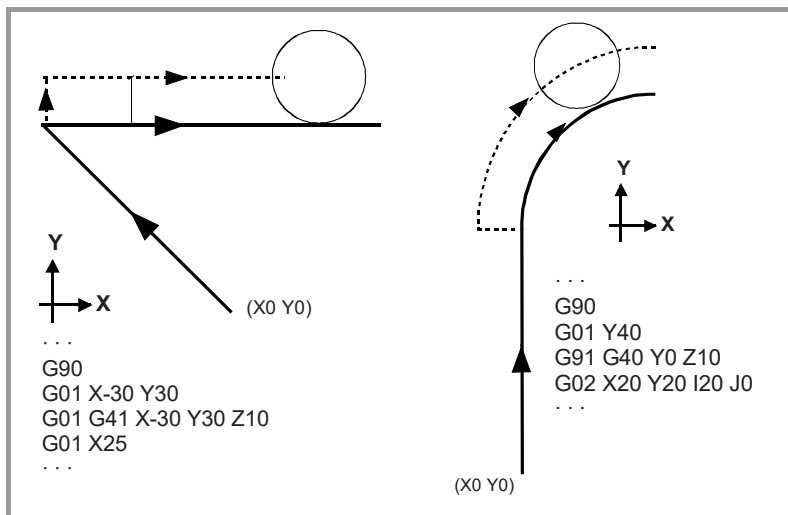
The following tables show the different ways tool compensation may begin, depending on the selected functions. The programmed path is shown with solid line and the compensated path with dashed line.

#### Beginning of the compensation without programmed movement

After activating the compensation, it may occur that the axes of the plane will not be involved in the first motion block. For example, because they have not been programmed, or the current tool position has been programmed or an incremental movement has been programmed.

In this case, the compensation is applied at the tool position as follows. Depending on the first movement programmed in the plane, the tool moves perpendicular to the path to its starting point.

The first movement programmed in the plane may be either linear or circular.



# 13.

TOOL COMPENSATION  
Tool radius compensation



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## STRAIGHT-TO-STRAIGHT PATH

When the angle between paths is smaller than or equal to 180°, the way radius compensation is activated is independent from the functions G136/G137 or G138/G139 selected.

$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between paths is greater than 180°, the way radius compensation is activated depends on the functions selected for type of beginning (G138/G139) and type of transition (G136/G137).

G139/G136	G139/G137	G138
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
$\alpha = 270^\circ$	$\alpha = 270^\circ$	$\alpha = 270^\circ$
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

**TOOL COMPENSATION**  
Tool radius compensation

**FAGOR**

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### STRAIGHT - ARC PATH

When the angle between the straight path and the tangent of the arc is smaller than or equal to 180°, the way radius compensation is activated is independent from the functions G136/G137 and G138/G139 selected.

$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between the straight path and the tangent of the arc is greater than 180° the way radius compensation is activated depends on the type of beginning (G138/G139) and type of transition (G136/G137) selected.

G139/G136	G139/G137	G138
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
$\alpha = 270^\circ$	$\alpha = 270^\circ$	$\alpha = 270^\circ$
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

## TOOL COMPENSATION

Tool radius compensation



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



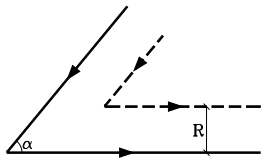
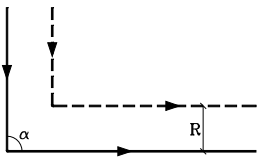
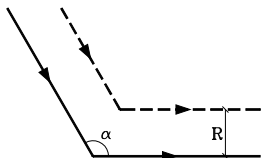
### 13.1.4 Sections of tool radius compensation

The way the compensated paths are joined depends on the type of transition selected (G136/G137).

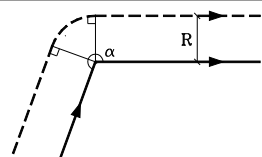
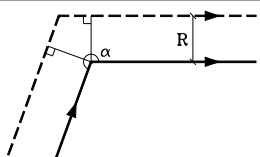
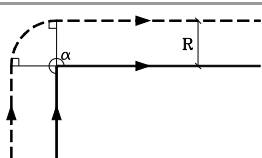
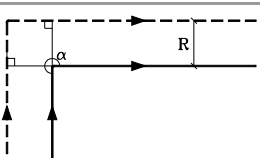
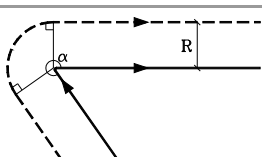
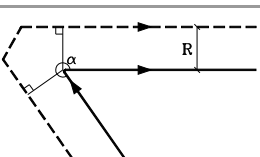
The following tables show the different transition possibilities between various paths depending on the selected function G136 or G137. The programmed path is shown with solid line and the compensated path with dashed line.

#### STRAIGHT-TO-STRAIGHT PATH

When the angle between paths is smaller than or equal to  $180^\circ$ , the transition between paths is independent from the G136/G137 function selected.

	
$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
	
$90^\circ < \tau_0 < 180^\circ$	

When the angle between paths is greater than  $180^\circ$ , the way the compensated paths are joined depends on the type of transition selected (G136/G137).

G136	G137
	
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
	
$\alpha = 270^\circ$	$\alpha = 270^\circ$
	
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

## STRAIGHT - ARC PATH

When the angle between the straight line and the tangent of the arc is smaller than or equal to 180°, the transition between the paths is independent from the selected G136/G137 function.

$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between the straight path and the tangent of the arc is greater than 180°, the way the compensated paths are joined depends on the type of transition selected (G136/G137).

G136	G137
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
$\alpha = 270^\circ$	$\alpha = 270^\circ$
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

**TOOL COMPENSATION**  
Tool radius compensation



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## ARC-TO-STRAIGHT PATH

When the angle between the tangent of the arc and the straight line is smaller than or equal to 180°, the transition between the paths is independent from the selected G136/G137 function.

$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between the tangent of the arc and the straight line is greater than 180°, the way the compensated paths are joined depends on the type of transition selected (G136/G137).

G136	G137
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
$\alpha = 270^\circ$	$\alpha = 270^\circ$
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

**TOOL COMPENSATION**  
Tool radius compensation

**FAGOR**

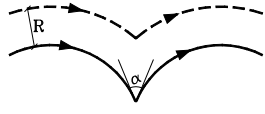
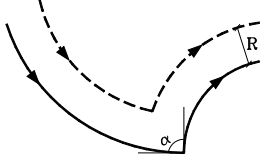
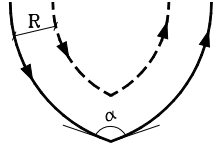
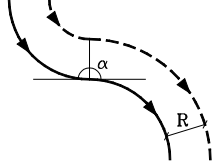
FAGOR AUTOMATION

CNC 8070

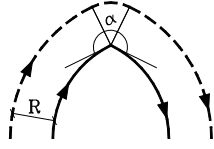
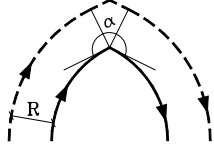
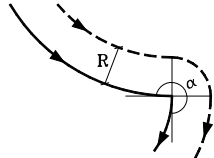
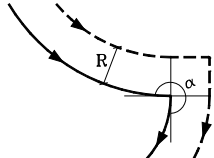
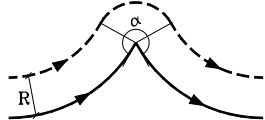
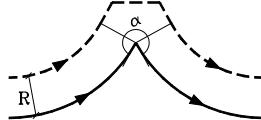
(REF: 2008)

## ARC-TO-ARC PATH

When the angle between the tangents of the arcs is smaller than or equal to  $180^\circ$ , the transition between the paths is independent from the selected G136/G137 function.

	
$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
	
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between the tangents of the arcs is greater than  $180^\circ$ , the way the compensated paths are joined depends on the type of transition selected (G136/G137).

G136	G137
	
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
	
$\alpha = 270^\circ$	$\alpha = 270^\circ$
	
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

**TOOL COMPENSATION**  
Tool radius compensation

### 13.1.5 Change of type of radius compensation while machining

The compensation may be changed from G41 to G42 or vice versa without having to cancel it with G40. It may be changed in any motion block or even in a motionless one; i.e. without moving the axis of the plane or by programming the same point twice.

It compensates independently the last movement before the change and the first one after the change. To change the type of compensation, the different cases are solved according to the following criteria:

**A** The compensated paths cut each other.

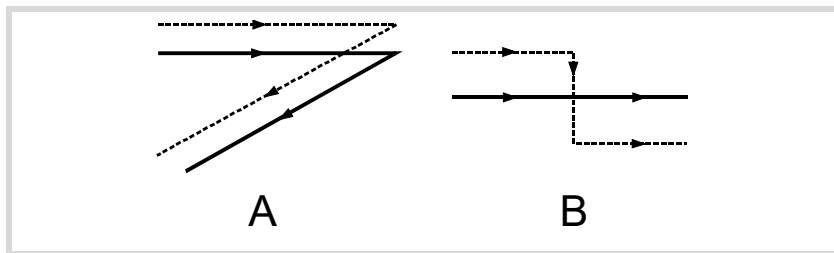
The programmed paths are compensated each on its corresponding side. The side change takes place in the intersection point between both paths.

**B** The compensated paths do not cut each other.

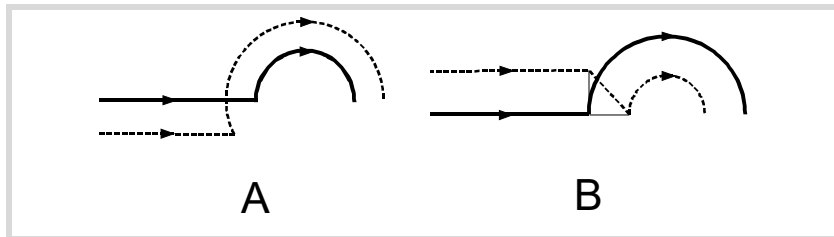
An additional section is inserted between the two paths. From the point perpendicular to the first path at the end point up to the point perpendicular to the second path at the starting point. Both points are located at a distance R from the programmed path.

Here is a summary of the different cases:

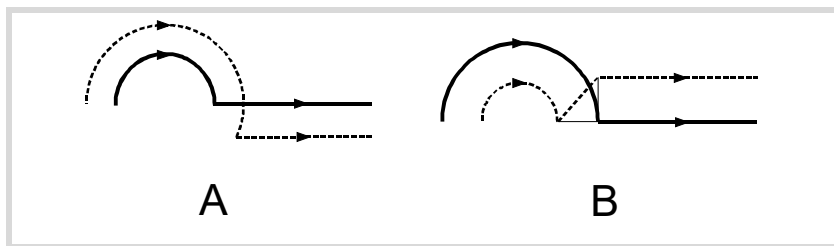
- Straight - straight path:



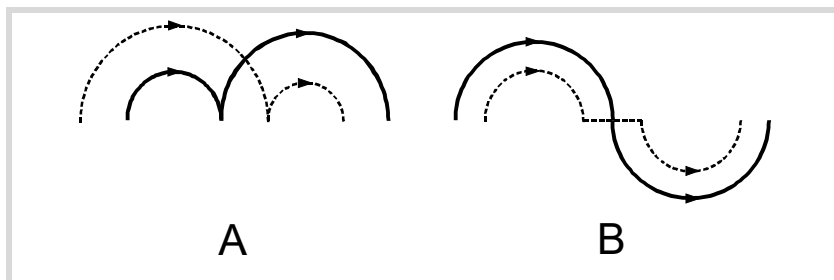
- Straight - circle path:



- Circle - straight path:



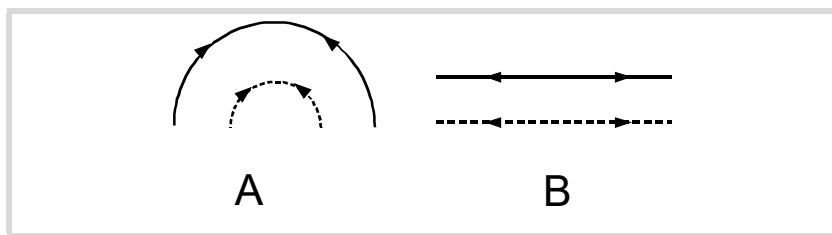
- Circle - circle path:



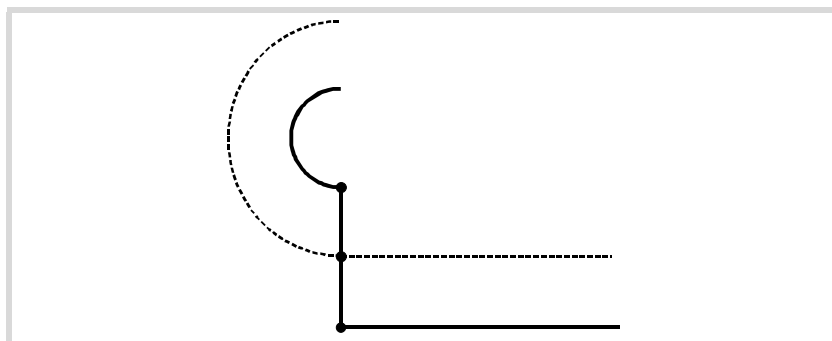
# 13.

**TOOL COMPENSATION**  
Tool radius compensation

- Back-and-forth path along the same way.



- Intermediate path as long as the tool radius:



# 13.

**TOOL COMPENSATION**  
Tool radius compensation



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 13.1.6 Cancellation of tool radius compensation

Tool radius compensation is canceled with function G40.

After executing one of this function, radius compensation will be canceled during the next movement in the work plane, that must be a linear movement.

The way this compensation is canceled depends on the type of cancellation end (G138/G139) and the type of transition G136/G137 selected:

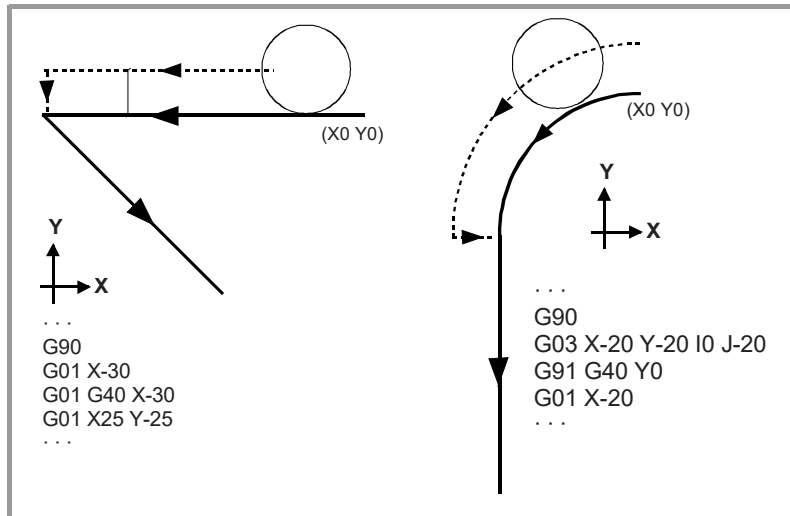
- G139/G136  
The tool goes to the endpoint, contouring the corner along a circular path.
- G139/G137  
The tool goes to the endpoint, contouring the corner along linear paths.
- G138  
The tool goes straight to the endpoint. Regardless of the type of transition (G136/G137) programmed.

The following tables show the different possibilities of canceling tool radius compensation depending on the selected functions. The programmed path is shown with solid line and the compensated path with dashed line.

#### End of the compensation without programmed movement

After canceling the compensation, it may occur that the axes of the plane will not be involved in the first motion block. For example, because they have not been programmed, or the current tool position has been programmed or an incremental movement has been programmed.

In this case, the compensation is canceled at the same point where the tool is, as follows. Depending on the last movement made in the plane, the tool moves to the end point (uncompensated) of the programmed path.



# 13.

TOOL COMPENSATION  
Tool radius compensation

## STRAIGHT-TO-STRAIGHT PATH

When the angle between the paths is smaller or equal to 180°, the way radius compensation is canceled is independent from the G136/G137 and G138/G139 functions selected.

$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between paths is greater than 180°, the way radius compensation is canceled depends on the functions selected for type of ending (G138/G139) and type of transition (G136/G137).

G139/G136	G139/G137	G138
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
$\alpha = 270^\circ$	$\alpha = 270^\circ$	$\alpha = 270^\circ$
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

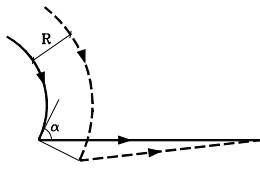
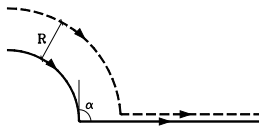
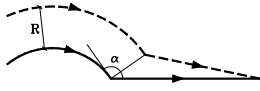
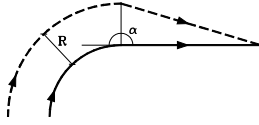
## TOOL COMPENSATION

Tool radius compensation

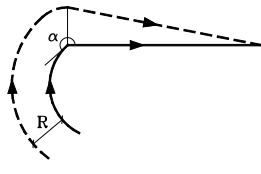
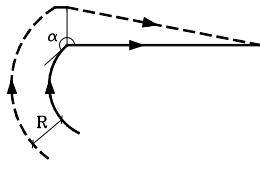
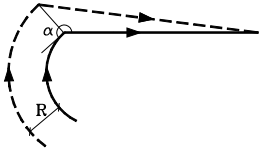
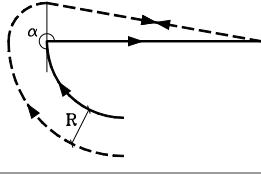
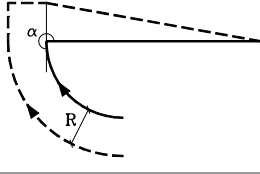
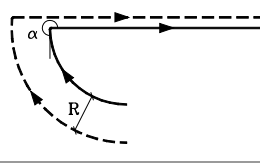
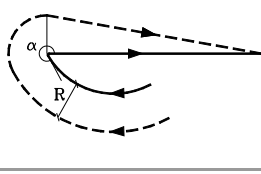
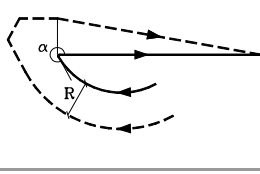
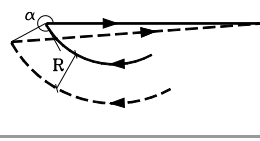


## TOOL PATH ARC-STRAIGHT

When the angle between the tangent of the arc and the straight path is smaller or equal to  $180^\circ$ , the way radius compensation is canceled is independent from the G136/G137 and G138/G139 functions selected.

	
$0^\circ < \tau_0 < 90^\circ$	$\alpha = 90^\circ$
	
$90^\circ < \tau_0 < 180^\circ$	$\alpha = 180^\circ$

When the angle between the tangent of the arc and the straight line is greater than  $180^\circ$ , the way radius compensation is canceled depends on the type of ending (G138/G139) and type of transition selected (G136/G137).

G139/G136	G139/G137	G138
		
$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$	$180^\circ < \tau_0 < 270^\circ$
		
$\alpha = 270^\circ$	$\alpha = 270^\circ$	$\alpha = 270^\circ$
		
$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$	$270^\circ < \tau_0 < 360^\circ$

# 13.

**TOOL COMPENSATION**  
Tool radius compensation

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 13.2 Tool length compensation

### Tool length compensation on a milling machine.

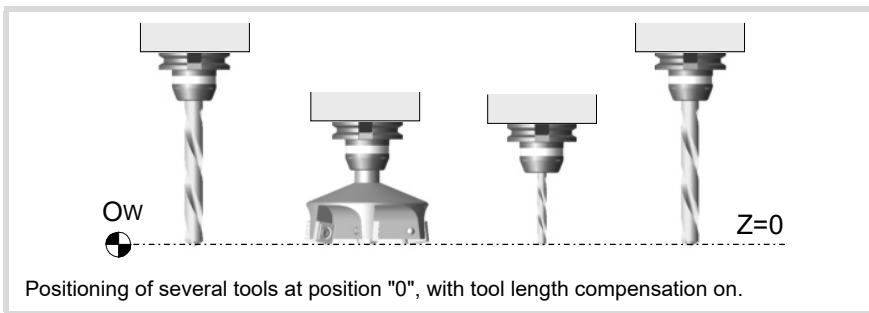
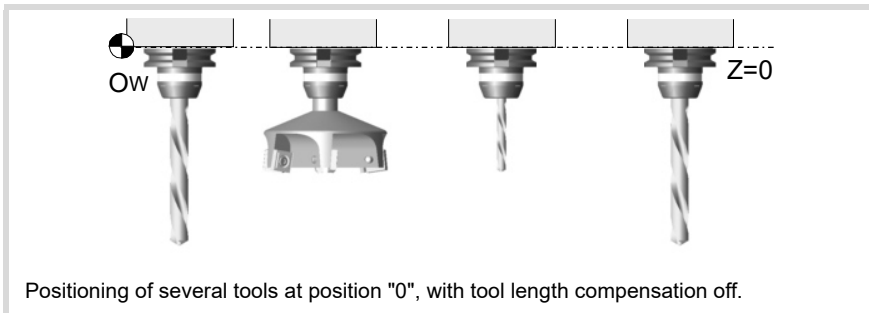
On a milling machine, tool length compensation is applied to the longitudinal axis; i.e. on the axis indicated by the instruction "#TOOL AX", or when missing, to the longitudinal axis designated by selecting the plane.

If G17, tool length compensation is applied to Z axis.

If G18, tool length compensation is applied to Y axis.

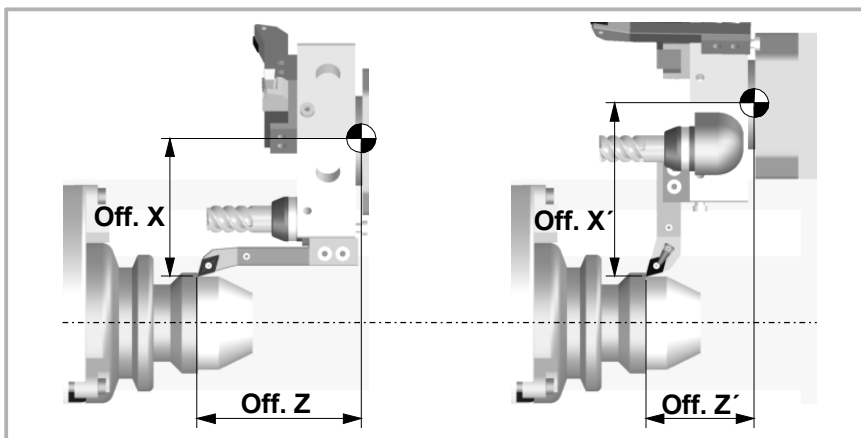
If G19, tool length compensation is applied to X axis.

Whenever any of functions G17, G18 or G19 is executed, the CNC assumes the axis perpendicular to the selected plane as the new longitudinal axis. If, then, "#TOOL AX" is executed, the new selected longitudinal axis replaces the previous one.



### Tool length compensation on a lathe.

In a turning operation, the CNC takes into account the dimensions of the new tool, defined in the corresponding tool offset, and moves the tool holding turret so the tip of the new tool occupies the same position as the previous one.



### Programming

Tool length compensation is activated when selecting a tool offset.

13.

TOOL COMPENSATION  
Tool length compensation

## Programming manual.

- To activate this compensation, program "D<n>", where <n> is the tool offset number that contains the tool dimensions that will be used as compensation values.
- To cancel this compensation, program "D0".

Once one of these codes has been executed, tool length compensation will be activated or cancel during the next movement of the longitudinal axis.

# 13.

**TOOL COMPENSATION**  
Tool length compensation

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 13.3 3D tool compensation.

In tool radius compensation (G41/G42) tool orientation is constant. 3D tool compensation allows changing the tool orientation during the path considering the dimensions and the shape of the tool.

There are two kinds of 3D compensation; paraxial compensation (or compensation factors) or the compensation calculated from the normal vector. In the first case, the CAM generates the program with the necessary blocks to generate the tool paths. In the second case, the CAM generates the blocks with a vector normal (perpendicular) to the surface and the CNC runs the relevant calculations to generate the paths. Both kinds of 3D compensation are incompatible with tool radius compensation (G41/G42).

# 13.

TOOL COMPENSATION  
3D tool compensation.

#### Programming. Activating 3D compensation.

This instruction must be programmed alone in the block. When programming this instruction, you must define the type of 3D compensation to activate.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

```
#COMP3D <ON>
```

```
#COMP3D <ON> [<{mode}>]
```

{mode} Optional. Type of compensation. Program one of the following commands.

- PARAX; paraxial 3D compensation (by default).
- NORMAL; 3D compensation with normal vector.

```
#COMP3D
#COMP3D ON
#COMP3D [PARAX]
#COMP3D ON [PARAX]
#COMP3D [NORMAL]
#COMP3D ON [NORMAL]
```

The paraxial mode is the default mode; however, inside a program, it keeps the mode selected last.

#### Type of compensation. Paraxial 3D compensation.

The CAM calculates the paths and gives the CNC a program with the necessary information to generate the paths at the corners. The CAM takes into account the tool shape and consequently the program may be executed with any type of tool.

The CAM adds to the motion blocks a vector (unnormalized) like N[P,Q,R]. The vector generated by the CAM is a compensation vector (paraxial vector) on the programmed coordinate, an offset vector. This vector is the equivalent to the one that the CNC would generate taking into account the vector normal to the surface, the type of tool and the intersection with the next tool path. Based on this vector, the CNC calculates the offset to be added to the programmed coordinate depending on the tool radius.

$$X \text{ offset} = \text{Tool radius} * P$$

$$Y \text{ offset} = \text{Tool radius} * Q$$

$$Z \text{ offset} = \text{Tool radius} * R$$

Paraxial compensation is a full 3D compensation for 5-axis machines, and it is applied for small movements and 3D surfaces. When working with rotary axes, we recommend activating the RTCP.

With this mode, the CNC can compensate profiles formed by segments, by segments and arcs tangents to each other and also by arcs as long they still are after being compensated for.

(REF: 2008)



FAGOR AUTOMATION

CNC 8070

**Type of compensation. 3D compensation with normal vector.**

The CAM generates a program with the necessary information for the CNC to generate the paths at the corners, depending on the type of tool, if necessary. This type of compensation can only be executed with cylindrical, toric or spherical tools.

The CAM adds to the motion blocks a unit vector normal to the surface like N[P,Q,R]. Based on this vector, the CNC calculates the offset to be added to the programmed coordinate depending on the type of tool and the intersection with the next path.

**Programming. Canceling 3D compensation.**

This instruction must be programmed alone in the block.

**Programming format.**

The programming format is:

```
#COMP3D OFF
```

```
#COMP3D OFF
```

**Considerations.**

- 3D compensation are incompatible with tool radius compensation (G41/G42).
- 3D compensation affects linear movements (G00, G01), circular movements (G02, G03, G08, G09) and threading (G33, G63).
- 3D compensation does not affect probing movements (G100, G103), home search (G74) or polynomials (#POLY).
- During tool inspection, the CNC temporarily cancels 3D compensation; in other words, it does not apply the normal vector to jog or MDI movements. The CNC restores 3D compensation when resuming the program after tool inspection.

**Properties of the function and Influence of the reset, turning the CNC off and of the M30 function**

On power-up, after executing an M02 or M30, and after an emergency or reset, the CNC cancels 3D compensation and initializes it to the paraxial compensation mode. When 3D compensation is active, the window for active G codes shows "C3D".

**13.**

**TOOL COMPENSATION**  
3D tool compensation.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 13.3.1 Programming the vector in the block.

The vector MUST BE programmed in all linear and circular motion blocks; if not programmed and 3D compensation is active, the CNC will issue an error message. If the vector is programmed but 3D compensation is not active, the CNC ignores that the vector has been programmed. This way, the same blocks may be used to generate compensated surfaces or not depending on whether #COMP3D is active or not.

## 13.

TOOL COMPENSATION  
3D tool compensation.

#### Programming.

The vector may be programmed anywhere in the block. For paraxial compensation, the vector may be unnormalized whereas for compensation with normal vector, the vector must be a unit vector.

#### Programming format.

The vector may be programmed anywhere in the block. The programming format is:

$N[\{p,q,r\}]$   
 $\{p,q,r\}$                       Components of the vector.

```
N[1,0,1]
N[-1,0,-1]
N[-1.4,-0.4,1.333]
N[P1,-P10,10]
N[P1+3,-P10-P2,10*P100]
```

#### Programming the vector.

The vector (paraxial or normal) is programmed like  $N[P,Q,R]$ , where the three components of the vector MUST BE programmed. Vector components may be numerical values, parametric or the result of math expressions.

#### Considerations about the vector (paraxial or normal).

Vector programming is not affected by the following coordinate transformations; but it is affected by mirror images.

- Programming in mm/inches (G70/G71).
- Programming in radius/diameters (G150/G151).
- Absolute/incremental coordinates (G90/G91).
- Scaling factor (G72).
- Zero offsets (G159).
- Coordinate rotation in the plane (G73).

The components of the vector  $N[p,q,r]$  are applied to the first three axes of the main trihedron of the channel regardless of the active plane (G17, G18, G19 or G20). If the first three axes of the channel are XYZ and the vector is  $N[A,B,C]$ , the A component is always applied to the X axis; B to the Y axis; C to the Z axis.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 14.1 Conditional block skip (/)



*In order to use this feature, the machine manufacturer must have defined the corresponding PLC operation. Refer to the machine's documentation for further information.*

The block skip condition (/) is governed by the mark BLKSKIP1 of the PLC; the user may do so using the keypad if the OEM has been provided a button or key for this purpose. If the mark is active, the CNC will skip the blocks where it is programmed (not executing them) and will go on to the next block.

### Programming.

The block skip condition must always be programmed at the beginning of the block.

### Programming format.

The programming format is:

/

```
/N10 X10 Y20 F1000
```

### Considerations.

The CNC analyzes the block skip condition during the block preparation. If the block skip needs to be analyzed at the time of execution, it is necessary to interrupt the block preparation, by programming the instruction #FLUSH in the previous block. See "[14.6 Interrupt block preparation \(#FLUSH\).](#)" on page 289.

```
#FLUSH  
/N10 X10 Y20 F1000
```

## 14.2 Abort the execution of the program and resume it in another block or program.



*In order to use this feature, the machine manufacturer must have defined the corresponding PLC operation. Refer to the machine's documentation for further information.*

The CNC offers a special interruption mode, handled via PLC, that allows aborting the execution of the program and resuming it either from a particular previously defined block or in another program. If the execution resumes in a different program, it will be executed from the beginning; it will not be possible to select the starting block.

The point where the execution is resumed is defined and canceled with the #ABORT instruction.

#ABORT            Define the execution resuming block or program.

#ABORT OFF      Canceling the execution resuming point.

Within the same program, it is possible to define several resume points; when interrupting the program, the CNC will use the one that is active at the time; in other words, the one executed last.

### Considerations.

#### Abort the program.

Usually, this feature is turned on and off using an external push button or key configured for that purpose. This interruption mode is not applied when pressing the [CYCLE STOP] key.

When interrupting the program from the PLC, the CNC channel aborts the execution of the program but without affecting the spindle, initializes the program history and resumes the execution in the point indicated by the active #ABORT instruction.

#### Abort a threading process and other machining operations that cannot be interrupted.

If the CNC aborts the program during a threading operation that cannot be interrupted, the CNC behaves like when doing a reset. After receiving the command to abort the execution, the CNC interrupts the execution once the operation has been properly completed. While the program is interrupted, it will be necessary to repeat the command to abort the program in order for the CNC to do so.

#### Considerations when resuming the program.

When the program is interrupted, the history is initialized. Thus, it is recommended to define the minimum machining conditions such as feedrate, ·M· functions, etc in the block where the execution will be resumed.

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE

Abort the execution of the program and resume it in another block or program.



## 14.2.1 Define the execution resuming block or program (#ABORT).

The point where the execution is resumed is defined with the #ABORT instruction. Within the same program, it is possible to define several resume points; when interrupting the program, the CNC will use the one that is active at the time; in other words, the one executed last. If no resume point has been defined, the execution continues in the #ABORT OFF instruction; if the instruction has not been defined, the program will jump to the end of the program (M30).

### Programming.

Program the instruction alone in the block. When defining this instruction, there is an option to define a block or program where it will resume execution.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
#ABORT
#ABORT {label}
#ABORT [{"path\name}"]
{label}           Block label.
{path\name}      Program name and location (path).
```

```
#ABORT
  (Cancels the active point).
  (Execution continues at #ABORT OFF; if there isn't one, resumes at M30).
#ABORT N120
  (The execution continues in block N120).
#ABORT [LABEL]
  (The execution continues in block [LABEL]).
#ABORT ["PRG.NC"]
  (The execution continues in program PRG.NC).
#ABORT ["C:\FAGORCNC\USERS\PRG\EXAMPLE.NC"]
  (The execution continues in program EXAMPLE.NC).
```

#### Label programming.

The labels that identify the blocks may be of a number type or a name type. In the program, the ":" character must be added to the number type labels after the block number.

```
#ABORT N50
.
.
N50: G01 G91 X15 F800
#ABORT [LABEL]
.
.
[LABEL] G01 G91 F800
```

#### Program name and location (path).

The program to be executed may be defined by either writing the full path or without it. When indicating the whole path, the CNC only looks for the program in the indicated folder. If the path has not been indicated, the CNC looks for the program in the following folders and in this order.

- 1 Directory selected with the #PATH instruction.
- 2 Directory of the program that executes the #ABORT instruction.
- 3 Directory defined by machine parameter SUBPATH.

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Abort the execution of the program and resume it in another block or program.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Considerations.

It is recommended to program the target labels at the beginning of the program, outside the main program. Otherwise, and depending on the length of the program, if the jump labels are defined at the end of it, the #ABORT instruction may take longer to find them.

### 14.2.2 Default point to continue the execution (#ABORT OFF).

If no resume point has been defined or it has been canceled, the execution continues in the #ABORT OFF instruction; if the instruction has not been defined, the program will jump to the end of the program (M30).

#### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

```
#ABORT OFF
```

```
#ABORT OFF
```

# 14.

**CONTROLLING THE EXECUTION AND DISPLAYING THE**

Abort the execution of the program and resume it in another block or program.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 14.3 Block repetition (NR).

### 14.3.1 Movement block repetition n times (NR/NR0).

The command NR indicates the number of times that programmed block movement is to be executed. If NR0 is programmed, the CNC executes the block only once.

#### Programming.

This command must be added to a movement block. When programming this command, the number of repetitions must be defined.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

NR{repetitions}

NR0

{repetitions}      Number of repetitions.

```
G91 G01 X34.678 F150 NR4
```

(The CNC repeats the block 4 times).

```
G91 G01 X34.678 F150 NR0
```

(The CNC executes the block once, without any repetitions).

#### Considerations.

##### Movement blocks under the influence of a canned cycle or modal subroutine.

If inside the influence zone of a canned cycle or modal subroutine, a movement block is program containing an NR command, the CNC executes the programmed movement and the canned cycle or subroutine the programmed number of times.

##### Do not execute the canned cycle or modal subroutine after the movement.

If the number of repetitions is zero (NR0), the CNC will only execute the programmed movement.

```
T11 M6
```

(Tool change).

```
F100 S800 M3
```

(Initial conditions).

```
G0 G90 X0 Y0 Z20
```

(Positioning).

```
G81 I-20 Z1
```

(Execution of the canned cycle for X0 Y0).

```
G91 X15 NR3
```

(Repeat the movement and canned cycle 3 times).

```
G90 X30 Y30 NR0
```

(Movement without executing the canned cycle).

```
G80
```

(End of canned cycle).

```
M30.
```

# 14.

### 14.3.2 Prepare a subroutine without executing it (NR0).

The command NR0 (always with a value set to 0) prevents the execution of the modal subroutines or modal cycle programmed in the block, but leaves it ready to execute during the following motion blocks. Movements can be defined in the following blocks or a subroutine.

#### Programming.

Add the NR0 command (always with a value set to 0) to a block with a modal subroutine or modal cycle.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

NR0

```
T11 M6
  (Tool change).
F100 S800 M3
  (Initial conditions).
G0 G90 Z100
G81 Z5 I-20 NR0
  (Canned cycle definition; without execution).
Y0 X20
  (Movement and execution of the canned cycle).
X40 NR0
  (Movement and execution of the canned cycle).
X60 NR0
  (Movement without executing the canned cycle).
X80
X100 NR0
Y50
X80 NR0
X60
X40
X20 NR0
G80
G0 G90 Z100
M30
```

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Block repetition (NR).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 14.4 Block group repetition (#RPT).

This instruction #RPT may be used to execute a portion of the program defined between two blocks must will be identified with labels. The number of times for the blocks to be repeated can be configured; if it is not programmed, the CNC repeats the block group only once. Once the repetition is done, the execution resumes at the block after the one containing the instruction #RPT.

The block group to be repeated must be defined in the same program or subroutine executing this instruction. They may also be after the program (after function M30)

For the block group to be repeated, where a second block repetition may have been defined and within this a third could have also been and so forth, the CNC limits this type of calls to a maximum of 20 nesting levels.

### Programming.

Program the instruction alone in the block. When programming this instruction, the start and end block repetition must be defined. Optionally, the number of times the blocks are to be repeated can be defined.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

```
#RPT [{label_begin}, {label_end}, <{repetitions}>]
```

{label_begin}	First block label.
{label_end}	Last block label.
{repetitions}	Number of repetitions. Optional (by default, 1).

```
#RPT [N100, N200]
```

(The CNC repeats the blocks N100 through N200 once).

```
#RPT [N18, N19, 7]
```

(The CNC repeats the blocks N18 through N19 seven times).

```
#RPT [[BEGIN], [END]]
```

(The CNC repeats the blocks [BEGIN] through [END] once).

### Label programming.

The labels that identify the blocks may be of a number type or a name type. The label of the first block may be part of the block being repeated, but the label of the last block must be alone in the block. The labels of the first and last blocks must be different.

- Programming with number labels. In the program, for the begin and end block labels, the character “:” must be added after the block number.

```
#RPT [N50,N70]
```

```
.
.
N50: G01 G91 X15 F800 (first block)
      X-10 Y-10
      X20
      X-10 Y10
N70: (last block)
```

# 14.

# 14.

- Programming with name labels.

```
#RPT [[BEGIN],[END]]
.
.
[BEGIN] G01 G91 F800    (first block)
  X-10 Y-10
  X20
  X-10 Y10
  G90
[END]    (last block)
```

## Considerations.



### Repeating the execution of a single block.

The execution of a single block repetition, program as follows: Movement blocks can also be repeated using the "NR" command. See "14.3.1 Movement block repetition n times (NR/NRO)." on page 283.

```
N10 #RPT [N10,N20,4]
N10: G01 G91 F800    (first block)
N20:    (last block)
```

### Block repetitions and execution loops (\$IF, \$WHILE, etc).

The block group to be repeated can include execution loops, such as \$IF, \$WHILE, etc. In this case, a loop closing instruction must always be accompanied with the corresponding opening instruction. If within the block group to be repeated there is only the loop closing instruction, the CNC will display the corresponding error.

Correct.	Incorrect.
	
<pre>#RPT [N10,N20] . N10: \$FOR P1=1,10,1 . . \$ENDFOR . N20:</pre>	<pre>#RPT [N10,N20] . \$FOR P1=1,10,1 . N10: . \$ENDFOR N20:</pre>

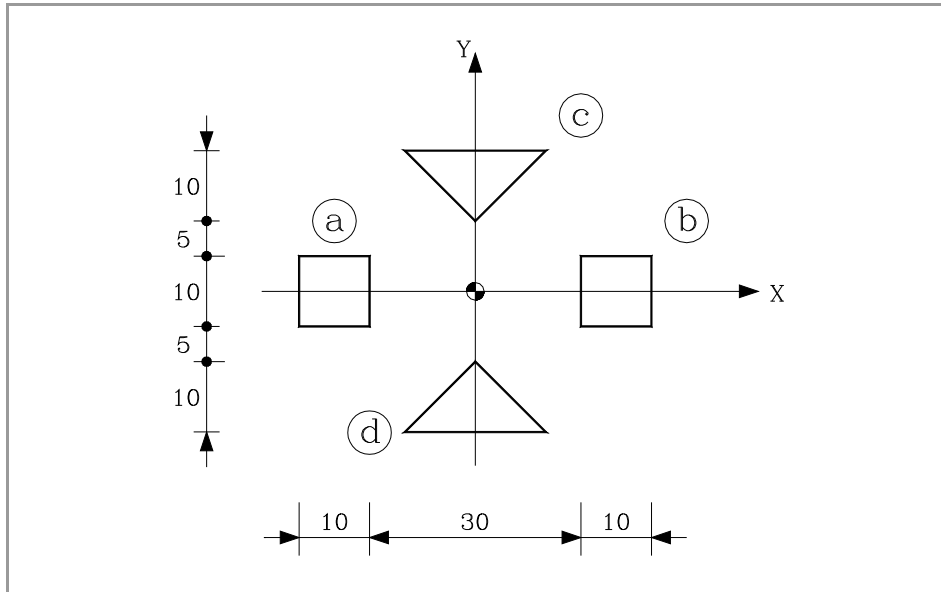


FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 14.4.1 Programming example.



```

%PROGRAM
G00 X-25 Y-5
N10: G91 G01 F800    (Profile "a")
      X10
      Y10
      X-10
      Y-10
      G90
N20:
G00 X15
#RPT [N10, N20]    (Profile "b")
#RPT [[INIT], [END], 2]    (Profiles "c" and "d")
M30

[INIT]
  G1 G90 X0 Y10
  G1 G91 X10 Y10
  X-20
  X10 Y-10
  G73 Q180
[END]
    
```

# 14.

**CONTROLLING THE EXECUTION AND DISPLAYING THE**  
Block group repetition (#RPT).

## 14.5 Interrupt block preparation until an event is caused (#WAIT FOR).

The instruction #WAIT FOR interrupts block preparation until the programmed condition is met.

### Programming.

Program the instruction alone in the block. When programming this instruction, the condition to resume the block preparation must be defined.

### Programming format.

The programming format is the following; the arguments appear between curly brackets.

#WAIT FOR [{condition}]

{condition}      True or false comparison.

```
#WAIT FOR [V.PLC.O[1] == 1]
```

(The CNC waits until the variable is set to ·1· to resume the block preparation).

### Considerations.

This instruction does not synchronize block preparation and execution; for synchronizing, use function #FLUSH. See "[14.6 Interrupt block preparation \(#FLUSH\)](#)." on page 289.

```
P100=1
```

```
#FLUSH
```

```
#WAIT FOR [P100==0]
```

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Interrupt block preparation until an event is caused (#WAIT FOR).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 14.6 Interrupt block preparation (#FLUSH).

The CNC reads several blocks ahead of the one being executed in order to calculate in advance the path to follow. This prior reading is known as "block preparation". The instruction #FLUSH interrupts block preparation, executes up to the last prepared block, synchronizes the preparation and execution of blocks and then goes on with the execution of the program and block preparation.

The blocks have data that is analyzed by the CNC when it is read; to analyze it when it is executed, then use the #FLUSH instruction.

### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

```
#FLUSH
```

```
#FLUSH
```

### Influence of the block preparation during the execution of some functions.

#### Radius compensation.

Careful must be taken with the programming of the instruction #FLUSH, since when they are inserted between machining blocks with compensation it may cause undesired profiles. It must be borne in mind that interrupting block preparation may result in compensated paths different from the one programmed, undesired joints when working with very short moves, jerky axis movements, etc.

#### Block skip condition.

The CNC analyzes the block skip condition during the block preparation. The instruction #FLUSH is used to evaluate a block skip condition at the time of the execution.

```
N110 #FLUSH
/N120 G01 X100
```

#### Variables.

The CNC checks certain variables during block preparation and others during execution.

- Variables that use the execution value. These variables the block preparation temporarily and the CNC resumes it when it is done reading/writing the variable.
- Variables that use the preparation value. Program the instruction #FLUSH from the previous block to force the evaluation of a variable when it is being executed.

## 14.7 Enable/disable the single-block treatment (#ESBLK/ #DSBLK).

The #ESBLK and #DSBLK instructions activate and deactivate the single block treatment.

#ESBLK      Activate the single-block treatment.

#DSBLK      Cancel the single-block treatment.

When the program is executed in "block-by-block" mode, the block group between both instructions is executed in a continuous cycle; that is, the execution does not stop at the end of the block, but rather continues onto the following block until the instruction #DSBLK has been reached.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is:

#ESBLK

#DSBLK

```
G01 X20 Y0 F850
G01 X20 Y20
#ESBLK      (Beginning of single block)
G01 X30 Y30
G02 X20 Y40 I-5 J5
G01 X10 Y30
G01 X20 Y20
#DSBLK      (End of single block)
G01 X20 Y0
M30
```

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Enable/disable the single-block treatment (#ESBLK/ #DSBLK).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 14.8 Enable/disable the stop signal (#DSTOP/#ESTOP).

The instruction #DSTOP disables the stop signal whether it comes from the operator panel ([STOP] key) or from the PLC. The instruction #ESTOP re-enables the stop signal.

#ESTOP      Enable the cycle stop signal.

#DSTOP      Disable the cycle stop signal.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is:

#ESTOP

#DSTOP

#DSTOP

#ESTOP

# 14.

## CONTROLLING THE EXECUTION AND DISPLAYING THE

Enable/disable the stop signal (#DSTOP/#ESTOP).

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 14.9 Enable/disable the feed-hold signal (#DFHOLD/#EFHOLD).

The instruction #DFHOLD disables the feed-hold signal coming from the PLC. The instruction #EFHOLD re-enables the free-hold signal.

#EFHOLD    Disable the feed-hold signal.

#DFHOLD    Disable the feed-hold signal.

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Enable/disable the feed-hold signal (#DFHOLD/#EFHOLD).

### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

#EFHOLD

#DFHOLD

```
#EFHOLD
```

```
#DFHOLD
```

### Operating the feed-hold signal of the PLC.

If the PLC activates the feed-hold input, the CNC channel interrupts momentarily the feedrate of the axes (keeps the spindle turning). If the PLC enables the feed-hold signal in a motionless block, the CNC continues executing the program until it detects a motion block. If the PLC deactivates the feed-hold, the movement of the axes continues. All the stops and starts of the axes are carried out with the corresponding acceleration and deceleration.

On Fagor supplied screens, the text "Freal" on the screens during automatic and jog modes appears in red when the feed-hold is active.

## 14.10 Block skip (\$GOTO).

The \$GOTO instruction continues the execution of the program for the defined block, which can be at a previous or later point of the program. The instruction \$GOTO and the destination block must be in the same program or subroutine; programming jumps to subroutines are permitted, nor between subroutines.

### Programming.

Program the instruction alone in the block or with the statement \$IF.

#### Programming format.

The programming format is:

```
$GOTO {label}
{label}          Block label.
```

#### Block label.

The labels that identify the blocks may be of a number type or a name type. In the program, the ":" character must be added to the number type labels after the block number.

```
$GOTO N50 (o $GOTO N50:)
.
.
N50: G01 G91 X15 F800
$GOTO [LABEL]
.
.
[LABEL] G01 G91 F800
```

### Considerations.

- There cannot be jumps to blocks contained in another instruction (\$IF, \$FOR, \$WHILE, etc).
- Although the flow controlling instructions must be programmed alone in the block, the \$GOTO instruction may added to an \$IF instruction in the same block. This way, it is possible to exit the blocks contained in an instruction (\$IF, \$FOR, \$WHILE, etc) without having to end the loop.

```
N10 P0=10
N20 $WHILE P0<=10
N30 G01 X[P0*10] F400
N40 P0=P0-1
N50 $IF P0==1 $GOTO N100
N60 $ENDWHILE
N100: G00 Y30
M30
```

# 14.

## 14.11 Conditional execution (\$IF).

### 14.11.1 Conditional execution (\$IF).

The instruction \$IF analyzes the programmed condition and, if it is true, it executes the blocks contained between the instructions \$IF and \$ENDIF. If the condition is false, the execution continues in the block after \$ENDIF.

#### Programming.

Program the instruction alone in the block. The \$IF instruction always ends with a \$ENDIF.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$IF {condition}
$ENDIF
{condition}      True or false comparison.
```

```
N20 $IF P1==1
N30 ...
N40 ...
N50 $ENDIF
N60 ...
(Si P1=1, the CNC executes blocks N30 to N40; if not, the execution continues at N60).
```

#### Considerations.

The \$IF instruction always ends with \$ENDIF, except when a \$GOTO instruction has been added, then it is not needed.

```
N20 $IF P1==1 $GOTO N40
N30...
N40: ...
N50...
(If P1=1, the execution will continue at block N40; if not, the execution continues at N30).
```

# 14.

### 14.11.2 Conditional execution (\$IF - \$ELSE).

The instruction \$IF analyzes the programmed condition and, if it is true, it executes the blocks contained between the instructions \$IF and \$ELSE. If the condition is false, the instruction \$IF executes the blocks contained between \$ELSE and \$ENDIF. The execution continues in the block after \$ENDIF.

#### Programming.

Program the instruction alone in the block. The \$IF instruction always ends with a \$ENDIF.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$IF {condition}
$ELSE
$ENDIF
{condition}      True or false comparison.
```

```
N20 $IF P1==1
    N30...
    N40...
N50 $ELSE
    N60...
    N70...
N80 $ENDIF
N90 ...
(Si P1=1, the CNC executes blocks N30 to N40; if not, it executes blocks N60 to N70).
```

# 14.

### 14.11.3 Conditional execution (\$IF - \$ELSEIF).

The instruction \$IF analyzes the programmed condition and, if it is true, it executes the blocks contained between the instructions \$IF and \$ELSEIF. The execution continues in the block after \$ENDIF.

If the condition \$IF is false, the instruction \$ELSEIF analyzes the programmed condition and, if it is true, it executes the blocks contained between the instructions \$ELSEIF and \$ENDIF (or the next \$ELSEIF if any). As many \$ELSEIF instructions as necessary may be programmed. The execution continues in the block after \$ENDIF.

The instruction \$ELSE is optional. In this case, if all the conditions are false, it will execute the blocks contained between \$ELSE and \$ENDIF.

#### Programming.

Program the instruction alone in the block. The \$IF instruction always ends with a \$ENDIF.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$IF {condition}
$ELSEIF {condition}
$ELSE
$ENDIF

{condition}      True or false comparison.
```

```
N20 $IF P1==1
    N30...
    N40...
N50 $ELSEIF P2==[-5]
    N60...
N70 $ELSE
    N80...
N90 $ENDIF
N100 ...
(If P1 = 1, it will execute blocks N30 through N40; the execution continues at N100)
(If P1 ≠ 1 and P2 = -5, it executes block N60; the execution continues at N100).
(If P1 ≠ 1 and P2 = -5, it executes block N80; the execution continues at N100).
```

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Conditional execution (\$IF).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 14.12 Conditional execution (\$SWITCH).

The instruction \$SWITCH calculates the value of an expression, and executes the block group associated with this value (blocks contained between \$CASE and \$BREAK). This instruction can have several groups of nested blocks (\$CASE), each associated with a value.

The instruction \$DEFAULT is optional. If the expression calculated for \$SWITCH does not coincide with any \$CASE, the CNC executes the group of blocks contained between the instructions \$DEFAULT and \$ENDSWITCH.

### Programming.

Program the instructions alone in the block. The \$SWITCH instruction always ends with a \$ENDSWITCH. The \$CASE instruction always ends with a \$BREAK.

### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$SWITCH {expression}
$CASE {value}
$BREAK
$DEFAULT
$ENDSWITCH
```

{expression}      Parameter, variable, arithmetic expression or rational expression.

{value}             Number, parameter, variable, arithmetic expression or rational expression.

```
N20 $SWITCH [P1+P2/P4]
  N30 $CASE 10
  .
  .
  N60 $BREAK
  N70 $CASE [P5+P6]
  .
  .
  N100 $BREAK
  N110 $DEFAULT
  .
  .
N140 $ENDSWITCH
N150 ...
```

The instruction \$SWITCH calculates the expression [P1+P2/P4].

- If the result is 10, the CNC executes blocks N40 to N50.
- If the result is [P5+P6], the CNC executes blocks N80 to N90.
- If the result does not coincide with any option, the CNC executes blocks N120 to N130.

The execution continues at N150.

# 14.

## 14.13 Block repetition (\$FOR)

The CNC repeats the execution of the blocks contained between \$FOR and \$ENDFOR for the number of times it is programmed. When \$FOR is executed, a counter takes the initial value and either increases or decreases this value according to what has been defined, until reaching the final value.

The \$BREAK instruction is optional, and ends the loop even if it has not finished the number of repetitions. The execution continues in the block after \$ENDFOR.

The instruction \$CONTINUE is optional, and starts the next repetition even when the current one has not finished. The blocks programmed after the instruction \$CONTINUE up to \$ENDFOR will be ignored.

### Programming.

Program the instructions alone in the block. The \$FOR instruction always ends with a \$ENDFOR.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$FOR {counter} = {value_begin},{value_end},{increment}
$ENDFOR
```

{counter}	Arithmetic parameter or write variable.
{value_begin}	Number, parameter, variable or arithmetic expression.
{value_end}	Number, parameter, variable or arithmetic expression.
{increment}	Number, parameter, variable or arithmetic expression.

```
N30 $FOR P1=0,10,2
.
.
.
N50 $ENDFOR
(The CNC repeats blocks N30 through N50, from P1=0 until P1=10, in increments of 2 (6 times)).
N12 $FOR V.P.VAR_NAME=20,15,-1
.
.
.
N42 $ENDFOR
(The CNC repeats blocks N22 through N32, from V.P.VAR_NAME=20 until V.P.VAR_NAME=15, in increments of -1 (5 times)).
```

#### Programming format.

The programming format is:

```
$BREAK
```

```
$FOR P1= 1,10,1
.
.
.
$IF P2==2
$BREAK
$ENDIF
.
.
$ENDFOR
(The loop stops when P1 is greater than 10, or if P2 is equal to 2).
```

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Block repetition (\$FOR)



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Programming format.**

The programming format is:

```
$CONTINUE
```

```
$FOR P1= 1,10,1  
.  
.  
  $IF P0==2  
  $CONTINUE  
  $ENDIF  
.  
.  
$ENDFOR  
(If P0=2, another repetition begins).
```

# 14.

**CONTROLLING THE EXECUTION AND DISPLAYING THE**  
Block repetition (\$FOR)



FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 14.14 Conditional block repetition (\$WHILE).

The CNC repeats the execution for the blocks contained between \$WHILE and \$ENDWHILE for as long as the defined condition is valid. The condition is analyzed at the beginning of each new loop.

The instruction \$BREAK is optional, and ends the loop even if the stop condition is not met. The execution continues in the block after \$ENDWHILE.

The instruction \$CONTINUE is optional, and starts the next loop even when the current one has not finished. The blocks programmed after the instruction \$CONTINUE up to \$ENDWHILE will be ignored.

### Programming.

Program the instructions alone in the block. The \$WHILE instruction always ends with a \$ENDWHILE.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$WHILE {condition}
$ENDWHILE
{condition}          True or false comparison.
```

```
$WHILE P1 <= 10
  P1=P1+1
  .
  .
  .
$ENDWHILE
(The loop repeats for as long as P1 is less than or equal to 10).
```

#### Programming format.

The programming format is:

```
$BREAK
$WHILE P1 <= 10
  .
  .
  $IF P2 == 2
  $BREAK
  $ENDIF
  .
  .
$ENDWHILE
(The loop stops when P1 is greater than 10, or if P2 is equal to 2).
```

#### Programming format.

The programming format is:

```
$CONTINUE
$WHILE P1 <= 10
  .
  .
  $IF P0 == 2
  $CONTINUE
  $ENDIF
  .
  .
$ENDWHILE
(If P0=2, a new loop begins).
```

# 14.

CONTROLLING THE EXECUTION AND DISPLAYING THE  
Conditional block repetition (\$WHILE).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 14.15 Conditional block repetition (\$DO).

The CNC repeats the execution of the blocks contained between \$DO and \$ENDDO for as long as the defined condition is valid. The condition is analyzed at the end of each loop, therefore the block group is executed at least once.

The instruction \$BREAK is optional, and ends the loop even if the stop condition is not met. The execution continues in the block after \$ENDDO.

The instruction \$CONTINUE is optional, and starts the next loop even when the current one has not finished. The blocks programmed after the instruction \$CONTINUE up to \$ENDDO will be ignored.

### Programming.

Program the instructions alone in the block. The \$DO instruction always ends with a \$ENDDO.

### Programming format.

The programming format is the following; the arguments appear between curly brackets.

```
$DO
$ENDDO {condition}
{condition}      True or false comparison.
```

```
$DO
  P1=P1+1
  .
  .
  .
$ENDDO P1 <= 10
(The loop repeats for as long as P1 is less than or equal to 10).
```

### Programming format.

The programming format is:

```
$BREAK
```

```
$DO
  .
  .
  $IF P2==2
  $BREAK
  $ENDIF
  .
  .
$ENDDO P1 <= 10
(The loop stops when P1 is greater than 10, or if P2 is equal to 2).
```

### Programming format.

The programming format is:

```
$CONTINUE
```

```
$DO
  .
  .
  $IF P0==2
  $CONTINUE
  $ENDIF
  .
  .
$ENDDO P1 <= 10
(If P0=2, a new loop begins).
```

# 14.

# 14.

## CONTROLLING THE EXECUTION AND DISPLAYING THE

Conditional block repetition (\$DO).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

A subroutine is a set of blocks that, once properly identified, may be called upon several times from another subroutine or from the program. Subroutines are normally used for defining a bunch of operations or movements that are repeated several times throughout the program. The CNC allows up to a total of seven subroutines to be executed per block (G180, G380, G500, functions M with subroutine, etc.).

## Types of subroutines.

The CNC has two types of subroutines, namely local and global. There is also a third type available, OEM subroutines, that are a special case of a global subroutine defined by the OEM. See "[15.5 OEM subroutine execution.](#)" on page 316.

### Global subroutines.

The global subroutine is stored in CNC memory as an independent program. This subroutine may be called upon from any program or subroutine being executed.

### Local subroutines.

The local subroutine is defined as part of a program. This subroutine may only be called upon from the program where it has been defined.

A program can have several local subroutines; but they all must be defined before the body of the program. A local subroutine can call a second local subroutine with the condition that the calling subroutine be defined after the one being called.

## Subroutine nesting levels and parameters.

Since a subroutine may be called upon from the main program (or a subroutine) and another subroutine from this one and so on. The CNC limits the number of these calls to a maximum of 20 nesting levels.

## Arithmetic parameters in the subroutines

### Local parameters.

The CNC has global parameters (accessible from the program or any subroutine) and local parameters (accessible only from the program or subroutine where they have been programmed).

Local parameters may be assigned to more than one subroutine up to 7 parameter nesting levels within the 20 subroutine nesting levels. Not all the subroutine call types change the nesting level; only the #PCALL, #MCALL calls and functions G180 to G189 and G380 to G399.

### Global parameters.

Global parameters will be shared by the program and the subroutines of the channel. They may be used in any block of the program and of the subroutine regardless of the nesting level they may be at.

**Common parameters.**

Common parameters will be shared by the program and the subroutines of any channel. They may be used in any block of the program and of the subroutine regardless of the nesting level they may be at.

**15.**

**SUBROUTINES.**



## 15.1 Executing subroutines from RAM memory.

If the same subroutines are executed repeatedly during execution, it is more efficient to load them into the RAM memory of the CNC because this way, they may be accessed faster and execution time may be consequently optimized. This option is valid for OEM as well as for user subroutines. To load a subroutine into RAM, it must have the extension fst. The reserved RAM space for subroutines is 5 Mb.

### **fst user subroutines.**

Subroutines with the fst extension that are not saved in the folder.\MTB\Sub. User routines having a fst extension are loaded into RAM memory during block preparation. The CNC checks whether or not it is loaded into RAM memory; if it is not and there is room for it, it loads it.

At the end of the program (M02/M30) or after a reset, if no other channel is executing subroutines, the CNC deletes them from its RAM memory. This way, if a user subroutine having a fst extension is edited or modified, the CNC assumes the changes the next time it executes it.

### **OEM (manufacturer's) subroutines.**

Subroutines with the fst extension saved in the folder.\MTB\Sub.

- Being the CNC in USER mode, the OEM routines having a fst extension are loaded into RAM memory when starting up the CNC application.

When the OEM is debugging his subroutines, they must have another extension so the changes are assumed without having to restart the application. Once they are debugged, the OEM must change the extension of the subroutines to fst so they are loaded into RAM memory.

- Being the CNC in SETUP mode, the OEM subroutines with an fst extension will be loaded into RAM memory during the preparation of the blocks. The CNC checks whether or not it is loaded into RAM memory; if it is not and there is room for it, it loads it. At the end of the program (M02/M30) or after a reset, if no other channel is executing subroutines, the CNC deletes them from its RAM memory. This way, the changes made in the subroutine will be taken into account the next time the program is executed.

**15.****SUBROUTINES.**  
Executing subroutines from RAM memory.**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 15.2 Definition of the subroutines

Like the body of the program, a subroutine has a header, a body and an end-of-subroutine function.

### Header of a local subroutine.

The header of the subroutine is a block consisting of the "%L" characters followed by a blank space and the name of the subroutine. The name of the subroutine may be up to 14 characters long and may consist of uppercase and lowercase characters as well as numbers (no blank spaces are allowed).

```
%L 0123456789
%L SUBROUTINE
%L SUB234S
```

The header must be programmed. The name of the header is used when calling a subroutine.

### Header of a global subroutine.

The header of a global subroutine is the same as that of a program; in other words, it is a block consisting of the "%" character followed by the name of the subroutine. The name may be up to 14 characters long and may consist of uppercase and lowercase characters as well as numbers (no blank spaces are allowed).

```
%0123
%GLOBSUBROUTINE
%PART923R
```

It is optional to program the header. The header name must not be used when calling a global subroutine, use the name of the file as it is stored at the CNC, instead.

The name defined in the header has nothing to do with the name of the file. The two may be different.

### End of a global or local subroutine.

The end of the subroutine is defined with M17, M29 or the #RET instruction and they all are equivalent. The #RETDSBLK instruction ends the subroutine and cancels the single block treatment. One of them must be programmed to end the subroutine.

```
M17
M29
#RET
#RETDSBLK
```

# 15.

SUBROUTINES.  
Definition of the subroutines



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 15.3 Subroutine execution.

The CNC offers the following types of commands to call the subroutines.

Command.	Call type.
L	Call to a global subroutine. Parameters cannot be initialized with this command.
LL	Call to a local subroutine. Parameters cannot be initialized with this command.
#CALL	Call to a global or local subroutine. Parameters cannot be initialized with this command.
#PCALL	Call to a global or local subroutine. Local parameters can be initialized with this command.
#MCALL	Modal call to a local or global subroutine. Local parameters can be initialized with this command.
#MDOFF	It makes a function non-modal.

When executing one of these commands, the CNC executes the selected subroutine. When the subroutine has ended, the program execution resumes at the calling statement.

### Location (path) of the global subroutines.

When calling a global subroutine its path (location) may be defined. When indicating the whole path, the CNC only looks for the subroutine in the indicated directory. If the path has not been indicated, the CNC looks for the subroutine in the following directories and in this order.

- 1 Directory selected with the #PATH instruction.
- 2 Directory of the program being executed.
- 3 Directory defined by machine parameter SUBPATH.

# 15.

**SUBROUTINES.**  
Subroutine execution.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 15.3.1 LL. Call to a local subroutine.

The command LL calls a local subroutine. This type of call allows initializing local parameters of the subroutine.

#### Programming format.

The programming format is:

LL sub  
sub                    Name of the subroutine

```
LL sub2.nc
```

### 15.3.2 L. Call to a global subroutine.

The command L calls a global subroutine. This type of call allows initializing local parameters of the subroutine. When it is a global subroutine, its whole path may be defined.

#### Programming format.

The programming format is:

L <path> sub  
path                    Optional. Subroutine location.  
sub                    Name of the subroutine

```
L C:\Cnc8070\Users\Prg\sub1.nc  
L C:\Cnc8070\Users\sub2.nc  
L Sub3.nc
```

# 15.

**SUBROUTINES.**  
Subroutine execution.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 15.3.3 #CALL. Call to a global or local subroutine.

The #CALL instruction calls a local or global subroutine. This type of call allows initializing local parameters of the subroutine. When it is a global subroutine, its whole path may be defined.

When there are two subroutines, one local and the other one global, with the same name, the following criteria is applied. If the path has been defined in the call, the CNC will execute the global subroutine, otherwise, it will execute the local one.

#### Programming format.

The programming format is:

```
#CALL <path> sub
```

path           Optional. Subroutine location.

sub             Name of the subroutine

```
#CALL C:\Cnc8070\Users\Prg\sub1.nc  
#CALL C:\Cnc8070\Users\sub2.nc  
#CALL Sub3.nc
```

#### Path definition.

Defining the path is optional. If defined, the CNC only looks for the subroutine in that folder; if not defined, the CNC looks for the subroutine in the default folders. See "[Location \(path\) of the global subroutines.](#)" on page 307.

# 15.

**SUBROUTINES.**  
Subroutine execution.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 15.3.4 #PCALL. Call to a global or local subroutine initializing parameters.

The #PCALL instruction calls a local or global subroutine. This type of call allows initializing local parameters of the subroutine. When it is a global subroutine, its whole path may be defined.

When there are two subroutines, one local and the other one global, with the same name, the following criteria is applied. If the path has been defined in the call, the CNC will execute the global subroutine, otherwise, it will execute the local one.

# 15.

SUBROUTINES.  
Subroutine execution.

#### Programming format.

The programming format is:

```
#PCALL <path> sub <P0..Pn>
```

path           Optional. Subroutine location.

sub             Name of the subroutine

P0..Pn         Optional. Initializing parameters.

```
#PCALL C:\Cnc8070\Users\Prg\sub1.nc
#PCALL C:\Cnc8070\Users\sub2.nc A12.3 P10=6
#PCALL Sub3.nc A12.3 F45.3 P10=6
```

#### How to set local parameters.

Calling a subroutine allows for the 57 local parameters (P0 to P57) to be initialized. The values of the parameters must be defined following the call statement and may be defined in two ways. Both ways of setting local parameters are equivalent and may be combined in the same block.

- The parameters P0 to P25 can also be defined using the letters from A to Z, so that "A" is equal to P0, "B" to P1 and so forth and where "Z" is equal to P25.
- The parameters P26 to P52 can also be defined using the letters from "D0=" to "D31=", so that "D0=" is equal to P26, "D1=" to P27 and so forth and where "D31=" is equal to P57.

#### Programming example.

```
#PCALL subroutine.nc A12.3 F45.3 P10=6 D0=34.12 D1=5 P28=0
```

#### Path definition.

Defining the path is optional. If defined, the CNC only looks for the subroutine in that folder; if not defined, the CNC looks for the subroutine in the default folders. See "[Location \(path\) of the global subroutines.](#)" on page 307.

#### Local parameter nesting levels.

If local parameters are initialized in the #PCALL instruction, this instruction generates a new nesting level for the local parameters. Remember that up to 7 parameter nesting levels are possible within 20 subroutine nesting levels.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 15.3.5 #MCALL. Modal call to a local or global subroutine.

The #MCALL instruction calls a local or global subroutine. This type of call allows initializing local parameters of the subroutine. When it is a global subroutine, its whole path may be defined.

When there are two subroutines, one local and the other one global, with the same name, the following criteria is applied. If the path has been defined in the call, the CNC will execute the global subroutine, otherwise, it will execute the local one.

With this type of call, the subroutine becomes modal; i.e. the subroutine remains active in successive movements and it is repeated at the end of each move. See "[Considerations about the modal character of the subroutine.](#)" on page 312.

#### Programming format.

The programming format is:

```
#MCALL <path> sub <P0..Pn>
```

path            Optional. Subroutine location.

sub             Name of the subroutine

P0..Pn         Optional. Initializing parameters.

```
#MCALL C:\Cnc8070\Users\Prg\sub1.nc
#MCALL C:\Cnc8070\Users\sub2.nc A12.3 P10=6
#MCALL Sub3.nc A12.3 F45.3 P10=6
```

#### How to set local parameters.

Calling a subroutine allows for the 57 local parameters (P0 to P57) to be initialized. The values of the parameters must be defined following the call statement and may be defined in two ways. Both ways of setting local parameters are equivalent and may be combined in the same block.

- The parameters P0 to P25 can also be defined using the letters from A to Z, so that "A" is equal to P0, "B" to P1 and so forth and where "Z" is equal to P25.
- The parameters P26 to P52 can also be defined using the letters from "D0=" a "D31=", so that "D0=" is equal to P26, "D1=" to P27 and so forth and where "D31=" is equal to P57.

#### Programming example.

```
#MCALL subroutine.nc A12.3 F45.3 P10=6 D0=34.12 D1=5 P28=0
```

#### Path definition.

Defining the path is optional. If defined, the CNC only looks for the subroutine in that folder; if not defined, the CNC looks for the subroutine in the default folders. See "[Location \(path\) of the global subroutines.](#)" on page 307.

#### Turning the subroutine into non-modal.

A subroutine stops being modal with the instruction #MDOFF and in the following cases: See "[15.4 #PATH. Define the location of the global subroutines.](#)" on page 315.

- After executing an M02 or an M30 and after a RESET.
- When changing the work plane.
- When programming a probing move (G100).
- When changing the configuration of the axes (#FREE AX, #CALL AX and #SET AX).
- When calling another subroutine (#PCALL, #CALL, L, LL, G180-G189).
- When activating a canned cycle.

## Local parameter nesting levels.

If local parameters are initialized in the #MCALL instruction, this instruction generates a new nesting level for the local parameters. Remember that up to 7 parameter nesting levels are possible within 20 subroutine nesting levels.

## Considerations about the modal character of the subroutine.

The modal subroutine will not be executed in the motion blocks programmed inside the subroutine itself or in the subroutines associated with T or M6. It will not be executed either, when programming a number of block repetitions using a NR value of ·0·.

If a motion block contains a number of repetitions NR other than 0 while a modal subroutine is active, both the movement and the subroutine will be repeated NR times.

If, when a subroutine is selected as modal, a block containing the #MCALL mnemonic is executed, the present subroutine will lose its modal quality and the new subroutine selected will be changed to modal.

**15.****SUBROUTINES.**  
Subroutine execution.

FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### 15.3.6 #MDOFF. Turning the subroutine into non-modal.

The subroutine stops being modal with the instruction #MDOFF. .

#### Programming format.

The programming format is:

#MDOFF

```
#MDOFF
```

15.

**SUBROUTINES.**  
Subroutine execution.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

### 15.3.7 #RETDSBLK. Execute subroutine as a single block.

The #RETDSBLK instruction ends the subroutine and cancels the single block treatment.

#### Programming format.

Program the instruction alone in the block and at the end of the subroutine.

```
#RETDSBLK
```

```
#RETDSBLK
```

#### How to create the subroutine.

For a subroutine to be executed as a single block, it usually has the following structure.

```
%Sub.nc
#ESBLK; Beginning of the single-block treatment.
.
.
#DSBLK; End of the single-block treatment.
#RET; End of subroutine.
```

When executing this subroutine in single-block mode, the [START] key must be pressed twice because the execution stops in the #RET block. To prevent this so the subroutine is executed with a single [START], the subroutine must begin with #ESBLK and end with #RETDSBLK.

```
%Sub.nc
#ESBLK; Beginning of the single-block treatment.
.
.
#RETDSBLK; End of subroutine and end of the single-block treatment.
```

# 15.

**SUBROUTINES.**  
Subroutine execution.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 15.4 #PATH. Define the location of the global subroutines.

The instruction #PATH defines the pre-determined location of the global subroutines. If no path is defined in the global subroutine call, the CNC will look for the subroutine in the path defined using the instruction #PATH.

If the path is defined when calling a global subroutine, the CNC looks for the subroutine in that location, it ignores the location defined in the #PATH statement.

### Programming format.

The programming format is:

```
#PATH ["path"]
```

path            Pre-determined subroutine location.

```
#PATH ["C:\Cnc8070\Users\Prg\"]
```

```
#PATH ["C:\Cnc8070\Users\"]
```

15.

**SUBROUTINES.**

#PATH. Define the location of the global subroutines.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 15.5 OEM subroutine execution.

The machine manufacturer may define up to 30 subroutines per channel and associate them with functions G180 through G189 and G380 through G399 in such a way that when a channel executes one of these functions, it will execute the subroutine associated with that function for that channel. These OEM subroutines may be executed either in a non-modal or modal way and it also allows resetting the local parameters of the subroutine.

# 15.

### Programming format.

These functions may be programmed anywhere in the program and they don't have to go alone in the block and allow resetting the local parameters of the subroutine.

#### Programming format. Executing the subroutine in a non-modal way.

The programming format is the following; the list of arguments appears inside the curly brackets. To execute the subroutine in a non-modal way, call it using the G code (G180, G181, etc).

G180

G380

G180 {P0..Pn}

G380 {P0..Pn}

P0..Pn      Optional. Local parameters of the subroutine.

```
G180
G183 P1=12.3 P2=6
G388 A12.3 B45.3 P10=6
```

#### Programming format. Executing the subroutine in a modal way.

The programming format is the following; the list of arguments appears inside the curly brackets. To execute the subroutine in a modal way, call it using the MG code (MG180, MG181, etc).

MG180

MG380

MG180 {P0..Pn}

MG380 {P0..Pn}

P0..Pn      Optional. Local parameters of the subroutine.

```
G180
G183 P1=12.3 P2=6
G388 A12.3 B45.3 P10=6
```

#### How to set local parameters.

Calling a subroutine allows for the 57 local parameters (P0 to P57) to be initialized. The values of the parameters must be defined following the call statement and may be defined in two ways. Both ways of setting local parameters are equivalent and may be combined in the same block.

- The parameters P0 to P25 can also be defined using the letters from A to Z, so that "A" is equal to P0, "B" to P1 and so forth and where "Z" is equal to P25.
- The parameters P26 to P52 can also be defined using the letters from "D0=" to "D31=", so that "D0=" is equal to P26, "D1=" to P27 and so forth and where "D31=" is equal to P57.

#### Programming example.

```
G180 A12.3 F45.3 P10=6 D0=34.12 D1=5 P28=0
```

**Additional data in the block.**

Besides initializing the parameters, any other type of additional information may be added to these functions, even movements. This information must be programmed before the subroutine calling function; otherwise, the data will be considered as for initializing the parameters. The associated subroutine is executed once the execution of the rest of the information programmed in the block has ended.

```
G01 X50 F450 G180 P0=15 P1=20
```

It first executes the programmed movement to point X50 and then the subroutine associated with G180 initializing parameters P0 and P1.

```
G180 P0=15 P1=20 G01 X50 F450
```

All the data is interpreted as parameter setting, where P6(G)=1, P23(X)=50 and P5(F)=450.

**Cancelling a modal subroutine.**

A subroutine stops being modal and in the following cases.

- When programming G80 or #MDOFF.
- When changing the work plane.
- When programming a probing move (G100).
- When executing another subroutine (#PCALL, #CALL, #MCALL, L, LL, G180-G189, G380-G399).
- When executing a canned cycle.
- After executing an M02 or an M30 and after a RESET.
- When changing the configuration of the axes (#FREE AX, #CALL AX and #SET AX).

**Local parameter nesting levels.**

If these functions initialize local parameters, this instruction generates a new nesting level for the local parameters. Remember that up to 7 parameter nesting levels are possible within 20 subroutine nesting levels.

**Considerations about the modal character of the subroutine.**

The modal subroutine will not be executed in the motion blocks programmed inside the subroutine itself or in the subroutines associated with T or M6. It will not be executed either, when programming a number of block repetitions using a NR value of ·0·.

If a motion block contains a number of repetitions NR other than 0 while a modal subroutine is active, both the movement and the subroutine will be repeated NR times.

If while a subroutine is selected as modal, another modal OEM subroutine is executed, the current subroutine will stop being modal and the new selected subroutine will become modal.

**Properties of the function and influence of the reset, turning the CNC off and of the M30 function.**

Functions G180-G189 and G380-G399 are not modal. Functions MG180-MG189 and MG380-MG399 are modal.

## 15.6 Generic user subroutines (G500-G599).

The CNC lets the users define up to 100 subroutines, common to all channels and that will be associated with functions G500 through G599, so when the CNC executes one of these functions, it will execute its associated subroutine. These OEM subroutines may be executed either in a non-modal or modal way and it also allows resetting the local parameters of the subroutine.

These subroutines are loaded into RAM memory when they are executed for the first time. If there is no room in RAM, the CNC will issue a warning and it will execute the subroutine from the disk. At the end of the program (M30), if no other channel is executing subroutines, the CNC deletes them from its RAM memory. This way, if a user subroutine is edited or modified, the CNC assumes the changes the next time it executes it.

When updating the version, only the subroutines supplied by Fagor will be updated when choosing the third installation level "rename previous version and install completely".

### Programming format.

These functions may be programmed anywhere in the program and they don't have to go alone in the block and allow resetting the local parameters of the subroutine.

#### Programming format. Executing the subroutine in a non-modal way.

The programming format is the following; the list of arguments appears inside the curly brackets. To execute the subroutine in a non-modal way, call it using the G code (G500, G501, etc).

```
G500
G500 {P0..Pn}
P0..Pn      Optional. Local parameters of the subroutine.
```

```
G500
G583 P1=12.3 P2=6
G588 A12.3 B45.3 P10=6
```

#### Programming format. Executing the subroutine in a modal way.

The programming format is the following; the list of arguments appears inside the curly brackets. To execute the subroutine in a modal way, call it using the MG code (MG500, MG501, etc).

```
MG500
MG500 {P0..Pn}
P0..Pn      Optional. Local parameters of the subroutine.
```

```
G500
G583 P1=12.3 P2=6
G588 A12.3 B45.3 P10=6
```

#### How to set local parameters.

Calling a subroutine allows for the 57 local parameters (P0 to P57) to be initialized. The values of the parameters must be defined following the call statement and may be defined in two ways. Both ways of setting local parameters are equivalent and may be combined in the same block.

- The parameters P0 to P25 can also be defined using the letters from A to Z, so that "A" is equal to P0, "B" to P1 and so forth and where "Z" is equal to P25.
- The parameters P26 to P52 can also be defined using the letters from "D0=" a "D31=", so that "D0=" is equal to P26, "D1=" to P27 and so forth and where "D31=" is equal to P57.

```
Programming example.
G588 P0=12.3 P5=45.3 K6 P26=34.12 P27=5 D2=0
```

15.

SUBROUTINES.  
Generic user subroutines (G500-G599).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Additional data in the block.**

Besides initializing the parameters, any other type of additional information may be added to these functions, even movements. This information must be programmed before the subroutine calling function; otherwise, the data will be considered as for initializing the parameters. The associated subroutine is executed once the execution of the rest of the information programmed in the block has ended.

```
G01 X50 F450 G500 P0=15 P1=20
    It first executes the programmed movement to point X50 and then the subroutine associated with G500 initializing parameters P0 and P1.
```

```
G500 P0=15 P1=20 G01 X50 F450
    All the data is interpreted as parameter setting, where P6(G)=1, P23(X)=50 and P5(F)=450.
```

**User subroutines (G500-G599) and modal subroutines.**

User routines do not alter the modal/non-modal status of other possible active subroutines, this means should there be an active subroutine with #MCALL among the user subroutines, it will remain modal.

For this case, if the program activates a local subroutine as a modal subroutine and within the user subroutine there are motion blocks, the CNC will display a subroutine not found error. Modal subroutines must be global, so they may be used outside the range of influence of the program.

**Cancelling a modal subroutine.**

A subroutine stops being modal and in the following cases.

- When programming G80 or #MDOFF.
- After executing an M02 or an M30 and after a RESET.

**Define the subroutines.**

The subroutines associated with these functions will be global and will have the same name as the function, without extension. The subroutines must be defined in the folder ..\Users\Sub. If the CNC executes a function and there is no subroutine, the CNC will issue an error message.

- G500 will have subroutine G500 associated with it.
- G501 will have subroutine G501 associated with it.
- ...
- G599 will have subroutine G599 associated with it.

**Subroutines supplied by Fagor.**

Subroutine.	Meaning.
G500	HSC cancellation.
G501	HSC activation for roughing operations.

**Execute the subroutines.**

The CNC executes the subroutine after executing the function with which they are associated. To execute the subroutine as a single block, program the instructions #ESBLK and #RETDSBLK. After executing the instruction #ESBLK, the CNC executes the blocks programmed next as a single block until it reaches the end of the subroutine (#RETDSBLK).

If the file that contains the subroutine has the "hidden" attribute, the CNC does not display the contents of the subroutine while it is executed. The attributes of the files may be modified from the utilities mode (see the operating manual).

## Local parameter nesting levels.

If these functions initialize local parameters, this instruction generates a new nesting level for the local parameters. Remember that up to 7 parameter nesting levels are possible within 20 subroutine nesting levels.

## Properties of the function and Influence of the reset, turning the CNC off and of the M30 function.

Functions G500 - G599 are not modal. Functions MG500 - MG599 are modal.

**15.****SUBROUTINES.**

Generic user subroutines (G500-G599).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 15.7 Assistance for subroutines.

### 15.7.1 Subroutine help files.

Help files may be associated with each OEM subroutine (G180, G380, etc), user subroutine (G500, G800, etc) and each global subroutine called upon using #MCALL or #PCALL and they will be displayed while editing.

The help window is displayed while editing, after a blank space or tab following a G function or following the name of the subroutine. The help window is only informative, it cannot be accessed with the cursor nor browse through it. The help window closes with [ESC], deleting the key word or going on to another line of the program.

The help window of the subroutines is only available when the editor uses the CNC language; when the editor is enabled for the 8055 CNC, these helps are not available. The help window of the subroutines is available even when the contextual helps of the editor are disabled.

When the help file is displayed, its text may be inserted into the part-program using the [INS] key.

#### Edit the help files.

Each subroutine may have two help files; a text file (txt) and an image file (bmp). There's no need to define both files; either one may be defined alone.

When the help window is only informative, it cannot be accessed with the cursor nor browse through it with the page-up-down keys. This is why it is recommended to use short help files; for example, that only contain the description of the parameters of the subroutine. Also, since the text of the help file can be inserted into the program ([INS] key), the following is recommended.

- That the help file contains the line calling the subroutine. Since the user must have written part of the call to display the help window, the editor deletes the call before inserting the help text.
- That all the lines of the help file follow the format of a CNC comment, except the line containing the call to the subroutine.

Help file example for a subroutine.

```
G180 P0= P1= P2= P3= P4= P5=
#COMMENT BEGIN
----- G180 -----
P1 = Movement in X
P2 = Movement in Y
P3 = Movement in Z
P4 = Feedrate F
P5 = Speed S
-----
#COMMENT END
```

#### Name and location of files.

##### Name of the help files.

The name of the files must follow the following rule:

Subroutine.	Name of the help files.
G180-G189 G380-G399 G500-G599 G800-G899 G8000-G8999	The name of the files must be the function it is associated with. For example <i>G180.txt</i> y <i>G180.bmp</i> .
#MCALL #PCALL	The name of the files must be the name of the subroutine. For example <i>subroutine.txt</i> and <i>subroutine.bmp</i> .

**Where to save the help files.**

The machine manufacturer can save the help files in the folders `..\Mtb\Sub\Help` and `..\Mtb\Sub\Help\{language}`. Since the modifications to the MTB directory in the "User" work mode disappear when turning the unit off, the user must save his help files in the folder `..\Users\Sub\Help` and `..\Users\Sub\Help\{language}`.

The CNC looks for the files in the following order and displays the first that it finds, this is the reason why it is recommended that the user does not define subroutines and/or help files with the same name as those of the OEM. If there are no help files, the CNC will not show any help and it will not display an error.

```

..\Users\Sub\Help\{idioma}
..\Users\Sub\Help
..\Mtb\Sub\Help\{idioma}
..\Mtb\Sub\Help\

```



*For versions V1.60 (8060) y V5.60 (8065), the CNC no longer searches for help files in the following folders.*

```

..\Users\Session\Help\{idioma}
..\Mtb\Sub\Help\{idioma}.
..\Users\Help\{idioma}.

```



*For versions prior to V1.60 (8060) and V5.60 (8065), the CNC first searched for the help files in the manufacturer folders and then in the user folders. For these and later versions, it does so in the opposite order.*

**15.****SUBROUTINES.**

Assistance for subroutines.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 15.7.2 List of available subroutines.

The editor allows having in a text file (txt) a list that is displayed while editing the part-program, every time a #PCALL or an #MCALL instruction is edited.

The editor shows the list of subroutines while editing, after a blank space or tab following a G180-G189 or following a #PCALL or #MCALL instruction. This list works the same way as the lists of variables, it is possible to use the arrow keys to scroll through the various elements. Pressing [ENTER], the editor inserts the selected line in the current position of the cursor. The list of subroutines disappears with [ESC], deleting the key word or going on to another line of the program.

This help is always active, regardless of the editing help softkey, "Prog. help".

### List of subroutines.

The list of subroutines must be in a text (txt) file. The file must be edited so each line is the name of a possible subroutine to be called.

Example of a file with a list of subroutines.

```
C:\CNC8070\USERS\SUB\FAGOR.NC
SUBROUTINE.NC
EXAMPLE.NC
POSITIONING.NC
```

### Name and location of files.

The name of the file should be *pcall.txt*.

### Where to save the list of subroutines.

The machine manufacturer must save the *pcall.txt* file in the folder `..\Mtb\Sub\Help`. Since the modifications to the MTB directory in the "User" work mode disappear when turning the unit off, the user must save his *pcall.txt* file in the folder `..\Users\Sub\Help`.

The CNC looks for the help files in both files; if the files are not there, the CNC will not show any help. If there is a *pcall.txt* file in both directories, the list will show the names of the subroutines contained in both of them.



*For versions V1.60 (8060) y V5.60 (8065), the CNC no longer searches for help files in the following folders.*

```
..\Users\Session\Help\{idioma}
..\Mtb\Sub\Help\{idioma}.
..\Users\Help\{idioma}.
```

## 15.8 Interruption subroutines.

The interruption subroutines are defined by the machine manufacturer and will be executed from the PLC. When the PLC commands the execution of one of these subroutines, the channel interrupts the execution of the program and executes the corresponding interruption subroutine.

If the program is already interrupted (STOP) or no program is in execution (channel in READY state), the execution of the subroutine depends on parameter SUBINTSTOP. Also, in order to execute the subroutine when no program is in execution, the channel must be in automatic mode; the subroutine cannot be executed from jog mode.

The CNC executes the subroutine with the current history of the interrupted program (G functions, feedrate, etc.). Once the subroutine is executed, the CNC resumes the execution of the program from the interruption point and maintaining the changes made to the history by the subroutine (G functions, etc.).

The execution of an interruption subroutine may be interrupted with a STOP, but not by another interruption subroutine. When a subroutine is interrupted, it is not possible to get into tool inspection mode.

**15.****SUBROUTINES.**

Interruption subroutines.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 15.8.1 Repositioning axes and spindles from the subroutine (#REPOS).

The #REPOS can only be used inside the interruption subroutines and allows repositioning axes and spindles before ending that subroutine. The CNC does not reposition the axes when the instruction is executed, it does when returning from the subroutine to the program as the last action associated with the subroutine.

An interruption subroutine may contain several #REPOS instructions, but they all must be programmed at the end of the subroutine, in the blocks before the end of the subroutine (#RET, M17, M29). Blocks programmed between the last #REPOS instruction and the end-of-subroutine block will cause an error.

### Programming.

This instruction must be programmed at the end of the subroutine, before the end-of-subroutine block. When programming this instruction, you must define the axes to be repositioned. Optionally, it is possible to indicate whether the reposition point for the axes is the point where the program was interrupted or the starting point of the interrupted block.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

#REPOS <{point}> X~C <X~C>

{point}            Optional. Reposition point. This parameter is set with one of the INT/INI values.

X~C                Sequence of axes and spindles to be repositioned.

```
#REPOS A1 A2 S1
```

The reposition point is the point where the program was interrupted.

```
#REPOS INT X A1 U Z S
```

The reposition point is the point where the program was interrupted.

```
#REPOS INI X Y Z
```

The reposition point is the starting point of the interrupted block.

#### Sequence of axes and spindles to be repositioned.

The CNC repositions the axes in the program sequence, except the axes of the active plane that reposition simultaneously when the first of them repositions. Since there may be several #REPOS instructions in the same subroutine, axes or spindles repeated (duplicated) in the same instruction or in a previous one are ignored.

#### Reposition point.

This parameter is defined with one of the following commands; if not programmed, the instruction assumes the INT value.

Value.	Meaning.
INT	The reposition point for the axes is the point where the program was interrupted when activating the subroutine.
INI	The reposition point for the axes is the starting point of the interrupted block.

The same subroutine may have several #REPOS instructions, but all of them must have the same INT/INI reposition point.

# 15.

**SUBROUTINES.**

Interruption subroutines.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

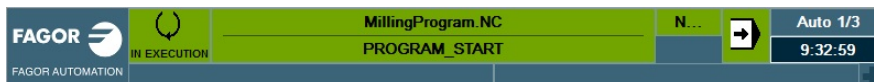
## 15.9 Subroutine associated with the start.

For each channel, a subroutine can be associated with the execution start, which can be executed by pressing the [START] key, in automatic mode, to start the execution of the entire program; that is if no program start point has been selected. The CNC does not call upon the subroutine either when executing a cycle from the conversational mode. This subroutine may be used, for example, to have certain machining conditions defined to determine the execution of user programs.

If the subroutine exists, the CNC will execute it immediately after pressing the [START] key, before starting the program execution. If that subroutine is missing, the CNC executes the program directly.

### Execution of subroutine

During execution, the CNC displays the name of the subroutine on the general status bar. The CNC does not display the blocks under execution, however, it executes the subroutine as a single block, this means that the block-by-block execution is not affected.



### Name and location of the subroutine.

The name of the subroutine must be PROGRAM\_START (without an extension) and it will be saved in the folder ..\Users\Sub. If there are several channels, there may be a different subroutine for each channel, whose name must be PROGRAM\_START\_Cn, where n is the channel number (between 1 and 4).

Name.	Channel.
PROGRAM_START PROGRAM_START_C1	Channel 1. The CNC accepts both names for the subroutine associated with the first channel and where both subroutines exist, the CNC will execute the PROGRAM_START.
PROGRAM_START_C2	Channel 2.
PROGRAM_START_C3	Channel 3.
PROGRAM_START_C4	Channel 4.

15.

SUBROUTINES.

Subroutine associated with the start.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

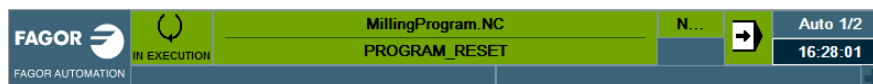
## 15.10 Subroutine associated with the reset.

For each channel, the reset may have an associated subroutine, which will be executed after pressing the [RESET] key on the operator panel or when the PLC activates the RESETIN mark. This subroutine, for example, establishes any initial conditions that are different from those set by the reset or conditioned to the configuration of the machine, activates operations/modes that disable the reset, etc.

If this subroutine exists, the CNC will execute it immediately after the reset. If this subroutine is missing, the CNC executes the reset directly.

### Execution of subroutine

During execution, the CNC displays the name of the subroutine on the general status bar. The CNC does not display the blocks under execution, however, it executes the subroutine as a single block, this means that the block-by-block execution is not affected.



### Name and location of the subroutine.

The name of the subroutine must be PROGRAM\_RESET (without an extension) and it will be saved in the folder ..\Users\Sub. If there are several channels, there may be a different subroutine for each channel, whose name must be PROGRAM\_RESET\_Cn, where n is the channel number (between 1 and 4).

Name.	Channel.
PROGRAM_RESET PROGRAM_RESET_C1	Channel 1. The CNC accepts both names for the subroutine associated with the first channel and where both subroutines exist, the CNC will execute the PROGRAM_RESET.
PROGRAM_RESET_C2	Channel 2.
PROGRAM_RESET_C3	Channel 3.
PROGRAM_RESET_C4	Channel 4.

# 15.

SUBROUTINES.  
Subroutine associated with the reset.

## 15.11 Subroutines associated with the kinematics calibration cycle.

The kinematics calibration cycle has two associated subroutines (KinCal\_Begin.nc and KinCal\_End.nc), which the CNC executes before and after the cycle.

Name.	Meaning.
KinCal_Begin.nc	Subroutine associated with the beginning of the kinematics calibration cycle.
KinCal_End.nc	Subroutine associated with the end of the kinematics calibration cycle.

Fagor supplies both subroutines as incomplete and it is the manufacturer's responsibility to define both subroutines. Software updates do not modify any existing subroutines.

### Name and location of the subroutine.

The names of the subroutines are KinCal\_Begin.nc and KinCal\_End.nc. Both subroutines must be saved in the folder ..\Mtb\Sub. All channels use the same subroutines.

# 15.

**SUBROUTINES.**

Subroutines associated with the kinematics calibration cycle.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

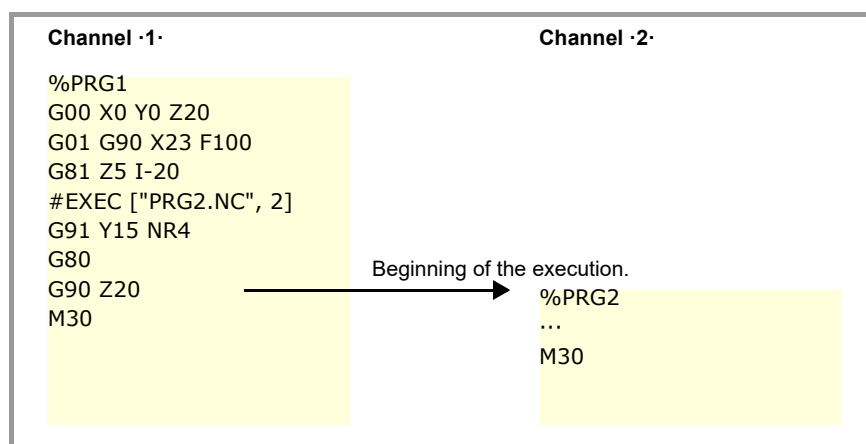


# EXECUTING BLOCKS AND PROGRAMS

# 16

## 16.1 Executing a program in the indicated channel.

With the #EXEC instruction, it is possible, from a program in execution, to begin the execution of a second program in another channel. The execution of the program starts in the indicated channel in parallel (at the same time) with the block following the #EXEC instruction. If the channel where the program to be executed is busy, the CNC waits for the operation in progress to end.



### Programming format.

The programming format is: Optional parameters are indicated between angle brackets.

```
#EXEC [{"prg}"<,{channel}>]
```

{prg} Part-program location.

{channel} Optional. Channel where the block is to be executed.

```
#EXEC ["PRG1.NC",2]
    (It executes in channel 2 the indicated program)
#EXEC ["MYPRG.NC"]
    (It executes the program as a subroutine).
#EXEC ["C:\CNC8070\USERS\PRG\EXAMPLE.NC",3]
    (It executes in channel 3 the indicated program)
```

### Program location (path).

The program to be executed may be defined by either writing the full path or without it. When indicating the whole path, the CNC only looks for the program in the indicated folder. If the path has not been indicated, the CNC looks for the program in the following folders and in this order.

- 1 Directory selected with the #PATH instruction.
- 2 Directory of the program that executes the #EXEC instruction.
- 3 Directory defined by machine parameter SUBPATH.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Channel where the block is to be executed.**

Programming the channel is optional. If the channel is not indicated or it coincides with the channel where the #EXEC instruction is executed, the second program will be executed as a subroutine. In this case, functions M02 and M30 will carry out all the associated actions (initialization, sending to the PLC, etc.) except the one for finishing the program. After executing function M02 or M30, it goes on executing the blocks programmed after the #EXEC instruction.

**Considerations.**

A program containing the #EXEC instruction may be executed, simulated, syntax checked or searched for a particular block. In all the cases, programs called upon using the #EXEC instruction are executed in the same conditions as the original program

**16.****EXECUTING BLOCKS AND PROGRAMS**

Executing a program in the indicated channel.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 16.2 Executing a block in the indicated channel.

With the #EXBLK instruction, it is possible, from a program in execution or via MDI, to execute a block in another channel.

If the channel where the block to be executed is busy, the CNC waits for the operation in progress to end. After executing the block, the channel goes back to the previous work mode.

### Programming format.

The programming format is: Optional parameters are indicated between angle brackets.

#EXBLK [{block}<,{channel}>]

{block}           Block to be executed.

{channel}        Optional. Channel where the block is to be executed.

```
#EXBLK [G01 X100 F550, 2]
      (The block is executed in channel ·2·)
#EXBLK [T1 M6]
      (The block is executed in the current channel)
```

### Channel where the block is to be executed.

Programming the channel is optional. If the channel is not indicated and the instruction is executed from the program, the block is executed in its own channel. If the instruction is executed in MDI and the channel is not indicated, the block is executed in the active channel.

# 16.

**EXECUTING BLOCKS AND PROGRAMS**  
Executing a block in the indicated channel.

# 16.

## EXECUTING BLOCKS AND PROGRAMS

Executing a block in the indicated channel.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

The CNC allows activating axes and spindles as C axis, that interpolated with a linear axis makes it possible to mill the cylindrical surface or the face of a turning part. Although the machine may have several axes or spindle defined as "C" axis, only one of them may be active.

## ·C· axis on a lathe.

On a lathe, it is most common to activate the spindle as ·C· axis and use a live tool to machine the part.

## ·C· axis on a milling machine.

On a milling machine, it is most common to activate a rotary axis as ·C· axis and use the spindle to machine the part.

## Configuration of a ·C· axis.

To activate an axis or spindle as "C" axis, it must have been defined as such by the machine manufacturer. To know whether an axis or spindle may be activated as a ·C· axis, refer to parameter CAXIS in the machine parameter table or to its variable.

(V.)MPA.CAXIS.Xn

Variable indicating whether the axis or spindle can be enabled as the C axis; a value of ·1· is yes and a value ·0· is no.

In the machine parameter table, parameter CAXNAME indicates the default name of the ·C· axis of the channel. This is the name that a spindle enabled as ·C· axis will take if not indicated otherwise via part-program.

## The zero offsets on the C axis.

Once the zero offsets have been defined in the table, they may be activated by program using functions G54 to G59 and G159. The zero offsets on the C axis have the following peculiarities.

- If there is an active zero offset and, then, a C axis is activated, the zero offset for the C axis is not assumed.
- When the spindle works as C axis (#CAX instruction) the zero offset is applied in degrees.
- When face machining is active (#FACE instruction) or cylinder machining (#CYL instruction) is active, the zero offset is applied in the active units, millimeters or inches.

## 17.1 Activating the spindle as "C" axis.

To use a spindle as ·C· axis, it must be enabled as such first. Once this is done, it will be possible to program machining operation on the face or on the side using the instructions #FACE or #CYL respectively.

### Activating the spindle as ·C· axis.

The #CAX instruction activates a spindle as ·C· axis.

The programming format is: Optional parameters are indicated between angle brackets.

#CAX [<{spdl}><, {name}>]

{spdl}            Optional. Spindle to be activated as C axis.

{name}            Optional. Name of the C axis.

```
#CAX
#CAX [S1]
#CAX [S,C]
```

The spindle needs only be indicated when a spindle other than the master is to be activated as a C axis. Otherwise, there is no need to program it.

The {name} parameter sets the name that will identify the C axis. This name will be used in the part program to define the movements. If the name is not defined, the CNC assigns a name by default. See "[Configuration of a ·C· axis.](#)" on page 333.

Programming	Spindle that is activated as ·C· axis.	Axis name.
#CAX	Master spindle.	By default.
#CAX [S1]	Spindle S1 (it may be the master).	By default.
#CAX [S,C]	Spindle S (it may be the master).	C
#CAX [S3,B2]	Spindle S3 (it may be the master).	B2

### Considerations about working with the C axis

Activating a running spindle as C axis stops the spindle. While being a spindle active as "C" axis, no speed may be programmed for it.

When activating the spindle as "C" axis, the CNC carries out a home search of the "C" axis.

### Accessing the variables of a spindle activated as C axis.

After activating a spindle as C axis, the new name of the spindle must be used to access its variables from the part-program or MDI. The access to the variables from the PLCA or from an interface does not change; the original name of the spindle remains unchanged.

### Influence of the M3/M4/M5 functions.

If the spindle is working as C axis, executing an M3, M4 or M5 function will automatically set the spindle in open loop (same as programming #CAX OFF).

### Deactivating the spindle as C axis.

The C axis is deactivated with the #CAX instruction and the spindle goes back to working as a normal spindle.

```
#CAX OFF
```

```
#CAX OFF
```

### Programming the spindle as ·C· axis.

When the spindle works as ·C· axis, it will be programmed as if it were a rotary axis (in degrees).

17.

C AXIS

Activating the spindle as "C" axis.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Programming the master spindle as ·C· axis.**

```
#CAX  
G01 Z50 C100 F100  
G01 X20 C20 A50  
#CAX OFF
```

**Programming any spindle as ·C· axis.**

```
#CAX [S1,C1]  
  (The spindle "S1" is activated as "C" axis under the name of "C1")  
G01 Z50 C1=100 F100  
G01 X20 C1=20 A50 S1000  
#CAX OFF
```

17.

**C AXIS**

Activating the spindle as "C" axis.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 17.2 Machining of the face of the part

Either a rotary axis or a spindle may be used as C axis for this type of machining operation. When using a spindle, it must be activated as "C" axis in advance using the #CAX instruction. See "17.1 Activating the spindle as "C" axis." on page 334.

### Activating the machining of the face of the part.

The #FACE instruction activates the machining of the face and it also defines the work plane. The axis to be activated as "C" axis will be determined by the work plane defined.

The programming format is: Optional parameters are indicated between angle brackets.

```
#FACE [{abs},{ord}<,<{long}>>]<[<{kin}>]>
```

{abs} Abscissa axis of the work plane.  
 {ord} Ordinate axis of the work plane.  
 {long} Optional. Longitudinal axis of the tool.  
 {kin} Optional. Number of the kinematics.

```
#FACE [X,C]
#FACE [X,C][1]
#FACE [X,C,Z]
#FACE [X,C,Z][1]
```

Programming the kinematics is optional; if not programmed, the CNC applies the first kinematics that has been defined in the machine parameters and is valid for this type of machining.

### Cancel the machining of the face of the part.

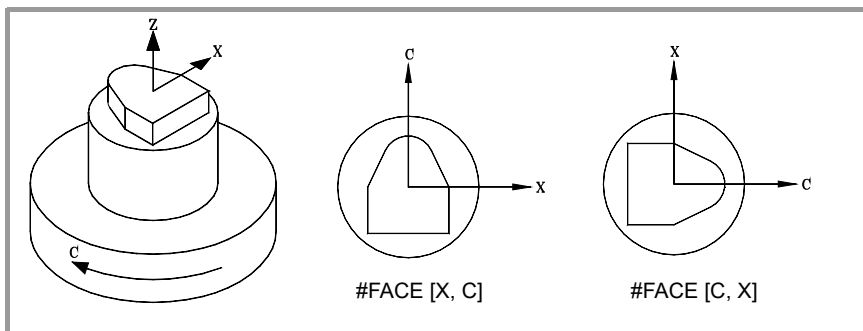
Machining is canceled with the #FACE instruction as follows.

```
#FACE OFF
```

```
#FACE OFF
```

### C axis programming.

The "C" axis will be programmed as if it were a linear axis (in millimeters or inches) and the CNC will calculate the corresponding angular movement depending on the selected radius. When the machining is activated, the CNC switches to operate in radius and G94 (mm/min).



17.

C AXIS

Machining of the face of the part

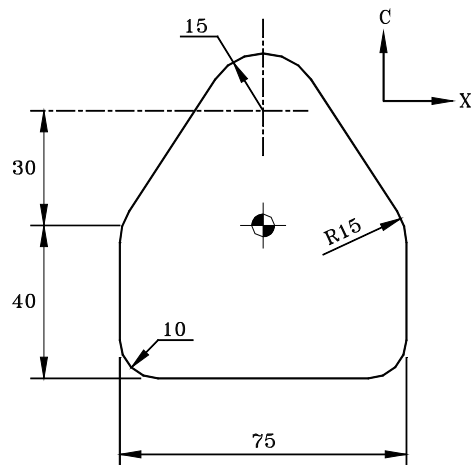


FAGOR AUTOMATION

CNC 8070

(REF: 2008)





```
#FACE [X,C]
G90 X0 C-90
G01 G42 C-40 F600
G37 I10
X37.5
G36 I10
C0
G36 I15
X12.56 C38.2
G03 X-12.58 C38.2 R15
G01 X-37.5 C0
G36 I15
C-40
G36 I10
X0
G38 I10
G40 C-90
#FACE OFF
M30
```

# 17.

**C AXIS**  
Machining of the face of the part

### 17.3 Machining of the turning side of the part

Either a rotary axis or a spindle may be used as C axis for this type of machining operation. When using a spindle, it must be activated as "C" axis in advance using the #CAX instruction. See "17.1 Activating the spindle as "C" axis." on page 334.

#### Activating the machining of the side of the part.

The #CYL instruction activates the machining of the side and it also defines the work plane. The axis to be activated as "C" axis will be determined by the work plane defined.

The programming format is: Optional parameters are indicated between angle brackets.

#CYL [{abs},{ord},{long}]<{radius}>[<{kin}>]

- {abs}           Abscissa axis of the work plane.
- {ord}           Ordinate axis of the work plane.
- {long}          Longitudinal axis of the tool.
- {radius}        Radius of the cylinder that will be machined.
- {kin}           Optional. Number of the kinematics.

```
#CYL [X,C,Z45]
#CYL [C,Y,Z30]
#CYL [X,C,Z45][3]
```

If the radius is programmed with a value of ·0·, it assumes as the cylinder radius the distance between the rotation center and the tool tip. This makes it possible to develop the surface on cylinders with variable radius without having to indicate the radius.



*In versions older than V3.10, programming the radius was optional. If the software is updated from an older version, the programs will have to be corrected.*

Programming the kinematics is optional; if not programmed, the CNC applies the first kinematics that has been defined in the machine parameters and is valid for this type of machining.

#### Cancel the machining of the side of the part.

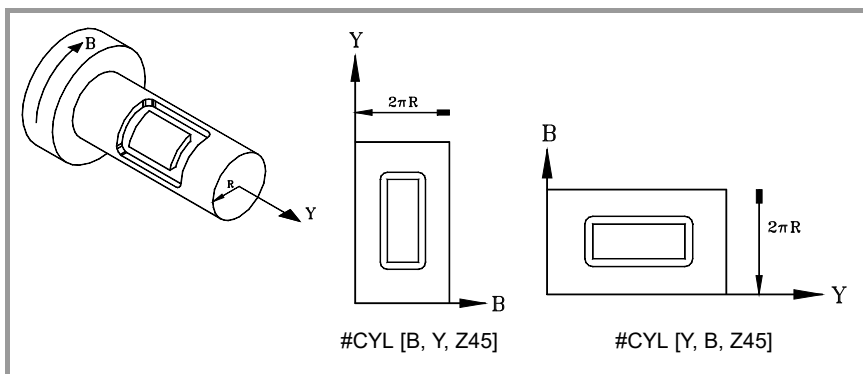
Machining is canceled with the #CYL instruction as follows.

#CYL OFF

```
#CYL OFF
```

#### C axis programming.

The "C" axis will be programmed as if it were a linear axis (in millimeters or inches) and the CNC will calculate the corresponding angular movement depending on the selected radius. When the machining is activated, the CNC switches to operate in radius and G94 (mm/min).



# 17.

## C AXIS

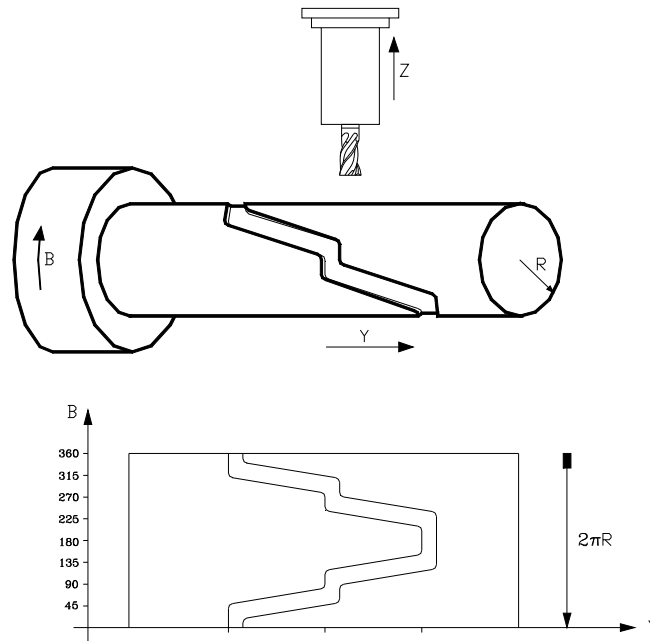
Machining of the turning side of the part



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



```
#CYL [Y,B,Z20]
G90 G42 G01 Y70 B0
G91 Z-4
G90 B15.708
G36 I3
Y130 B31.416
G36 I3
B39.270
G36 I3
Y190 B54.978
G36 I3
B70.686
G36 I3
Y130 B86.394
G36 I3
B94.248
G36 I3
Y70 B109.956
G36 I3
B125.664
G91 Z4
#CYL OFF
M30
```

# 17.

**C AXIS**  
Machining of the turning side of the part

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

# 17.

## C AXIS

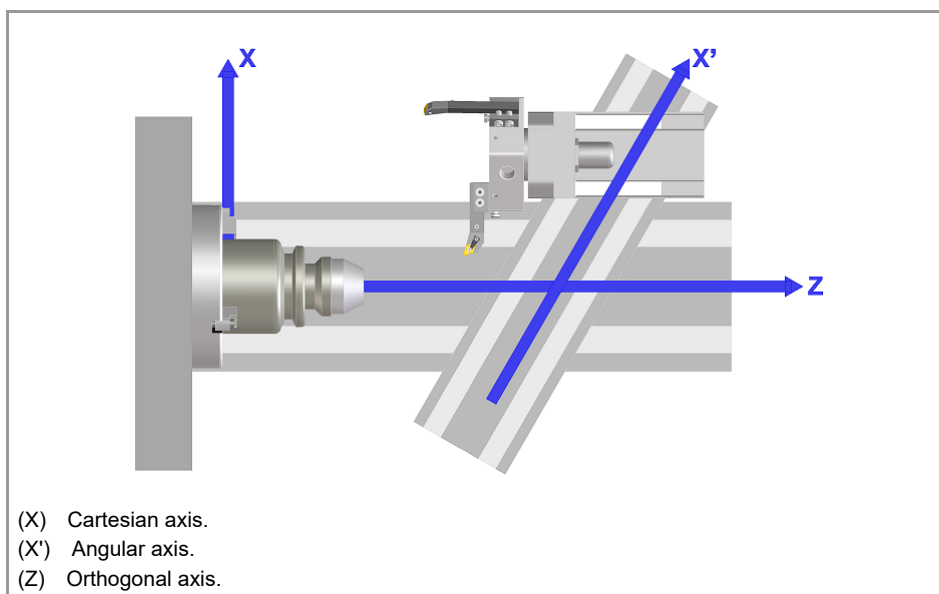
Machining of the turning side of the part

# ANGULAR TRANSFORMATION OF AN INCLINE AXIS.

# 18

With the angular transformation of an incline axis, it is possible to make movements along an axis that is not perpendicular to another. The movements are programmed in the Cartesian system and to make the movements, they are transformed into movements on the real axes.

On certain machines, the axes are configured in a Cartesian way, they are not perpendicular to each other. A typical case is the X axis of a lathe that for sturdiness reasons is not perpendicular to the Z axis.



Programming in the Cartesian system (Z-X) requires activating an angular transformation of an inclined plane that converts the movements of the real (non-perpendicular) axes (Z-X'). This way, a movement programmed on the X axis is transformed into movements on the Z-X' axes; i.e. it then moves along the Z axis and the angular X' axis.

## Turning angular transformation on and off.

The CNC assumes no transformation on power-up; the angular transformations are activated via part-program. Several angular transformations may be active at the same time.

The angular transformations are turned off via part-program. Optionally, a transformation may be "frozen" (suspended) to move angular axis by programming in Cartesian coordinates.

## Influence of the reset, turning the CNC off and of the M30.

The angular transformation of an incline axis is kept active after a RESET or an M30. Turning the CNC off turns the active angular transformation off.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Considerations for the angular transformation of an incline axis.

The axes that make up the angular transformation must meet the following requirements:

- Both axes must belong to the same channel.
- Both axes must be linear.
- Both axes may be masters in a pair of slaved (coupled) axes or gantry axes.

Home search is not possible when the angular transformation is active.

If the angular transformation is active, the coordinates displayed will be those of the Cartesian system. Otherwise, it will display the coordinates of the real axes.

# 18.

ANGULAR TRANSFORMATION OF AN INCLINE AXIS.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 18.1 Turning angular transformation on and off

### Turn angular transformation on.

When the transformation is on, the movements are programmed in the Cartesian system and to make the movements, the CNC transforms them into movements on the real axes. The coordinates displayed on the screen will be those of the Cartesian system.

The angular transformation is turned on with the instruction #ANGAX. This instruction may be used to activate the transformation in one or several axes.

```
#ANGAX ON [1,...,n]
```

1,...,n     Angular transformation to be activated.

At least one transformation must be programmed in the activating instruction, otherwise the relevant error message will be issued. The number of the angular transformation is determined by the order in which they have been defined in the machine parameter table.

```
#ANGAX ON [1]
```

```
#ANGAX ON [5.7]
```

Angular transformations may be activated either all of them at the same time or one by one. Activating one transformation does not cancel the previous ones.

This instruction turns a "frozen" (suspended) transformation on again. See "[18.2 Freezing \(suspending\) the angular transformation.](#)" on page 344.

### Turn angular transformation off.

If the transformation is off, the movements are programmed and executed in the system of the real axes. The coordinates displayed on the screen will be those of the real axes.

The angular transformation is turned on with the instruction #ANGAX. The programming format is: Optional parameters are indicated between angle brackets.

```
#ANGAX OFF <[1,...,n]>
```

1,...,n     Optional. Angular transformation to be activated.

If no transformation has been defined, all the transformations of the channel are turned off.

```
#ANGAX OFF
```

```
#ANGAX OFF [1]
```

```
#ANGAX OFF [5.7]
```

The angular transformation of an incline axis is kept active after a RESET or an M30. Turning the CNC off turns the active angular transformation off.

# 18.

ANGULAR TRANSFORMATION OF AN INCLINE AXIS.  
Turning angular transformation on and off

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 18.2 Freezing (suspending) the angular transformation.

Freezing the angular transformation is a special way to make movements along the angular axis, but programming it in the Cartesian system. The angular transformation cannot be "frozen" (suspended) while jogging.

The angular transformation is "frozen" (suspended) using the instruction `#ANGAX SUSP` whose programming format is.

`#ANGAX SUSP [1,...,n]`

1,...,n Angular transformation to be activated.

If no angular transformation has been programmed, all the transformations of the channel are "frozen" (suspended). The number of the angular transformation is determined by the order in which they have been defined in the machine parameter table.

<code>#ANGAX SUSP</code>	Freezing all the transformations of the channel.
<code>#ANGAX SUSP [1]</code>	Freezing transformation ·1·.
<code>#ANGAX SUSP [5.7]</code>	Freezing transformations ·5· and ·7·.

### Programming movements after "freezing" the angular transformation.

If an angular transformation is "frozen" (suspended), only the coordinate of the angular axis must be programmed in the motion block. If the coordinate of the orthogonal axis is programmed, the movement is carried out according to the normal angular transformation.

### Canceling the freezing of a transformation.

The "freezing" of an angular transformation is canceled after a reset or an M30.

Programming `#ANGAX ON` on the frozen transformation turns it back on.

# 18.

ANGULAR TRANSFORMATION OF AN INCLINE AXIS.  
Freezing (suspending) the angular transformation.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 18.3 Obtaining information on angular transformation.

### Checking the configuration of the angular transformation.

The configuration data of the angular transformation may be checked directly in the machine parameter table or using the following variables.

#### Number of angular transformations defined.

(V.)MPK.NANG

Variable that can only be read from the PRG, PLC and INT.

It returns the number of angular transformations defined in the machine parameter table.

#### Axes that make up the angular transformation.

These variables refer to the angular transformation n. Programming using the curly brackets is mandatory.

(V.)MPK.ANGAXNA[n]

(V.)MPK.ORTGAXNA[n]

Variable that can only be read from the PRG, PLC and INT.

The first one returns the name of the angular axis. The second one returns the name of the orthogonal axis.

#### Geometry of the angular transformation.

These variables refer to the angular transformation n. Programming using the curly brackets is mandatory.

(V.)MPK.ANGANTR[n]

Variable that can only be read from the PRG, PLC and INT.

Angle between the Cartesian angle and the angular axis it is associated with. Positive angle when the angular axis has been rotated clockwise and negative if counterclockwise.

(V.)MPK.OFFANGAX[n]

Variable that can only be read from the PRG, PLC and INT.

Offset of the origin of the angular transformation. Distance between machine zero and the origin of the coordinate system of the incline axis.

### Checking the status of the angular transformation.

#### Status of the angular transformation.

(V.)[n].G.ANGAXST

Variable that can only be read from the PRG, PLC and INT.

It returns the status of angular transformation defined in the channel.

(V.)[n].G.ANGIDST

Variable that can only be read from the PRG, PLC and INT.

It returns the status of angular transformation defined in the [i] position in the machine parameters.

Both variables return the following values:

Value	Meaning
0	The transformation is deactivated.
1	The transformation is activated.
2	The transformation is frozen (suspended).

# 18.

ANGULAR TRANSFORMATION OF AN INCLINE AXIS.  
Obtaining information on angular transformation.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 18.

## ANGULAR TRANSFORMATION OF AN INCLINE AXIS.

Obtaining information on angular transformation.

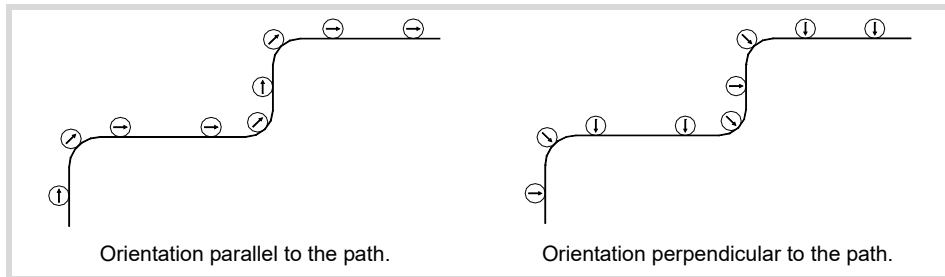


FAGOR AUTOMATION

CNC 8070

(REF: 2008)

"Tangential Control" Keeps a rotary axis always in the same orientation with respect to the programmed path. The machining path is defined in the axes of the active plane and the CNC keeps the orientation of the rotary axis along the whole path.



## Turning tangential control on and off.

The CNC does not activate tangential control on power-up; it is activated via part-program. Tangential control may be active on several axes. Once tangential control is active, the tangential axis cannot be moved in jog or by program; the CNC is in charge of orienting this axis.

Optionally, tangential control can also be "frozen" so it can be re-activated in the same conditions later on.

There are two ways to program tangential control; using either ISO-coded functions or high level commands. Both modes are equivalent and may be combined in the same part-program.

## Influence of the reset, turning the CNC off and of the M30.

Tangential control is modal. Tangential control is canceled on power-up, after executing an M02 or M30 and after an emergency or reset.

## Considerations about tangential control.

Tangential control is compatible with tool radius and length compensation. Mirror image may also be applied while tangential control is active.

### Axes allowed in tangential control.

Tangential control can only be activated on module type rotary axes. An axis of the plane or the longitudinal axis cannot be set as tangential axis. Likewise, a tangential axis can also be a gantry axis, including the gantry axis associated with the rotary axis.

### Tool inspection.

Tool inspection is possible while tangential control is active. When going into tool inspection, the CNC cancels tangential control so the axes can move. After quitting tool inspection, the CNC re-activates tangential control in the same conditions as before.

### Jogging the axes.

The tangential axis cannot be moved while tangential control is active. The axes not affected by tangential control may be moved freely.

When jogging the axes, the CNC cancels tangential control. Once the movement has ended, the CNC re-activates tangential control in the same conditions as before.

**MDI mode.**

The MDI mode may be accessed from jog mode to activate tangential control and move the axes using blocks programmed in MDI. The tangential axis cannot be moved while tangential control is active.

# 19.

## TANGENTIAL CONTROL.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 19.1 Turning tangential control on and off.

There are two ways to manage tangential control; using either ISO-coded functions or high level commands. Both modes are equivalent and may be combined in the same part-program.

### Activating tangential control.

While tangential control is active, the movements are programmed for the axes of the active work plane. It is not possible to program movements of the tangential axis; it is up to the CNC to orient this axis.

Tangential control is activated with function G45 or with the instruction #TANGCTRL. These commands also restore a "frozen" tangential control, but the angle must be programmed again. See "[19.2 Freezing tangential control.](#)" on page 352.

#### Programming format (1).

This function may be used to activate tangential control on one or several axes; it does not allow to set the positioning feedrate of the tangential axis. At least one tangential axis must be defined in this function.

G45 X~C

X~C      Axis upon which tangential control is applied and angular position with respect to the path. The angle is defined in degrees ( $\pm 359.9999$ ).

```
G45 A90
G45 B45 W15.123 B2=-34.5
```

#### Programming format (2).

This statement may be used to activate tangential control on one or several axes and to set the positioning feedrate of the tangential axis. There is no need to activate any axis in order to set the feedrate.

The programming format is: Optional parameters are indicated between angle brackets.

#TANGCTRL ON [<X~C>, <F>]

X~C      Optional. Axis upon which tangential control is applied and angular position with respect to the path. The angle is defined in degrees ( $\pm 359.9999$ ).

F          Optional. Tangential axis orienting feedrate.

Although both parameters are optional, at least one of them must be programmed.

```
#TANGCTRL ON [A34.35]
#TANGCTRL ON [A90, F300]
#TANGCTRL ON [B-45, W15.123, F300]
#TANGCTRL ON [F300]
```

#### Combining both programming formats.

The two programming formats may be combined in the same part-program. For example, the instruction may be used to set the positioning feedrate and function G45 to active tangential control.

```
#TANGCTRL ON [F1000]
G45 W45
```

#### Programming the positioning angle.

The positioning angle is defined in degrees ( $\pm 359.9999$ ). The angle is defined with respect to the path to follow, positive angle for counterclockwise positioning and negative angle for clockwise positioning.

# 19.

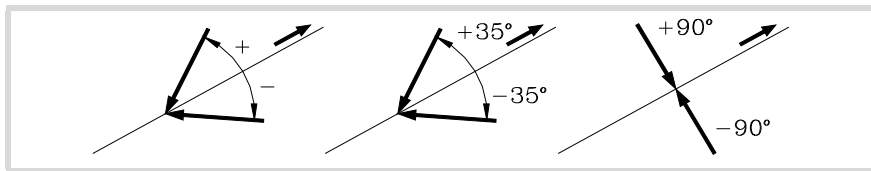
**TANGENTIAL CONTROL.**  
Turning tangential control on and off.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)



The positioning angle is only maintained when tangential control is "frozen"; in the rest of the cases, it must be programmed every time tangential control is activated. See "19.2 Freezing tangential control." on page 352.

**Positioning feedrate for the tangential axis.**

The feedrate for the tangential axes is set with the instruction #TANGCTRL. This feedrate is only applied to the movements of the tangential axes; not to the axes of the plane whose feedrate is F.

```
#TANGCTRL ON [F1000]
```

The tangential feedrate stays active even when tangential control is canceled. This means that the feedrate will be applied the next time tangential control is activated.

If no feedrate has been defined for the tangential axis, it behaves as follows. In any case, the maximum feedrate for each tangential axis will be limited by its machine parameter MAXFEED.

- If the tangential axis must move alone, it moves at the feedrate set by machine parameter MAXFEED.
- If the tangential axis moves with the axes of the plane, its feedrate is the same as the one for those axes.

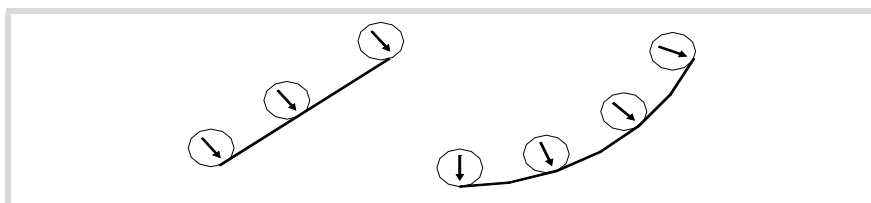
**Operation of the tangential control.**

Every time tangential control is activated, the CNC acts as follows:

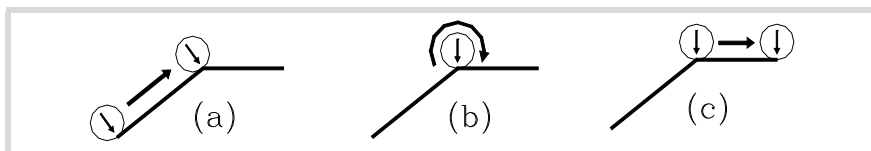
- 1 The CNC orients the tangential axis with respect to the first section and it puts it in the programmed position.



- 2 The interpolation of the plane axes starts once the tangential axis has been positioned. The orientation of the tangential axis is maintained in linear sections whereas in circular interpolations the programmed orientation is maintained throughout the whole travel.



- 3 Blending two sections requires a new orientation of the tangential axis, the CNC ends the section in progress, then orients the tangential axis with respect to the next section and goes on with the execution.



19.

TANGENTIAL CONTROL: Turning tangential control on and off.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Cancel tangential control.

Tangential control is canceled with function G45 or with the instruction #TANGCTRL.

### Programming format (1).

This function cancels tangential control in all the axes of the channel.

G45

G45

### Programming format (2).

This instruction cancels tangential control in one or several axes. If no axis is programmed, it cancels tangential control in all the axes of the channel.

The programming format is: Optional parameters are indicated between angle brackets.

#TANGCTRL OFF <[X~C]>

X~C      Optional. Axis where tangential control is canceled.

```
#TANGCTRL OFF
#TANGCTRL OFF [A]
#TANGCTRL OFF [B, W, V]
```

### Cancellation of tangential control during tool radius compensation.

Tangential control may be canceled even if tool radius compensation is active. However, it is recommended to freeze tangential control instead of canceling it. This is because the instruction #TANGCTRL OFF, besides canceling tangential control, it generates some additional blocks at the beginning and end of the tool compensation.

19.

TANGENTIAL CONTROL.

Turning tangential control on and off.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 19.2 Freezing tangential control.

Freezing tangential control is a special cancellation where the CNC remembers the programmed angle. When restoring tangential control, the CNC orients the axis with the same angle it had when tangential control was frozen. Freezing tangential control does not cancel tool radius compensation.

# 19.

TANGENTIAL CONTROL.  
Freezing tangential control.

### Turning tangential control on.

While tangential control is frozen, the movements are programmed for the axes of the active work plane. No movement of the tangential axis can be programmed.

Tangential control is frozen with function G145 or with the instruction #TANGCTRL.

#### Programming format (1).

This instruction freezes tangential control in one or several axes. If no axis is programmed, it freezes tangential control in all the axes of the channel.

The programming format is: Optional parameters are indicated between angle brackets.

G145 <K0> <X~C>

K0           Optional. Freeze tangential control.

X~C          Optional. Axis where tangential control is frozen.

Parameter *K* may take the values ·0· and ·1·. When set to ·1·, it means that a previously frozen tangential axis is to be resumed. If the parameter *K* is not programmed, the CNC assumes *K*0.

```
G145 K0
G145 K0 A
G145 K0 B W C
G145 B A
```

#### Programming format (2).

This statement freezes tangential control in one or several axes. If no axis is programmed, it freezes tangential control in all the axes of the channel.

The programming format is: Optional parameters are indicated between angle brackets.

#TANGCTRL SUSP <[X~C]>

X~C          Optional. Axis where tangential control is frozen.

```
#TANGCTRL SUSP
#TANGCTRL SUSP [A]
#TANGCTRL SUSP [B, W]
```

### Canceling tangential control.

Tangential control is resumed with function G145 or with the instruction #TANGCTRL.

#### Programming format (1).

This function resumes tangential control in one or several axes. If no axis is programmed, it resumes tangential control in all the axes of the channel.

The programming format is: Optional parameters are indicated between angle brackets.

G145 K1 <X~C>

K1           Restoring tangential control.

X~C          Optional. Axis where tangential control is resumed.

Parameter *K* may take the values ·0· and ·1·. When set to ·0·, it means that tangential control is to be frozen.

```
G145 K1
G145 K1 A
G145 K1 B W C
```



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### Programming format (2).

This instruction resumes tangential control in one or several axes. If no axis is programmed, it resumes tangential control in all the axes of the channel.

The programming format is: Optional parameters are indicated between angle brackets.

#TANGCTRL RESUME <[X~C]>

X~C      Optional. Axis where tangential control is resumed.

```
#TANGCTRL RESUME  
#TANGCTRL RESUME [A]  
#TANGCTRL RESUME [B, W, C]
```

19.

**TANGENTIAL CONTROL.**  
Freezing tangential control.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 19.3 Obtaining information on tangential control.

# 19.

### Checking the configuration of the angular transformation.

The configuration data of the tangential control may be checked directly in the machine parameter table or using the following variables.

#### Is it a module type of rotary axis?

(V.)[n].MPA.AXISMODE.Xn

The variable indicates the type of rotary axis; if it is a module type, the variable must return the value of ·0·.

### Consulting the tangential control data.

(V.)A.TANGAN.Xn

This variable returns the angle programmed in the Xn axis.

(V.)G.TANGFEED

This variable returns the positioning feedrate programmed for tangential control.

### Consulting the tangential control status.

(V.)PLC.TANGACTIVCn

This variable indicates whether tangential control is active in channel n nor not. A ·1· if the tangential control is active and a ·0· if otherwise.

(V.)PLC.TANGACTx

This variable indicates whether tangential control is active in the x axis or not. A ·1· if the tangential control is active and a ·0· if otherwise.

(V.)[n].G.TGCTRLST

It returns the status of tangential control in the channel. A value of ·0· if tangential control is off, a ·1· if it is on and a ·2· if frozen (suspended).

(V.)[n].A.TGCTRLST.Xn

It returns the status of tangential control in the axis. A value of ·0· if tangential control is off, a ·1· if it is on and a ·2· if frozen (suspended).

### Initialization of the variables.

Canceling tangential control initializes all the variables except (V.)A.TANGFEED, because the programmed feedrate is maintained for a possible future tangential control.

When freezing (suspending) tangential control, the variables act as follows.

(V.)A.TANGAN.Xn	It maintains the value of the programmed angle.
(V.)G.TANGFEED	It is not initialized.
(V.)PLC.TANGACTIVCn	It is not initialized.
(V.)PLC.TANGACTx	It is initialized.

The description of the general coordinate transformation is divided into these basic functions:

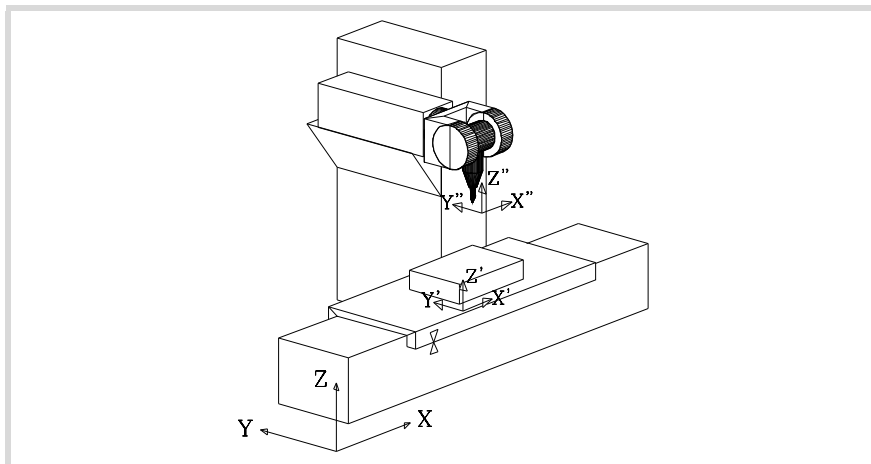
Instruction.	Meaning.
#KIN ID.	Select a kinematics.
#CS.	Define a machining coordinate system (Inclined plane).
#ACS.	Define a fixture coordinate system.
#RTCP.	RTCP (Rotating Tool Center Point) transformation.
#TLC.	Correct the implicit tool length compensation of the program.
#CSROT ON	Activate tool orientation in the part coordinate system.
#CSROT OFF	Cancel tool orientation in the part coordinate system and, therefore, activate tool orientation in the machine coordinate system.
#DEFROT	How to manage the discontinuities in the orientation of rotary axes.
#SELECT ORI	Select onto which rotary axes of the kinematics the tool orientation is calculated for a given direction on the work piece (part).
#KINORG	Transform the current part zero considering the position of the table kinematics.
#TOOL ORI	Tool perpendicular to the inclined plane.

## 20.1 Coordinate systems.

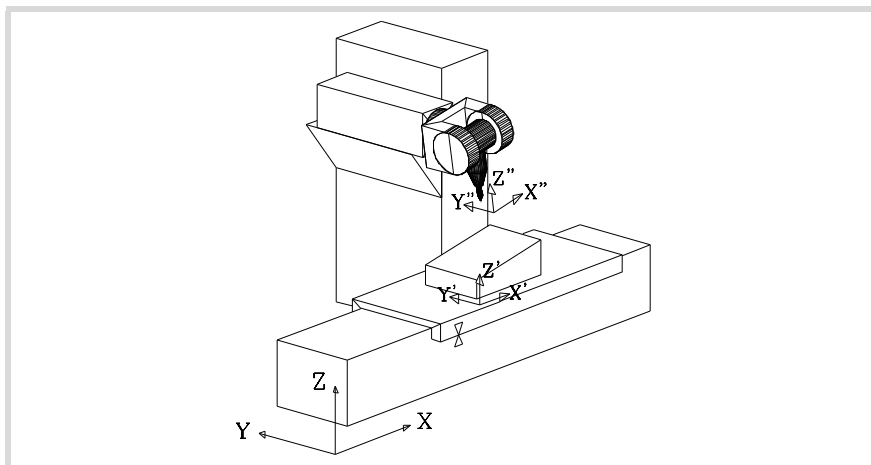
For clarity's sake, the following examples show three coordinate systems:

- X Y Z            Machine coordinate system.
- X' Y' Z'        Part coordinate system.
- X'' Y'' Z''      Tool coordinate system.

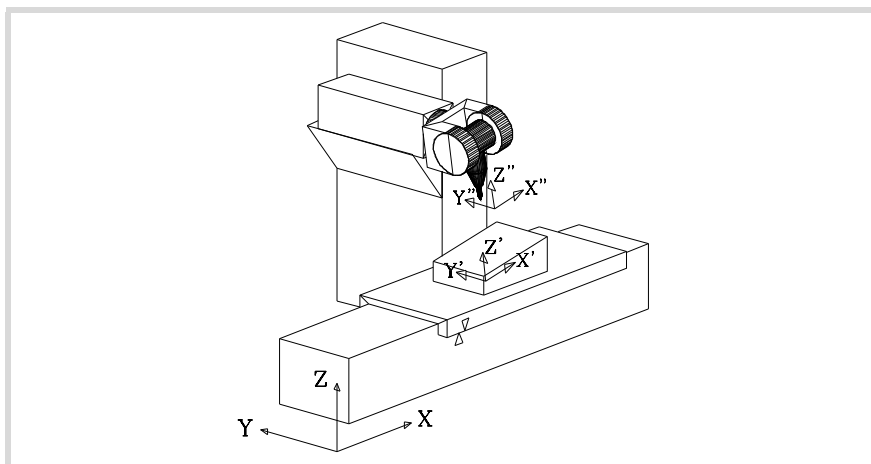
When no transformation has been made and the spindle is in the starting position, the three coordinate systems coincide.



When turning the spindle, the tool coordinate system (X'' Y'' Z'') changes.



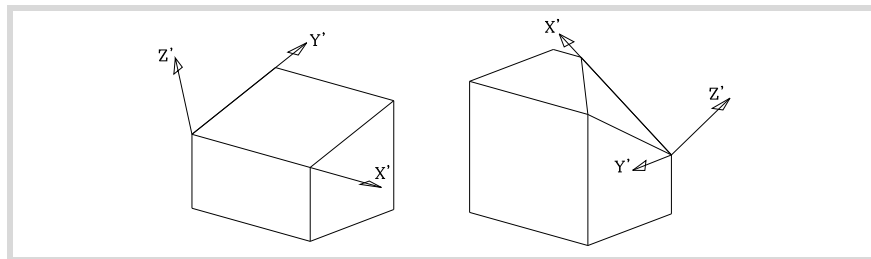
If besides this, a new machining coordinate system is selected (#CS instruction) or fixture coordinate system (#ACS instruction) the part coordinate system will also change (X' Y' Z').



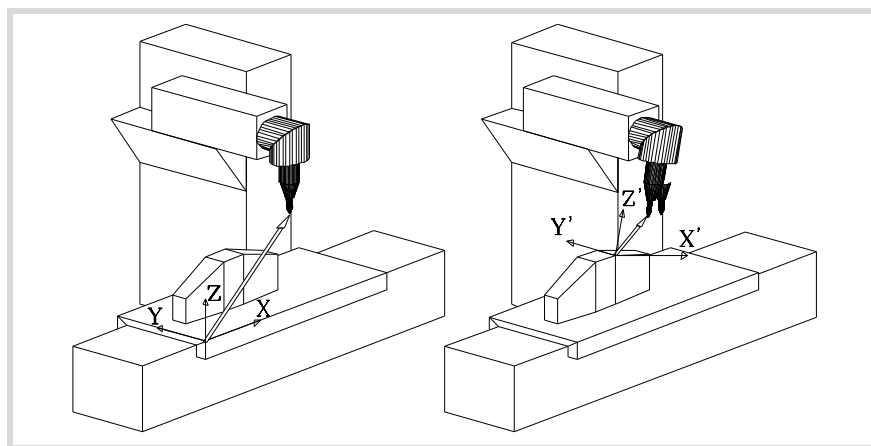
# 20.

## 20.2 Movement in an inclined plane.

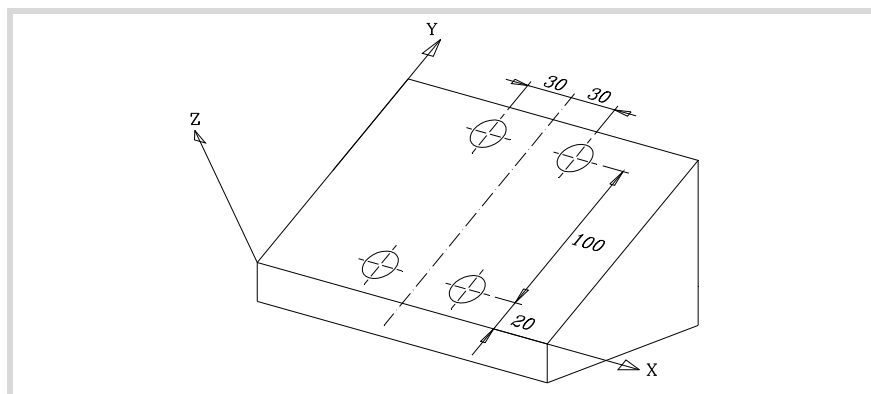
An inclined plane is any plane resulting from a coordinate transformation of the first three axes of the channel (XYZ in the following examples). Any plane in space may be selected to carry out machining operations in it. To define the inclined plane corresponding to the machining operations, use the #CS and #ACS instructions that are described later on in this chapter.



To orient the tool perpendicular to the inclined plane, use the #TOOL ORI instruction or the kinetics related variables that indicate the position that each rotary axis of the spindle head must occupy. The new coordinates (right figure) are referred to the new part zero assuming that the tool is positioned perpendicular to the new plane.



From this moment on, the programming and the X, Y movements are carried out along the selected plane and those of the Z axis will be perpendicular to it.



20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Movement in an inclined plane.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.3 Select a kinematics (#KIN ID).

The manufacturer can set up to 6 kinematics for the machine each one indicating the type of spindle or table, its characteristics and dimensions. Usually, the OEM defines the kinematics number being used by default by means of general machine parameter KINID.

To work with coordinate transformation, the kinematics being used must be indicated. If several kinematics have been defined, it is possible to activate the desired one via part-program using the instruction #KIN ID. If there is only one and it has been set as the default kinematics, there is no need to program this instruction.

### Programming.

When defining this instruction, define the number of the kinematics to activate out of the possible 6.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets.

```
#KIN ID [{kin}]
```

{kin} Kinematics number (between 0 and 6). A value of 0 deactivates the kinematics.

```
#KIN ID [2]
(Activate kinematics number 2)
```

### Deactivating the kinematics.

Programming the #KIN ID instruction with a 0 value deactivates the active kinematics.

```
#KIN ID [0]
(Deactivate active kinematics)
```

### Considerations.

- Functions #RTCP, #TLC and #TOOL ORI must always be activated after selecting a kinematics.

```
N50 #KIN ID [2]
(Activate kinematics number 2)
N60 #RTCP ON
(Activate RTCP transformation with kinematics 2)
.
.
N70 #RTCP OFF
(Turn RTCP off)
N80 M30
```

- The kinematics cannot be changed while function #RTCP or #TLC is active.

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Select a kinematics (#KIN ID).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.4 Coordinate systems (#CS / #ACS).

There are two different types of coordinate systems, namely the machining coordinate system and the fixture coordinate system. Each one is handled with its associated instruction.

#CS	With the #CS instruction, up to 5 machining coordinate systems may be defined, stored, activated and deactivated.
#ACS	With the #ACS instruction, up to 5 fixture coordinate systems may be defined, stored, activated and deactivated. This system is used to compensate for workpiece inclination due to the fixtures used to secure them.

Both instructions use the same programming format and may be used independently or combined as indicated in the following sections.

Several #ACS and #CS coordinate systems may be combined. When activating a new one, it is added to the current coordinate system. See "[20.4.8 How to combine several coordinate systems.](#)" on page 372.

It is recommended to start the program with #CS NEW or #ACS NEW to avoid undesired planes. This happens, for example, after interrupting the program and resuming execution.

### Programming.

Both instructions (#CS and #ACS) use the same programming format. There are various programming formats depending on the operations to carry out with the coordinate systems; define, activate, save, cancel and delete.



*In EDISIMU mode, the user can access an editor that makes programming inclined planes easier using the instructions #CS and #ACS. See the operating manual for further information on the editor of inclined planes.*

### Programming format.

The general programming format is the following; later on in this chapter, there is a more detailed description of the different formats for each operation. See "[Detailed programming.](#)" on page 360.

#CS <DEF/ON/NEW/OFF> <ALL> <ACT> [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]

#ACS <DEF/ON/NEW/OFF> <ALL> <ACT> [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND> <SOL2>]

DEF/ON/ NEW/OFF	Operation to be carried out. DEF: Define and save a coordinate system. ON: Define, save and activate a coordinate system. NEW: Cancel and delete all coordinate systems. OFF: Cancel a coordinate system.
ACT	Together with the DEF command, it assumes and saves the current coordinate system.
ALL	Together with the ALL command, cancels all coordinate systems.
{nb}	Number of the coordinate system (from 1 to 5).
MODE {mode}	Definition mode (from 1 to 6).
{V1}...{V3}	Components of the translation vector.
{φ1}...{φ3}	Rotation angles.
{align}	Optional. Plane alignment (value 0/1). Only in modes 3, 4 and 5.
KEEP	Optional. Command to maintain the part zero when canceling the coordinate system.
FIRST/SECOND	Optional. Command to define the orientation of the axes. Only in mode 6.
<SOL2>	Optional. On Huron type spindles, use the second solution to orient the spindle; if not programmed, use the first one.

**Definition mode.**

The MODE definition mode sets the order in which the axes rotate to reach the desired plane. In some cases, the resolution of the plane presents two solutions; the selection is carried out defining which axis of the coordinate system stays aligned with the plane.

**Maintain the part zero when canceling the coordinate system.**

Turning a transformation off, if not defined otherwise, restores the part zero that was defined before activating the inclined plane. To maintain the current part zero, the one defined with the coordinate system, program the command KEEP. This command is only admitted in instructions that deactivate a coordinate system.

**45° spindles (Huron type).**

Huron type spindles have two solutions for orienting the tool perpendicular to the new work plane. For this type of spindles, it is possible to select which one of the two solutions is to be applied (SOL2 command). See *"20.4.7 Operation with 45° spindles (Huron type)."* on page 370.

**Detailed programing.**

Here are all the possible programming formats; the list of arguments appears between curly brackets and the optional ones between angle brackets.

**Format to define and save (without activating) a coordinate system.**

If the coordinate system has already been defined earlier, these instructions will redefine it.

```
#CS DEF [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>,
<FIRST/SECOND>]
#ACS DEF [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>,
<FIRST/SECOND>]
```

{nb}                    Number of the coordinate system (from 1 to 5).  
MODE {mode}        Definition mode (from 1 to 6).  
{V1}...{V3}        Components of the translation vector.  
{φ1}...{φ3}        Rotation angles.  
{align}             Optional. Plane alignment (value 0/1). Only in modes 3, 4 and 5.  
FIRST/SECOND    Optional. Command to define the orientation of the axes. Only in mode 6.

```
#CS DEF [2] [MODE 1,0,15,5,30,15,4.5]
(Defines and saves a new coordinate system as CS2)
#CS DEF [3] [MODE 3,0,15,5,30,15,4.5,1]
(Defines and saves a new coordinate system as CS3)
#CS DEF [4] [MODE 6,20,105,50,30,FIRST]
(Defines and saves a new coordinate system as CS4)
```

**Format to define, save and activate a coordinate system.**

```
#CS ON [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>,
<FIRST/SECOND>]
#ACS ON [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>,
<FIRST/SECOND>]
```

{nb}                    Number of the coordinate system (from 1 to 5).  
MODE {mode}        Definition mode (from 1 to 6).  
{V1}...{V3}        Components of the translation vector.  
{φ1}...{φ3}        Rotation angles.  
{align}             Optional. Plane alignment (value 0/1). Only in modes 3, 4 and 5.  
FIRST/SECOND    Optional. Command to define the orientation of the axes. Only in mode 6.

```
#CS ON [2] [MODE 1,0,15,5,30,15,4.5]
(Defines, saves and activates a new coordinate system as CS2)
```

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Coordinate systems (#CS / #ACS).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### Format to define and activate (without saving) a coordinate system.

Only one of them may be defined; to define another one, the previous one must be canceled. The coordinate system may be used, until canceled, as any other coordinate system saved in memory.

```
#CS ON [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]
```

```
#ACS ON [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]
```

MODE {mode}	Definition mode (from 1 to 6).
{V1}...{V3}	Components of the translation vector.
{φ1}...{φ3}	Rotation angles.
{align}	Optional. Plane alignment (value 0/1). Only in modes 3, 4 and 5.
FIRST/SECOND	Optional. Command to define the orientation of the axes. Only in mode 6.

```
#CS ON [MODE 1,0,15,5,30,15,4.5]
(Defines and activates a new coordinate system)
```

### Format to deactivate and delete all current coordinate systems and define, save and activate a new one.

```
#CS NEW <KEEP> [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]
```

```
#ACS NEW <KEEP> [{nb}] [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]
```

{nb}	Number of the coordinate system (from 1 to 5).
MODE {mode}	Definition mode (from 1 to 6).
{V1}...{V3}	Components of the translation vector.
{φ1}...{φ3}	Rotation angles.
{align}	Optional. Plane alignment (value 0/1). Only in modes 3, 4 and 5.
KEEP	Optional. Command to define whether the part zero defined in the coordinate system will be maintained (kept) or not when canceling the coordinate system.
FIRST/SECOND	Optional. Command to define the orientation of the axes. Only in mode 6.

```
#CS NEW KEEP [2] [MODE 1,0,15,5,30,15,4.5]
(Cancels and deletes all coordinate systems)
(Defines, saves and activates a new coordinate system as CS2)
(Keeps the part zero)
```

### Format to cancel and delete all current coordinate systems and define and activate a new one (without saving).

```
#CS NEW <KEEP> [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]
```

```
#ACS NEW <KEEP> [MODE {mode}, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>, <FIRST/SECOND>]
```

MODE {mode}	Definition mode (from 1 to 6).
{V1}...{V3}	Components of the translation vector.
{φ1}...{φ3}	Rotation angles.
{align}	Optional. Plane alignment (value 0/1). Only in modes 3, 4 and 5.
KEEP	Optional. Command to define whether the part zero defined in the coordinate system will be maintained (kept) or not when canceling the coordinate system.
FIRST/SECOND	Optional. Command to define the orientation of the axes. Only in mode 6.

```
#CS NEW [2] [MODE 1,0,15,5,30,15,4.5]
(Cancels and deletes all coordinate systems)
(Defines and activates a new coordinate system)
```

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Coordinate systems (#CS / #ACS).

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Format to assume and save the current coordinate system.**

#CS DEF ACT [{nb}]

#ACS DEF ACT [{nb}]

{nb}                    Number of the coordinate system (from 1 to 5).

#CS DEF ACT [2]

(Assumes and saves a new coordinate system as CS2)

**Format to activate a saved coordinate system.**

#CS ON [{nb}]

#ACS ON [{nb}]

{nb}                    Number of the coordinate system (from 1 to 5).

#CS ON [2]

(Activates the CS2 coordinate system)

**Format to activate the coordinate system saved last.**

#CS ON

#ACS ON

#CS ON

(Activates the coordinate system saved last)

**Format to cancel the coordinate system activated last.**

#CS OFF &lt;KEEP&gt;

#ACS OFF &lt;KEEP&gt;

KEEP

Optional. Command to define whether the part zero defined in the coordinate system will be maintained (kept) or not when canceling the coordinate system.

#CS OFF

(Cancels the coordinate system activated last)

#CS OFF KEEP

(Cancels the coordinate system activated last)

(Keeps the part zero)

**Format to cancel all the active coordinate systems.**

#CS OFF ALL

#ACS OFF ALL

#CS OFF ALL

(Cancels all active coordinate systems)

**The coordinate systems and part zero.**

The origin of the coordinate system is referred to the current part zero. While a CS or ACS coordinate system is active, new part zeros may be preset in the inclined plane.

Deactivating an inclined plane, if not defined otherwise, restores the part zero that was defined before activating the inclined plane. Optionally, it is possible to define whether to keep or not the current part zero.

Sometimes, it could happen that when activating a CS or a ACS coordinate system saved earlier, the coordinate origin of the plane may not be desired one. This happens if the part zero is modified between the definition and the application of the coordinate system.

**Considerations for the two functions.**

Both coordinate systems, (#CS and #ACS) are kept active after a Reset or after executing an M02 or M30. On power-up, the CNC either keeps or cancels the coordinate system according to the setting of machine parameter CSCANCEL.

20.

### 20.4.1 Define a coordinate system (MODE1).

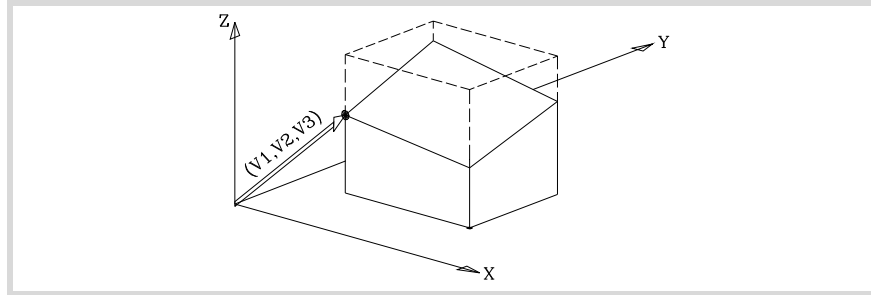
Both instructions use the same programming format and may be used together or separately.

```
#CS DEF [{n}] [MODE 1, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}]
#ACS DEF [{n}] [MODE 1, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}]
```

This mode defines an inclined plane as a result from rotating the amounts indicated in  $\phi_1$ ,  $\phi_2$ ,  $\phi_3$  first around the first axis, then around the second axis and finally around the third axis respectively.

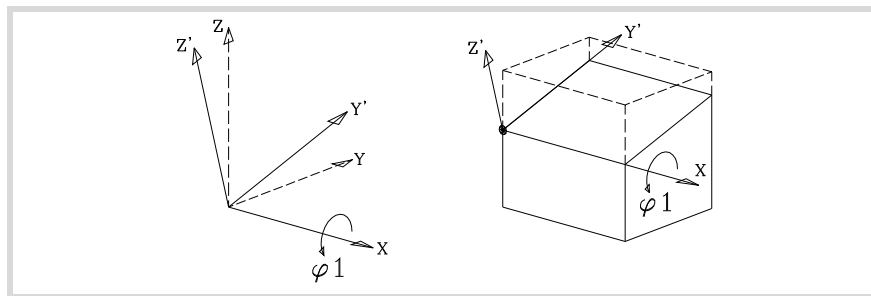
#### V1, V2, V3 Components of the translation vector.

Coordinate origin of the inclined plane with respect to the current part zero.

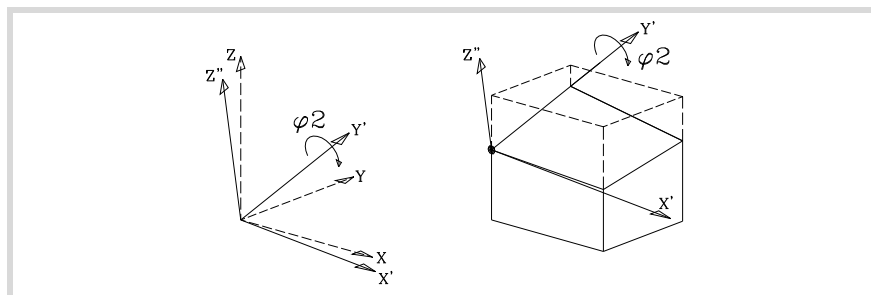


#### $\phi_1, \phi_2, \phi_3$ Rotation angles to build the inclined plane.

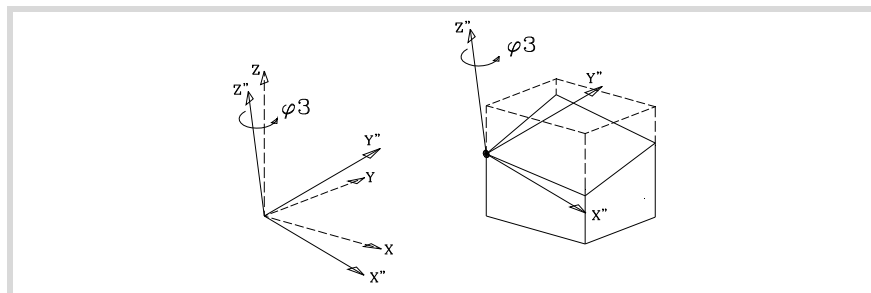
First, rotate around the (X) axis the amount indicated by  $\phi_1$ . In the figure, the new coordinate system resulting from this transformation is called X' Y' Z' because the Y, Z axes have been rotated.



Then, rotate around the 2nd axis (Y'), the amount indicated by  $\phi_2$ . In the figure, the new coordinate system resulting from this transformation is called X' Y' Z'' because the X, Z axes have been rotated.



And last, rotate around the third axis (Z''), the amount indicated by  $\phi_3$ .



### 20.4.2 Define a coordinate system (MODE2).

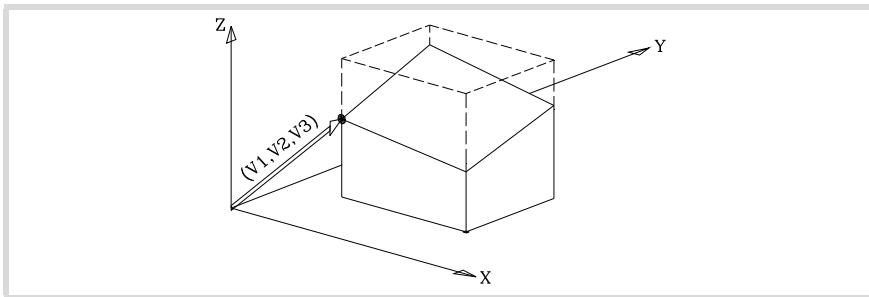
Both instructions use the same programming format and may be used together or separately.

```
#CS DEF [{n}] [MODE 2, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}]
#ACS DEF [{n}] [MODE 2, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}]
```

This mode defines, in spherical coordinates, an inclined plane as a result from rotating the amounts indicated in  $\varphi_1$ ,  $\varphi_2$ ,  $\varphi_3$  respectively first around the third axis, then around the second axis and then again around the third axis respectively.

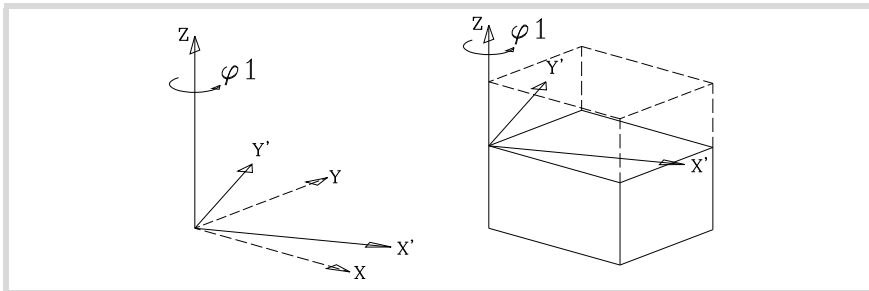
#### V1, V2, V3 Components of the translation vector.

Coordinate origin of the inclined plane with respect to the current part zero.

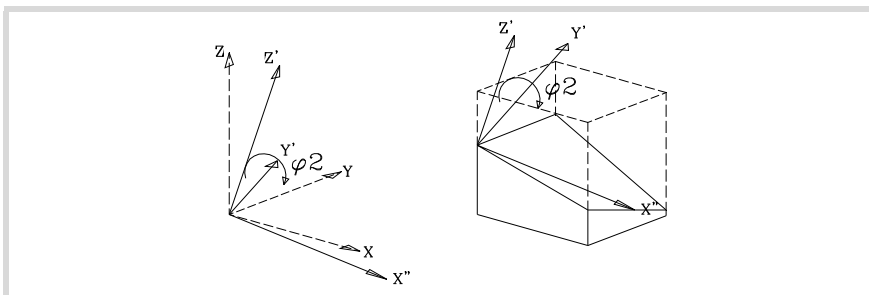


#### $\varphi_1$ , $\varphi_2$ , $\varphi_3$ Rotation angles to build the inclined plane.

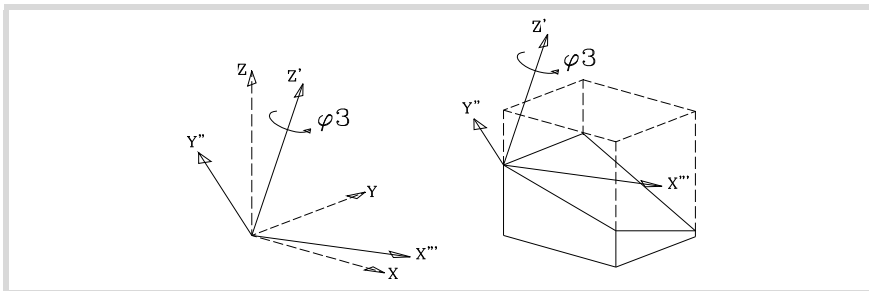
First rotate around the third axis (Z), the amount indicated by  $\varphi_1$ . In the figure, the new coordinate system resulting from this transformation is called X' Y' Z because the X, Y axes have been rotated.



Then, rotate around the 2nd axis (Y'), the amount indicated by  $\varphi_2$ . In the figure, the new coordinate system resulting from this transformation is called X'' Y' Z' because the X, Z axes have been rotated.



And last, rotate around the third axis (Z'), the amount indicated by  $\varphi_3$ .



**20.**  
**KINEMATICS AND COORDINATE TRANSFORMATION**  
 Coordinate systems (#CS / #ACS).

### 20.4.3 Define a coordinate system (MODE3).

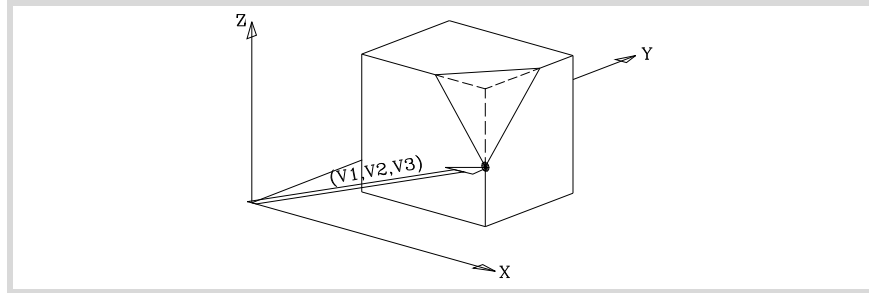
Both instructions use the same programming format and may be used together or separately.

```
#CS DEF [{n}] [MODE 3, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>]
#ACS DEF [{n}] [MODE 3, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>]
```

In this mode, the inclined plane is defined by the angles formed by the plane with respect to the first and second axes (X Y) of the machine coordinate system.

#### V1, V2, V3 Components of the translation vector.

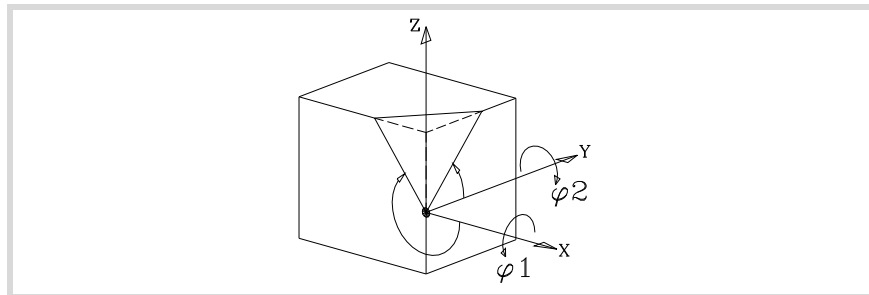
Coordinate origin of the inclined plane with respect to the current part zero.



#### φ1, φ2 Angles of the plane.

Angles that the inclined plane forms with the 1st and 2nd axes (X Y) of the machine coordinate system.

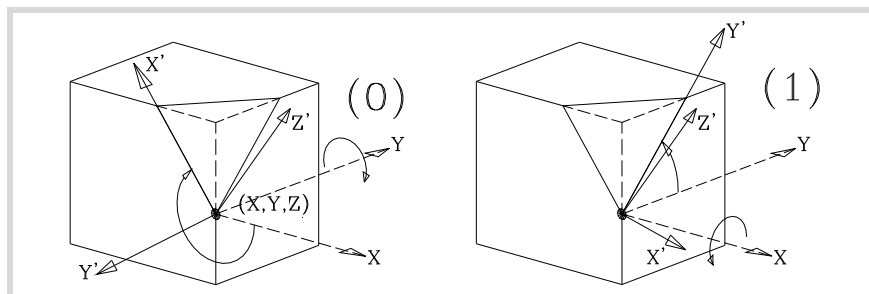
- φ1 Angle of the plane with the first axis (X).
- φ2 Angle of the plane with the second axis (Y).



#### {align} Plane alignment (value 0/1).

This argument defines which of the axes of the new plane (X' Y') is aligned with the edge. If not programmed, it assumes 0.

- {align} = 0 Alignment of the X' axis.
- {align} = 1 Alignment of the Y' axis.



#### φ3 Coordinate (pattern) rotation.

This argument permits defining and applying a coordinate rotation in the new Cartesian plane X' Y'.

### 20.4.4 Define a coordinate system (MODE4).

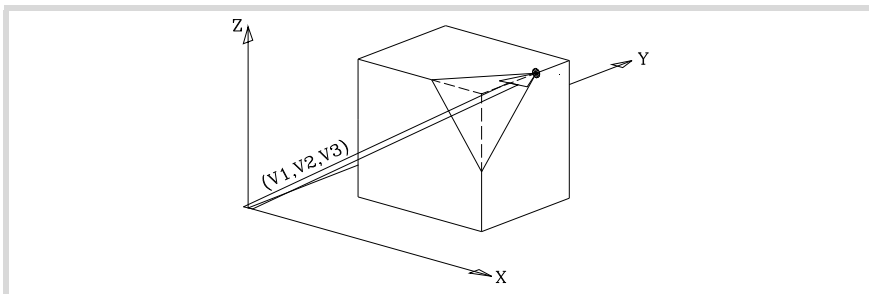
Both instructions use the same programming format and may be used together or separately.

```
#CS DEF [{n}] [MODE 4, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>]
#ACS DEF [{n}] [MODE 4, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>]
```

In this mode, the inclined plane is defined by the angles formed by the plane with respect to the first and third axes (X Z) of the machine coordinate system.

#### V1, V2, V3 Components of the translation vector.

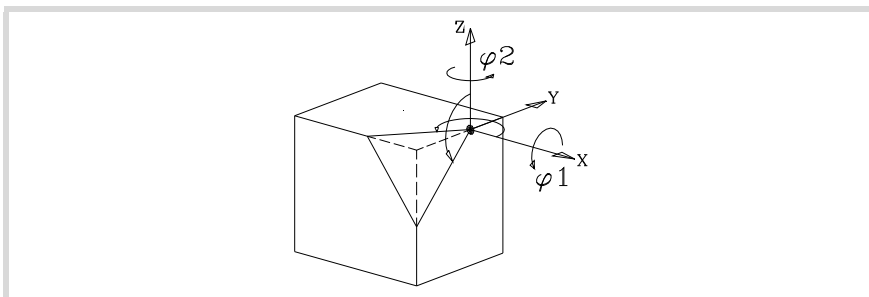
Coordinate origin of the inclined plane with respect to the current part zero.



#### φ1, φ2 Angles of the plane.

Angles that the inclined plane forms with the 1st and 3rd axes (X Z) of the machine coordinate system.

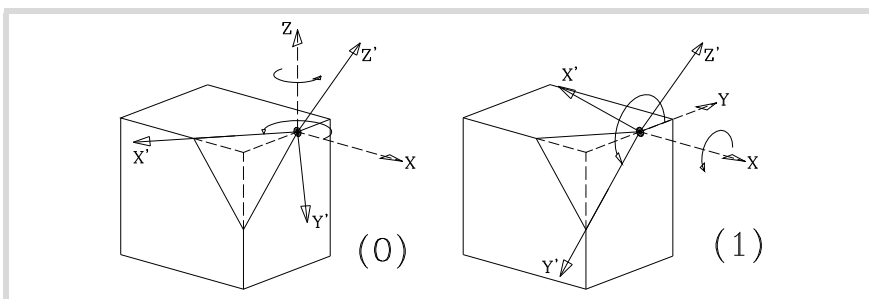
- φ1 Angle of the plane with the first axis (X).
- φ2 Angle of the plane with the third axis (Z).



#### {align} Plane alignment (value 0/1).

This argument defines which of the axes of the new plane (X' Y') is aligned with the edge. If not programmed, it assumes 0.

- {align} = 0 Alignment of the X' axis.
- {align} = 1 Alignment of the Y' axis.



#### φ3 Coordinate (pattern) rotation.

This argument permits defining and applying a coordinate rotation in the new Cartesian plane X' Y'.

### 20.4.5 Define a coordinate system (MODE5).

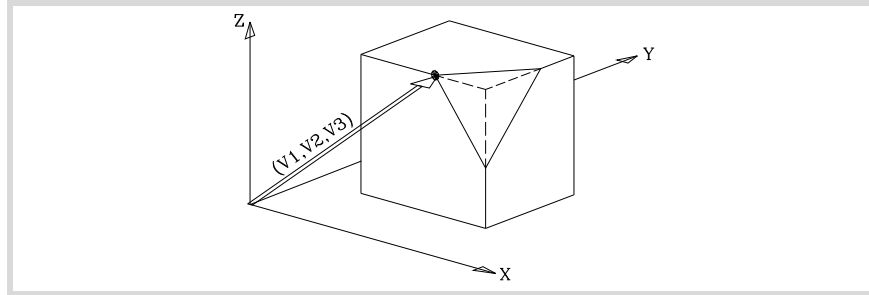
Both instructions use the same programming format and may be used together or separately.

```
#CS DEF [{n}] [MODE 5, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>]
#ACS DEF [{n}] [MODE 5, {V1}, {V2}, {V3}, {φ1}, {φ2}, {φ3}, <{align}>]
```

In this mode, the inclined plane is defined by the angles formed by the plane with respect to the second and third axes (Y Z) of the machine coordinate system.

#### V1, V2, V3 Components of the translation vector.

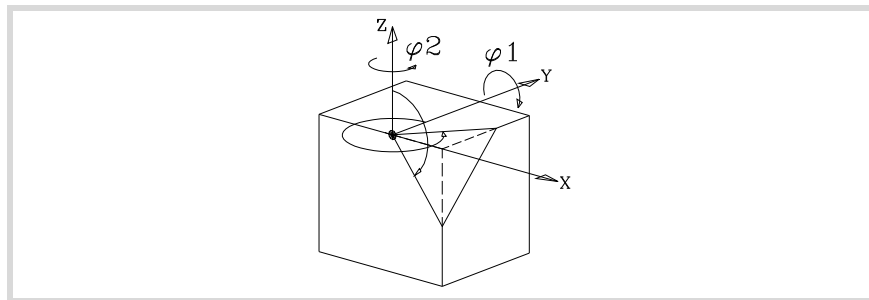
Coordinate origin of the inclined plane with respect to the current part zero.



#### φ1, φ2 Angles of the plane.

Angles that the inclined plane forms with the 2nd and 3rd axes (Y Z) of the machine coordinate system.

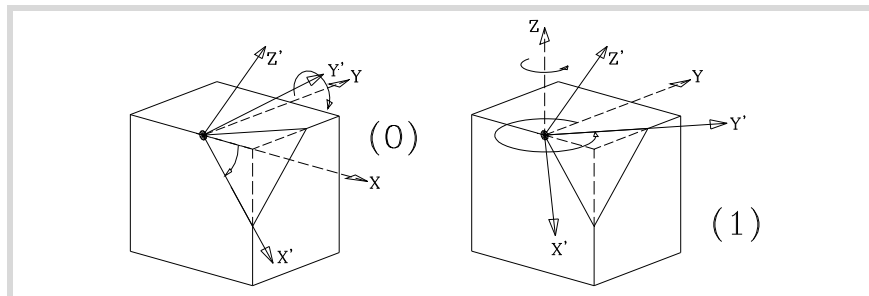
- φ1 Angle of the plane with the second axis (Y).
- φ2 Angle of the plane with the second axis (Y).



#### {align} Plane alignment (value 0/1).

This argument defines which of the axes of the new plane (X' Y') is aligned with the edge. If not programmed, it assumes 0.

- {align} = 0 Alignment of the X' axis.
- {align} = 1 Alignment of the Y' axis.



#### φ3 Coordinate (pattern) rotation.

This argument permits defining and applying a coordinate rotation in the new Cartesian plane X' Y'.

### 20.4.6 Define a coordinate system (MODE6).

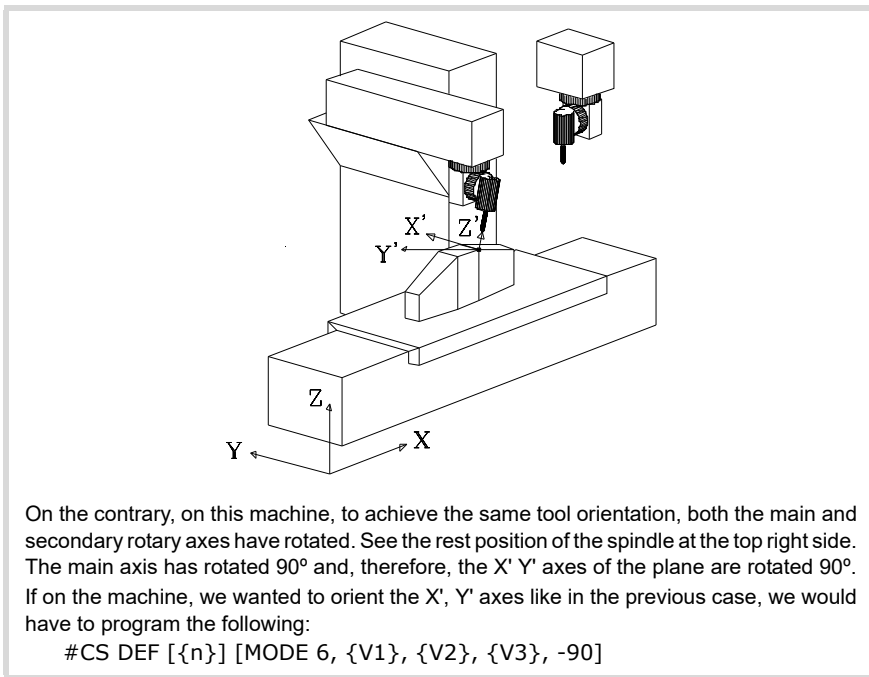
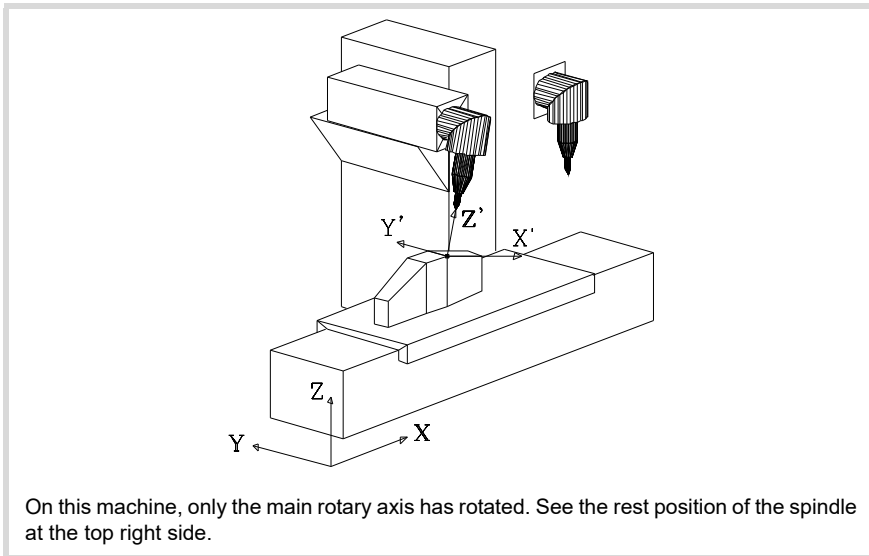


*In order to use this definition, while setting up the machine, the tool position when it is parallel to the Z axis of the machine must be set as the spindle's rest position.*

Both instructions use the same programming format and may be used together or separately.

```
#CS DEF [{n}] [MODE 6, {V1}, {V2}, {V3}, {φ1}, <FIRST/SECOND>]
#ACS DEF [{n}] [MODE 6, {V1}, {V2}, {V3}, {φ1}, <FIRST/SECOND>]
```

This mode defines a new work plane (inclined plane) perpendicular to the direction of the tool. The new work plane assumes the orientation of the tool's coordinate system.



# 20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Coordinate systems (#CS / #ACS).



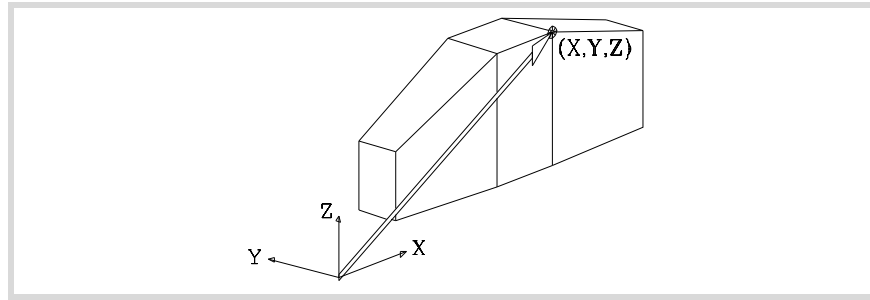
CNC 8070

(REF: 2008)



### V1, V2, V3 Components of the translation vector.

Coordinate origin of the inclined plane with respect to the current part zero.



### φ1 Coordinate (pattern) rotation.

This argument permits defining and applying a coordinate rotation in the new Cartesian plane X' Y'.

### <FIRST/SECOND> Plane alignment.

When defining the inclined plane perpendicular to the tool, the third axis of the plane is fully defined with the tool orientation. On the other hand, the location of the first and second axes of the new plane depends on the type of spindle; being 45° spindles, it is hard to predict.

Depending on the programmed option, it behaves as follow.

- When programming the FIRST command, the projection of the new first axis of the inclined plane is oriented with the first axis of the machine.
- When programming the SECOND command, the projection of the new second axis of the inclined plane is oriented with the second axis of the machine.
- If none of them is programmed, it is impossible to determine the orientation of the axes in advance, it will depend on the type of spindle.

## 20.4.7 Operation with 45° spindles (Huron type).

Huron type spindles have two solutions for orienting the tool perpendicular to the new work plane.

- The first solution is the one that involves the smallest movement of the main rotary axis (the articulation closest to the ram or furthest away from the tool) referred to the zero position.
- The second solution is the one that involves the largest movement of the main rotary axis referred to the zero position.

The selected solution will be applied to calculate the offsets of the spindle and for the instruction `#TOOL ORI`, placing the tool perpendicular to the work plane. See ["20.5 Tool perpendicular to the inclined plane \(#TOOL ORI\)." on page 374.](#)

### Selecting one of the solutions to orient the spindle.

When defining a new coordinate system, it is possible to select which one of the two solutions to be applied. For this type of spindles, if the command `SOL2` is programmed together with the instruction `#CS` or `#ACS`, the CNC applies the second solution; otherwise, if nothing is programmed, the CNC applies the first solution.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

```
#CS DEF [{n}] [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#CS ON [{n}] [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#CS ON [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#CS NEW [{n}] [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#CS NEW [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
```

```
#ACS DEF [{n}] [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#ACS ON [{n}] [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#ACS ON [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#ACS NEW [{n}] [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
#ACS NEW [MODE {mode},{V1},{V2},{V3},{φ1},{φ2},{φ3},<{align}>,<SOL2>]
```

### Consulting the position for each axis to occupy.

The position for each rotary axis to be perpendicular to the inclined plane may be consulted in the following variables. The CNC updates these variables every time a new plane is selected, using the instruction `#CS` or `#ACS`.

#### Variables for the first solution.

Variables.	Meaning.
V.G.TOOLORIF1	Position (machine coordinates) of the first rotary axis.
V.G.TOOLORIS1	Position (machine coordinates) of the second rotary axis.
V.G.TOOLORIT1	Position (machine coordinates) of the third rotary axis.
V.G.TOOLORIO1	Position (machine coordinates) of the fourth rotary axis.

#### Variables for the second solution.

Variables.	Meaning.
V.G.TOOLORIF2	Position (machine coordinates) of the first rotary axis.
V.G.TOOLORIS2	Position (machine coordinates) of the second rotary axis.
V.G.TOOLORIT2	Position (machine coordinates) of the third rotary axis.
V.G.TOOLORIO2	Position (machine coordinates) of the fourth rotary axis.

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Coordinate systems (#CS / #ACS).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

In order that the tool is perpendicular to the defined plane, the positioning must be performed using machine coordinates (#MCS), since the CNC provides the solution in machine coordinates, or by using the instruction #TOOL ORI and the movement of an axis.

Option 1. Movement in machine dimensions with the given solution.

```
#MCS ON  
G01B[V.G.TOOLORIF1] C[V.G.TOOLORIS1] F1720  
#MCS OFF
```

Option 2. Orientate the work plane so that it is perpendicular to the tool for the next movement after #TOOL ORI.

```
#TOOL ORI  
G01 X0 Y0 Z40
```

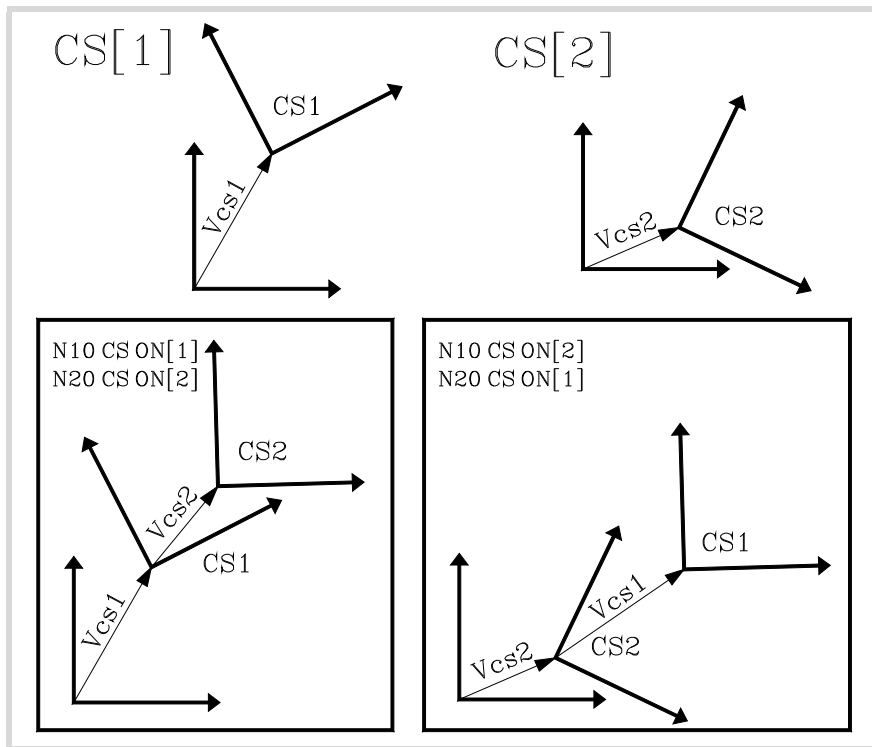
20.

### 20.4.8 How to combine several coordinate systems.

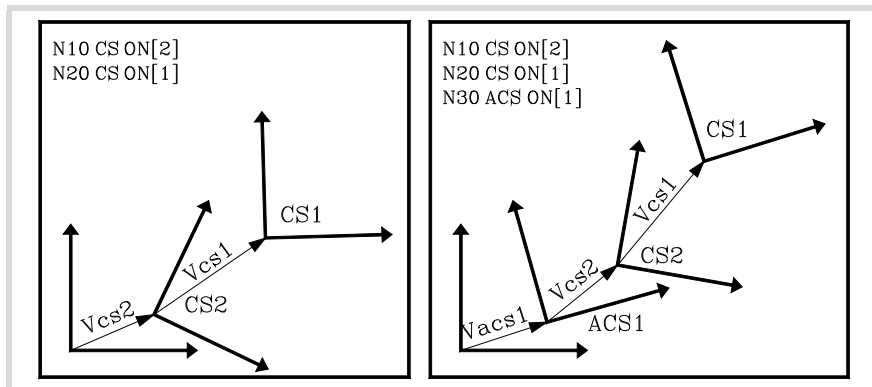
Up to 10 ACS and CS coordinate systems may be combined to build new coordinate systems. For example, the ACS coordinate system generated by a fixture on the part may be combined with the #CS coordinate system that defines the inclined plane of the part to be machined. When combining several coordinate systems, the CNC acts as follows.

- 1 First, the CNC checks the ACS and applies them sequentially in the programmed order, resulting in an ACS transformation.
- 2 Then, the CNC checks the CS and applies them sequentially in the programmed order, resulting in a CS transformation.
- 3 And last, the CNC applies the resulting CS onto the ACS to obtain the new coordinate system.

The result of the combination depends on the order they are activated as may be observed in the figure below.



Every time a #ACS or #CS is activated, the resulting coordinate system is recalculated as can be observed in the figure below.



# 20.

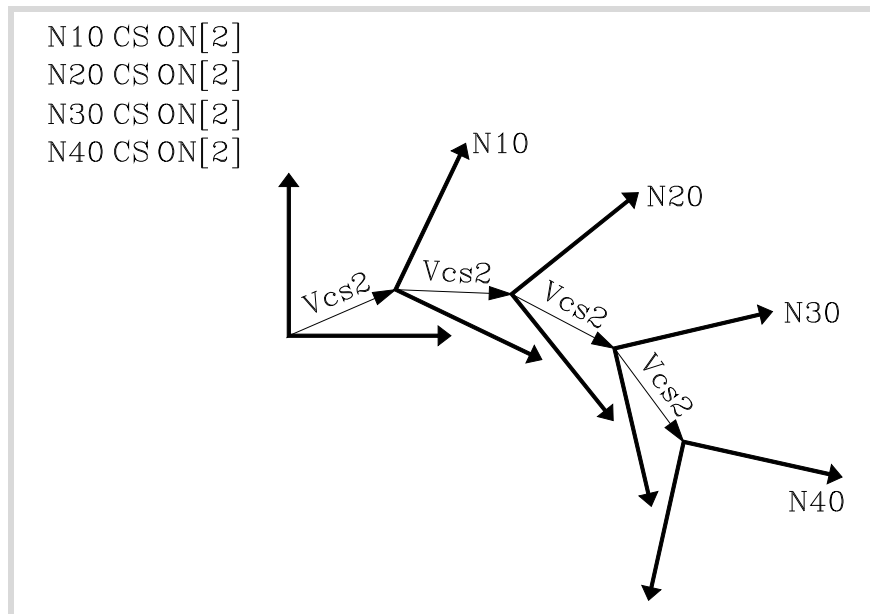
KINEMATICS AND COORDINATE TRANSFORMATION  
Coordinate systems (#CS / #ACS).

The #ACS OFF and #CS OFF instructions deactivate the last #ACS or #CS activated, respectively.

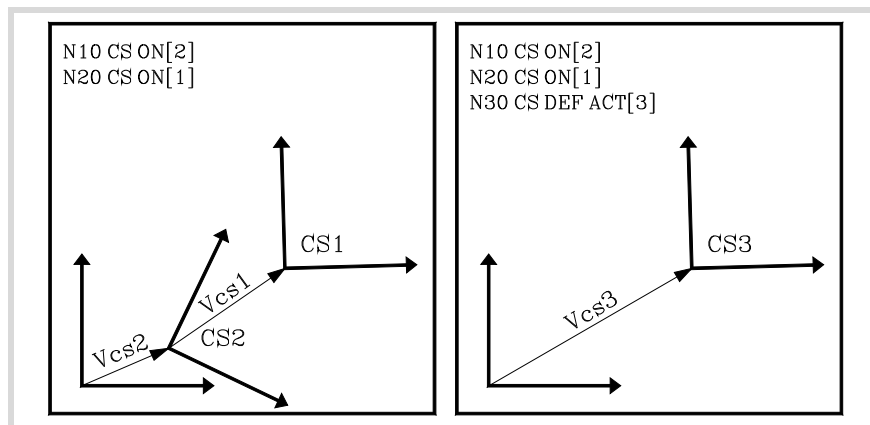
```

N100 #CS ON [1]
      (CS[1])
N110 #ACS ON [2]
      (ACS[2] + CS[1])
N120 #ACS ON [1]
      (ACS[2] + ACS[1] + CS[1])
N130 #CS ON [2]
      (ACS[2] + ACS[1] + CS[1] + CS[2])
N140 #ACS OFF
      (ACS[2] + CS[1] + CS[2])
N140 #CS OFF
      (ACS[2] + CS[1])
N150 #CS ON [3]
      (ACS[2] + CS[1] + CS[3])
N160 #ACS OFF ALL
      (CS[1] + CS[3])
N170 #CS OFF ALL
M30
    
```

A #ACS or #CS coordinate system may be activated several time.



The figure below shows an example of the instruction #CS DEF ACT [n] to assume and store the current coordinate system as a #CS.



## 20.5 Tool perpendicular to the inclined plane (#TOOL ORI).

The #TOOL ORI instruction is used to position the tool perpendicular to the active inclined plane. After executing this instruction, the tool is positioned perpendicular to the inclined plane, parallel to the third axis of the active coordinate system at the first motion programmed next.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is:

```
#TOOL ORI
```

```
#TOOL ORI
```

```
(Tool perpendicular to the inclined plane; request)
```

```
G1 X_ Y_ Z_
```

```
(Position at the defined point, with the tool perpendicular to the inclined plane)
```

# 20.

**KINEMATICS AND COORDINATE TRANSFORMATION**  
Tool perpendicular to the inclined plane (#TOOL ORI).

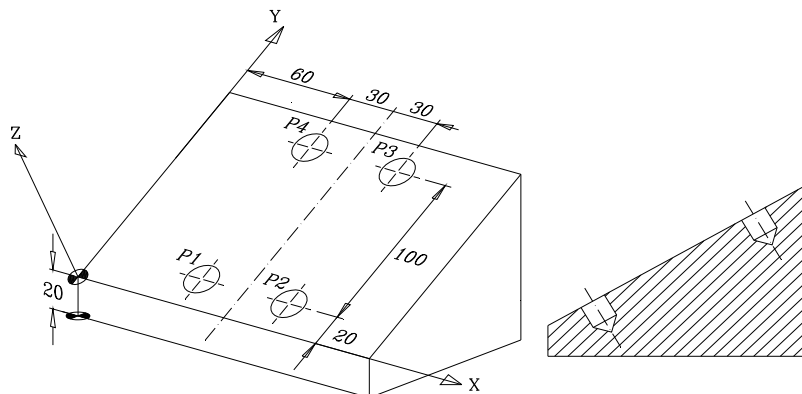


FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.5.1 Programming examples.



```
#CS ON [1] [MODE 1, 0, 0, 20, 30, 0, 0]
  (Define the inclined plane)
#TOOL ORI
  (Tool perpendicular to the inclined plane; request)
G90 G90 X60 Y20 Z3
  (Position at point P1)
  (The spindle orients perpendicular to the plane during this positioning move)
G1 G91 Z-13 F1000 M3
  (Drilling)
G0 Z13
  (Withdrawal)
G0 G90 X120 Y20
  (Position at point P2)
G1 G91 Z-13 F1000
  (Drilling)
G0 Z13
  (Withdrawal)
G0 G90 X120 Y120
  (Position at point P3)
G1 G91 Z-13 F1000
  (Drilling)
G0 Z13
  (Withdrawal)
G0 G90 X60 Y120
  (Position at point P4)
G1 G91 Z-13 F1000
  (Drilling)
G0 Z13
  (Withdrawal)
M30
```

20.

**KINEMATICS AND COORDINATE TRANSFORMATION**

Tool perpendicular to the inclined plane (#TOOL ORI).

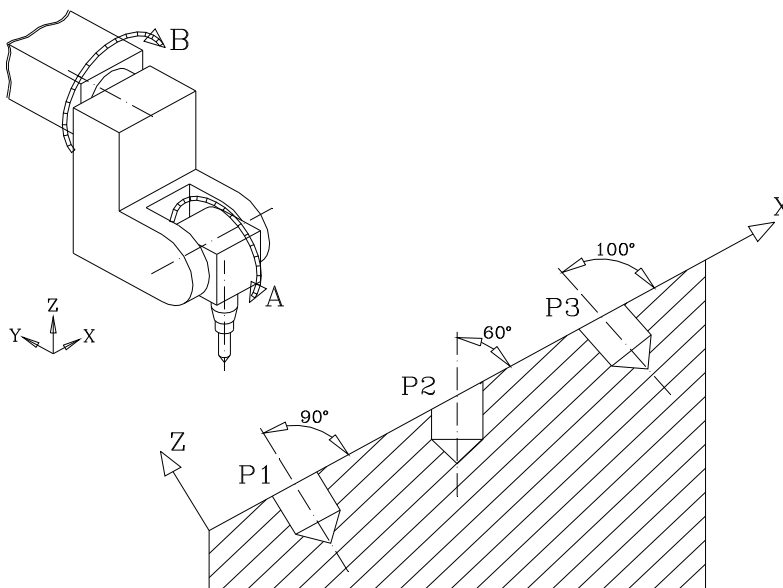
**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

The following example shows how to drill three holes with different inclination in the same plane:



```
#CS ON [1] [MODE .....]
  (Define the inclined plane)
#TOOL ORI
  (Tool perpendicular to the inclined plane; request)
G1 G90 X{P1} Y{P1} Z{P1+5}
  (Movement to point P1)
  (The spindle orients perpendicular to the plane during this positioning move)
G1 G91 Z-13 F1000 M3
  (Drilling)
G1 Z13
  (Withdrawal)
G1 X{P2} Y{P2}
  (Movement to point P2)
G90 B0
  (Orient to the tool in the machine coordinate system)
#MCS ON
  (Programming in machine coordinates)
G1 G91 Z-13 F1000
  (Drilling)
G1 Z13
  (Withdrawal)
#MCS OFF
  (End of programming in machine coordinates)
  (Restore the coordinate system of the plane)
G1 X{P3} Y{P3}
  (Movement to point P3)
G90 B-100
  (Positions the tool at 100°)
#CS OFF
#CS ON [2] [MODE6 .....]
  (Define the inclined plane perpendicular to the tool)
G1 G91 Z-13 F1000
  (Drilling)
G1 Z30
  (Withdrawal)
#CS OFF
M30
```

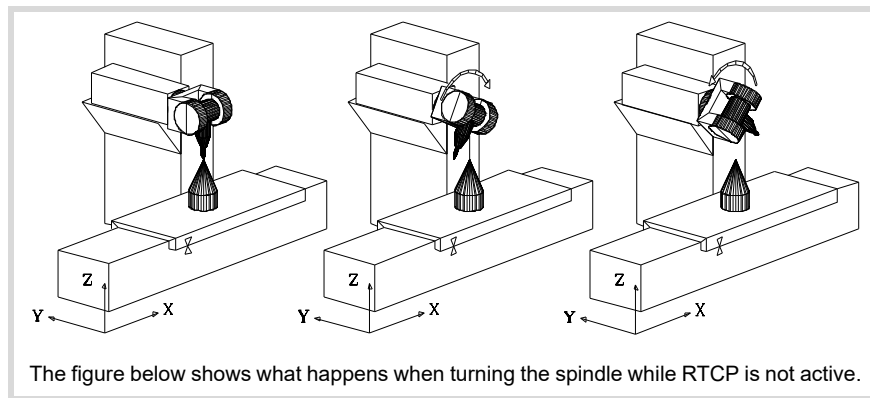
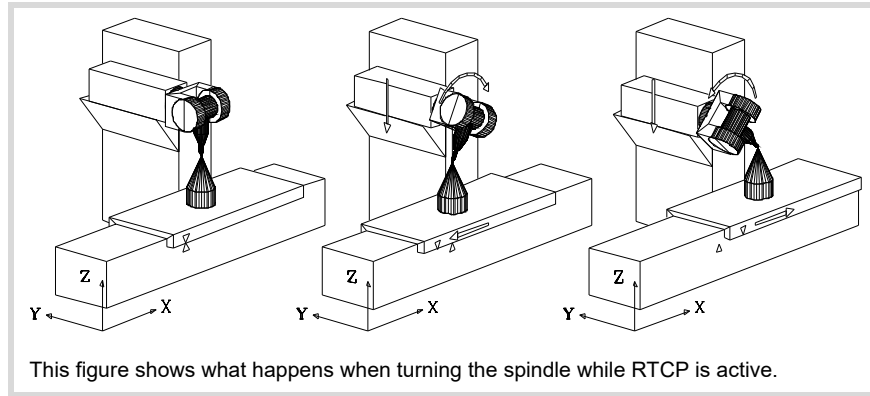
# 20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Tool perpendicular to the inclined plane (#TOOL ORI).



## 20.6 Using RTCP (Rotating Tool Center Point).

The RTCP represents a length compensation in space. The RTCP orientation of the tool may be changed without modifying the position occupied by its tip on the part. Obviously, the CNC must move several axes in order to maintain the tool tip position at all times.



Once RTCP transformation is active, spindle positioning may be combined with linear and circular interpolations.

### Programming. Turn RTCP on.

'This instruction must be programmed alone in the block'

#### Programming format.

The programming format is:

#RTCP ON

#RTCP ON

### Programming. Turn RTCP transformation off.

'This instruction must be programmed alone in the block'

#### Programming format.

The programming format is:

#RTCP OFF

#RTCP OFF

### Function properties.

The RTCP transformation is kept active even after executing M02 or M30, after an emergency or a reset and after turning the CNC off.

## Considerations about the RTCP transformation.

- In order to work with RTCP transformation, the first three axes of the channel (for example X, Y, Z) must be defined, they must form a trihedron and be linear. These axes may be GANTRY axes.
- Being RTCP transformation active, it is possible to apply zero offsets (G54-G59, G159) and preset coordinates (G92).
- Being RTCP transformation active, it is possible to carry out movements in continuous job, incremental jog and handwheel mode.
- RTCP transformation being active, the CNC only allows home search (G74) on the axes not involved in RTCP .
- The RTCP transformation cannot be selected while the TLC compensation is active.
- RTCP transformation being active, the CNC does not allow modifying the active kinematics (#KIN ID).
- RTCP transformation being active, the CNC does not allow modifying the software limits (G198/G199).

## Recommended programming order.

When working with inclined planes and RTCP transformation, it is recommended to follow this programming order (sequence). RTCP should be turned on first because it allows orienting the tool without modifying the tool tip position.

```
#RTCP ON
  (Turn RTCP on)
#CS ON
  (Activate the inclined plane)
#TOOL ORI
  (Position the tool perpendicular to the plane)
G_ X_ Y_ Z_
  (Machine on the inclined plane)
.
.
.
#CS OFF
  (Cancel the inclined plane)
#RTCP OFF
  (Turn RTCP off)
M30
  (End of part-program)
```

# 20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Using RTCP (Rotating Tool Center Point).



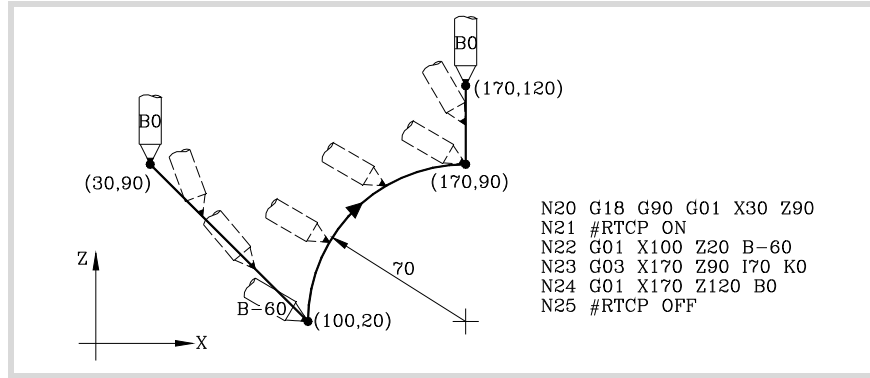
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

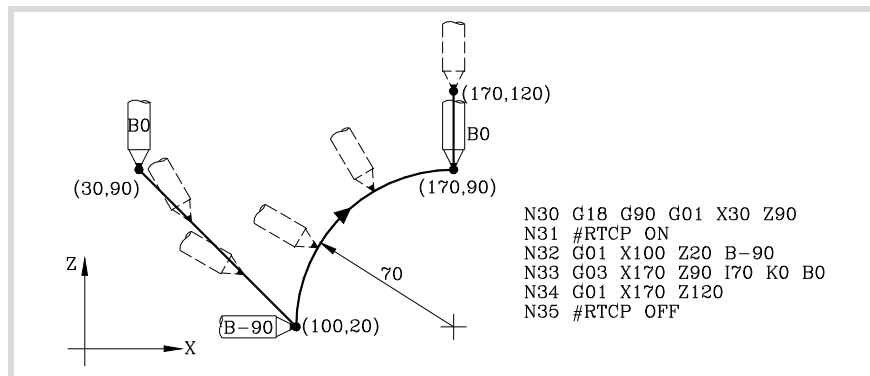
## 20.6.1 Programming examples.

### Example 1. Circular interpolation maintaining tool orientation.



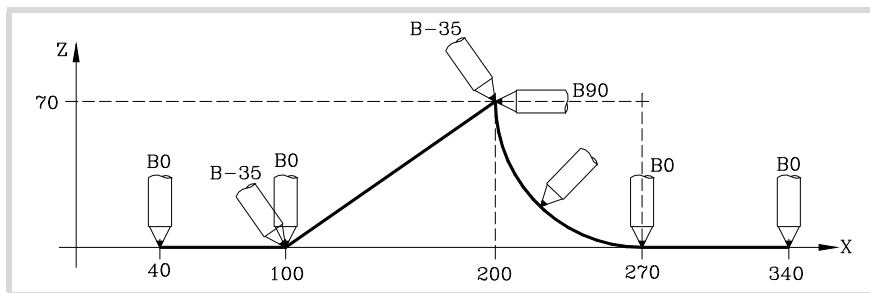
- Block N20 selects the ZX plane (G18) and positions the tool at the starting point (30,90).
- Block N21 turns RTCP on.
- Block N22 contains a movement to point (100,20) and a tool orientation from 0° to -60°. The CNC interpolates the X, Z and B axes in such a way that the tool is being oriented along the movement.
- Block N23 makes a circular interpolation to point (170,90) maintaining the same tool orientation along the whole path.
- Block N24 contains a movement to point (170,120) and a tool orientation from -60° to 0°. The CNC interpolates the X, Z and B axes in such a way that the tool is being oriented along the movement.
- Block N25 turns RTCP off.

### Example 2. Circular interpolation with tool perpendicular to its path.



- Block N30 selects the ZX plane (G18) and positions the tool at the starting point (30,90).
- Block N31 turns RTCP on.
- Block N32 contains a movement to point (100,20) and a tool orientation from 0° to -90°. The CNC interpolates the X, Z and B axes in such a way that the tool is being oriented along the movement.
- Block N33 contains a circular interpolation to point (170,90) maintaining the tool perpendicular to the path at all times.
- At the starting point, it is oriented to -90° and at the endpoint, it must end be 0°. The CNC interpolates the X, Z and B axes maintaining the tool perpendicular to its path at all times.
- Block N34 moves the tool to point (170,120) maintaining the orientation of 0°.
- Block N35 cancels RTCP.

## Example 3. Machining a profile.



```

G18 G90
  (Selects the ZX plane (G18))
#RTCP ON
  (Turn RTCP on)
G01 X40 Z0 B0 F1000
  (Position the tool at X40 Z0, oriented at 0°)
X100
  (Movement to X100 with tool oriented at 0°)
B-35
  (Orient the tool at -35°)
X200 Z70
  (Movement to X200 Z70 with tool oriented at -35°)
B90
  (Orient the tool to 90°)
G02 X270 Z0 R70 B0
  (Circular interpolation to X270 Z0) maintaining the tool perpendicular to the path)
G01 X340
  (Movement to X340 with tool oriented at 0°)
#RTCP OFF
  (Turn RTCP off)

```

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Using RTCP (Rotating Tool Center Point).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.7 Correct the implicit tool length compensation of the program (#TLC).

CAD-CAM programs take the tool length into consideration and generate the coordinates for the tool base. The instruction #TLC must be used with CAD-CAM generated programs and the CNC does not have a tool with the same dimensions. When using the #TLC function (Tool Length Compensation) the CNC compensates the length difference between both tools, the actual (real) one and the theoretical (calculated) one. The #TLC compensates for the length difference; but it does not compensate for the radius difference.

### Programming. Turn tool length compensation (TLC) on.

When defining this instruction, one must define the length difference between the real tool and the theoretical one used to write the program.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets.

```
#TLC ON [{long}]
{long}          Tool length difference (real - theoretical).
```

```
#TLC ON [1.5]
  (Turn on with a tool that is 1.5 mm longer)
#TLC ON [-2]
  (Turn on with a tool that is 2 mm shorter)
```

### Programming. Cancel tool length compensation (TLC off).

'This instruction must be programmed alone in the block'

#### Programming format.

The programming format is:

```
#TLC OFF
```

```
#TLC OFF
```

### Considerations about the TLC.

- TLC compensation being active, the CNC only allows home search (G74) on the axes not involved in TLC.
- TLC cannot be selected while the RTCP transformation is active.
- TLC compensation being active, the CNC does not allow modifying the active kinematics (#KIN ID).
- TLC compensation being active, the CNC does not allow modifying the software limits (G198/G199).

# 20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Correct the implicit tool length compensation of the program (#TLC).

**FAGOR** 

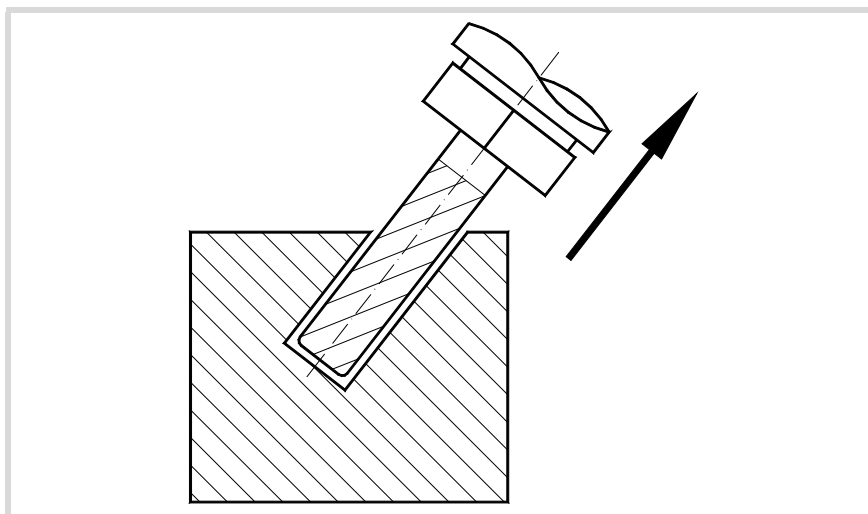
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.8 How to withdraw the tool when losing the plane.

If the CNC is turned off and back on while working with kinematics, the work plane that was selected gets lost. If the tool is inside the part, proceed as follows to withdraw it:



- 1 Use the #KIN ID [n] instruction to select the kinematics that was being used.
- 2 Use the coordinate system definition MODE6 so the CNC selects a plane perpendicular to the direction of the tool as the work plane.  
#CS ON [n] [MODE 6, 0, 0, 0, 0]
- 3 Move the tool along the longitudinal axis until it is away from the part. This movement may be made in jog mode or by program, for example, G0 G91 Z20.

# 20.

KINEMATICS AND COORDINATE TRANSFORMATION

How to withdraw the tool when losing the plane.

## 20.9 Tool orientation in the part coordinate system.

### 20.9.1 Activate tool orientation in the part coordinate system.

Currently at the CNC, in order to orient the tool considering an active kinematics, one has to program the angles of the rotary axes (the position that those axes take). With this instruction, it is possible to add the defined orientation due to the inclined plane to the tool orientation defined in the program; in other words, to refer the tool orientation to either the machine coordinate system or to the part coordinate system (#CS/#ACS) defined with the inclined plane.

Usually, the axis orienting process provides two possible solution in the positioning of the rotary axes for a particular tool orientation. The CNC applies the one resulting in the shortest distance with respect to the current position. If a little change in the programmed angle results in a huge change of angle due to the inclined plane, it is possible to define different actions depending on the angle (instruction #DEFROT).

#### Programming.

When defining this instruction, there is an option to set when the CNC orients the tool.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets. Programming the ON command is optional.

```
#CSROT <ON>
#CSROT <ON> [ROTATE]
```

```
#CSROT
#CSROT ON
#CSROT [ROTATE]
#CSROT ON [ROTATE]
```

#### ROTATE command.

Using the ROTATE command, the CNC orients the tool in the new coordinate system together with the first motion block, even if the rotary axes have not been programmed. If the ROTATE option is not programmed, the CNC orients the tool together with the first motion block where the rotary axes have been programmed.

#### Considerations.

Once activated, this instruction stays active until an M02 or M30 is executed, after a reset or it is canceled (#CSROT OFF).

# 20.

## 20.9.2 Cancel tool orientation in the part coordinate system.

The instruction #CSROT OFF cancels the programming of the rotary axes in the active ACS/CS coordinate system and, therefore, activates the programming of those axes in the machine coordinate system.

After executing an M30 and after a reset, it also cancels the programming of the rotary axes of the kinematics in the part coordinate system.

### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

```
#CSROT OFF
```

```
#CSROT OFF
```

# 20.

**KINEMATICS AND COORDINATE TRANSFORMATION**  
Tool orientation in the part coordinate system.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



### 20.9.3 How to manage the discontinuities in the orientation of rotary axes.

Usually, the axis orienting process provides two possible solution in the positioning of the rotary axes for a particular tool orientation. The CNC applies the one resulting in the shortest distance with respect to the current position.

A discontinuity means that a little change in the programmed angle results in a huge change of angle on the rotary axes due to the inclined plane. When the CNC detects a discontinuity, the instruction #DEFROT defines what the CNC must do depending on the angle difference between the one programmed and the one calculated.

#### Programming.

##### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

```
#DEFROT [<{action}>,><{criterion}>,><Q{angle}>]
```

{action}            Optional. CNC action when it detects a discontinuity.

{criterion}        Optional. Criterion to solve the discontinuity.

Q{angle}           Optional. Comparison angle.

```
#DEFROT
#DEFROT [ERROR, Q5]
#DEFROT [WARNING, DNEGF, Q10]
#DEFROT [NONE, LOWF]
```

The CNC admits programing any combination of the three parameters (a minimum of one and a maximum of three), maintaining the order.

##### CNC action when it detects a discontinuity.

These values define what the CNC must do when it detects a discontinuity.

Command.	Meaning.
ERROR	Show an error and interrupt the execution.
WARNING	Show a warning and interrupt the execution. The CNC displays a screen for the user to choose the solution to be applied, either the solution programmed in the instruction (the {criterion} argument) or the second solution.
NONE	Ignore the discontinuity and resume program execution. The CNC applies the solution programmed in the instruction (argument {criterion}), without showing to the user the screen to choose a solution. If no criterion has been programmed, the CNC applies the last one active.

If not programmed, the CNC assumes the last value programmed. After executing an M30 and after a reset, the CNC assumes the WARNING value (show a warning and interrupt the execution).

**Criterion to solve the discontinuity.**

The possible criteria are:

Command.	Meaning.
LOWF	The shortest way for the main rotary axis, then the secondary axis.
LOWS	The shortest way for the secondary rotary axis, then the main axis.
DPOSF	Positive direction of the main rotary axis.
DPOSS	Positive direction of the secondary rotary axis.
DNEGF	Negative direction of the main rotary axis.
DNEGS	Negative direction of the secondary rotary axis.
VPOSF	Positive value of the main rotary axis.
VPOSS	Positive value of the secondary rotary axis.
VNEGF	Negative value of the main rotary axis.
VNEGS	Negative value of the secondary rotary axis.

If not programmed, the CNC assumes the last value programmed. After executing an M30 and after a reset, the CNC assumes the LOWF value (the shortest distance for the main rotary axis and then secondary axis).

**Comparison angle.**

This value indicates the maximum travel difference between the programmed angle and the calculated angle, from which the actions and criteria are applied to choose the solution.

If not programmed, the CNC assumes the last value programmed. After executing an M30 and after a reset, the assumes the value of 5°.

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Tool orientation in the part coordinate system.



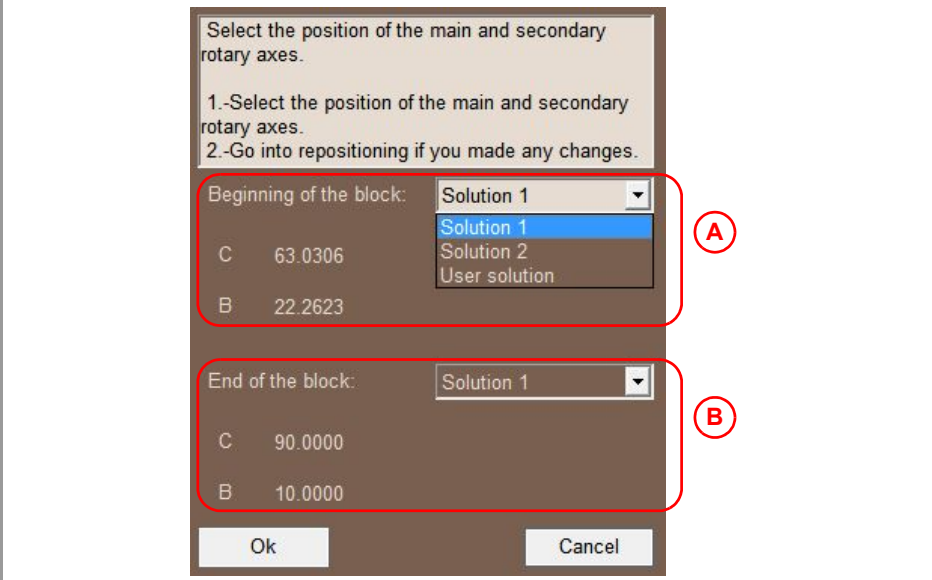
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.9.4 Screen for choosing the desired solution.

When the instruction #CSROT is programmed with the WARNING option (show a warning and interrupt the execution) the CNC shows the following screen for the user to choose the solution to be applied, both for the beginning of the block and for the end. The screen offers the two solutions calculated by the CNC, plus a third one that lets program the position of the rotary axes on the screen itself. The position of the axes is given in machine coordinates.



Select the position of the main and secondary rotary axes.

1.-Select the position of the main and secondary rotary axes.  
2.-Go into repositioning if you made any changes.

Beginning of the block: Solution 1 Solution 1 Solution 2 User solution A

C 63.0306  
B 22.2623

End of the block: Solution 1 B

C 90.0000  
B 10.0000

Ok Cancel

(A)Solution for the position of the axes at the beginning of the block.  
(B)Solution for the position of the axes at the end of the block.

By default, the CNC suggests one solution. If the user chooses the solution suggested by the CNC, it resumes execution. If the user chooses a solution other than the one suggested by the CNC, it goes into tool inspection to reposition the axes.

Once in tool inspection, the process will be as follows.

- 1 Move the tool away from the part by moving the linear axes or the virtual tool axis if it is active.
- 2 Orient the rotary axes of the kinematics.
- 3 Reposition the tool by moving the linear axes or the virtual tool axis if it is active.

## 20.9.5 Execution example. Selecting a solution.

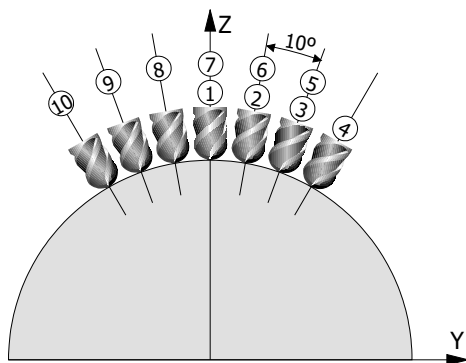
The example assumes a CB spindle type kinematics. The starting program will be a circle in the XZ plane.

```
N1 X.. Y.. Z.. C0 B0
N2 X.. Y.. Z.. C0 B10
N3 X.. Y.. Z.. C0 B20
N4 X.. Y.. Z.. C0 B30
N5 X.. Y.. Z.. C0 B20
N6 X.. Y.. Z.. C0 B10
N7 X.. Y.. Z.. C0 B0
N8 X.. Y.. Z.. C0 B-10
N9 X.. Y.. Z.. C0 B-20
N10 X.. Y.. Z.. C0 B-30
```

And specifically for a circle with a radius of 10.

```
N1 X0 Z10 C0 B0
N2 X1.736 Z9.8480 C0 B10
N3 X3.420 Z9.3969 C0 B20
N4 X5 Z8.660 C0 B30
...
```

If the part rotates 90° with respect to the C axis, the result will be a circle in the YZ plane.



```
#CS NEW[MODE1,0,0,0,0,0,90]
; 90° rotation on the C axis.
#CSROT ON
N1 X0 Z10 C0 B0
N2 X1.736 Z9.8480 C0 B10
Discontinuity point.
; Solution 1: C90 B10.
; Solution 2: C-90 B-10.
N3 X3.420 Z9.3969 C0 B20
N4 X5 Z8.660 C0 B30
M30
```

In block N2 there is a path discontinuity greater than 5° between what has been programmed and what has been calculated which is the default value for the angle that may be programmed in the instruction DEFROT. Depending on the criterion chosen, one may select solution 1 or 2 and from there on keep positioning in the rest of the blocks.

- With #DEFROT [DPOSF] (positive direction of the main axis), solution 1 is chosen and the resulting positioning of the rotary axes will be the following.

```
N2 C90 B10
N3 C90 B20
N4 C90 B30
```

- With #DEFROT [DNEF] (negative direction of the main axis), solution 2 is chosen and the resulting positioning of the rotary axes will be the following.

```
N2 C-90 B-10
N3 C-90 B-20
N4 C-90 B-30
```

If WARNING (issue a warning and generate a stop) is chosen when defining the criterion in #DEFROT, the CNC will select the solution according to the chosen criterion. The CNC will also display an interactive screen offering the option to change from one solution to another in that motion block, both in its starting orientation and in the final one.

# 20.

## 20.10 Selecting the rotary axes that position the tool in type-52 kinematics.

The instruction #SELECT ORI may be used to choose onto which rotary axes of the kinematics the tool orientation is calculated for a given direction on the work piece (part).

Kinematics 52 has at the most 2 rotary axes on the spindle and two rotary axes on the table, which means that there may be up to 4 rotary axes to orient the tool on the work piece. Therefore, there are always several solutions when calculating the position of the rotary axes to orient the tool. The tool orientation on the work piece is calculated in the following instructions:

- #CS. Define and select the machining coordinate system (Inclined plane).
- #ACS. Define and select the fixture coordinate system.
- #TOOL ORI. Orient the tool perpendicular to the work plane.
- #CSROT. Tool orientation in the part coordinate system.

### Programming.

When defining this instruction, you must define the rotary axes involved in the position calculation.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets.

```
#SELECT ORI [{ROT1}, {ROT2}]
{ROT1}      Rotary axis of the kinematics.
{ROT2}      Rotary axis of the kinematics.
```

Both arguments are defined using the following commands; HEAD1 (first axis of the spindle), HEAD2 (second axis of the spindle), TABLE1 (first axis of the table), TABLE2 (second axis of the table). Any programming sequence (order) is permitted.

```
#SELECT ORI [HEAD1, HEAD2]
  Default value.
  The tool orientation instructions work on the first and second axes of the spindle,
  leaving the table axis in their current position.

#SELECT ORI [HEAD1, TABLE1]
  The tool orientation instructions work on the first axis of the spindle and the first axis
  of the table, leaving the other two rotary axes of the kinematics in their current
  position.

#SELECT ORI [HEAD2, TABLE1]
  The tool orientation instructions work on the second axis of the spindle and the first
  axis of the table, leaving the other two rotary axes of the kinematics in their current
  position.
```

### Considerations.

This instruction is modal. On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the instruction assumes its default value; #SELECT ORI [HEAD1, HEAD2].

# 20.

KINEMATICS AND COORDINATE TRANSFORMATION

Selecting the rotary axes that position the tool in type-52 kinematics.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 20.11 Transform the current part zero considering the position of the table kinematics.

On 7-axis kinematics (spindle-table) or 5-axis table kinematics, without coordinate system rotation, it may be necessary to get a part zero with the axes of the table in any position, to use it later on when activating the RTCP of the kinematics with the option to keep the part zero without coordinate system rotation.

The instruction #KINORG may be used to transform the active part zero into a new part zero that considers the position of the table. After executing this instruction, the following variables offer the values of the transformed part zero considering the position of the table.

Variable.	Meaning.
(V.)[ch.]G.KINORG1	Position of the part zero transformed by the instruction #KINORG, considering the table position, on the first axis of the channel.
(V.)[ch.]G.KINORG2	Position of the part zero transformed by the instruction #KINORG, considering the table position, on the second axis of the channel.
(V.)[ch.]G.KINORG3	Position of the part zero transformed by the instruction #KINORG, considering the table position, on the third axis of the channel.

Save the value of these variables in the zero offset table for later in order to have that part zero available and be able to activate it at any time.

### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is:

```
#KINORG
```

```
#KINORG
```

# 20.

KINEMATICS AND COORDINATE TRANSFORMATION

Transform the current part zero considering the position of the table kinematics.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 20.11.1 Process of saving a part zero with the table axes in any position.

The following steps are valid for the type-51 table kinematics, type-52 spindle-table kinematics and the standard tables with parameter TDATA17=1.

- 1 Activate the kinematics (#KIN ID [ ]).
- 2 If it is easier for measuring, in type-52 kinematics (table-spindle) it is possible to activate only the RTCP of the spindle side.
- 3 Position the rotary axes of the spindle in the desired position for measuring the part zero. Make the measurement and activate the part zero at the desired point on X-Y-Z (G92).
- 4 From the current part zero and without moving the rotary axes of the table, calculate the variables referred to the part zero that consider the current position of the spindle and of the table (#KINORG).
- 5 At any time after executing #KINORG, save the calculated new part zero in the zero offset table.

V.A. ORGT[n].X = V.G.KINORG1

V.A. ORGT[n].Y = V.G.KINORG2

V.A. ORGT[n].Z = V.G.KINORG3

The necessary steps to activate and work with that part zero, with the spindle-table or table kinematics, without coordinate system rotation keeping the part zero are the following.

- 1 Activate the part zero where the values have been saved (G159 =n).
- 2 Activate the kinematics.
- 3 Activate the RTCP.
  - Type-52 kinematics: Activate the full RTCP (TDATA52=0) and without coordinate system rotation (TDATA51=1).
  - Type-51 kinematics: Activate the RTCP without coordinate system rotation (TDATA31=1).



*The variable of the kinematics that are applied for each TDATA are the result of adding the value and the offset, defined in the machine parameter tables. The value is defined by the OEM and the offset is a value that may be modified by the user.*

# 20.

**KINEMATICS AND COORDINATE TRANSFORMATION**  
 Transform the current part zero considering the position of the table kinematics.

## 20.11.2 Example to maintain the part zero without rotating the coordinate system.

The following example shows a possible sequence of steps so the measured part zero may be saved and restored after activating the RTCP with the option to maintain the part zero without rotating the coordinate system. The part zero may be activated with the rotary axes in any position, those of the spindle as well as those of the table.

The example uses a vectorial spindle-table type-52 kinematics on the third kinematics table. The rotary axes of the spindle are A-B and the rotary axes of the table U-V.

- 1 Activate the kinematics.

```
#KIN ID [3]
```

- 2 Activate the RTCP only on the spindle side (optional). This option may be used to work easily considering the tool tip and moving the X-Y-Z axes aligned with respect to the machine axes.

```
V.G.OFTDATA3[52]=1
  (Apply RTCP only to the spindle side)
#RTCP ON
```

- 3 Measure the reference point. Position the rotary axes of the spindle and table in the desired position for measuring the part zero on X-Y-Z.

```
A_ B_ U_ V_
X_ Y_ Z_
```

- 4 Activate the part zero at the desired point on X-Y-Z.

```
G92 X_ Y_ Z_
```

- 5 Transform the current part zero, without moving the rotary axes of the table into a new set of values that consider the table position.

```
#KINORG
```

- 6 Save the calculated values in to the zero offset table; for example, in G55 (G159=2).

```
V.A.ORG[2].X = V.G.KINORG1
V.A.ORG[2].Y = V.G.KINORG2
V.A.ORG[2].Z = V.G.KINORG3
```

- 7 Move the axes to any position and keep doing all the desired processes .

To activate the RTCP maintaining the measured part zero and without rotating the coordinate system, with the rotary and linear axes in any position, follow these steps.

- 1 Cancel the RTCP, if it is active.

```
#RTCP OFF
```

- 2 Activate the kinematics if there is another one active.

```
#KIN ID [3]
```

- 3 Activate the part zero where the KINORG has been saved; in this case, G55.

```
G55
```

- 4 Activate the full RTCP, considering the spindle and the table and without rotating the coordinate system.

```
V.G.OFTDATA3[52]=0
  (Apply full RTCP; table and spindle)
V.G.OFTDATA3[51]=1
  (RTCP without rotation of the coordinate system)
#RTCP ON
```

# 20.



## 20.12 Summary of kinematics related variables.

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

### Variables related to the active kinematics.

Variables.	R/W	Meaning.
V.G.KINTYPE	R	Type of the active kinematics.
V.G.NKINAX	R	Number of axes of the active kinematics.
V.G.SELECTORI	R	Rotary axes selected for positioning the tool (instruction #SELECT ORI)
V.G.CSROTST	R	Status of the #CSROT function.

### Variables related to the position of the rotary axes of kinematics (1).

These variables indicate the current position of the rotary axes of the kinematics.

Variables.	R/W	Meaning.
V.G.POSROTF	R/W	Current position of the first rotary axis of the kinematics.
V.G.POSROTS	R/W	Current position of the second rotary axis of the kinematics.
V.G.POSROTT	R/W	Current position of the third rotary axis of the kinematics.
V.G.POSROTO	R/W	Current position of the fourth rotary axis of the kinematics.

### Variables related to the position of the rotary axes of kinematics (2).

These variables that indicate the position the rotary axes of the kinematics must occupy in order for the tool to be perpendicular to the define inclined plane. These variables are very useful when the spindle is not fully motorized (mono-rotary or manual spindles). The CNC updates these variables every time a new plane is selected, using the instruction #CS or #ACS.

Since there isn't a single solution for angular spindle, it offers two possible solutions; the one involving the smallest movement of the main rotary axis with respect to the zero position (solution 1) and the one involving the greater movement of the main rotary axis with respect to the zero position (solution 2).

Variables.	R/W	Meaning.
V.G.TOOLORIF1	R	Position (machine coordinates) to be occupied by the first rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 1.
V.G.TOOLORIF2	R	Position (machine coordinates) to be occupied by the first rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 2.
V.G.TOOLORIS1	R	Position (machine coordinates) to be occupied by the second rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 1.
V.G.TOOLORIS2	R	Position (machine coordinates) to be occupied by the second rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 2.
V.G.TOOLORIT1	R	Position (machine coordinates) to be occupied by the third rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 1.
V.G.TOOLORIT2	R	Position (machine coordinates) to be occupied by the third rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 2.
V.G.TOOLORIO1	R	Position (machine coordinates) to be occupied by the fourth rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 1.
V.G.TOOLORIO2	R	Position (machine coordinates) to be occupied by the fourth rotary axis in order to position the tool perpendicular to the inclined plane, according to solution 2.

20.

In order that the tool is perpendicular to the defined plane, the positioning must be performed using machine coordinates, since the CNC provides the solution in machine coordinates, or by using the instruction #TOOL ORI and the movement of an axis.

Option 1. Movement in machine dimensions with the given solution.

```
#MCS ON
G01B[V.G.TOOLORIF1] C[V.G.TOOLORIS1] F1720
#MCS OFF
```

Option 2. Orientate the work plane so that it is perpendicular to the tool for the next movement after #TOOL ORI.

```
#TOOL ORI
G01 X0 Y0 Z40
```

**Variables related to the CSROT option (tool orientation in the part coordinate system).**

Variables.	R/W	Meaning.
V.G.CSROTST	R	Status of the #CSROT function.
V.G.CSROTF1[1]	R	Position (machine coordinates) calculated for the first rotary axis of the kinematics at the beginning of the block, for solution 1 of the #CSROT mode.
V.G.CSROTF1[2]	R	Position (machine coordinates) calculated for the first rotary axis of the kinematics at the end of the block, for solution 1 of the #CSROT mode.
V.G.CSROTS1[1]	R	Position (machine coordinates) calculated for the second rotary axis of the kinematics at the beginning of the block, for solution 1 of the #CSROT mode.
V.G.CSROTS1[2]	R	Position (machine coordinates) calculated for the second rotary axis of the kinematics at the end of the block, for solution 1 of the #CSROT mode.
V.G.CSROTT1[1]	R	Position (machine coordinates) calculated for the third rotary axis of the kinematics at the beginning of the block, for solution 1 of the #CSROT mode.
V.G.CSROTT1[2]	R	Position (machine coordinates) calculated for the third rotary axis of the kinematics at the end of the block, for solution 1 of the #CSROT mode.
V.G.CSROTO1[1]	R	Position (machine coordinates) calculated for the fourth rotary axis of the kinematics at the beginning of the block, for solution 1 of the #CSROT mode.
V.G.CSROTO1[2]	R	Position (machine coordinates) calculated for the fourth rotary axis of the kinematics at the end of the block, for solution 1 of the #CSROT mode.
V.G.CSROTF2[1]	R	Position (machine coordinates) calculated for the first rotary axis of the kinematics at the beginning of the block, for solution 2 of the #CSROT mode.
V.G.CSROTF2[2]	R	Position (machine coordinates) calculated for the first rotary axis of the kinematics at the end of the block, for solution 2 of the #CSROT mode.
V.G.CSROTS2[1]	R	Position (machine coordinates) calculated for the second rotary axis of the kinematics at the beginning of the block, for solution 2 of the #CSROT mode.
V.G.CSROTS2[2]	R	Position (machine coordinates) calculated for the second rotary axis of the kinematics at the end of the block, for solution 2 of the #CSROT mode.
V.G.CSROTT2[1]	R	Position (machine coordinates) calculated for the third rotary axis of the kinematics at the beginning of the block, for solution 2 of the #CSROT mode.
V.G.CSROTT2[2]	R	Position (machine coordinates) calculated for the third rotary axis of the kinematics at the end of the block, for solution 2 of the #CSROT mode.
V.G.CSROTO2[1]	R	Position (machine coordinates) calculated for the fourth rotary axis of the kinematics at the beginning of the block, for solution 2 of the #CSROT mode.

20.

KINEMATICS AND COORDINATE TRANSFORMATION  
Summary of kinematics related variables.



(REF: 2008)

Variables.	R/W	Meaning.
V.G.CSROTO2[2]	R	Position (machine coordinates) calculated for the fourth rotary axis of the kinematics at the end of the block, for solution 2 of the #CSROT mode.
V.G.CSROTF[1]	R/W	Position (machine coordinates) to be occupied by the first rotary axis of the kinematics at the beginning of the block, for the #CSROT mode.
V.G.CSROTF[2]	R/W	Position (machine coordinates) to be occupied by the first rotary axis of the kinematics at the end of the block, for the #CSROT mode.
V.G.CSROTS[1]	R/W	Position (machine coordinates) of the second rotary axis at the beginning of the block, for the #CSROT mode.
V.G.CSROTS[2]	R/W	Position (machine coordinates) to be occupied by the second rotary axis at the end of the block, for the #CSROT mode.
V.G.CSROTT[1]	R/W	Position (machine coordinates) to be occupied by the third rotary axis at the beginning of the block, for the #CSROT mode.
V.G.CSROTT[2]	R/W	Position (machine coordinates) to be occupied by the third rotary axis at the end of the block, for the #CSROT mode.
V.G.CSROTO[1]	R/W	Position (machine coordinates) to be occupied by the fourth rotary axis at the beginning of the block, for the #CSROT mode.
V.G.CSROTO[2]	R/W	Position (machine coordinates) to be occupied by the fourth rotary axis at the end of the block, for the #CSROT mode.

**Variables related to the KINORG option (position of the current transformed part zero, considering the position of the table kinematics).**

Variable.	R/W	Meaning.
(V.)[ch.]G.KINORG1	R	Position of the part zero transformed by the instruction #KINORG, considering the table position, on the first axis of the channel.
(V.)[ch.]G.KINORG2	R	Position of the part zero transformed by the instruction #KINORG, considering the table position, on the second axis of the channel.
(V.)[ch.]G.KINORG3	R	Position of the part zero transformed by the instruction #KINORG, considering the table position, on the third axis of the channel.

20.

KINEMATICS AND COORDINATE TRANSFORMATION

Summary of kinematics related variables.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 20.

## KINEMATICS AND COORDINATE TRANSFORMATION

Summary of kinematics related variables.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Nowadays, lots of parts are designed using CAD-CAM systems. This type of information is later post-processed to generate a CNC program, usually made up of a large number of very short blocks of several mm or just a few tenths of a micron.

In this type of parts, the CNC must be capable to analyze a large number of points in advance so it can generate a continuous path that goes through (or near) the points of the program while keeping (the best way possible) the programmed feedrate and the restrictions of maximum acceleration, jerk, etc of each axis and of the path.

## **Default HSC mode.**

The command to execute programs made up of lots of small blocks, typical of high speed machining, is carried out with a single instruction #HSC. This function offers several ways to work; optimizing part surface finish (SURFACE mode) optimizing the contour error (CONTERROR mode) or the machining feedrate (FAST mode).

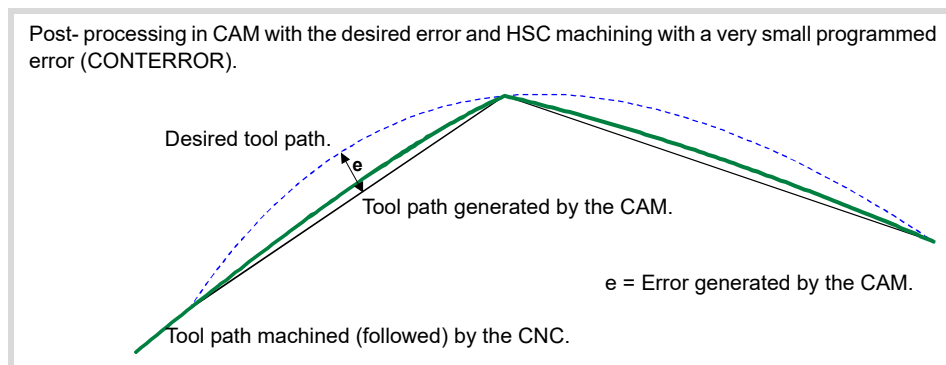
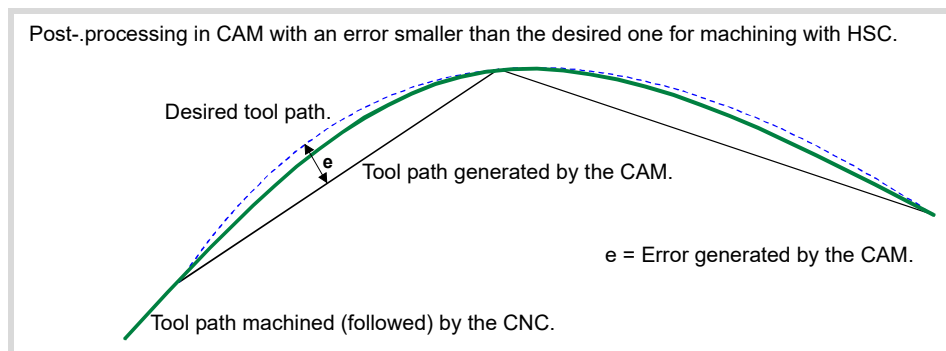
The default machining mode is defined by parameter HSCDEFAULTMODE, where Fagor offers the SURFACE mode as default. The more sophisticated algorithms of the SURFACE mode obtain more accurate machining. In parallel, the CNC controls machine movements more smoothly reducing considerably the vibrations due to part geometry or machine dynamics. Reducing machine vibrations results in higher surface quality of the machined parts.

## 21.1 Recommendations for machining.

### Selecting the chordal error at the CNC and at the post-processed CAM.

As mentioned earlier, the error caused by the CNC between the programmed part and the resulting part is never greater than the programmed value. On the other hand, the CAM system also generates an error when processing the original part and converting the paths into a CNC program. The resulting error may be the sum of the two; therefore, the desired maximum error must be spread between both processes.

Selecting a large chordal error when generating the program and a small chordal error when executing it, the execution is slower and of lower quality. In this case ridges will appear because the CNC perfectly follows the CAM generated polyhedron. It is recommended to post-process at the CAM with a smaller error than the desired one for high speed cutting HSC (between 10% and 20%). For example, for a maximum error of 50 microns, we should post-process with an error of 5 or 10 microns and program the 50 microns in the HSC command (#HSC ON [CONTERROR E0.050]). This way of programming lets CNC modify the profile while respecting the dynamics of each axis without causing undesired effects like ridges. If CAM post-processing is done with the desired error and a very small error is programmed in HSC CONTERROR, as a result, the CNC will follow faithfully the facets generated by the CAM.



### The part-program.

Since the CNC works with an accuracy of nanometers, better results may be obtained if the coordinate have 4 or 5 decimals than if they only have 2 or 3.

**21.**  
**HSC. HIGH SPEED MACHINING.**  
 Recommendations for machining.



CNC 8070

(REF: 2008)

## 21.2 User subroutines G500-G501 to turn HSC on/off.

The CNC lets the users define up to 100 subroutines, common to all channels and that will be associated with functions G500 through G599, so when the CNC executes one of these functions, it will execute its associated subroutine.

Subroutines G500 and G501 are pre-configured by Fagor to turn HSC on and off in SURFACE mode (mode recommended by Fagor). Both subroutines may be modified by the user.

Subroutine.	Meaning.
G500	HSC cancellation.
G501	Turning HSC on in SURFACE mode.

### Subroutines supplied by Fagor.

The subroutines associated with these functions will be global and will have the same name as the function, without extension. The subroutines must be defined in the folder ..\Users\Sub. If the CNC executes a function and there is no subroutine, the CNC will issue an error message.

- G500 will have subroutine G500 associated with it.
- G501 will have subroutine G501 associated with it.

These functions may be programmed anywhere in the program and allow resetting the local parameters of the subroutine.

### Programming the subroutines.

The programming format is the following; the list of arguments is shown between curly brackets, they will be the parameters to be initialized by the local parameters of the subroutine. The angular brackets indicate that all the arguments are optional.

G501 <A{ %acceleration}> <E{error}> <J{ %jerk}> <M{mode}>

- A Optional. Percentage of acceleration.
- E Optional. Maximum chordal error allowed (millimeters or inches).
- J Optional. Percentage of jerk.
- M Optional. HSC mode (1=SURFACE; 2=FAST; 3=CONTEERROR).

```
G501
(Acceleration = 100%)
(Chordal error = machine parameter HSCROUND)
(Jerk = 100%)
(Mode = machine parameter HSCDEFAULTMODE)
```

```
G501 A97.5 E0.01 M1
(Acceleration = 97.5%)
(Chordal error = 0.01)
(Jerk = 100%)
(Mode = SURFACE)
```

### Subroutine G500 supplied by Fagor (may be modified by the user).

```
; HSC off
#ESBLK
G131 100 ; % of global acceleration.
G133 100 ; % of global jerk.
#HSC OFF
#RETDSBLK
```

# 21.

HSC. HIGH SPEED MACHINING.  
User subroutines G500-G501 to turn HSC on/off.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Subroutine G501 supplied by Fagor (may be modified by the user).

21.

**HSC. HIGH SPEED MACHINING.**  
 User subroutines G500-G501 to turn HSC on/off.

```

;-----
;-----
; HSC ACTIVATION
;
; OPTIONAL PARAMETERS
;
; E - CONTOUR TOLERANCE
; A - % ACCELERATION
; J - % JERK
; M - HSCMODE
;   1 SURFACE
;   2 FAST
;   3 CONTERROR
;
;-----
;-----
#ESBLK
#HSC OFF
#PATHND ON
;-----HSC MODE -----
$IF V.C.PCALLP_M
  $IF [P12 == 1]
    #HSC ON [SURFACE]
  $ELSEIF [P12 == 2]
    #HSC ON [FAST]
  $ELSEIF [P12 == 3]
    #HSC ON [CONTERROR]
  $ENDIF
$ELSE
  #HSC ON
$ENDIF
;-----CONTOUR TOLERANCE----
$IF V.C.PCALLP_E
  #HSC ON [EP4]
$ENDIF
;-----ACCELERATION -----
$IF V.C.PCALLP_A
  G131 P0
$ELSE
  G131 100
$ENDIF
;-----JERK-----
$IF V.C.PCALLP_J
  G133 P9
$ELSE
  G133 100
$ENDIF
#RETDSBLK

```



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 21.2.1 Alternative example for functions G500-G501 supplied by Fagor.

G500 subroutines supplied by Fagor may be modified by the user. Here is another example for turning HSC ON/OFF using three subroutines.

Subroutine.	Meaning.
G500	Turn HSC off.
G501	Turn HSC on in FAST mode.
G502	Turn HSC on in SURFACE mode.

### Programming the subroutines.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets. In these subroutines, all the arguments are optional.

G501 <A{ %acceleration} > <E{error} >

G502 <A{ %acceleration} > <E{error} >

A Optional. Percentage of acceleration.

E Optional. Maximum chordal error allowed (millimeters or inches).

```
G501
(Acceleration = 100%)
(Chordal error = twice the value set in machine parameter HSCROUND)
G501 A97.5 E0.01
(Acceleration = 97.5%)
(Chordal error = 0.01)
```

```
G502
(Acceleration = 100%)
(Chordal error = machine parameter HSCROUND)
```

### Example of a G500 subroutine. Turn HSC off.

```
-----
;
;
; HSC DEACTIVATION
;
;
;
#ESBLK
G131 100 ;% acceleration
G133 100 ;% deceleration
V.G.DYNOVR = 100 ;%Dynamic override
#PATHND OFF
#HSC OFF

#RETDSBLK
```

# 21.

**HSC. HIGH SPEED MACHINING.**  
User subroutines G500-G501 to turn HSC on/off.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Example of a G501 subroutine. Turn HSC on in FAST mode.**

```

;-----
;-----
; HSC ROUGHING ACTIVATION
; E - Contour Tolerance
; A - % Acceleration
;-----
;-----
#ESBLK
#HSC OFF
#PATHND ON

$IF V.C.PCALLP_A
  G131 P0
$ELSE
  G131 100
$ENDIF

$IF V.C.PCALLP_E == 0
  P4 = 2 * V.MPG.HSCROUND
$ENDIF

#HSC ON [FAST, EP4]
V.G.DYNOVR = 120

#RETDSBLK

```

**Example of a G502 subroutine. Turn HSC on in SURFACE mode.**

```

;-----
;-----
; HSC FINISHING ACTIVATION
; E - Contour Tolerance
; A - % Acceleration
;-----
;-----
#ESBLK
#HSC OFF
V.G.DYNOVR = 100
#PATHND ON

$IF V.C.PCALLP_E == 0
  P4 = V.MPG.HSCROUND
$ENDIF
$IF V.C.PCALLP_A
  G131 P0
$ELSE
  G131 100
$ENDIF

#HSC ON [SURFACE, EP4]

#RETDSBLK

```

**21.**

**HSC. HIGH SPEED MACHINING.**  
User subroutines G500-G501 to turn HSC on/off.

## 21.3 HSC SURFACE mode. Optimization of surface finish.

It is the recommended way (mode) to work. This mode optimizes the velocity profile through intelligent algorithms that detect curvature changes.

This mode offers excellent results in time and surface quality solving jerk problems that may appear depending on the profile to be machined. This is the best mode for roughing and semi-finishing operations besides finishing operations where surface quality is most important.

### Activating the HSC mode.

Program the instruction alone in the block. This mode is turned on with the instruction #HSC ON and the SURFACE command.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#HSC ON [<SURFACE> <,E{error}> <,CORNER{angle}> <,RE{error}>
<,SF{frequency}> <,AXF{frequency}> <,OS{frequency}>]
```

SURFACE	Optional. HSC mode.
E{error}	Optional. Maximum chordal error allowed. Units: mm or inches.
CORNER{angle}	Optional. Maximum angle for square corner. Units: from 0 to 180°.
RE{error}	Optional. Maximum error on rotary axes. Units: Degrees
SF{frequency}	Optional. Path filter frequency for linear slope. Units: Herz.
AXF{frequency}	Optional. Axis filter frequency. Units: Herz.
OS{frequency}	Optional. Orientation smoothing of the rotary axes working with RTCP. Units: ms.

```
#HSC ON
(SURFACE mode, if it is the default mode)
(Chordal error = machine parameter HSCROUND)
(Angle = machine parameter CORNER)
(RE error = machine parameter MAXERROR)
(SF filter = machine parameter SOFTFREQ)
(AXF filter = machine parameter SURFFILTFREQ)
(OS filter = machine parameter ORISMOOTH)
```

```
#HSC ON [SURFACE]
(Chordal error = machine parameter HSCROUND)
(Angle = machine parameter CORNER)
```

```
#HSC ON [SURFACE, E0.01]
(Chordal error = 0.01)
(Angle = machine parameter CORNER)
```

```
#HSC ON [SURFACE, E0.01, CORNER150]
(Chordal error = 0.01)
(Angle = 150°)
```

```
#HSC ON [SURFACE, CORNER150]
(Chordal error = machine parameter HSCROUND)
(Angle = 150°)
```

### HSC mode.

The work mode must only be selected when it is not the default mode (parameter HSCDEFAULTMODE).

# 21.

**HSC. HIGH SPEED MACHINING.**  
HSC SURFACE mode. Optimization of surface finish.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

**Maximum chordal error allowed.**

The E command sets the maximum contouring error allowed between the programmed path and the resulting path (mm or inches). This command is applied to the first three linear axes of the channel. Programming it is optional; if not programmed, the CNC assumes as maximum contouring error the value set in machine parameter HSCROUND.

**Maximum angle for square corner.**

The CORNER command sets the maximum angle between two paths (between 0° and 180°), under which the CNC machines in square corner mode. Programming it is optional; if not programmed, the CNC assumes the angle set in machine parameter CORNER.

**Maximum error on rotary axes.**

The RE command defines the error in all the rotary axes and linear axes (except the first three axes of the channel). Programming it is optional; if not programmed, the CNC assumes as maximum error the highest value between machine parameter MAXERROR and the E command.

**Path filter frequency for linear slope.**

The SF command allows applying different filters to those set in the machine parameters. Lower the value of this command to obtain a smoother movement and slightly slower without losing accuracy.

Programming the SF command is optional; if not programmed, the CNC assumes as frequency of the filter the one defined in machine parameter SOFTFREQ.

**Axis filter frequency in HSC mode.**

The AXF command allows applying different filters to those set in the machine parameters. Lower the value of this command to obtain a smoother path and faster but with lower accuracy.

Programming the AXF command is optional; if not programmed, the CNC assumes as frequency of the filter the one defined in machine parameter SURFFILTFREQ.

**Orientation smoothing of the rotary axes working with RTCP.**

The OS command may be used to smooth the orientation of the rotary axes, without tool tip error, when working with RTCP in HSC SURFACE mode. Increase the value of this command to obtain smoother RTCP movements.

Programming the OS command is optional; if not programmed, the CNC assumes the value set in machine parameter ORISMOOTH.

**Considerations.****Commands E and CORNER.**

The CNC maintains the value of the commands programmed until a different one is programmed, the HSC mode is canceled, a reset is done or the program ends.

When switching HSC modes, the CNC keeps the values programmed in the previous mode for the commands that are not programmed (for example, the contouring error). If no HSC mode has been programmed earlier, the CNC takes the default values for the commands that are not programmed.

**Example 1.**

```
#HSC ON [CONTEERROR, E0.050]
.
#HSC ON [SURFACE]
(Chordal error = 0,050)
```

**Commands RE, SF and AXF.**

The CNC maintains the value of the commands programmed until a different one is programmed, the HSC mode is changed or canceled, a reset is done or the program ends.

21.

**HSC. HIGH SPEED MACHINING.**  
HSC SURFACE mode. Optimization of surface finish.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

When changing HSC modes, the CNC assumes the default values set in the machine parameters.

**Execute an HSC mode starting with initial conditions.**

To execute in HSC mode starting with initial conditions, first cancel the previous mode. See "[21.6 Canceling the HSC mode.](#)" on page 410.

**Example 2.**

```
#HSC ON [CONTERROR, E0.050]
.
#HSC OFF
.
#HSC ON [SURFACE]
(Chordal error = machine parameter HSCROUND)
```

# 21.

**HSC. HIGH SPEED MACHINING.**  
HSC SURFACE mode. Optimization of surface finish.

## 21.4 HSC CONTERROR mode. Optimizing the contouring error.

From this instruction on, the CNC modifies the geometry through intelligent algorithms for eliminating unnecessary points and automatically generating polynomials. This way, the contour is traveled at a variable feedrate according to the curvature and the programmed parameters (acceleration and feedrate) but respecting the set error limits.

### Programming.

Program the instruction alone in the block. This mode is turned on with the instruction #HSC ON and the CONTERROR command.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#HSC ON [<CONTERROR> <,E{error}> <,CORNER{angle}> <,RE{error}>
<,AXF{frequency}>]
```

CONTERROR	Optional. HSC mode.
E{error}	Optional. Maximum chordal error allowed. Units: mm or inches.
CORNER{angle}	Optional. Maximum angle for square corner. Units: from 0 to 180°.
RE{error}	Optional. Maximum error on rotary axes. Units: Degrees
AXF{frequency}	Optional. Axis filter frequency. Units: Herz.

```
#HSC ON [CONTERROR]
  (Chordal error = machine parameter HSCROUND)
  (Angle = machine parameter CORNER)
  (RE error = machine parameter MAXERROR)
  (AXF filter = machine parameter HSCFILTFREQ)

#HSC ON [CONTERROR, E0.01]
  (Chordal error = 0.01)
  (Angle = machine parameter CORNER)

#HSC ON [CONTERROR, E0.01, CORNER150]
  (Chordal error = 0.01)
  (Angle = 150°)

#HSC ON [CONTERROR, CORNER150]
  (Chordal error = machine parameter HSCROUND)
  (Angle = 150°)
```

### HSC mode.

The work mode must only be selected when it is not the default mode (parameter HSCDEFAULTMODE).

### Maximum chordal error allowed.

The E command sets the maximum contouring error allowed between the programmed path and the resulting path (mm or inches). This command is applied to the first three linear axes of the channel. Programming it is optional; if not programmed, the CNC assumes as maximum contouring error the value set in machine parameter HSCROUND.

### Maximum angle for square corner.

The CORNER command sets the maximum angle between two paths (between 0° and 180°), under which the CNC machines in square corner mode. Programming it is optional; if not programmed, the CNC assumes the angle set in machine parameter CORNER.

# 21.

HSC. HIGH SPEED MACHINING. HSC CONTERROR mode. Optimizing the contouring error.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Maximum error on rotary axes.**

The RE command defines the error in all the rotary axes and linear axes (except the first three axes of the channel). Programming it is optional; if not programmed, the CNC assumes as maximum error the highest value between machine parameter MAXERROR and the E command.

**Axis filter frequency in HSC mode.**

The AXF command allows applying different filters to those set in the machine parameters. Programming it is optional; if not programmed, the CNC assumes as maximum contouring error the value set in machine parameter HSCFILTFREQ.

**Considerations.****Commands E and CORNER.**

The CNC maintains the value of the commands programmed until a different one is programmed, the HSC mode is canceled, a reset is done or the program ends.

When switching HSC modes, the CNC keeps the values programmed in the previous mode for the commands that are not programmed (for example, the contouring error). If no HSC mode has been programmed earlier, the CNC takes the default values for the commands that are not programmed.

**Example 1.**

```
#HSC ON [CONTERROR, E0.050]
.
#HSC ON [SURFACE]
  (Chordal error = 0,050)
```

**Commands RE, SF and AXF.**

The CNC maintains the value of the commands programmed until a different one is programmed, the HSC mode is changed or canceled, a reset is done or the program ends.

When changing HSC modes, the CNC assumes the default values set in the machine parameters.

**Execute an HSC mode starting with initial conditions.**

To execute in HSC mode starting with initial conditions, first cancel the previous mode. See ["21.6 Canceling the HSC mode."](#) on page 410.

**Example 2.**

```
#HSC ON [CONTERROR, E0.050]
.
#HSC OFF
.
#HSC ON [SURFACE]
  (Chordal error = machine parameter HSCROUND)
```

## 21.5 HSC FAST mode. Optimizing the machining feedrate.

In spite of the recommendations for generating CAM programs, it is possible to have programs already generated that do not show a continuity between the error generated by the CAM, the block size and the error required by the HSC. For this type of programs, the HSC mode has a fast mode where the CNC generates paths trying to restore this continuity and thus be able to work on a smoother surface and obtain a more continuous feedrate.

### Programming.

Program the instruction alone in the block. This mode is turned on with the instruction #HSC ON and the FAST command.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#HSC ON [<FAST> <,E{error}> <,CORNER{angle}> <,RE{error}> <,SF{frequency}> <,AXF{frequency}>]
```

FAST	Optional. HSC mode.
E{error}	Optional. Maximum chordal error allowed. Units: mm or inches.
CORNER{angle}	Optional. Maximum angle for square corner. Units: from 0 to 180°.
RE{error}	Optional. Maximum error on rotary axes. Units: Degrees
SF{frequency}	Optional. Path filter frequency for linear slope. Units: Herz.
AXF{frequency}	Optional. Axis filter frequency. Units: Herz.

```
#HSC ON [FAST]
  (Chordal error = machine parameter HSCROUND)
  (Angle = machine parameter CORNER)
  (RE error = machine parameter MAXERROR)
  (SF filter = machine parameter SOFTFREQ)
  (AXF filter = machine parameter FASTFILTREQ)
```

```
#HSC ON [FAST, E0.05]
  (Chordal error = 0.05)
  (Angle = machine parameter CORNER)
```

```
#HSC ON [FAST, E0.01, CORNER130]
  (Chordal error = 0.01)
  (Angle = 130°)
```

#### HSC mode.

The work mode must only be selected when it is not the default mode (parameter HSCDEFAULTMODE).

#### Maximum chordal error allowed.

The E command sets the maximum contouring error allowed between the programmed path and the resulting path (mm or inches). This command is applied to the first three linear axes of the channel. Programming it is optional; if not programmed, the CNC assumes as maximum contouring error the value set in machine parameter HSCROUND.

Programming the chordal error improves the accuracy in curved segments or circles; however, and considering the peculiarities of executing in FAST mode, the contour error at the corners is not ensured.

# 21.

HSC. HIGH SPEED MACHINING.  
HSC FAST mode. Optimizing the machining feedrate.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



**Maximum angle for square corner.**

The CORNER command sets the maximum angle between two paths (between 0° and 180°), under which the CNC machines in square corner mode. Programming it is optional; if not programmed, the CNC assumes the angle set in machine parameter CORNER.

**Maximum error on rotary axes.**

The RE command defines the error in all the rotary axes and linear axes (except the first three axes of the channel). Programming it is optional; if not programmed, the CNC assumes as maximum error the highest value between machine parameter MAXERROR and the E command.

**Path filter frequency for linear slope.**

The SF command allows applying different filters to those set in the machine parameters. Programming it is optional; if not programmed, the CNC assumes as frequency of the filter the one defined in machine parameter SOFTFREQ.

**Axis filter frequency in HSC mode.**

The AXF command allows applying different filters to those set in the machine parameters. Programming it is optional; if not programmed, the CNC assumes as maximum contouring error the value set in machine parameter FASTFILTFREQ.

**Considerations.****Percentage of acceleration in the transition between blocks.**

*From versions V1.30 (8060) and V5.30 (8065/8070), the #HSC instruction does not let program the percentage of acceleration for transition between blocks.*

The percentage of acceleration in the transition between blocks may be modified using functions G130/G131. The CNC assumes by default the value of machine parameter ACCEL.

**Commands E and CORNER.**

The CNC maintains the value of the commands programmed until a different one is programmed, the HSC mode is canceled, a reset is done or the program ends.

When switching HSC modes, the CNC keeps the values programmed in the previous mode for the commands that are not programmed (for example, the contouring error). If no HSC mode has been programmed earlier, the CNC takes the default values for the commands that are not programmed.

**Example 1.**

```
#HSC ON [CONTERROR, E0.050]
.
#HSC ON [SURFACE]
(Chordal error = 0,050)
```

**Commands RE, SF and AXF.**

The CNC maintains the value of the commands programmed until a different one is programmed, the HSC mode is changed or canceled, a reset is done or the program ends.

When changing HSC modes, the CNC assumes the default values set in the machine parameters.

**Execute an HSC mode starting with initial conditions.**

To execute in HSC mode starting with initial conditions, first cancel the previous mode. See ["21.6 Canceling the HSC mode."](#) on page 410.

**21.**

**HSC. HIGH SPEED MACHINING.**  
HSC FAST mode. Optimizing the machining feedrate.

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 21.6 Canceling the HSC mode.

The HSC mode is canceled with the instruction #HSC OFF. HSC is also canceled when programming any of the functions G05, G07 or G50. Functions G60 and G61 do not cancel the HSC mode. Activating a second HSC mode does not cancel the previous HSC mode.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is:

```
#HSC OFF
```

```
#HSC OFF
```

### Influence of the reset, turning the CNC off and of the M30.

The HSC mode is canceled on power-up, after executing an M02 or M30 and after an emergency or reset.

# 21.

**HSC. HIGH SPEED MACHINING.**  
Canceling the HSC mode.



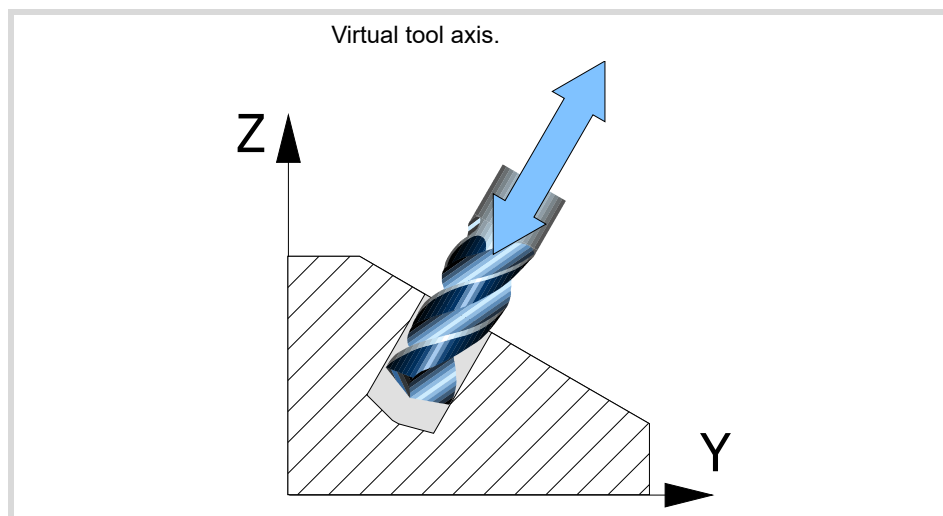
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

A virtual axis of the tool is a fictitious axis that always moves in the direction in which the tool is oriented. This virtual tool axis facilitates the movement in the tool direction when it is not aligned with the axes of the machine, but it is oriented in any other direction depending on the position of the bi-rotary or tri-rotary spindle.

This way and depending on the applied kinematics, it will move the corresponding X Y Z axes so the tool moves along its axis. This function facilitates drilling operations, withdrawing the tool in its direction as well as increasing or decreasing the depth of the pass while machining a part.



## Considerations about the virtual tool axis.

- There can be one virtual tool axis per channel.
- The virtual axis of the tool must be linear and it must belong to the channel. The virtual tool axis cannot be part of the main trihedron when it is active.
- The virtual tool axis, since it is an axis of the channel, may be moved like any other axis in the various work modes, automatic, jog, tool inspection, axis repositioning, etc.
- The virtual tool axis has travel limits that may be set by machine parameter and by program.

## 22.1 Activate the virtual tool axis.

The instruction #VIRTAX is used to activate the transformation of the virtual tool axis.

### Programming.

When defining this instruction, it is also possible to define the position value (coordinate) where the axis is located.

### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

```
#VIRTAX ON
#VIRTAX ON <[{}pos{}]>
{}pos{}          Optional. Axis position.
```

```
#VIRTAX
  Activate the transformation of the virtual tool axis, in its current position.
#VIRTAX ON
  Activate the transformation of the virtual tool axis, in its current position.
#VIRTAX ON [15]
  Activate the transformation of the virtual tool axis considering that it is positioned
  in position 15.
#VIRTAX [0]
  Activate the transformation of the virtual tool axis considering that it is positioned
  in position 0.
```

Programming the ON command is optional.

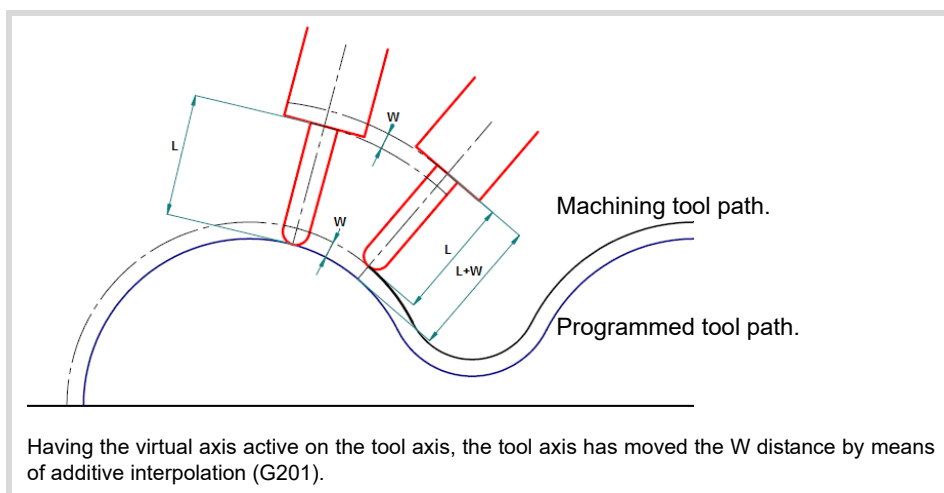
### Axis position.

This parameter may be used to activate the transformation of the virtual tool axis considering that it is positioned in a specific position. If 0 is programmed, the CNC considers that the virtual tool axis is positioned in the 0 position.

If the axis position is not programmed, the CNC activates the virtual axis taking its current position into account.

#### Example 1. Increase or decrease the machining pass depth while machining

Functions #VIRTAX and G201 are active in the program being executed. In this case, the virtual tool axis may be moved while the program is running.



# 22.

**VIRTUAL TOOL AXIS.**  
Activate the virtual tool axis.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Example 2. Increase or decrease the machining pass depth while machining**

Functions #VIRTAX and G201 are not active in the program being executed. The steps to change the machining pass are the following.

- (1) Interrupt program execution with the [STOP] key.
- (2) Go into tool inspection mode.
- (3) From MDI mode, execute #VIRTAX[0].
- (4) Move the axis the desired distance via MDI, jog, etc.
- (5) Resume execution without repositioning the axes.

## 22.2 Cancel the virtual tool axis.

The instruction #VIRTAX OFF cancels the transformation of the virtual tool axis. The way the virtual tool axis will behave after executing an M30 or after a reset depends on the setting of parameter VIRTAXCANCEL.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is:

#VIRTAX OFF

```
#VIRTAX OFF
```

22.

**VIRTUAL TOOL AXIS.**  
Cancel the virtual tool axis.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 22.3 Variables associated with the virtual tool axis.

The following variables may be accessed via part-program or via MDI/MDA mode. Each of them indicates whether it may be read (R) or written (W).

Variable.	R/W	Meaning.
(V.)[ch].G.VIRTAXIS	R	Logic number of the virtual tool axis.
(V.)[ch].G.VIRTAXST	R	Status of the virtual tool axis: (0) inactive / (1) active.
(V.)[ch].A.VIRTAXOF.xn	R	Distance traveled by the axis due to the movement of the virtual axis of the tool.

### Syntax of the variables.

- ch· Channel number.
- xn· Name, logic number or index of the axis.

V.[2].G.VIRTAXS	Channel ·2·.
V.A.VIRTAXOF.Z	Z axis.
V.A.VIRTAXOF.4	Axis with logic number ·4·.
V.[2].A.VIRTAXOF.1	Axis with index ·1· in the channel ·2·.

# 22.

## VIRTUAL TOOL AXIS.

Variables associated with the virtual tool axis.

The display of messages provides a simple way to follow the execution of a program, by showing messages at certain times during the program that the operator considers important, such as at the beginning of each operation, etc. The CNC only shows the last executed message. Messages, in combination with the function M0 (program interruption), are also useful when requesting the intervention of the operator.

The warning and error windows are a CNC resource for informing of undesired situations during program execution. All warnings and errors remain active, visible or minimized on the status bar until they are removed by the operator. Warning windows offer the possibility to continue program execution, whereas an error window stops the execution of the program.

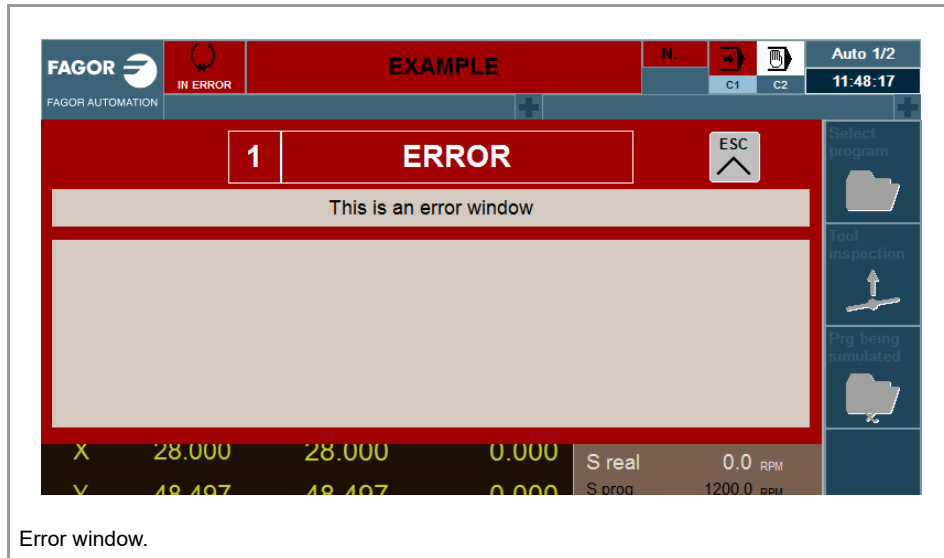
## Programming messages, warnings and errors.

The CNC provides the following instructions to display messages, warning windows or error windows. The table below shows a summary of the properties of each of the instructions.

Instruction.	Meaning and properties.
#MSG	<b>Show a message on the status bar.</b> <ul style="list-style-type: none"><li>• The CNC only displays the last executed message.</li><li>• The CNC does not interrupt program execution.</li><li>• An empty message, a CNC reset or when starting the execution of a program will remove the message.</li></ul>
#WARNING	<b>Shows a warning window.</b> <ul style="list-style-type: none"><li>• The CNC displays all executed warning windows.</li><li>• The CNC does not interrupt program execution.</li><li>• The [ESC] key closes the window.</li></ul>
#WARNINGSTOP	<b>Shows a warning window and interrupts the execution.</b> <ul style="list-style-type: none"><li>• The CNC displays all executed warning windows.</li><li>• The CNC interrupts program execution. The user can continue executing the program or stop it.</li><li>• The [ESC] key closes the window.</li></ul>
#ERROR	<b>Shows a message in a warning window.</b> <ul style="list-style-type: none"><li>• The CNC displays all executed error windows.</li><li>• The CNC interrupts program execution and enters an error state.</li><li>• The [ESC] key closes the window and CNC reset will remove the error status.</li></ul>

## 23.1 #ERROR. Display an error on the screen.

The #ERROR instruction interrupts program execution and displays the indicated error message. The error can be defined either by text or by a number that corresponds to the list of CNC, OEM or user errors. Errors and warnings are found on the same list; depending on the programmed instruction, the CNC will show a warning or an error.



### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#ERROR [{number}]
#ERROR [{"text"}]

{number}      Number of errors
               Units: -.

{text}        Error text.
               Units: -.

```

```
#ERROR [1254]
#ERROR [P100]
#ERROR [P10+34]
#ERROR ["Error text"]
```

#### Number of errors

The error number, which must be an integer, may be defined with a numerical constant, a parameter or an arithmetic expression. Local parameters must be programmed as P0, P1, etc.

The texts in the cncError.txt file, defined by the OEM or the user, can include up to 5 parameter values and variables using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the instruction #ERROR after the number and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#ERROR [10214, P20, V.G.FREAL]
      (Show 10214 error, defined in cncError.txt)
      (Substitute the first format identifier with the P20 value)
      (Substitute the first format identifier with the V.G.REAL value)
```

**23.** **DISPLAYING MESSAGES, WARNINGS AND ERRORS.** #ERROR. Display an error on the screen.



**Error text.**

The text must be defined between quote marks. If no text is defined, an empty window is displayed. The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined after the text and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#ERROR ["The current tool is %D", V.G.TOOL]
#ERROR [10214, V.G.TOOL, V.G.FREAL, P1]
(The 10214 error must be defined in the cncError.txt file)
```

**Considerations.****Fagor text and OEM/USER text.**

Errors and warnings between 0 and 9999 and between 23000 and 23999 are reserved by Fagor. The range of errors and warnings from 10000 to 20000 is available to the OEM and the user, so that other text can be created. See ["23.5 cncError.txt file. List of OEM and user errors and warnings."](#) on page 423.

**Format identifiers.**

See ["23.4 Format identifiers and special characters."](#) on page 422.

**23.****DISPLAYING MESSAGES, WARNINGS AND ERRORS.**

#ERROR. Display an error on the screen.

**FAGOR** 

FAGOR AUTOMATION

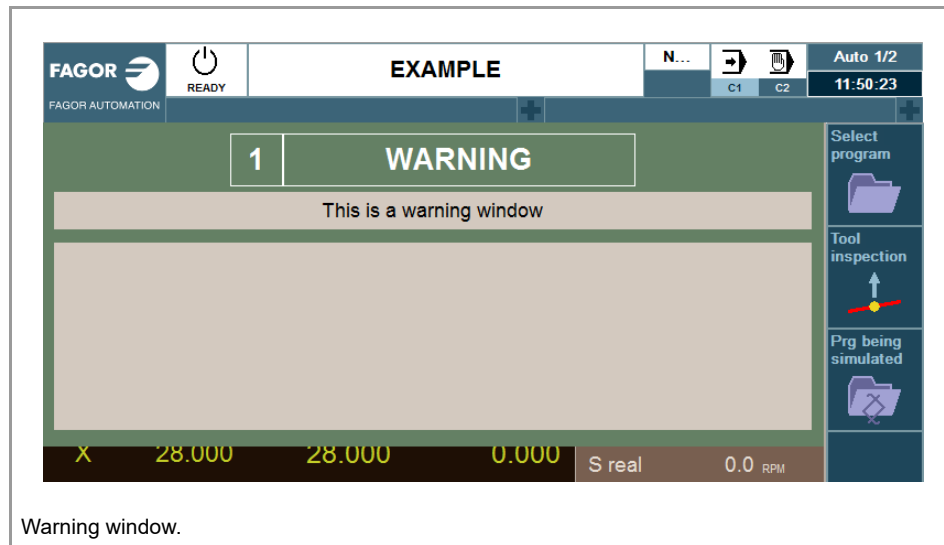
**CNC 8070**

(REF: 2008)

## 23.2 #WARNING / #WARNINGSTOP. Displaying a warning on the screen.

The display of warnings on the screen may be programmed using the instruction #WARNINGSTOP or #WARNING depending on whether the program execution is to be interrupted or not. For both cases, the CNC displays the warning during block preparation, and in the case of #WARNINGSTOP, the CNC interrupts the execution when the instruction is executed. The warning can be defined either by text or by a number that corresponds to the list of CNC, OEM or user errors. Errors and warnings are found on the same list; depending on the programmed instruction, the CNC will show a warning or an error.

Instruction.	Meaning.
#WARNING	Show a warning without interrupting the program execution.
#WARNINGSTOP	Show a warning and interrupt the program execution at the point where the instruction is located. The user decides whether to resume the execution at this point, with the [START] key, or abort the program, with the [RESET] key.



### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#WARNING [{number}]
#WARNING [{"text}"]
#WARNINGSTOP [{number}]
#WARNINGSTOP [{"text}"]
```

{number}          Warning number.  
Units: -.

{text}             Warning text.  
Units: -.

```
#WARNING [1254]
#WARNINGSTOP [1254]
#WARNING [P100]
#WARNINGSTOP [P100]
#WARNING [P10+34]
#WARNINGSTOP [P10+34]
#WARNING ["Warning text"]
#WARNINGSTOP ["Warning text"]
#WARNING [0]
(Clears all warnings on the screen)
```

# 23.

DISPLAYING MESSAGES, WARNINGS AND ERRORS.  
#WARNING / #WARNINGSTOP. Displaying a warning on the screen.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Warning number.**

The warning number, which must be an integer, may be defined using a numerical constant, a parameter or an arithmetic expression. Local parameters must be programmed as P0, P1, etc.

The texts in the cncError.txt file, defined by the OEM or the user, can include up to 5 parameter values and variables using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the call instruction (for example, #WARNING) after the number and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#WARNING [10214, P20, V.G.FREAL]
  (Show the 10214 warning, defined in cncError.txt)
  (Substitute the first format identifier with the P20 value)
  (Substitute the first format identifier with the V.G.REAL value)
```

**Warning text.**

The text must be defined between quote marks. If no text is defined, an empty window is displayed. The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined after the text and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#WARNING ["The current tool is %D", V.G.TOOL]
```

**Clears all displayed warnings.**

Programming a warning with a value of 0 clears all the warnings that are being displayed.

```
#WARNING [0]
  (Clears all warnings on the screen)
```

**Considerations.****Fagor text and OEM/USER text.**

Errors and warnings between 0 and 9999 and between 23000 and 23999 are reserved by Fagor. The range of errors and warnings from 10000 to 20000 is available to the OEM and the user, so that other text can be created. See ["23.5 cncError.txt file. List of OEM and user errors and warnings."](#) on page 423.

**Format identifiers.**

See ["23.4 Format identifiers and special characters."](#) on page 422.

**23.**

**DISPLAYING MESSAGES, WARNINGS AND ERRORS.**  
#WARNING / #WARNINGSTOP. Displaying a warning on the screen.

**FAGOR** 

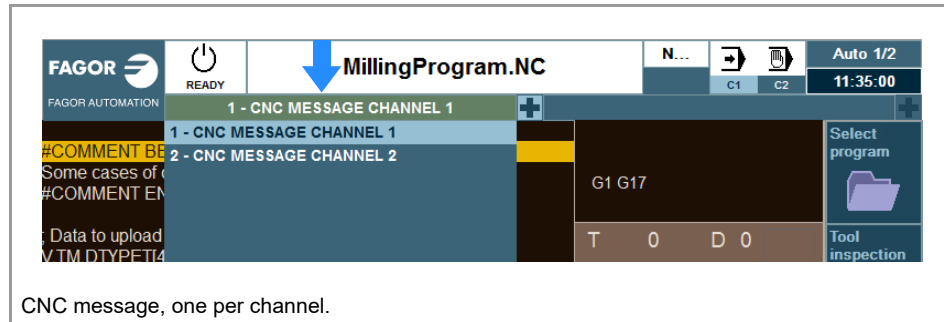
FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 23.3 #MSG. Display a message on the screen.

The instruction #MSG appears at the top of the screen and does not interrupt the execution of the program. The message will remain active until a new message is activated or it is cleared. The message can be defined either by text or by a number that corresponds to the list of OEM or user errors.



### Programming.

Program the instruction alone in the block.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#MSG [{number}]
#MSG [{"text"}]

{number}      Message number.
               Units: -.

{text}        Message text.
               Units: -.
```

```
#MSG ["User message"]
#MSG [100]
#MSG [P20]
#MSG [""]
      (Clear message)
```

#### Message number.

The message number, which must be an integer, may be defined with a numerical constant, a parameter or an arithmetic expression. Local parameters must be programmed as P0, P1, etc.

The texts in the cncMsg.txt file, defined by the OEM or the user, can include up to 5 parameter values and variables using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the instruction #MSG after the number and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#MSG [100, V.G.TOOL]
      (Show message number 100, defined in cncMsg.txt)
      (Substitute the first format identifier with the V.G.TOOL value)

#MSG [P10, P20]
      (Show message number P10, defined in cncMsg.txt)
      (Substitute the first format identifier with the P20 value)
```

# 23.

DISPLAYING MESSAGES, WARNINGS AND ERRORS.  
#MSG. Display a message on the screen.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Message text.**

The text must be defined between quote marks. If no text is defined, the message is erased from the screen. The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined after the text and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#MSG ["Part number %D", P2]
#MSG ["Finishing F=%D mm/min and S=%D RPM", P21, 1200]
#MSG ["Tool %u has expired", V.G.TOOL]
```

**Clears the displayed message.**

Programming an empty message clears the message on the screen. A reset or “end of program” will not clear the message on the screen)

```
#MSG ["" ]
(Clear message on the screen)
```

**23.****DISPLAYING MESSAGES, WARNINGS AND ERRORS.**

#MSG. Display a message on the screen.

## 23.4 Format identifiers and special characters.

### Format identifiers.

Writing text with %letter that is not found in this list or the following list, the CNC will include it as %letter.

Identifier.	Meaning.
%d %D	Integer or floating point number (with or without decimals). With this identifier, the number of integers and decimals to be displayed (default 5.5) can be defined; this format is defined between the symbol % and a letter; for example %5.5d.
%i	Integer in base 10 with positive or negative sign (int).
%u	Integer in base 10 without positive or negative sign (int).
%o	Integer in base 8 without positive or negative sign (int).
%x	Integer in base 16 using lowercase letters (int).
%X	Integer in base 16 using uppercase letters (int).
%f %F	Simple precision floating decimal (float). With this identifier, the number of integers and decimals to be displayed (default 5.5) can be defined; this format is defined between the symbol % and a letter; for example %5.5f.
%e	Scientific notation (coefficient / exponent), lower case (single precision or double precision).
%E	Scientific notation (coefficient / exponent), upper case (single precision or double precision).
%c	Write a character using its ASCII code (decimal number).
%s	Write a string (string of characters) from a string. This identifier can only be used with the variables (V.)A.AXISNAME.xn and (V.)A.SPDLNAME.sn.

### Special characters.

Writing text with %letter that is not found in this list or the previous list, the CNC will include it as %letter.

Identifier.	Meaning.
%%	Character %.
\"	Quotation marks.

```
#WARNING ["Difference between P12 and P14 > 40%%"]
#ERROR ["Parameter \"P100\" is wrong"]
#MSG ["The tool \"T1\" is a finishing tool"]
#MSG ["80%% of feedrate"]
#WARNING ["%s", V.A.AXISNAME.1]
      (Display the name of the axis).
#WARNING ["%c", 65]
      (Show character A, because 65 is its ASCII code).
```

# 23.

DISPLAYING MESSAGES, WARNINGS AND ERRORS.  
Format identifiers and special characters.

## 23.5 cncError.txt file. List of OEM and user errors and warnings.

The range of errors and warnings from 10000 to 20000 is available to the OEM and the user, so that other text can be created. These errors and warnings are saved in the cncError.txt file. Both the OEM and the user can create one of these files according to language.

### File location.

The CNC looks for the messages in the following order and displays the one it finds first. Therefore, it is recommended that the user does not define messages using the same number as the manufacturer.

```
..\Users\Data\Lang\{language}
..\Users\Data\Lang
..\Mtb\Data\Lang\{language}
..\Mtb\Data\Lang\English
..\Mtb\Data\Lang
```

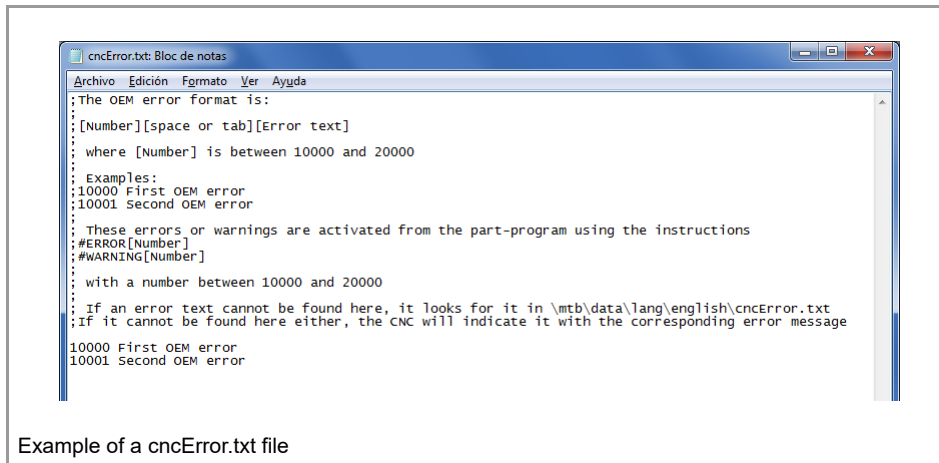
```
#WARNING [10032]
(Find 10032 message in cncError.txt file)
```

### File format.

In the folder ..\Mtb\Data\Lang\{idioma} there is an example of a cncError.txt file. The same file contains errors and warnings. The call type (#ERROR/#WARNING) determines whether the CNC displays an error or a warning.

- Any comments must start with the character ";".
- Errors follow the structure; number + space or tab key + text.

```
; Comment.
10000 First OEM or user error/warning.
10001 Second OEM or user error/warning.
```



### Format identifiers.

The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the warning or error call.

```
10002 The current tool is %D.
10003 Spindle speed %u excessive.
```

23.

DISPLAYING MESSAGES, WARNINGS AND ERRORS.  
cncError.txt file. List of OEM and user errors and warnings.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 23.6 cncMsg.txt file. List of OEM and user messages.

The file cncMsg.txt contains the OEM and user defined messages for the instruction #MSG. Both the OEM and the user can create one of these files according to language.

### File location.

The CNC looks for the messages in the following order and displays the one it finds first. Therefore, it is recommended that the user does not define messages using the same number as the manufacturer. If the message is not found in any file, the CNC will not show any message.

```
..\Users\Data\Lang\{language}
..\Users\Data\Lang
..\Mtb\Data\Lang\{language}
..\Mtb\Data\Lang
```

```
#MSG [1234]
(Find 1234 message in cncMsg.txt file)
```

### File format.

The file format will be:

- Any comments must start with the character ";".
- Errors follow the structure; number + space or tab key + text.

```
; Comment.
1 First message.
2 Second message.
3 Third message.
```

### Format identifiers.

The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the message call.

```
12 The current tool is %D.
13 Spindle speed %u excessive.
```

# 23.

DISPLAYING MESSAGES, WARNINGS AND ERRORS.  
cncMsg.txt file. List of OEM and user messages.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 23.7 Summary of the variables.

The following variables may be accessed from (PRG) the part-program and from the MDI/MDA mode, PLC and (INT) from an external application. For each variable, the table indicates whether the access is reading (R) or writing (W). PLC access to the variables, for both reading and writing, will be synchronous. Access to the variables from the part program returns the value for the block preparation (it does not stop the preparation), except when indicated otherwise.

Variables.	PRG	PLC	INT
<b>(V.)[ch].G.CNCERR</b> Error number of the highest priority. If several channels are in the same group, an error in a channel causes the same error in all of them; in this case, this variable will have the same value for all the channels of the group. Units: -.	R	R	R
<b>(V.)[ch].G.CNCWARNING</b> Displayed warning number. If there are several warnings, the value of the variable is updated as they are eliminated. When the last warning is eliminated, this variable is initialized to zero. Units: -.	R(*)	R	R
<b>(V.)[ch].A.AXISNAME.xn</b> Axis name. This variable can be used to include the name of the axis in the instructions #WRITE, #MSG, #WARNING, #WARNINGSTOP and #ERROR, while using the identifier %s. Units: -.	R	---	R
<b>(V.)[ch].A.SPDLNAME.sn</b> <b>(V.)[ch].SP.SPDLNAME.sn</b> Spindle name. This variable can be used to include the name of the spindle in the instructions #WRITE, #MSG, #WARNING, #WARNINGSTOP and #ERROR, while using the identifier %s. Units: -.	R	---	R

(\*) The CNC evaluates the variable during execution (it stops the block preparation).

This variable is initialized with a reset. If several channels are in the same group, the reset of a channel means resetting all of them, thus initializing the variables of all the channels of the group.

### Syntax of the variables.

- ch· Channel number.
- xn· Name, logic number or index of the axis.
- sn· Name, logic number or index of the spindle.

V.[2].G.CNCERR	
V.[2].G.CNCWARNING	
V.A.AXISNAME.4	Axis name with logic number ·4·.
V.[2].A.AXISNAME.1	Name of first axis on channel ·2·.
V.SP.SPDLNAME.1	Name of first spindle for spindle system.
V.[2].SP.SPDLNAME.1	Name of the first spindle on channel ·2·.
V.A.SPDLNAME.1	Name of spindle with logic number ·1·.
V.[2].A.SPDLNAME.1	Name of spindle with logic number ·1·.

# 23.

DISPLAYING MESSAGES, WARNINGS AND ERRORS.  
Summary of the variables.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 23.

## DISPLAYING MESSAGES, WARNINGS AND ERRORS.

Summary of the variables.



*The DMC is only available for the master spindle using Fagor digital servo systems. The spindle must be enabled for the DMC in the machine parameter DMCSDDL.*

The choice of machining feedrate depends on the material to be machined, the tool (material, number of teeth, etc.) and the depth of pass. The feedrate for machining is set at the start and remains constant until the end. If the machining conditions change (for example, due to tool wear), the programmed feedrate may no longer be appropriate, which will have a negative effect on the tool life, machining time, etc.

DMC (dynamic machining control) adapts the feedrate during machining to maintain the cutting power as close as possible to ideal machining conditions. DMC adapts the feedrate by changing the override.

DMC optimizes the use of the machine and the tool, which makes it possible to increase the material removal rate (MRR) without impairing the life of the tool, since it is operating according to its rated conditions. Optimization is based on the cutting power to be reached, which is also referred to as target power. This power not only depends on the tool, but also on the material and the cutting conditions (feedrate, spindle turning speed, depth of pass and sideways step (pass)), and therefore its value must be linked to the entire combination of the tool+material+cutting conditions.

DMC is only available for milling operations with "Milling" and "Surface milling" operations. This function can be applied to roughing and finishing operations, but this function provides the most benefits in terms of machining time and tool life in roughing operations.

## 24.1 Activating the DMC.

The instruction #DMC ON activates DMC, always on the master spindle. DMC is only available for milling operations with "Milling" and "Surface milling" operations. DMC can be applied to roughing and finishing operations, but this function provides the most benefits in terms of machining time and tool life in roughing operations.

### Programming.

This instruction must be programmed alone in the block. All commands are optional when defining this instruction.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones, between angle brackets.

```
#DMC ON [<PWRSP {power}> <, OVRMIN{<percent>> <, OVRMAX{<percent>> <, FZMIN{feed}> <, FZMAX{feed}>]
```

PWRSP{power}	Target power or ideal cutting power is defined as the percentage of the rated power of the spindle. <ul style="list-style-type: none"> <li>Optional; if not programmed, it is calculated by the CNC.</li> <li>Values: 0 — 100 %.</li> </ul>
OVRMIN{<percent>}	Minimum override permitted for DMC. <ul style="list-style-type: none"> <li>Optional (by default, the machine parameter value MINDMCOVR).</li> <li>Values: 10 — 100 %.</li> </ul>
OVRMAX{<percent>}	Maximum override permitted for DMC. <ul style="list-style-type: none"> <li>Optional (by default, the machine parameter value MAXDMCOVR).</li> <li>Values: 100 — 255 %.</li> </ul>
FZMIN{feed}	Minimum feedrate per tooth permitted during DMC. <ul style="list-style-type: none"> <li>Optional; if not programmed or FZMIN &gt; FZMAX, the CNC does not monitor the minimum feedrate per tooth.</li> <li>Values: 0 — 99999.9999 mm/tooth 0 — 3937.00787 inches/tooth.</li> </ul>
FZMAX{feed}	Maximum feedrate per tooth permitted during DMC. <ul style="list-style-type: none"> <li>Optional; if not programmed or FZMIN &gt; FZMAX, the CNC does not monitor the maximum feedrate per tooth.</li> <li>Values: 0 — 99999.9999 mm/tooth 0 — 3937.00787 inches/tooth.</li> </ul>

```
#DMC ON
(The CNC activates DMC with the default values.)
(DMC initiates the learning phase to calculate the target power.)

#DMC ON [PWRSP 80, OVRMIN 90, OVRMAX 110, FZMIN 0.8, FZMAX 1.3]
(The CNC activates DMC with the programmed values.)

#DMC ON [OVRMIN 90, OVRMAX 110, FZMIN 0.8, FZMAX 1.3]
(The CNC activates DMC with the programmed values.)
(DMC initiates the learning phase to calculate the target power.)
```

#### Target power or ideal cutting power.

The target power is programmed as a percentage of the rated power of the spindle. The programming of the target power is optional; if it is not programmed, the CNC conducts a learning phase to determine it. See "[24.4.1 DMC operation.](#)" on page 433.

#### Feed per tooth.

The feedrate modified with the override obeys the minimum and maximum feedrate per tooth set for the tool. So that the CNC can oversee the feedrate per tooth, the number of teeth on the tool must be determined from the tool table.

# 24.

DMC (DYNAMIC MACHINING CONTROL).  
Activating the DMC.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Considerations and limitations.

### Work modes.

- DMC can only be active when running; in a simulation, the CNC will analyze the activation and deactivation instructions, but will not start up DMC.
- DMC can only be active in automatic mode; it is not activated in manual mode. The DMC deactivates when entering into tool inspection and resumes operation after the axes have been repositioned.
- DMC is incompatible with some types of operations, such as threading, drilling, etc. In addition, DMC will be automatically deactivated if the active tool is not a "Milling" or "Surface milling" type.

### Tool change or offset.

The following tool-related actions deactivate DMC. The user is responsible for reprogramming DMC with the appropriate values for the new tool.

- Changing the tool.
- Changing the offset.
- Executing an M6.
- Changing any of the offset characteristics that affect the power consumption of the spindle (by writing the corresponding variable).

### Tool inspection.



When entering the tool inspection mode, DMC is deactivated. At the end of the tool inspection, the DMC is automatically activated and allows learning phase may be repeated by pressing the "DMC learning" softkey.

### DMC behavior after being deactivated.

The tool paths in G0, axis stops using [STOP] and the softkey "Set DMC off" temporarily deactivate the DMC. In these cases, DMC acts as follows:

- If DMC has not begun to measure the no-load power, the process will not begin.
- If DMC is waiting for the spindle to reach the programmed speed or is measuring the no-load power, DMC is deactivated when it finishes measuring the no-load power.
- During the learning phase, DMC does not consider the time that the cause of the deactivation persists as learning time.
- During the execution without a load, DMC is deactivated, but it will continue to detect entries into the part.
- If the tool is starting to enter the part, DMC will be deactivated once it has entered.
- If the tool is inside the part, DMC is deactivated, but it will continue to detect the part exits and excessive power consumption.
- If the tool is exiting from the part, DMC is deactivated once the exit is completed.

# 24.

DMC (DYNAMIC MACHINING CONTROL).  
Activating the DMC.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 24.2 Deactivating the DMC.

The #DMC°OFF instruction deactivates DMC. The M02 or M30 (end of program) functions and the reset also deactivate DMC. A spindle stop, function M5, deactivates DMC.

### Programming.

This instruction must be programmed alone in the block. This instruction has no commands.

### Programming format.

The programming format is:

#DMC OFF

```
#DMC OFF
(The CNC deactivates DMC).
```

# 24.

DMC (DYNAMIC MACHINING CONTROL).

Deactivating the DMC.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 24.3 Summary of the variables.

The following variables may be accessed from (PRG) the part-program and from the MDI/MDA mode, PLC and (INT) from an external application. For each variable, the table indicates whether the access is reading (R) or writing (W). PLC access to the variables, for both reading and writing, will be synchronous. Access to the variables from the part program returns the value for the block preparation (it does not stop the preparation), except when indicated otherwise.

Variables.	PRG	PLC	INT
<b>(V.)[ch].MPG.MINDMCOVR</b> Minimum DMC override of all channel axes. Units: Percentage.	R	R	R
<b>(V.)[ch].MPG.MAXDMCOVR</b> Maximum DMC override of all channel axes. Units: Percentage.	R	R	R
<b>(V.)[ch].MPA.DMCSPDL.sn</b> Spindle with enabled power control. Units: -.	R	R	R
<b>(V.)[ch].G.FRO</b> Active feedrate (feed override) percentage. Units: Percentage.	R(*)	R	R
<b>(V.)[ch].G.PRGFRO</b> Feedrate (feed override) percentage defined by the program. Units: Percentage.	R/W	R	R
<b>(V.)[ch].PLC.FRO</b> Feedrate (feed override) percentage defined by the PLC. Units: Percentage.	R(*)	R/W	R
<b>(V.)[ch].G.CNCFRO</b> Feedrate percentage (feed override) selected on the switch of the operator panel. Units: Percentage.	R(*)	R	R/W
<b>(V.)[ch].G.NCUTTERS</b> Tool being prepared. Number of teeth. Units: Teeth.	R/W	R	R
<b>(V.)[ch].TM.NCUTTERS[offset]</b> [Offset] of the active tool. Number of teeth. Units: Teeth.	R/W(*)	R/W	R/W
<b>(V.)TM.NCUTTERST[tool][offset]</b> [Offset] of the [tool]. Number of teeth. Units: Teeth.	R/W(*)	R/W	R/W
<b>(V.)[ch].G.DMCPWRSP</b> Target power, defined as a percentage of the rated power of the spindle. Value programmed in the PWRSP command of the #DMC ON instruction. Units: Percentage.	R	R	R
<b>(V.)[ch].G.DMCOVRMIN</b> Minimum feedrate (feed override) percentage allowed for DMC. Value programmed in the OVRMIN command of the #DMC ON instruction. Units: Percentage.	R	R	R
<b>(V.)[ch].G.DMCOVRMAX</b> Maximum feedrate (feed override) percentage allowed for DMC. Value programmed in the OVRMAX command of the #DMC ON instruction. Units: Percentage.	R	R	R
<b>(V.)[ch].G.DMCFZMIN</b> Minimum feedrate per tooth permitted during DMC. Value programmed in the FZMIN command of the #DMC ON instruction. Units: Millimeters/tooth or inches/tooth.	R	R	R

(\*) The CNC evaluates the variable during execution (it stops the block preparation).

# 24.

DMC (DYNAMIC MACHINING CONTROL).  
Summary of the variables.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 24.

**DMC (DYNAMIC MACHINING CONTROL).**

Summary of the variables.

Variables.	PRG	PLC	INT
<b>(V.)[ch].G.DMCFZMAX</b> Maximum feedrate per tooth permitted during DMC. Value programmed in the FZMAX command of the #DMC ON instruction. Units: Millimeters/tooth or inches/tooth.	R	R	R
<b>(V.)[ch].G.DMCON</b> DMC status. Units: -.	R(*)	R	R
<b>(V.)[ch].G.LEARNEDPWRSP</b> Target power calculated for DMC in the learning phase (percentage of rated power). Units: Percentage.	R(*)	R	R
<b>(V.)[ch].G.DMCACTPWR</b> Active power on the spindle, measured by DMC (percentage of the rated power). Units: Percentage.	R(*)	R	R
<b>(V.)[ch].G.DMCOVR</b> Feedrate percentage (feed override) calculated by DMC. Units: Percentage.	R(*)	R	R
<b>(V.)[ch].G.DMCFZ</b> Feedrate per tooth, calculated by DMC. Units: Millimeters/tooth or inches/tooth.	R(*)	R	R
<b>(V.)[ch].G.DMCNOLOADPWR</b> Spindle power without a load, measured by DMC. Units: Kilowatts.	R(*)	R	R
<b>(V.)[ch].G.DMCSAVEDTIME</b> Time saved by the DMC action. Units: Seconds.	R(*)	R	R

(\*) The CNC evaluates the variable during execution (it stops the block preparation).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 24.4 Operating with DMC.

### 24.4.1 DMC operation.

After executing the instruction #DMC ON, the CNC will activate DMC as long as the spindle has a digital drive system, is turning in M3 or M4 and has reached the programmed turning speed (REVOK mark).

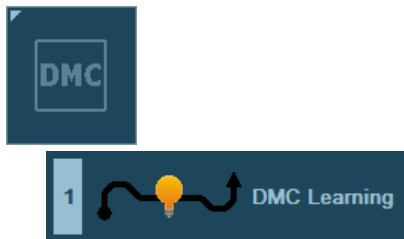
#### Calculating the no-load power.

The first time that this function is executed, the CNC stops the feedrate of the axes until the spindle reaches the programmed turning speed and has stabilized. Next, and with the axes stopped, the CNC measure the power consumed by the spindle without a load (without machining). The entire process to measure the no-load power can be completed in just a few seconds, during which time the CNC stops the feedrate of the axes.

Knowing the no-load power enables the CNC to detect the part entries and exits during the machining.

#### Learning phase.

Every time #DMC ON is programmed without target power (PWRSP command), DMC determines it through a learning phase that it initiates automatically. Once said value is obtained, the normal DMC operation will begin.



The learning phase can be repeated at any time, while the DMC is active, by pressing the "DMC learning" softkey in the automatic mode. After pressing the softkey, while the DMC is active, the next entry into the part will begin a learning phase, whether or not the target power had been programmed while activating the DMC.

With the axes in movement, the learning phase begins once DMC detects the entry into the part. DMC waits for the feedrate to reach the programmed value, and during the movement of the axes, it calculates the target power ("power consumed" – "no-load power"). The learning phase has a one minute duration, starting from when the tool enters the part at a distance equal to the radius. If the tool exits the part, it ceases counting the time until the tool re-enters the piece.

DMC will write the target power value obtained in the learning phase in the variable (V.)G.LEARNEDPWRSP, so that it can be used on the following parts that are machined with the same part program, avoiding the repetition of the learning phase.

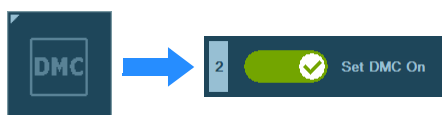
It is recommended to perform the learning phase with a depth of pass as close as possible to that which will be used during the machining; otherwise, if the depth of pass is shallower, the target power calculated may not be appropriate.

#### Deactivating the DMC.

The softkey menu allows the DMC to be deactivated.



The DMC can be activated from the same softkey menu.



## DMC operation.

Once the target power is known, and after detecting the entry into the part, DMC will begin to function as normal. During the machining, DMC adapts the feedrate so that the cutting power ("power consumed" – "no-load power") is as close as possible to the target power. DMC adapts the feedrate by changing the override. See "[24.4.3 Percentage of feedrate \(feedrate override\).](#)" on page 435.

### Processing of part entries and exits.

DMC detects the part entries and exits and carries out a special feedrate override process so that said transitions are smooth and do not damage the tool. For entries into the part, DMC use a 75% feedrate until the tool enters the part along a length equal to the tool radius. The CNC also attempts to optimize the time on the paths without a load.

### The target power and the part entries and exits.

To ensure the proper operation of the DMC, the target power (programmed or obtained by learning) must be at least 20% of the no-load power. If the DMC detects this type of situation, it will display a 3103 warning.

- If the actual values of the no-load power and the target power during machining are similar, it is likely that the DMC cannot accurately distinguish the part inputs and outputs or it may detect false inputs or outputs. In this case, it is recommended to check the value of the target power.
- Programming a higher target power, which is not actually going to be achieved, can mean that the DMC never detects part inputs and performs all the machining as if the tool were working in a vacuum.

For both these reasons, if the actual target power is less than 20% of the no-load power, it is recommended to deactivate the DMC function during this machining process.

## Monitoring of consumed power.

During operation, DMC continuously monitors the power consumed by the spindle to detect any problems with the tool or in the machining.

### Collision detection.

If the instantaneous power exceeds the target power in a predefined range, the CNC considers that a collision has occurred. In this case, the CNC displays warning 3101, stops the feedrate of the axes, while keeping the spindle turning (this behavior is the equivalent of having pressed the [STOP] button). After verifying the cause of the warning, the user can continue the machining (pressing [START]), enter in tool inspection or terminate the execution to replace the damaged tool.

### Worn tool. Excessive continuous power consumption.

If DMC detects excessive power consumption during a certain amount of time, it understands that the tool is worn or deteriorated and displays the warning 3100, without stopping the execution. The user decides if it is necessary to stop the execution.

# 24.

DMC (DYNAMIC MACHINING CONTROL).  
Operating with DMC.



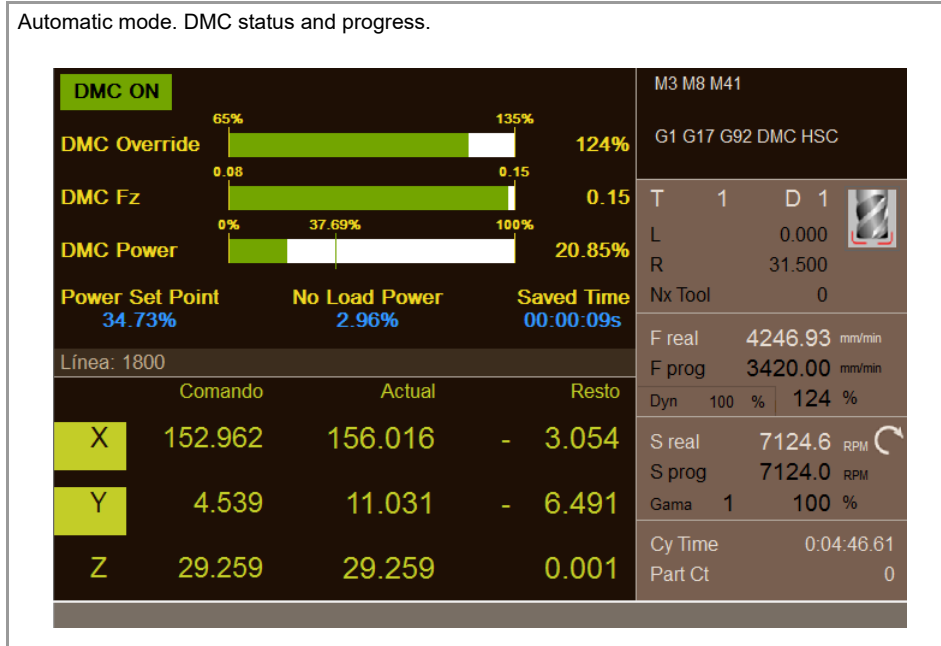
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### 24.4.2 DMC status and progress. Automatic mode.

During the execution of a program while DMC is active, the automatic mode can display the status and progress of this function; to do this, on the "Display" softkey of the horizontal menu, select the option "DMC". To return to the standard screen in automatic mode, select the "Standard" option on the same softkey.



# 24.

DMC (DYNAMIC MACHINING CONTROL).  
Operating with DMC.

### 24.4.3 Percentage of feedrate (feedrate override).

The feedrate (feed override) percentage can be set, from highest to lowest priority, by the program (variable V.G.PRGFRO), by the PLC (variable V.PLC.FRO) or from the switch on the operator panel. The value set by the program has the highest priority, while the values set from the switch on the operator panel have the lowest priority.

- The percentage set by the program or by the PLC has a higher priority than that set by DMC; both percentages inhibit DMC. To cancel the feedrate percentage set by program or PLC, define its variables with a value of 0 (zero).
- The percentage set by DMC affects the percentage set from the switch.

$$\text{Override}(\%) = \frac{\text{OverrideDMC}(\%) \times \text{OverrideJOG}(\%)}{100}$$

- DMC can change its override within the limits set by the instruction #DMC ON (commands OVRMIN and OVRMAX); if they are not defined, the limits will be set by the machine parameters MINDMCOVR and MAXDMCOVR. The final override (overrideDMC + overrideJOG) may exceed these limits.
- The CNC always abides by the maximum limit set in the machine parameter MAXOVR.
- If the user selects an override with the switch on the operator panel that is lower than MINDMCOVR, the CNC inhibits DMC (but does not deactivate it); when the override once again exceeds MINDMCOVR, DMC will once again operate as normal.
- The CNC abides by the minimum and maximum feedrate set by the tool in the instruction #DMC ON (commands FZMIN and FZMAX).



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

# 24.

## DMC (DYNAMIC MACHINING CONTROL).

Operating with DMC.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 25.1 #OPEN. Open file for writing.

The instruction #OPEN opens a file so that the part-program (#WRITE) can be written. The file must have write permission, otherwise the CNC will show the corresponding error. The execution and the simulation write on the same file.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#OPEN [{"file"}, <A/D>, <F{IdOfFile}>, <KEEPLINE>, <TYPE{format}>]
```

{“file”} Name of file and its path (optional). If the path is not programmed, the CNC saves the file in the same folder as the program executing it.

A/D Optional. Default access mode is D.  
A (APPEND): Add file  
D (DELETE): Clears content and write from scratch.

F{IdOfFile} Optional. File identifier (F1 to F4) for multi-channel access. From one channel it is possible to write to a file from any other channel when its identifier is determined. If no identifier is programmed, only the channel opening it can write to the file.

KEEPLINE Optional. Does not add a line break after each sentence. If it is not programmed, each instruction will create a line break.

TYPE{format} Optional. File format; by default UCS-2LE BOM.  
1 = COD\_ANSI  
2 = UCS-2LE BOM

```
#OPEN ["FileForWrite.txt", A, F2, KEEPLINE]  
#OPEN ["FileForWrite.txt", D]  
#OPEN ["FileForWrite.txt"]  
#OPEN ["FileForWrite.txt", A, F2, KEEPLINE, TYPE 2]
```

### File path and name.

Programming the path is optional. The path and the file name must be defined between inverted commas. The file name cannot contain any of the following characters: \ / : \* ? " < > |.

### File identifier for multi-channel access.

The F1 to F4 identifier allows for a channel to write to a file opened in an other channel when the identifier is determined. If no identifier is programmed, only the channel opening it can write to the file. If a file with a previously used identifier is opened, the CNC will show the corresponding error.

**Access mode.**

- The "D" (DELETE) option deletes the file and creates a new one in the format specified by the TYPE command (by default, UCS-2LE BOM).
- The "A" (APPEND) option adds content to an existing file. If the file does not exist, create a new one in the format specified by the command TYPE (by default, UCS-2LE BOM). If the file already exists, the command TYPE must match the file format.

**Line feed.**

When the command KEEPLINE is programmed, the line break manages the text of the #WRITE instruction using the \n identifier.

```
#OPEN ["FileForWrite.txt", A, KEEPLINE]
#WRITE ["Message \n"]
(The instruction #WRITE inserts a line break)
```

When the command KEEPLINE is not programmed, each #WRITE instruction will create a line break. If the identifier \n is appended to the text, two line breaks are inserted.

```
#OPEN ["FileForWrite.txt", A]
#WRITE ["Mensaje"]
(The instruction #WRITE inserts a line break)
#WRITE ["Message \n"]
(The instruction #WRITE inserts two line breaks)
```

**File format (command TYPE).**

COD_ANSI	Non-Unicode format, for files that are to be used in applications that do not support Unicode format.
UCS-2LE BOM	Unicode format (recommended).

**25.**

**OPENING AND WRITING FILES.**  
#OPEN. Open file for writing.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 25.2 #WRITE. Writing in a file.

The instruction #WRITE inserts text into the file opened with the instruction #OPEN. From one channel it is possible to write to a file from any other channel when its identifier is determined (comando "F"). The instruction is performed during the execution, but the CNC does not wait for the instruction to finish before continuing with the execution. The instructions will be stored until an error occurs or M30 is executed.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#WRITE [<F{IdOfFile}>, > "{Message}"/{IdOfMessage}, <{var1}>, >...<{var5}>]
```

F{IdOfFile}	Optional. File identifier (F1 to F4) for multi-channel access. If an identifier is not programmed, the program writes in the open file using an #OPEN instruction without the "F" file identifier in the channel executing the #WRITE instruction.
"{Message}" {IdOfMessage}	Message or message number predefined by the cncWrite.txt file. Both messages allow format identifiers.
{var1}...{var5}	Optional. Variables or parameters whose value is to be inserted in the message text (replaces the format identifiers).

```
#WRITE ["%d value incorrect", P21]
(Write in the file that was opened in the channel using #OPEN without an "F" file identifier)
(The line break depends on the programming of #OPEN instruction, with or without KEEPLINE instruction)
(The identifier %d is resolved with the P21 value)

#WRITE [F2, "Tool %u worn down\n", V.G.TOOL]
(Write #OPEN in the open file [F2])
(The format identifier \n inserts a line break) (Depending on the #OPEN programming, with or without the KEEPLINE instruction, there may be another line break)
(The identifier %u is addressed with the V.G.TOOL value)

#WRITE [F2, 10214, V.G.TOOL]
(Write #OPEN in the open file [F2])
(The line break depends on the programming of #OPEN instruction, with or without KEEPLINE instruction)
(Write the instruction 10214, defined in the cncWrite.txt file)
(The identifier %d of the 10214 text is addressed with the V.G.TOOL value)

#WRITE [F2, "%s = %d", V.A.AXISNAME.1, P100]
(Write #OPEN in the open file [F2])
(The line break depends on the programming of #OPEN instruction, with or without KEEPLINE instruction)
(The identifiers %s and %d are resolved with the values V.A.AXISNAME.1 and P100)
```

### Message number.

The error number, which must be an integer, may be defined with a numerical constant, a parameter or an arithmetic expression. Local parameters must be programmed as P0, P1, etc.

The texts in the cncWrite.txt file, defined by the OEM or the user, can include up to 5 parameter values and variables using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the instruction #WRITE after the number and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#WRITE [123, P20, V.G.FREAL]
(Write the instruction 123, defined in the cncWrite.txt file)
(Substitute the first format identifier with the P20 value)
(Substitute the first format identifier with the V.G.REAL value)
```

# 25.

OPENING AND WRITING FILES.  
#WRITE. Writing in a file.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Error text.**

The text must be defined between quote marks. The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined after the text and separated by commas. Up to five format identifiers can be defined for each message, and there must be as many variables or data parameters as identifiers.

```
#WRITE ["The current tool is %D", V.G.TOOL]
```

**Format identifiers and special characters.**

**Format identifiers.**

Writing text with %letter that is not found in this list or the following list, the CNC will include it as %letter.

Identifier.	Meaning.
%d %D	Integer or floating point number (with or without decimals). With this identifier, the number of integers and decimals to be displayed (default 5.5) can be defined; this format is defined between the symbol % and a letter; for example %5.5d.
%i	Integer in base 10 with positive or negative sign (int).
%u	Integer in base 10 without positive or negative sign (int).
%o	Integer in base 8 without positive or negative sign (int).
%x	Integer in base 16 using lowercase letters (int).
%X	Integer in base 16 using uppercase letters (int).
%f %F	Simple precision floating decimal (float). With this identifier, the number of integers and decimals to be displayed (default 5.5) can be defined; this format is defined between the symbol % and a letter; for example %5.5f.
%e	Scientific notation (coefficient / exponent), lower case (single precision or double precision).
%E	Scientific notation (coefficient / exponent), upper case (single precision or double precision).
%c	Write a character using its ASCII code (decimal number).
%s	Write a string (string of characters) from a string. This identifier can only be used with the variables (V.)A.AXISNAME.xn and (V.)A.SPDLNAME.sn.

The identifier %s can only be used with the variables (V.)A.AXISNAME.xn and (V.)A.SPDLNAME.sn.

**Special characters.**

If you write a text with %letter that is not included in this list or the previous one, the CNC will include it as %letter.

Identifier.	Meaning.
%%	Character %.
\"	Quotation marks.
\n	Line feed.

```
#WRITE ["Difference between P12 and P14 > 40%%"]
#WRITE ["Parameter \"P100\" is wrong"]
#WRITE ["Message with line break \n"]
#WRITE ["%s", V.A.AXISNAME.1]
    (Write the name of the axis).
#WRITE ["%c", 65]
    (Show the character A, because 65 is its ASCII code).
```

**25.**  
**OPENING AND WRITING FILES.**  
#WRITE. Writing in a file.



CNC 8070

(REF: 2008)



## 25.3 #CLOSE. Close a file.

The instruction #CLOSE closes the file opened with the instruction #OPEN.

### Programming.

Program the instruction alone in the block.

### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets.

```
#CLOSE [<F{IdOfFile}>]
```

F{IdOfFile}      Optional. File identifier (F1 to F4) for multi-channel access. If an identifier is not programmed, the instruction closes the file that opened in the channel using #OPEN without an "F" file identifier.

```
#CLOSE
#CLOSE [F2]
```

### Considerations.

- If the file has been opened without an identifier (F1 to F4), function M30 closes one from the file.
- If the file has been opened with an identifier (F1 to F4), the function M30 does not close the file to allow multi-channel management.
- If an error occurs in the channel opening the file, it will be closed with a reset.
- Closing the CNC will close all the open files.

# 25.

OPENING AND WRITING FILES.

#CLOSE. Close a file.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 25.4 cncWrite.txt file. List of OEM and user messages.

The file cncWrite.txt contains the OEM and user defined messages for the instruction #WRITE. Both the OEM and the user can create one of these files according to language.

### File location.

The CNC can have two cncWrite.txt files: that created by the OEM and another created by the user. The CNC looks for the messages in both files, in the following order, and displays the message it finds first. Therefore, it is recommended that the user does not define messages using the same number as the manufacturer. If the message is not found in any file, the CNC will not show any error.

```
..\Users\Data\Lang\{language}
..\Users\Data\Lang
..\Mtb\Data\Lang\{language}
..\Mtb\Data\Lang
```

### File format.

The file format will be:

- Any comments must start with the character ";".
- Errors follow the structure; number + space or tab key + text.

```
; Comment.
1 First message.
2 Second message.
3 Third message.
```

### Format identifiers.

The text can include 5 parameter values and variables in the message using the format identifiers (%D, %i, %u, etc). The variables or parameters whose value is to be displayed must be defined in the message call.

```
12 The current tool is %D.
13 Spindle speed %u excessive.
```

# 25.

OPENING AND WRITING FILES.  
cncWrite.txt file. List of OEM and user messages.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 26.1 Display instructions. Define the size of the graphics area

The instruction #DGWZ may be used to define cylindrical or prismatic parts at both CNC models. The defined parts are kept until a new one is defined, they are changed or the CNC is turned off. Up to four different parts may be programmed and each one may be assigned to several channels at the same time.

Instruction.	·M· model.	·T· model.	Dual-purpose machine.
#DGWZ	Prismatic part.	Cylindrical part.	(*)
#DGWZ RECT	Prismatic part.	Prismatic part.	Prismatic part.
#DGWZ CYL	Cylindrical part.	Cylindrical part.	Cylindrical part.

(\*) On an ·M· model, having the dual-purpose machine option, the instruction draws a prismatic part. On an ·T· model, having the dual-purpose machine option, the instruction draws a cylindrical part.

The origins for the parts will be the ones defined in the execution channel.

### Programming.

When programming this instruction, it is necessary to define the size of the part and optionally the part number and the channels it is associated with. Both parameters, part number and channels may be programmed in any order.

#### Programming format (1). Defining a prismatic part.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets. The RECT command may be left out at the mill model.

```
#DGWZ <RECT> [{Xmin},{Xmax},{Ymin},{Ymax},{Zmin},{Zmax}] <P{1-4}>
<C{1-4}>..<C{1-4}>
```

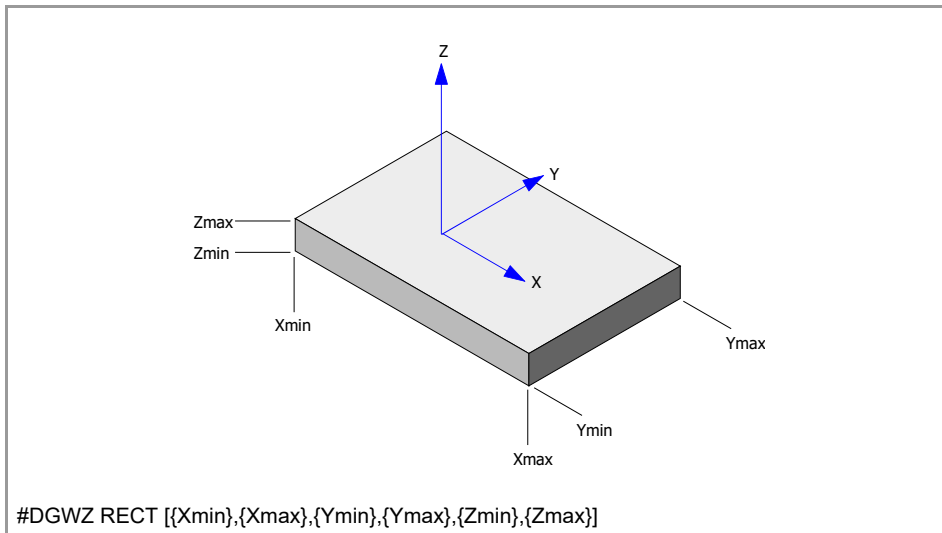
- <RECT> Optional at the mill model. Prismatic part.
- {Xmin}{Xmax} Minimum and maximum limit on the first axis of the channel.
- {Ymin}{Ymax} Minimum and maximum limit on the second axis of the channel.
- {Zmin}{Zmax} Minimum and maximum limit on the third axis of the channel.
- <P{1-4}> Optional. Part number (between 1 and 4).
- <C{1-4}> Optional. Number of channel associated with the part (between 1 and 4). The instruction lets associate several channels to the same part in any order.

```
#DGWZ [-10, 100, -15, 40, 0, 20]
    (It can only be programmed at the ·M· model)
#DGWZ RECT [-10, 100, -15, 40, 0, 20]
#DGWZ RECT [-10, 100, -15, 40, 0, 20] P1 C1 C2
#DGWZ RECT [-10, 100, -15, 40, 0, 20] C2 P1 C1 C3
```

# 26.

**PROGRAMMING STATEMENTS.**

Display instructions. Define the size of the graphics area



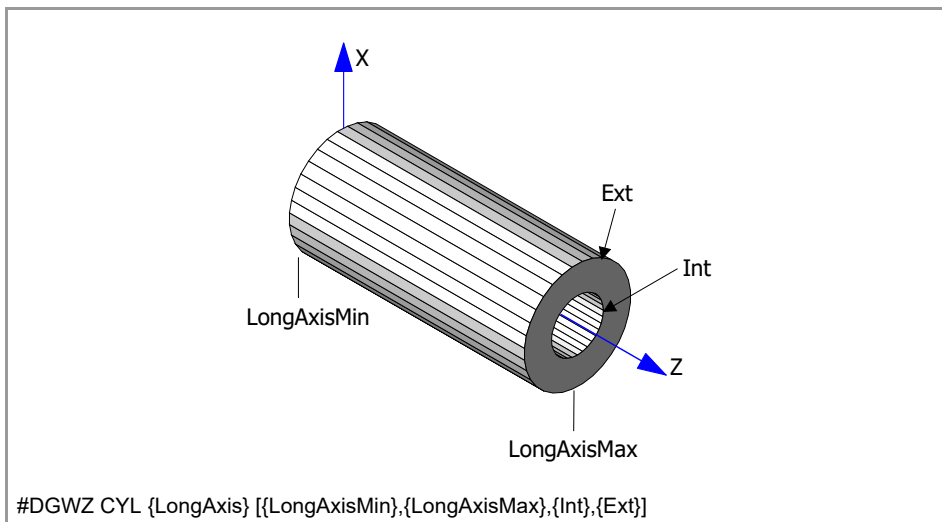
**Programming format (2). Defining a cylindrical part.**

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets. The CYL command may be left out at the lathe model.

```
#DGWZ <CYL> {LongAxis} [{LongAxisMin},{LongAxisMax},{Int},{Ext}] <P{1-4}>
<C{1-4}>..<C{1-4}>
```

- <CYL>           Optional at the lathe model. Cylindrical part.
- {LongAxis}       Longitudinal axis of the cylinder.
- {LongAxisMin}    Minimum and maximum limit on the longitudinal axis.
- {LongAxisMax}
- {Int}{Ext}        Inside and outside radius/diameter. The value will be in radius or diameter depending on machine parameter DIAMPROG and the active function G151/G152.
- <P{1-4}>         Optional. Part number (between 1 and 4).
- <C{1-4}>         Optional. Number of channel associated with the part (between 1 and 4). The instruction lets associate several channels to the same part in any order.

```
#DGWZ [-100, 0, 0, 40]
  (It can only be programmed at the ·T· model)
#DGWZ CYL Z [-100, 0, 0, 40]
#DGWZ CYL Z [-100, 0, 0, 40] P1 C1 C2
#DGWZ CYL Z [-100, 0, 0, 40] C1 C4 P1 C2
```



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**Part number and channel number.**

The graphics may represent up to 4 parts at the same time and each one of them may be associated with one or several channels. The part zeros are always associated with the execution channel.

#DGWZ CYL/RECT [...]

Modify or create the part with the same number as that of the execution channel and associated with the execution channel. For example, from channel 1, it will be part P1 associated with channel C1; from channel 2, it will be part P2 associated with C2, etc.

#DGWZ CYL/RECT [...] Pn Cm

Modify or create part Pn associated with channel Cm.

#DGWZ CYL/RECT [...] Pn

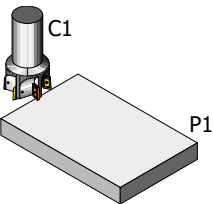
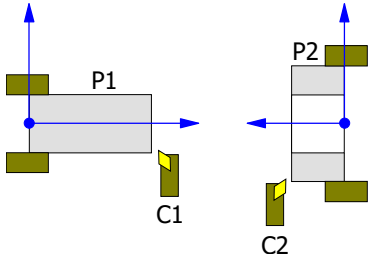
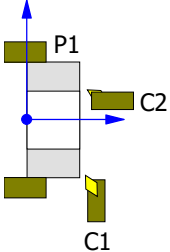
Modify or create part Pn associated with the execution channel.

#DGWZ CYL/RECT [...] Cm

Modify or create part Pm associated with channel Cm.

#DGWZ CYL/RECT [...] Cn Cm

Modify or create part Pn associated with channels Cn and Cm.

	<p>Programming from channel ·1·. #DGWZ RECT [...]</p>
	<p>Programming from channel ·1·. #DGWZ CYL Z [...] P1 C1</p> <p>Programming from channel ·2·. #DGWZ CYL Z2 [...] P2 C2</p>
	<p>Programming from channel ·1·. #DGWZ CYL Z [...] P1 C1 C2</p>

## 26.2 ISO generation.

ISO generation converts canned cycles, calls to subroutines, loops, etc. into their equivalent ISO code (G, F, S, etc functions), so the user can modify it and adapt it to his needs (eliminate unwanted movements, etc.).

Original block.	ISO generation.
ISO and conversational canned cycles.	The CNC breaks up the canned cycles into ISO blocks (G, F, S, etc. functions).
Local subroutines.	The CNC replaces the calls to local subroutines with the contents of the subroutine.
Conditional loops (\$IF, \$FOR, etc.) and block repetition (#RPT, NR).	The CNC breaks up the loops and repetitions into ISO blocks (G, F, S, etc. functions).
Parameters and variables.	The CNC replaces the arithmetic parameters and variables with their values.

The CNC generates the new ISO code while simulating the program, either from the DISIMU mode or from the conversational mode. Simulating a cycle from the cycle editor does not generate ISO code. While converting into ISO code, the CNC saves the new blocks into a new program (by default with extension .fiso) and, therefore, does not modify the original program.

To generate the ISO code during simulation, the program must include the following instructions. The CNC only generates the ISO code of the portion programmed between these two instructions and it ignores the rest.

```
#ISO ON      Enable ISO generation.
#ISO OFF     Disable ISO generation.
```

### Programming. Enable ISO generation.

When defining this instruction, there is an option to define the path and the name of the generated program. If a parameter is to be modified throughout the program, just program the instruction again with the new parameters.

#### Programming format.

The programming format is the following; the arguments appear between curly brackets and the optional ones between angle brackets. Programming the ON command is optional.

```
#ISO <ON> <[NAME="{path\name}"]>
```

NAME={path\name}    Optional. Path and name of the output program.

```
#ISO
  (Enable ISO generation)
#ISO ON [NAME="C:\Fagorcnc\Users\Prg\cycles.fiso"]
  (Enable ISO generation)
  (The CNC saves the program in the indicated folder)
  (The CNC saves the program with the name "cycles.fiso")
#ISO [NAME="cycles.nc"]
  (Enable ISO generation)
  (The CNC saves the program with the name "cycles.nc")
```

#### Path and name of the generated file.

The path and name are optional; If not programmed, the CNC assumes the last value used in the program. The CNC keeps the programmed values until the end of the program.

If the path is not indicated and no value has been programmed earlier, the generated program will be in the same folder as the original one. If the name is not indicated and no value has been programmed earlier, the generated program will have same name as the original one but with the extension .fiso.

# 26.

PROGRAMMING STATEMENTS.  
ISO generation.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Programming. Disable ISO generation.

'This instruction must be programmed alone in the block' Programming it is optional; if not programmed, the CNC generates ISO code up to the end of the program (M30).

### Programming format.

The programming format is:

#ISO OFF

```
#ISO OFF
  (Disable ISO generation)
```

### Considerations.

#### Converting pockets to ISO code.

ISO blocks created from the pockets are calculated with a specific tool radius. If these blocks are executed using another radius, the machining will not execute as expected.

#### Programming the tool in the canned cycles.

There are canned cycles, subroutines, etc. that need to have the tool radius to create the ISO blocks. In these cases, the tool number must be programmed after the statement #ISO.

#### Program two #ISO instructions in the same program.

- If in a program there are two or more #ISO instructions with the same name and an #ISO OFF instruction has been programmed between them, from the second #ISO instruction on the CNC resumes ISO block generation in the same program.
- If in a program there are two or more #ISO instructions with the same name and no #ISO OFF instruction has been programmed between them, from the second #ISO instruction will have no effect (it will be ignored).
- If in a program there are two or more #ISO instructions with different names, the ISO blocks generated since each instruction will go in the program indicated in each instruction. It doesn't matter whether an #ISO OFF instruction has been programmed or not between both instructions.

### Examples.

#### Example. Convert a subroutine.

```
%L SUBROUTINE
G90 G01 X80 Y0 F500
Z-2
G91 Y-25
G03 Y50 R25
G01 Y-25
G90 G01 Z5
M29

%PROGRAM
...
LL SUBROUTINE
...
```

#### Program after ISO generation.

```
...
(LL SUBROUTINE)
G90 G01 X80 Y0 F500
Z-2
G91 Y-25
G03 Y50 R25
G01 Y-25
G90 G01 Z5
(M29)
...
```

#### Example. Convert a canned cycle.

```
G0 X0 Y0 G81 I-10
```

#### Program after ISO generation.

```
G0 X0 Y0 G80
;----- G81 I-10 -----
G40
M3
G0 G61 G90 Z5
G1 G60 Z-10
G0 G50 Z5
G0 G139
;-----
```

**Example. Convert a repetition.**

```
G91 G01 Q60 NR6
```

**Program after ISO generation.**

```
G91 G01 Q60 ;NR6  
G91 G01 Q60 ;NR6  
G91 G01 Q60 ;NR6  
G91 G01 Q60 ;NR6  
G91 G01 Q60 ;NR6  
G91 G01 Q60 ;NR6
```

**Example. Convert parameters.**

```
$FOR P1=0,240,120  
G73 Q[P1]  
$ENDFOR
```

**Program after ISO generation.**

```
G73 Q[0]  
G73  
G73 Q[120]  
G73  
G73 Q[240]  
G73
```

**26.****PROGRAMMING STATEMENTS.**  
ISO generation.



## 26.3 Electronic axis slaving

Two axes may be slaved to each other so the movement of one of them (slave) depends on the movement of the other one (master).

It is possible to have several axis couplings (slaving) at the same time.

Axis coupling is activated with the #LINK instruction and canceled with the #UNLINK instruction. When reaching the end of program with a coupled pair of axes, this slaving is canceled after executing an M02 or M30.

### Considerations about axis coupling

Although the #LINK instruction admits several sets (pairs) of axes, the following limitations must be taken into account:

- The main axes (the first three axes of the channel) cannot be slaves.
- Both axes of the master-slave pair must be of the same type (linear or rotary).
- The master axis of a pair cannot be the slave of another pair.
- An axis cannot be slaved to more than one master axis.

Likewise, a new slaving (coupling) cannot be activated without deactivating the pairs previously slaved.

### #LINK

#### Activate the electronic coupling (slaving) of axes

This instruction defines and activates the electronic coupling of axes. Several couplings may be activated at the same time. When executing this instruction, all the axes defined as slaves depend on their relevant masters. On these slave axes, no movement may be programmed while they stay coupled.

This instruction may also be used to define the maximum following error difference allowed between the master axis and its slave.

The programming format is as follows:

```
#LINK [<master>,<slave>,<error>][...]
```

Parameter	Meaning
<master>	Master axis.
<slave>	Slave axis.
<error>	Optional. Maximum difference allowed between the following errors of both axes.

Programming the amount of error is optional; if not programmed, this test is not carried out. The maximum error will be defined in millimeters or inches for linear axes and in degrees for rotary axes.

```
#LINK [X,U][Y,V,0.5]
#LINK [X,U,0.5][Z,W]
#LINK [X,U][Y,V][Z,W]
```

### #UNLINK

#### Cancel the electronic coupling (slaving) of axes

This instruction deactivates the active axis slaving.

```
#UNLINK
(Cancels axis coupling)
```

When reaching the end of program with a coupled pair of axes, this slaving is canceled after executing an M02 or M30.

# 26.

PROGRAMMING STATEMENTS.

Electronic axis slaving

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 26.4 Axis parking

Some machines, depending on the type of machining, may have two different configurations (axes and spindles). In order to prevent the elements not present in one of the configurations from causing an error message (drives, feedback systems, etc.) the CNC allows parking them.

For example, a machine that swaps a normal spindle with a rectangular one may have the following axes configurations:

- With a normal spindle, X Y Z axes configuration.
- With an orthogonal spindle, X Y Z A B axes configuration.

In this case, when working with the normal spindle, the A and B axes may be parked to ignore their signals.

Several axes and spindles may stay parked at the same time, but they must always be parked (and unparked) one by one.

Use the #PARK instruction to park the axes and spindles and #UNPARK to cancel (unpark) them. The axes and spindles stay parked after executing an M02 or M30, after a RESET and even after turning the CNC off and back on.

### Considerations about axis parking

The CNC will not allow parking an axis in the following cases.

- If the axis belongs to the active kinematics.
- If the axis belongs to an active transformation #AC or #ACS.
- If the axis belongs to an active angular transformation #ANGAX .
- If the axis belongs to a gantry, tandem pair or is a slaved axis.
- If the axis belongs to an active tangential control #TANGCTRL.

### Considerations about spindle parking

The CNC will not allow parking a spindle in the following cases.

- If the spindle is not stopped.
- If the spindle is working as a C axis.
- If G96 or G63 is active and it is the master spindle of the channel.
- If G33 or G95 is active and it is the master spindle of the channel or the spindle is used to synchronize the feedrate.
- If the spindle belongs to a tandem pair or is a synchronized spindle, be it the master or the slave.

If after parking the spindles, there is only one spindle left in the channel, it will become the new master. If a spindle is unparked and it is the only spindle of the channel, it is also assumed as the new master spindle.

### #PARK Parks an axis

This instruction is used to park the selected axis or spindle. When any of them is parked, the CNC interprets that it no longer belongs to the machine configuration and no longer controls it (ignoring the signals from the drive and from the feedback systems, etc.).

Once an axis or spindle has been parked, the part-program cannot mention it (movements, speed, M functions, etc.).

The programming format is as follows:

```
#PARK <axis/spindle>
```

Each element (axis or spindle) must be parked separately. However, a second element can be parked without having to unpark the first one.

# 26.

PROGRAMMING STATEMENTS.  
Axis parking



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

When trying to park an axis or spindle that is already parked, the programming is ignored.

```
#PARK A
  (It parks the "A" axis)
#PARK S2
  (It parks spindle "S2")
```

### **#UNPARK** **Unparks an axis**

This instruction is used to unpark the selected axis or spindle. When unparking one of them, the CNC interprets that it belongs to the machine configuration and starts controlling it.

The programming format is as follows:

```
#UNPARK <axis/spindle>
```

The axes must be unparked one by one.

When trying to unpark an axis or spindle that is already unparked, the programming is ignored.

```
#UNPARK A
  (It unparks the "A" axis)
#UNPARK S
  (It unparks the "S" spindle)
```

26.

PROGRAMMING STATEMENTS.

Axis parking

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 26.5 Modifying the configuration of the axes of a channel

Initially, each channel has some axes assigned to it as set by the machine parameters. While executing a program, a channel may release its axes or request new axes. This possibility is determined by machine parameter `AXISEXCH`, which establishes whether an axis can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.



*It also restores the machine parameter settings if a checksum error occurs when powering up the CNC.*

### Knowing if an axis can change channels

Machine parameter `AXISEXCH` may be consulted using the following variable.

`V.MPA.AXISEXCH.Xn`

Replace "Xn" with the name or logic number of the axis.

Value	Meaning
0	It cannot change channels.
1	The change is temporary.
2	The change is permanent.

### Knowing in which channel the axis is

It is possible to know in which channel the axis is by using the following variable.

`V.[n].A.ACTCH.Xn`

Replace "Xn" with the name or logic number of the axis.

Replace the "n" letter with the channel number.

Value	Meaning
0	It is not in any channel.
1-4	Channel number.

### Commands for modifying the axis configuration via program

The following instructions are used to modify the configuration of the axes. It is possible to add or remove axes, change their names and even redefine the main axes of the channel by swapping their names.

Changing the configuration of the axes cancels the active polar origin, the pattern rotation, the mirror image and the scaling factor.

In the configuration of the axes (if G17 is active), the axis that occupies the first position must be the abscissa axis, the second will be the ordinate axis, the third will be the axis perpendicular to the work plane, the fourth will be the first auxiliary axis and so on.

#### #SET AX Sets the axis configuration

Defines a new axis configuration in the channel. The channel axes not programmed in the instruction and the nonexistent programmed ones will be added. The axes are placed in the channel in the positions as they are programmed in the instruction `#SET AX`. Optionally, one or several offsets may be applied to the defined axes.

It is the same as programming a `#FREE AX` of all the axes and then a `#CALL AX` of all the new axes.

The instruction `#SET AX` may also be used only to order the existing axes in the channel differently.

The programming format is as follows:

```
#SET AX [<Xn>,...] <offset> <...>
```

# 26.

PROGRAMMING STATEMENTS.

Modifying the configuration of the axes of a channel



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

Parameter	Meaning
<Xn>	Axes that make up the new configuration. If instead of defining an axis, a zero is written, an empty space (without an axis) appears in this position.
<offset>	Optional. It sets which offset is applied to the axes. Several offsets may be applied.

```
#SET AX [X,Y,Z]
#SET AX [X,Y,V1,0,A]
```

### Offset setting

The offsets that may be applied to the axes are identified with the following commands. To apply several offsets, program the relevant commands separated by a blank space.

Command	Meaning
ALL	Include all the offsets.
LOCOF	Include the offset of the reference search.
FIXOF	Include the fixture offset.
ORGOF	Include zero offset.
MEASOF	Include measurement offset.
MANOF	Include the offset of the manual operations.

```
#SET AX [X,Y,Z] ALL
#SET AX [X,Y,V1,0,A] ORGOF FIXOF
```

If when defining a new configuration only the order of the axes in the channel is swapped, the offsets are ignored.

### Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.

<b>Y 00000.0000</b> <b>? 00000.0000</b> <b>? 00000.0000</b> <b>Z 00000.0000</b> <b>A 00000.0000</b>	<b>X 00125.1500</b> <b>Y 00089.5680</b> <b>Z 00000.0000</b> <b>? 00000.0000</b> <b>? 00000.0000</b>
#SET AX [Y, 0, 0, Z, A]	#SET AX [X, Y, Z] FIXOF ORGOF

Screen display of the different configurations. Let us suppose a machine with 5 axes X-Y-Z-A-W.

### #CALL AX Adds an axis to the configuration

it adds one or more axes to the preset configuration and it also allows defining its position. If the axis already exists in the configuration, it is placed in the new position. If the axis already exists and no position is programmed, the axis stays in its original position. Optionally, one or several offsets may be applied to the defined axes.

The programming format is as follows:

```
#CALL AX [<Xn>,<pos>...] <offset> <...>
```

# 26.

**PROGRAMMING STATEMENTS.**

Modifying the configuration of the axes of a channel

Parameter	Meaning
<Xn>	Axes to be added to the configuration. If the axis already exists, it is placed in the new position.
<pos>	Optional. Position of the axis in the new configuration. If not programmed, the axis is placed after the last one. If the position is occupied, the relevant error message will be issued.
<offset>	Optional. It sets which offset is applied to the axes. Several offsets may be applied.

```
#CALL AX [X,A]
    (It adds the X and A axes to the configuration, after the last existing axis)
#CALL AX [V,4,C]
    (It adds the V axis to position 4 and the C axis after the last one)
```

### Offset setting

The offsets that may be applied to the axes are identified with the following commands. To apply several offsets, program the relevant commands separated by a blank space.

Command	Meaning
ALL	Include all the offsets.
LOCOF	Include the offset of the reference search.
FIXOF	Include the fixture offset.
ORGOF	Include zero offset.
MEASOF	Include measurement offset.
MANOF	Include the offset of the manual operations.

```
#CALL AX [X] ALL
#CALL AX [V1,4,Y] ORGOF FIXOF
```

### Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.

<b>Y 00000.0000</b>	<b>Axis configuration</b> #SET AX [Y, 0, 0, Z] Y: Abscissa axis. Z: First auxiliary axis.  #CALL AX [X,2, W, 3] Y: Abscissa axis. X: Ordinate axis. W: Axis perpendicular to the plane. Z: First auxiliary axis.
<b>X 00000.0000</b>	
<b>W00000.0000</b>	
<b>Z 00000.0000</b>	
<b>? 00000.0000</b>	

### #FREE AX Frees an axis from the configuration

Removes the programmed axes from the current configuration. After removing an axis, the position is free, but the order of the axes that remain in the channel does not change.

The programming format is as follows:

```
#FREE AX [<Xn>,...]
```

Parameter	Meaning
<Xn>	Axis to be removed from the configuration



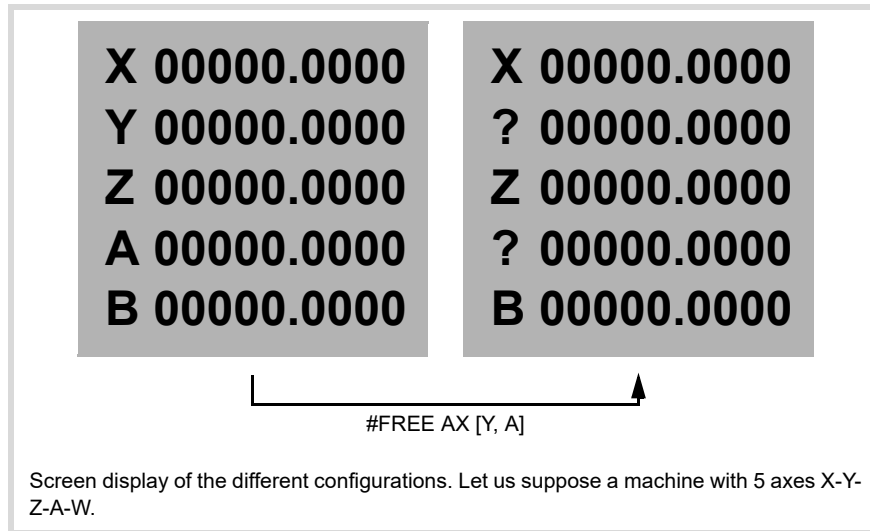
CNC 8070

(REF: 2008)

```
#FREE AX [X,A]
    (It removes the X and A axes from the configuration)
#FREE AX ALL
    (Removes all the axes from the channel)
```

**Screen display**

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.



**#RENAME AX**  
**Renames the axes**

It changes the name of the axes. For each programmed axis pair, the first axis takes the name of the second one. If the second axis is present in the configuration, it takes the name of the first one. Any axis may be renamed with any name, whether it is in any channel or not.

The programming format is as follows:

```
#RENAME AX [<Xn1>,<Xn2>][...]
```

Parameter	Meaning
<Xn1>	Axis whose name is to be changed
<Xn2>	new axis name.

```
#RENAME AX [X,X1]
    (The X axis is now called X1. If X1 already exists in the channel, it is called X)
#RENAME AX [X1,Y][Z,V2]
```

Parameter RENAMECANCEL indicates whether the CNC keeps or cancel the names of the axes and spindles (#RENAME) after executing M02 or M30, after a reset or at the beginning of a new part-program in the same channel.

After turning the CNC off and on, the axes and spindles always keep their new name, except after a checksum error or after validating the machine parameters that involve restoring the original configuration of channels, axes or spindles. In either case, the axes and the spindles will recover their original names.

When a channel releases (frees) an axis (instruction #SET or #FREE), the axis always recovers its original name.

Even if the #RENAME is kept (parameter RENAMECANCEL), the CNC cancels it if the channel recovers an axis with the same name after a reset or after a new program starts. This occurs if the #RENAME uses the name of an axis whose channel change permission is temporary or no\_exchange (parameter AXISEXCH), that is not in the channel at the time.

**Accessing the variables of a renamed axis.**

After changing the name of an axis, the new name of the axis must be used to access its variables from the part-program or MDI. The access to the variables from the PLCA or from an interface does not change; the original name of the axis remains unchanged.

**#RENAME AX OFF****Cancel rename.**

This instruction cancels the renaming of the indicated axes, regardless of what parameter RENAMECANCEL indicates; if no axis is defined, it cancels the renaming of all the axes of the channel.

The programming format is as follows:

```
#RENAME AX OFF [<Xn>, <Xn>, ...]
```

Parameter	Meaning
<Xn>	Renamed axis.

```
#RENAME AX OFF [X]
  (Cancel the renaming of the X axis).
#RENAME AX OFF
  (Cancel the renaming of all the axes).
```

**26.****PROGRAMMING STATEMENTS.**

Modifying the configuration of the axes of a channel



FAGOR AUTOMATION

CNC 8070

(REF: 2008)



## 26.6 Modifying the configuration of the spindles of a channel

The CNC can have up to four spindles distributed between the various channels of the system. A channel may have one, several or no spindles associated with it.

Initially, each channel has some spindles assigned to it as set by the machine parameters. While executing a program, a channel may release its spindles or request new spindles. This possibility is determined by machine parameter `AXISEXCH`, which establishes whether a spindle can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.



*It also restores the machine parameter settings if a checksum error occurs when powering up the CNC. .*

### Knowing if a spindle can change channels

Machine parameter `AXISEXCH` may be consulted using the following variable.

`V.MPA.AXISEXCH.Sn`

Replace "Sn" with the spindle name.

Value	Meaning
0	It cannot change channels.
1	The change is temporary.
2	The change is permanent.

### Knowing in which channel the spindle is

It is possible to know in which channel the spindle is by using the following variable.

`V.[n].A.ACTCH.Sn`

Replace "Sn" with the spindle name.

Replace the "n" letter with the channel number.

Value	Meaning
0	It is not in any channel.
1-4	Channel number.

### Commands for modifying the spindle configuration via program

The following instructions are used to modify the configuration of the spindles of the channel. It is possible to add or remove spindles, change the name of the spindles and define which one is the master spindle of the channel.

#### #FREE SP

#### Frees a spindle from the configuration

Removes the defined spindles from the current configuration.

The programming format is as follows:

```
#FREE SP [<Sn>,...]
#FREE SP ALL
```

Parameter	Meaning
<Sn>	Spindle name.
ALL	Frees all the spindles of the channel.

```
#FREE SP [S]
    (It removes the spindle S from the configuration)
#FREE SP [S1,S4]
    (It removes spindles S1 and S4 from the configuration)
#FREE SP ALL
    (It removes all the spindles from the configuration)
```

**#CALL SP****Add a spindle to the configuration**

It adds one or several spindles to the current configuration. The position of the spindles in the channel is not relevant. To add a spindle to the channel, the spindle must be free; it must not be in another channel.

The programming format is as follows:

```
#CALL SP [<Sn>,...]
```

Parameter	Meaning
<Sn>	Spindle name.

```
#CALL SP [S1]
```

(It adds spindle S1 to the configuration)

```
#CALL SP [S,S2]
```

(It adds spindles S and S2 to the configuration)

**#SET SP****Sets the spindle configuration**

Defines a new spindle configuration. The spindles existing in the channel and not programmed in `#SET SP` are removed and those programmed that are not already in the channel will be added. When a new configuration is defined, the order in which the spindles are defined is not relevant; the CNC always arranges them in an ascending order, according to the list of machine parameters.

It is the same as programming a `#FREE SP` of all the spindles and then a `#CALL SP` of all the new spindles. The programming format is as follows:

```
#SET SP [<Sn>,...]
```

Parameter	Meaning
<Sn>	Spindle name.

```
#SET SP [S]
```

(Configuring one spindle)

```
#SET SP [S1,S2]
```

(Configuring two spindles)

**#RENAME SP****Rename the spindles**

It changes the name of the spindles. For each programmed spindle pair, the first spindle takes the name of the second one. If the second spindle is present in the configuration, it takes the name of the first one. Any axis may be renamed with any name, whether it is in any channel or not.

The programming format is as follows:

```
#RENAME SP [<Sn>,<Sn>][...]
```

Parameter	Meaning
<Sn>	Spindle name.

```
#RENAME SP [S,S1]
```

```
#RENAME SP [S1,S2][S3,S]
```

Parameter `RENAMECANCEL` indicates whether the CNC keeps or cancel the names of the axes and spindles (`#RENAME`) after executing `M02` or `M30`, after a reset or at the beginning of a new part-program in the same channel.

After turning the CNC off and on, the axes and spindles always keep their new name, except after a checksum error or after validating the machine parameters that involve restoring the original configuration of channels, axes or spindles. In either case, the axes and the spindles will recover their original names.

26.

PROGRAMMING STATEMENTS.

Modifying the configuration of the spindles of a channel



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

When a channel releases (frees) a spindle (instruction #SET or #FREE), the axis always recovers its original name.

Even if the #RENAME is kept (parameter RENAMECANCEL), the CNC cancels it if the channel recovers a spindle with the same name after a reset or after a new program starts. This occurs if the #RENAME uses the name of a spindle whose channel change permission is temporary or no\_exchange (parameter AXISEXCH), that is not in the channel at the time.

**Accessing the variables of a renamed axis.**

After changing the name of an axis, the new name of the axis must be used to access its variables from the part-program or MDI. The access to the variables from the PLCA or from an interface does not change; the original name of the axis remains unchanged.

**#RENAME SP OFF  
Cancel rename.**

This instruction cancels the renaming of the indicated spindles, regardless of what parameter RENAMECANCEL indicates; if no spindle is defined, it cancels the renaming of all the spindles of the channel.

The programming format is as follows:

#RENAME SP OFF [<Sn>, <Sn>, ...]

Parameter	Meaning
<Sn>	Renamed spindle.

#RENAME SP OFF [S3]  
(Cancel the renaming of spindle S3).  
#RENAME SP OFF  
(Cancel the renaming of all the spindles).

## 26.7 Spindle synchronization

This mode may be used to set the movement of a spindle (slave) synchronized with that of another spindle (master) through a given ratio. The spindle synchronization is always programmed in the channel the slave spindle belongs to, both to activate it or deactivate it and to reset it.

There are two types of synchronization; in speed and in position. The activation and cancellation of the different types of synchronization are programmed with the following instructions.

- #SYNC - Spindle synchronization based on the real (actual) coordinate.
- #TSYNC - Spindle synchronization based on the theoretical coordinate.
- #UNSYNC - Cancellation of spindle synchronization.

### #SYNC Spindle synchronization based on the real (actual) coordinate

### #TSYNC Spindle synchronization based on the theoretical coordinate

The programming format for each of them is the following. Optional parameters are indicated between the <> characters.

- ```
#SYNC [{master}, {slave} <,N{nratio}, D{dratio}> <,O{posync}> <,{looptype}>
<,{keepsync}>][..]
#TSYNC [{master}, {slave} <,N{nratio}, D{dratio}> <,O{posync}> <,{looptype}>
<,{keepsync}>][..]
```

Each pair of brackets defines a synchronism between two spindles.

| Parameter            | Meaning                                                                                                                                                                                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {master}             | Master spindle of the synchronization.                                                                                                                                                                                                                                                                                                         |
| {slave}              | Master spindle of the synchronization.                                                                                                                                                                                                                                                                                                         |
| {nratio}<br>{dratio} | Optional. This pair of numbers define the gear ratio (nratio/dratio) between the synchronized spindles.<br>Both values can be positive or negative.                                                                                                                                                                                            |
| {posync}             | Optional. This parameter determines that they are synchronized in position and it also sets the offset (shift) between the two spindles.<br>The values may be positive or negative and greater than 360°.                                                                                                                                      |
| {looptype}           | Optional. This parameter indicates the type of loop for the master spindle. "CLOOP" for the spindle in closed loop. "OLOOP" for the spindle in open loop. If not programmed, the statement assumes the value "CLOOP".                                                                                                                          |
| {keepsync}           | Optional. This parameter indicates whether the CNC cancels spindle synchronization or not after executing M02, M30 or after an error or reset. With "CANCEL" value, the CNC cancels the synchronism; with "NOCANCEL" value, it doesn't cancel it.<br>If not programmed, the statement assumes the value set by the OEM (parameter SYNCCANCEL). |

- ```
#SYNC [S,S1]
    The spindles are synchronized in speed. The slave spindle S1 turns at the same speed as the master spindle S.

#SYNC [S,S1,N1,D2]
    The slave spindle S1 turns at half the speed (1/2) of the master spindle S.

#SYNC [S,S1,N1,D2,O15]
    After synchronizing in speed and in position, the slave spindle S1 follows the master S with the indicated offset, which could be for example 15°.

#SYNC [S,S1,O30,OLOOP]
    Synchronism in speed and position with 30° offset. The master spindle works in open loop.

#SYNC [S,S1,O30,CLOOP, CANCEL]
    Synchronism in speed and position with 30° offset. The master spindle works in closed loop. The CNC cancels the synchronism after M30, an error or a reset.
```

26.

PROGRAMMING STATEMENTS.  
Spindle synchronization

### Considerations for the synchronization

The #SYNC function may be executed either in open loop (M3 or M4) or in closed loop (M19). In the synchronization, the master spindle can work in either open or closed loop; the slave spindle is always in closed loop.

Several pairs of synchronized spindles may be programmed in the same instruction #SYNC or #TSYNC. Several consecutive #SYNC instructions can also be programmed with an additive effect as long as they do not conflict with the previous ones.

The slave spindle must be in the channel where the synchronization is activated whereas the master spindle may be in any channel. Several slave spindles may have the same master spindle, but a slave spindle cannot be the master of a third one; thus preventing synchronism loops.

It is possible to program either first the synchronism in speed and then in position or both at the same time. Once a pair has been synchronized, it is possible to change its speed ratio and/or its offset (shift); if necessary, the spindles will get out of synchronism and back in synchronism to make the change.

Both spindles must operate in closed loop in order to assure proper follow-up. Once they're both in closed loop, the slave spindle switches from its current speed to the synchronism speed. The master spindle may be turning when programming the synchronization and it will switch to closed loop maintaining its rotation.

### Programming the master and slave spindles

For the slave spindle, it is not possible to program the speed, the spindle functions M3, M4, M5 and M19, gear changes M41 through M44 or change the speed override.

The following functions may be programmed for the master spindle:

- Change the spindle speed via PLC or CNC.
- Execute the speed functions G94, G95, G96 and G97.
- Execute the auxiliary functions M3, M4, M5 and M19.
- Change the spindle speed override via PLC, CNC or keyboard.
- Change the spindle speed limit via PLC or CNC.
- If the C axis is activated, define the XC or ZC plane.

When defining the synchronization or when it is active, the master spindle can work as C axis or in G63. Functions G33, G95 or G96 may also be active at the master spindle. Functions G33 and G95 may also be active at the slave spindle, but function G96 will stay temporarily "frozen" and without effect during the synchronization.

On the other hand, the synchronized spindles cannot change channels and it is not possible to change gears M41 through M44. If the gear change is automatic and the new speed requires a gear change, the relevant error message will be issued.

### Work gear

The spindles may have different gears (ranges). If at the time of the synchronization, the spindles are not in the same state, the slave "freezes" its state, changes the gear indicated by machine parameter SYNCSET and is forced to follow the master.

If the master belongs to the same channel, it also changes to the gear indicated by its parameter SYNCSET. If the master is in another channel, the gear must be activated before activating the synchronization. Therefore, it is up to the user to prepare the master spindle so the slave can synchronize it with it.

### Machine reference (home) search

The slave spindle must be homed (referenced) before activating the position synchronism if it has been homed before. If the master spindle is in the same channel and has not been homed, it is also forced to be homed. An error message will be issued if the master spindle is in another channel and it has not been homed.

## #UNSYNC Uncouple one or several spindles

The programming format is: Optional parameters are indicated between the <> characters.

```
#UNSYNC
#UNSYNC [slave1 <,slave2> ...]
```

All the spindles are uncoupled if no parameter is defined.

Parameter	Meaning
slave	Slave spindle to be synchronized.

```
#UNSYNC
  All the spindles of the channel are uncoupled.
#UNSYNC [S1,S2]
  Slave spindles S1 and S2 are uncoupled from the master spindle with which they were
  synchronized.
```

### Considerations for uncoupling

The synchronism is also canceled with M30 and RESET.

When canceling the synchronism, the master spindle stays in its current state and the slave stops. The slave does not recover the M function prior to the synchronization, but it keeps the synchronism gear until a new S function is programmed.

### Variables associated with the synchronization movement

These variables are read/write (R/W) synchronous and are evaluated during execution. They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis.

#### Adjust the speed synchronism ratio

##### (V.)[n].A.GEARADJ.Xn

Read-only from the PRG, PLC and INT. The PLC reading comes in hundredths (x100)

Fine adjustment of the gear ratio during the synchronization itself. It is programmed as a percentage of the original adjustment value.

#### Speed synchronization

##### (V.)[n].A.SYNCVELW.Xn

Read-only from the PRG, PLC e INT.

When the spindles are synchronized in speed, the slave spindle turns at the same speed as the master spindle (taking the ratio into account). If the value defined in this variable is exceeded, the SYNSPEED signal goes low; the movement is not stopped and no error message is issued.

Its default value is that of machine parameter DSYNCVELW.

##### (V.)[n].A.SYNCVELOFF.Xn

Read-only from the PRG, PLC e INT.

Velocity offset of the slave spindle synchronization.

# 26.

PROGRAMMING STATEMENTS.  
Spindle synchronization



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

### **Position synchronization**

#### **(V.)[n].A.SYNCPOSW.Xn**

Read-only from the PRG, PLC e INT.

When the spindles are synchronized in position, the slave spindle follows the master keeping the programmed offset (taking the ratio into account). If the value defined in this variable is exceeded, the SYNCPOSI signal goes low; the movement is not stopped and no error message is issued.

Its default value is that of machine parameter `DSYNCPOSW`.

#### **(V.)[n].A.SYNCPOSOFF.Xn**

Read-only from the PRG, PLC e INT.

Position offset.

**26.**

**PROGRAMMING STATEMENTS.**

Spindle synchronization

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## 26.8 Selecting the loop for an axis or a spindle. Open loop or closed loop



*This function is not available for Sercos position-drives (axis or spindle). In this case, the CNC cannot open or close the loop, the drive controls the loop, instead.*

When working in open loop, the command does not depend on the feedback. When working in closed loop, the command is generated according to the feedback.

The spindle usually works in open loop when it is in M3 or M4 and in closed loop when it is in M19. In spindle synchronization, the slave always works in closed loop and the master can work either in open loop or closed loop depending on the programming parameters of the #SYNC instruction. However, it is possible to work in closed loop with functions M3 and M4 to make the following adjustments on a spindle:

- Adjust the loop for M19.
- Adjust a loop for when the spindle is the master of a synchronization.

The axes usually work in closed loop. A rotary axis may be controlled in open loop as if it were a spindle.

The following instructions may be used to open and close the loops and they are valid for axes as well as for spindles.

- #SERVO ON - Activates the closed loop mode.
- #SERVO OFF - Activates the open loop mode.

### #SERVO ON Activates the closed loop mode

Programming this instruction switches the axis or spindle to closed loop mode.

The spindle must be homed before switching to closed loop mode; otherwise the loop will be closed and a warning message will be issued.

The programming format is as follows:

#SERVO ON [axis/spindle]

Parameter	Meaning
axis/spindle	Name of the axis or spindle.

The loop of each axis or spindle must be closed separately.

```
#SERVO ON [S]
  Closes the loop of spindle S.
#SERVO ON [S2]
  Closes the loop of spindle S2.
#SERVO ON [X]
  Closes the loop of the X axis.
```

### #SERVO OFF Activates the open loop mode

Programming this instruction switches the axis or spindle to open loop mode. For a spindle, it cancels the closed-loop mode programmed #SERVO ON and restores the status that the spindle had before closing the loop.

- If the spindle was in M19, it stays in closed loop after programming this instruction.
- In spindle synchronization, the #SERVO OFF instruction cannot be programmed for the slave spindle; if programmed, the CNC will issue an error message.

If the synchronization has been defined with the master spindle working in closed loop, it stays in closed loop after programming #SERVO OFF. If the synchronization has been defined with the master spindle working in open loop and, later on, it is closed with #SERVO ON, the loop of the master spindle will be opened after programming #SERVO OFF.

- If the spindle was in M3, M4 or M5 without synchronism, the CNC opens the loop.

# 26.

PROGRAMMING STATEMENTS. Selecting the loop for an axis or a spindle. Open loop or closed loop



The programming format is as follows:

#SERVO ON [axis/spindle]

Parameter	Meaning
axis/spindle	Name of the axis or spindle.

The loop of each axis or spindle must be opened separately.

#SERVO OFF [S]  
It cancels the closed loop of spindle S.  
#SERVO OFF [Z2]  
The Z2 axis switches to open loop mode.

### Considerations for programming the loops

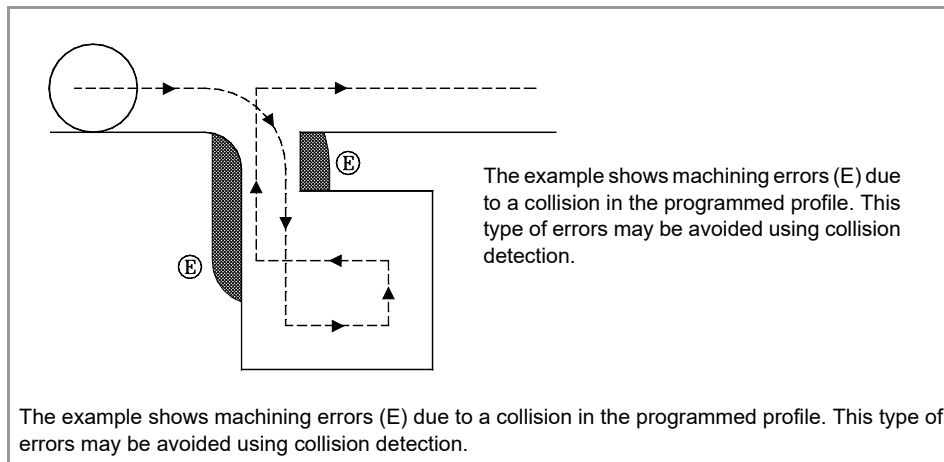
Function M19 involves always working in closed loop. Functions M3, M4 and M5 work in open loop by default , but they can also work in closed loop when programming a spindle synchronization or the instruction #SERVO ON .

When a spindle becomes a C axis or is interpolated with the rest of the axes (for rigid tapping, for example) does not lose its open-loop or closed-loop condition that it had. When done with these instructions, the previous situation is restored.

On power-up, the spindle is set in open loop. After executing an M30 or reset, the CNC opens the loop and cancels the instruction #SERVO ON, except when the reset is for the master spindle of a synchronization (which could be in a different channel from that of the slave); in that case, the synchronization is not canceled and it does not switch to open loop. This case issues a warning message.

## 26.9 Collision detection

The collision detection analyzes the machining tool paths beforehand in order to detect and avoid intersections of the profile with itself or collisions in the programmed profile. This function does not take into account the tool sizes, but rather only the tool paths. The number of analyzed blocks can be defined by the user, up to a maximum of 200 blocks (40 for an 8060).



When detecting a loop or a collision, the CNC will not execute the blocks that cause it and the screen will display a warning to let the operator know that the programmed profile has been modified. It will display a warning for each loop or collision eliminated. The information contained in the eliminated blocks, not being the moving in the active plane, will be executed (including the movements of the axes).

### Considerations for the collision detecting process.

- The collision detection can be applied even if there is no active tool radius compensation, but it will only detect the sections where the tool path is cut.
- Being collision detection active, it is possible to apply zero offsets, coordinate presetting and tool changes. However, home searches and measurements are NOT possible.
- Changing the work plane will interrupt the collision detecting process. The CNC checks for collisions in the blocks stored so far and resumes the process with the new plane starting with the new motion blocks.
- The collision detecting process will be interrupted when programming an instruction (explicit or implicit) that involves synchronizing block preparation and execution (e.g. #FLUSH). The process will resume after executing that instruction.
- Collision detection cannot be activated if a Hirth axis is active and it is part of the main plane. Likewise, while collision detection is active, an axis cannot be activated as a Hirth axis and the work plane cannot be changed if one of the axis is a Hirth axis.

### #CD ON Activate collision detection

It activates the collision detecting process. Being collision detection already active, it lets modify the number of blocks to be analyze.

The programming format is as follows:

#CD ON [<blocks>]

Parameter	Meaning
<blocks>	Optional. Number of blocks to analyze.

Defining the number of blocks to be analyzed is optional. If not programmed, the CNC assumes the maximum value. The horizon of blocks may be changed at any time, even while collision detection is active.

26.

PROGRAMMING STATEMENTS.  
Collision detection



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## #CD OFF Cancels collision detection

It cancels the collision detecting process.

The process will also be canceled automatically after executing an M02 or M30 and after an error or a reset.

### Example of a profile with a loop.

```
#CD ON [50]
G01 X0 Y0 Z0 F750
X100 Y0
Y-50
X90
Y20
X40
Y-50
X0
Y0
#CD OFF
```

### Example of profile collision.

```
#CD ON
G01 G41 X0 Y0 Z0 F750
X50
Y-50
X100
Y-10
X60
Y0
X150
Y-100
X0
G40 X0 Y0
#CD OFF
M30
```

26.

PROGRAMMING STATEMENTS.

Collision detection

**FAGOR** 

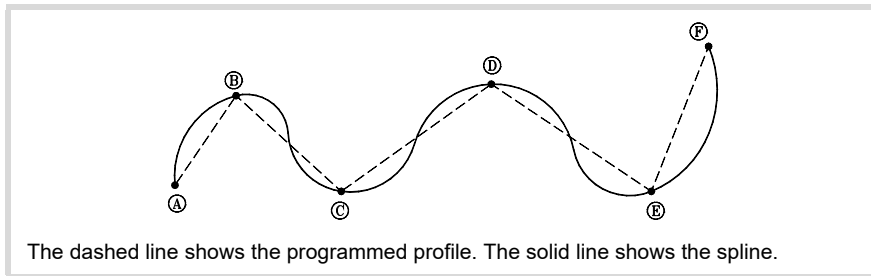
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 26.10 Spline interpolation (Akima)

This type of machining adapts the programmed contour to a spline type curve that goes through all the programmed points.



The contour to be splined is defined with straight paths (G00/G01). When defining an arc (G02/G03), the spline is interrupted while machining it and it resumes on the next straight path. The transitions between the arc and the spline is done tangentially.

### #SPLINE ON Activate spline adaptation.

When executing this instruction, the CNC interprets that the points programmed next are part of the spline and begins making the curve.

The programming format is as follows:

```
#SPLINE ON
```

The machining of splines cannot be activated if tool radius compensation (G41/G42) with linear transition between blocks (G137) or viceversa.

### #SPLINE OFF Cancel spline adaptation.

When executing this instruction, the CNC ends the spline and goes on machining as the path were programmed.

The programming format is as follows:

```
#SPLINE OFF
```

The spline can only be canceled if at least 3 points have been programmed. When defining the initial and final tangents of the spline, 2 points will be enough.

### #ASPLINE MODE Select type of tangent.

This instruction sets the type of initial and final tangents of the spline that determines the transition from the previous and to the next path. It is optional; if not defined, the tangent is calculated automatically.

The programming format is as follows:

```
#ASPLINE MODE [<initial> <, final>]
```

Parameter	Meaning
<initial>	Initial tangent.
<final>	Final tangent

The initial and final tangent of the spline may take one of the following values. If not programmed, it assumes a value of 1.

26.

PROGRAMMING STATEMENTS.  
Spline interpolation (Akima)

Value	Meaning
1	The tangent is calculated automatically.
2	Tangent to the previous /next block.
3	Tangent as specified.

If defined with a value of '3', the initial tangent is defined using the #ASPLINE STARTTANG instruction and the final tangent using the #ASPLINE ENDTANG instruction. If not defined, it applies the values used last.

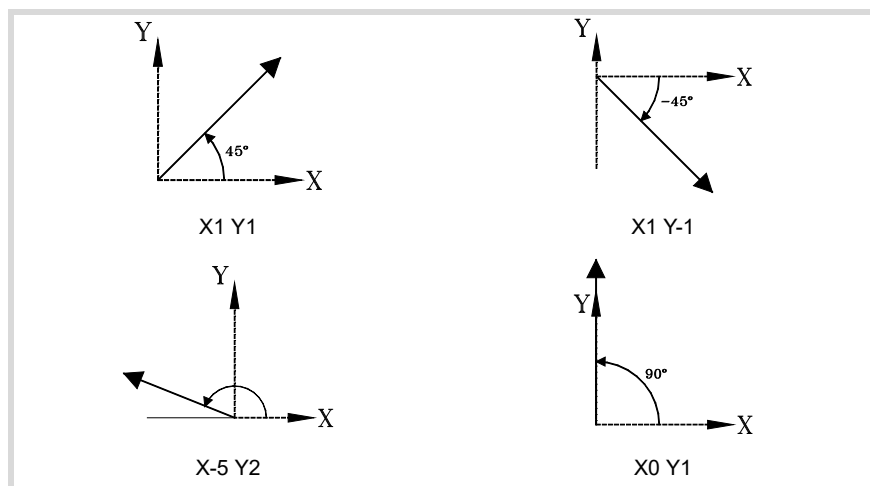
### #ASPLINE STARTTANG Initial tangent

### #ASPLINE ENDTANG Final tangent

These instructions define the initial and final tangents of the spline. The tangent is determined by giving its vectorial direction along the different axes.

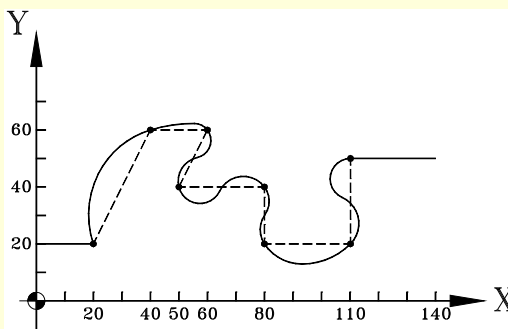
The programming format is as follows:

```
#ASPLINE STARTTANG <axes>
#ASPLINE ENDTANG <axes>
```



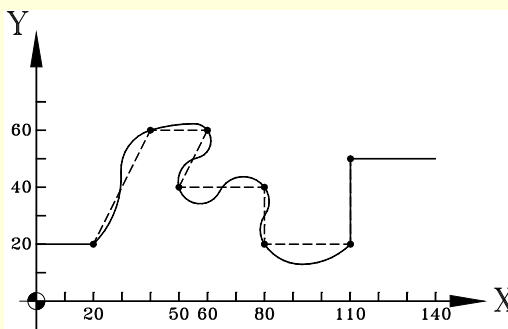
# 26.

**PROGRAMMING STATEMENTS.**  
Spline interpolation (Akima)



```

N10 G00 X0 Y20
N20 G01 X20 Y20 F750 (Starting point of the spline)
N30 #ASPLINE MODE [1,2] (Type of initial and final tangent)
N40 #SPLINE ON (Activation of the spline)
N50 X40 Y60
N60 X60
N70 X50 Y40
N80 X80
N90 Y20
N100 X110
N110 Y50 (Last point of the spline)
N120 #SPLINE OFF (Cancellation of the spline)
N130 X140
N140 M30
    
```



```

N10 G00 X0 Y20
N20 G01 X20 Y20 F750 (Starting point of the spline)
N30 #ASPLINE MODE [3,3] (Type of initial and final tangent)
N31 #ASPLINE STARTTANG X1 Y1
N32 #ASPLINE ENDTANG X0 Y1
N40 #SPLINE ON (Activation of the spline)
...
N120 #SPLINE OFF (Cancellation of the spline)
N130 X140
N140 M30
    
```

## 26.11 Polynomial interpolation

The CNC permits interpolating straight lines and arcs and the #POLY instruction may be used to interpolate complex curves, like a parabola.

### #POLY Polynomial interpolation

This type of interpolation lets machining a curve given by a polynomial of up to a 4th degree where the interpolation parameter is the length of the arc.

The programming format is as follows:

```
#POLY [<axis1>[a,b,c,d,e] <axis2>[a,b,c,d,e] .. SP<sp> EP<ep>]
```

Parameter	Meaning
<axis>	Axis to interpolate.
a,b,c,d,e	Coefficients of the polynomial.
<sp>	Initial parameter of the interpolation.
<ep>	Final parameter of the interpolation.

The coefficients define the axis path as a function for each axis.

```
#POLY [X[ax,bx,cx,dx,ex] Y[ay,by,cy,dy,ey] Z[az,bz,cz,dz,ez] .. SP<sp> EP<ep>]
```

$$X(p) = ax+bx*p+cx*p^2+dx*p^3+ex*p^4$$

$$Y(p) = ay+by*p+cy*p^2+dy*p^3+ey*p^4$$

$$Z(p) = az+bz*p+cz*p^2+dz*p^3+ez*p^4$$

Where "p" is the same parameter in all the axes. Parameters sp and ep define the initial and final values of "p", as the ends between which the path for each axis will be generated.

Programming a parabola. The polynomial may be represented as follows:

- Coefficients of the X axis: [0,60,0,0,0]
- Coefficients of the Y axis: [1,0,3,0,0]
- Starting parameter: 0
- End parameter: 60

The part program stays as follows.

```
G0 X0 Y1 Z0
G1 F1000
#POLY [X[0,60,0,0,0] Y[1,0,3,0,0] SP0 EP60]
M30
```

# 26.

**PROGRAMMING STATEMENTS.**  
Polynomial interpolation

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

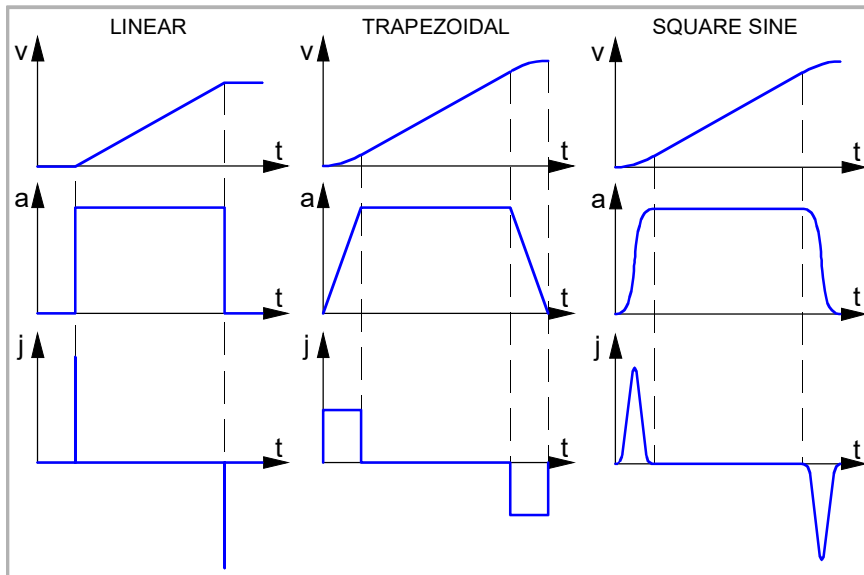
## 26.12 Acceleration control

The acceleration and the jerk (variation of acceleration) applied on the movements are set by machine parameters. However, those values may be changed from the program using the following functions.

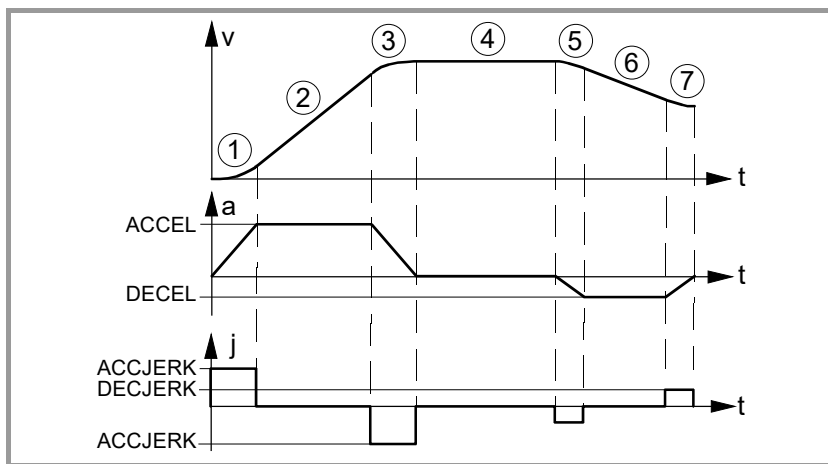
G130 or G131 Percentage of acceleration and deceleration to be applied.

G132 or G133 Percentage of acceleration jerk and deceleration jerk to be applied.

The next figure shows the feedrate (v), acceleration (a) and jerk (j) for each case.



We now show an example of the dynamics of the trapezoidal acceleration.



- 1 The axis starts moving with a uniform increasing acceleration, with a slope limited by the percentage of acceleration jerk indicated with function G132 or G133 until reaching the acceleration percentage indicated by function G130 or G131.
- 2 The acceleration is now constant.
- 3 Before reaching the programmed feedrate, there is a steady deceleration with a slope limited by the percentage of acceleration jerk.
- 4 It goes on at the programmed feedrate and with no acceleration.
- 5 To slow down or stop the axis, a deceleration is applied with a slope limited by the percentage of deceleration jerk.
- 6 The deceleration becomes constant and its value is the percentage of deceleration.
- 7 Before reaching the programmed feedrate, or stopping, there is a deceleration with a slope limited by the percentage of deceleration jerk.

26.

PROGRAMMING STATEMENTS.  
Acceleration control



## #SLOPE

### It sets the behavior of the acceleration

This instruction sets the influence of the values defined with functions G130, G131, G132 and G133 in the behavior of the acceleration.

The programming format is as follows:

```
#SLOPE [<type>,<jerk>,<accel>,<move>]
```

Parameter	Meaning
<type>	Acceleration type.
<jerk>	Optional. It sets the influence of the jerk.
<accel>	Optional. It sets the influence of the acceleration.
<mov>	Optional. It affects the movements in G00.

```
#SLOPE [1,1,0,0]
#SLOPE [1]
#SLOPE [2,,1]
```

It is not necessary to program all the parameters. The values that each parameter may take are the following.

- The <type> parameter determines the type of acceleration.

Value	Meaning
0	Linear acceleration.
1	Trapezoidal acceleration.
2	Square sine (bell shaped) acceleration.

By default, it assumes a value of ·0·.

- The optional <jerk> parameter sets the influence of the Jerk defined with functions G132 and G133. It will only be taken into account in trapezoidal and square-sine type acceleration.

Value	Meaning
0	It modifies the jerk of the acceleration and deceleration stage.
1	It modifies the jerk of the acceleration stage.
2	It modifies the jerk of the deceleration stage.

By default, it assumes a value of ·0·.

- The optional <accel> parameter sets the influence of the acceleration set with functions G130 and G131.

Value	Meaning
0	Always applied.
1	Only applied in the acceleration stage.
2	Only applied in the deceleration stage.

By default, it assumes a value of ·0·.

- The optional <move> parameter determines whether functions G130, G131, G132 and G133 affect the G00 movements or not.

Value	Meaning
0	They affect the G00 movements.
1	They do NOT affect the G00 movements.

By default, it assumes a value of ·0·.

## 26.13 Definition of macros

Macros may be used to define a program block or part of it with their own names in the format "MacroName" = "CNCblock". Once the macro has been defined, programming "MacroName" will be the same as programming "CNCblock". When executing a macro from the program (or MDI), the CNC will execute its associated program block.

The macros defined via program (or MDI) are stored in a CNC table; this way, they are available for the rest of the programs without having to define them again. This table is initialized on CNC power-up and it can also be initialized from the part-program using the #INIT MACROTAB instruction, thus deleting the macros saved.

### #DEF: Macro definition

Up to 50 different macros may be defined at the CNC. The defined macros may be accessed from any program. When trying to define too many macros, the CNC issues the relevant error message. The macro table may be initialized (erasing all the macros) using the instruction #INIT MACROTAB.

The definition of the macro must be programmed alone in the block.

The programming format is as follows:

```
#DEF "MacroName" = "BloqueCNC"
```

Parameter	Meaning
MacroName	Name used to identify the macro in the program. It may have up to 30 characters and consist of letters and numbers.
CNCBlock	Program block. It may be up to 140 characters long.

Several macros may be defined in a block as follows.

```
#DEF "Macro1"="Block1" "Macro2"="Block2" ...
```

(Definition of macros)

```
#DEF "READY"="G0 X0 Y0 Z10"  
#DEF "START"="SP1 M3 M41" "STOP"="M05"
```

(Execution of macros)

```
"READY" (same as programming G0 X0 Y0 Z10)  
P1=800 "START" F450 (same as programming S800 M3 M41)  
G01 Z0  
X40 Y40  
"STOP" (same as programming M05)
```

### Definition of arithmetic operations in the macros.

When including arithmetic operations in the definition of a macro, the whole arithmetic operation must be included.

Correct definition of a macro.

```
#DEF "MACRO1"="P1*3"  
#DEF "MACRO2"="SIN [\"MACRO1\"]"
```

The following macros are defined wrong.

```
#DEF "MACRO1"="56+"  
#DEF "MACRO2"="12"  
#DEF "MACRO3"="\"MACRO1\" \"MACRO2\""  
  
#DEF "MACRO4"="SIN["  
#DEF "MACRO5"="45]"  
#DEF "MACRO6"="\"MACRO4\" \"MACRO5\""
```

### Concatenating of macros. Including macros in the definition of other macros.

The definition of a macro can include other macros. In this case, one of the macros included in the definition must be delimited with the \" characters (\"macro\").

# 26.

PROGRAMMING STATEMENTS.  
Definition of macros



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

```
Example1
#DEF "MACRO1"="X20 Y35"
#DEF "MACRO2"="S1000 M03"
#DEF "MACRO3"="G01 \ "MA1\ " F100 \ "MA2\ ""
```

```
Example 2
#DEF "POS"="G1 X0 Y0 Z0"
#DEF "START"="S750 F450 M03"
#DEF "MACRO"="\ "POS\ " \ "START\ ""
```

### **#INIT MACROTAB** **Resetting the table of macros**

When defining a macro from a program (or MDI), it is stored in a CNC table so it is available for all the rest of the programs. This instruction resets the table of macros erasing the ones stored in it.

# 26.

## 26.14 Communication and synchronization between channels

Each channel may execute its own program simultaneously and independently from other channels. But, besides this, it can also communicate with other channels, transfer information or synchronize in specific points.

The communication takes place on the basis of a number of marks managed by the part-programs of each channel. These marks establish whether the channel is waiting to be synchronized or it may be synchronized, etc.

There are two different ways to synchronize, each offers a different solution.

- Using the #MEET instruction.
 

The easiest way to synchronize. It stops the execution in all the channels involved in the synchronization.

The set of marks being used are initialized after executing an M02 or an M30, after a reset or on power-up.
- Using the instructions #WAIT - #SIGNAL - #CLEAR.
 

This method is somewhat more complicated than the previous one, but more versatile. It does not stop the execution in all the channels in order to synchronize.

The set of marks being used are maintained after executing an M02 or an M30, after a reset or on power-up.

The synchronism marks of the two methods are independent from each other. The marks managed by the #MEET instruction neither affect nor are affected by the rest of the instructions.

### Other ways to synchronize channels

The common arithmetic parameters can also be used to communicate and synchronize channels. By writing a certain value from a channel and later reading it from another channel, it is possible to set the condition to follow up on the execution of a program.

Accessing the variables of a channel from another channel can also be used as a way to communicate.

Swapping axes between channels also makes it possible to synchronize processes, because a channel cannot grab an axis until it has been released by another one.

CHANNEL 1	CHANNEL 2	CHANNEL 3
G1 F1000	X1=0 Y1=0 Z1=0	G1 F1000
S3000 M3	G1 F1000	X2=20 Z2=10
#FREE AX [Z]	#FREE AX[Z1]	#FREE AX[Z2]
(Frees the Z axis)	(Frees the Z1 axis)	(Frees the Z2 axis)
X30 Y0	G2 X1=-50 Y1=0 I-25	X2=100 Y2=50
#CALL AX [Z1,Z2]	#CALL AX [Z]	#CALL AX[Z2]
(It adds the Z1 and Z2 axes)	(Adds the Z axis)	(Recovers the Z2 axis)
X90 Y70 Z1=-30 Z2=-50	G1 X1=50 Z20	G0 X2=0 Y2=0 Z2=0
#FREE AX [Z1,Z2]	#FREE AX[Z]	M30
(Frees the Z1 and Z2 axes)	(Frees the Z axis)	
X0	X1=20	
#CALL AX [Z]	#CALL AX [Z1]	
(Recovers the Z axis)	(Recovers the Z1 axis)	
G0 X0 Y0 Z0	G0 X1=0 Y1=0 Z1=0	
M30	M30	

### Consultation variables

The information about the status of the synchronization marks may be consulted using the following variables.

- MEET or WAIT type mark expected by the "n" channel from the "m" channel
  - V.[n].G.MEETCH[m]
  - V.[n].G.WAITCH[m]

Replace the letters "n" and "m" with the channel number.

# 26.

PROGRAMMING STATEMENTS. Communication and synchronization between channels



CNC 8070

(REF: 2008)

- Status of the MEET or WAIT type "m" mark in the "n" channel

V.[n].G.MEETST[m]

V.[n].G.WAITST[m]

### #MEET

**It activates the mark indicated in the channel and waits for it to be activated in the rest of the programmed channels.**

This instruction, after activating the mark in its own channel, waits for it to also be active in the programmed channels before resuming the execution. Each channel has 100 marks that are numbered from 1 to 100.

Programming the same instruction in several channels, all of them stop and wait for the rest to reach the indicated point before they all resume the execution at the same time from that point on.

The programming format is:

#MEET [<mark>, <channel>,...]

Parameter	Meaning
<mark>	Synchronization mark that is activated in the channel itself and must be activated in the rest of the channels before going on.
<channel>	Channel or channels where the same mark must be activated.

There is no need to include the number of its own channel in each instruction because the mark is activated when executing the #MEET instruction. However, it is recommended to program it in order to make the program more understandable.

### Operation

Programming the same instruction in each channel, all of them are synchronized at that point and the execution resumes from there on. It works as follows.

- 1 It activates the mark selected in its own channel.
- 2 It waits for the mark to be activated in all the indicated channels.
- 3 After synchronizing the channels, it deletes the mark from its own channel and goes on executing the program.

Each channel stops on its #MEET. When the last one of them reaches the command and checks that all the marks are active, the process unlocks for all of them at the same time.

In the following example, it waits for mark ·5· to be active in channels ·1·, ·2· and ·3· to synchronize the channels and resume the execution.

CHANNEL 1	CHANNEL 2	CHANNEL 3
%PRG_1	%PRG_2	%PRG_3
...	...	...
...	#MEET [5,1,2,3]	...
#MEET [5,1,2,3]	...	...
...	...	...
...	...	#MEET [5,1,2,3]
M30	M30	M30

### #WAIT

**It waits for the mark to be activated in the indicated channel**

The #WAIT instruction waits for the indicated mark to be active in the specified channels. If the mark is already active when executing the command, the execution is not interrupted and the program keeps running.

Each channel has 100 marks that are numbered from 1 to 100.

The programming format is:

#WAIT [<mark>, <channel>,...]

Parameter	Meaning
<mark>	Synchronization mark waited for to be activated.
<channel>	Channel or channels that must activate the mark.

As opposed to the #MEET instruction, it does not activate the indicated mark of its own channel. The marks of the channel are activated using the instruction #SIGNAL.

**#SIGNAL**  
**It activates the mark in its own channel**

The #SIGNAL instruction activates the indicated marks in its own channel. Each channel has 100 marks that are numbered from 1 to 100. These marks correspond to the #WAIT instructions.

This instruction does not perform any wait; it goes on executing. Once synchronized, the marks are deactivated, if so wished, using the #CLEAR instruction.

The programming format is:

```
#SIGNAL [<mark>,...]
```

Parameter	Meaning
<mark>	Synchronization marks that is activated in the channel.

**#CLEAR**  
**It clears the synchronism marks of the channel**

This instruction activates the indicated marks in its own channel. If no marks are programmed, it deletes all of them.

The programming format is:

```
#CLEAR  

#CLEAR [<mark>,...]
```

Parameter	Meaning
<mark>	Synchronization marks that is deleted in the channel.

In the following example, channels ·1· and ·2· wait for mark ·5· to be active in channel ·3· to synchronize. When mark ·5· is activated in channel ·3·, it resumes the execution in all three channels.

CHANNEL 1	CHANNEL 2	CHANNEL 3
%PRG_1	%PRG_2	%PRG_3
...	...	...
...	#WAIT [5,3]	...
#WAIT [5,3]	...	...
...	...	#SIGNAL [5]
...	...	...
...	...	#CLEAR [5]
M30	M30	M30

**26.**

PROGRAMMING STATEMENTS.  
 Communication and synchronization between channels



CNC 8070

(REF: 2008)

## 26.15 Movements of independent axes



*This function has a specific manual. This manual that you are reading now only offers some information about this function. Refer to the specific documentation to obtain further information regarding the requirements and operation of the independent axes.*

The CNC has the possibility of executing independent positioning and synchronization. For this type of movements, each CNC axis has an independent interpolator that keeps track of the current position on its own without depending on the tracking of the general interpolator of the CNC.

It is possible to execute an independent movement and general movement simultaneously. The result will be the sum of the two interpolators.

The CNC stores up to a maximum of two independent-motion instructions per axis. The rest of instructions sent when there are two pending execution imply a wait from the part-program.

### Treatment of a rotary axis as an infinite axis.

Axis synchronization permits treating a rotary axis as an infinite axis making it possible to increase the feedback count of the axis indefinitely (without limits) regardless of the value of the module. This type of axis is activated when programming by adding the prefix ACCU to the name of the master axis. From the moment this is programmed, the CNC uses the variable V.A.ACCUDIST.xn that may be initialized at any time to follow up on the axis.

This feature is useful, for example, for a rotary axis or encoder that moves an endless conveyor belt on where the part is. The infinite axis treatment allows synchronizing the coordinate of the conveyor belt with an external event and, this way, count the moving distance of the part beyond the value of the module of the rotary axis that moves the belt.

### Restrictions for the independent axes

Any axis of the channel may be moved independently using the associated instructions. However, this function presents the following restrictions.

- A spindle can only move independently when set in axis mode with the instruction #CAX . However, it can always be the master of a synchronization.
- A rotary axis may be of any module, but the lower limit must always be zero.
- A Hirth axis cannot move independently.

### Synchronizing the interpolators

In order for the incremental movements to take the real coordinate of the machine into account, each interpolator must be synchronized with that real coordinate. The synchronization is done from the part-program using the instruction #SYNC POS.

Resetting the CNC synchronizes the theoretical coordinates of both interpolators with the real coordinate. These synchronizations will only be necessary when inserting instructions of both types of interpolators.

Every time the program is initiated or an MDI block is executed, the coordinate of the general interpolator of the CNC is synchronized and every new independent instruction (without any one pending) also synchronizes the coordinate of the independent interpolator.

### Influence of the movements in block preparation

None of these blocks interrupt block preparation, but they do interrupt the interpolation. Therefore, it will not blend two blocks, there will be an intermediate one.

# 26.

**PROGRAMMING STATEMENTS.**

Movements of independent axes

**FAGOR** 

FAGOR AUTOMATION

**CNC 8070**

(REF: 2008)

## Positioning move (#MOVE)

The various types of positioning are programmed with the following instructions.

- #MOVE - Absolute positioning move.
- #MOVE ADD - Incremental positioning move.
- #MOVE INF - Infinite (endless) positioning move.

The programming format for each of them is the following. Optional parameters are indicated between the <> characters.

- #MOVE <ABS> [Xpos <,Fn> <,blend>]
- #MOVE ADD [Xpos <,Fn> <,blend>]
- #MOVE INF [X+/- <,Fn> <,blend>]

### [ Xpos ] Axis and position to reach

Axis and position to reach. With #MOVE ABS it will be defined in absolute coordinates whereas with #MOVE ADD it will be defined in incremental coordinates.

The moving direction is determined by the coordinate or the increment programmed. For rotary axes, the moving direction is determined by the type of axis. If normal, via the shortest path; if unidirectional, in the preset direction.

### [ X+/- ] Axis and moving direction

Axis (without coordinate) to position. The sign indicates the moving direction.

It is used with #MOVE INF to execute an endless (infinite) movement until the axis limit is reached or until the movement is interrupted.

### [ Fn ] Positioning feedrate

Positioning feedrate.

Feedrate given in mm/min, inches/min or degrees/min.

Optional parameter. If not defined, it assumes the feedrate set by machine parameter POSFEED.

### [ blend ] Dynamic blend with the next block

Optional parameter. The feedrate used to reach the position (dynamic blend with the next block) is defined by an optional parameter.

The feedrate used to reach the position is given by one of these elements:

[blend]	Type of dynamic blend
PRESENT	It reaches the indicated position at the positioning feedrate specified for the block itself.
NEXT	It reaches the indicated position at the positioning feedrate specified in the next block.
NULL	The indicated position is reached at zero feedrate.
WAITINPOS	The indicated position is reached at zero feedrate and it waits to be in position before executing the next block.

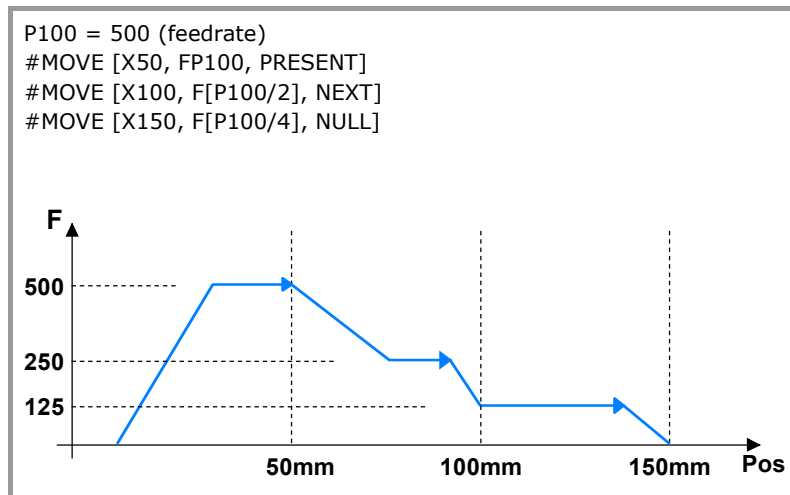
Programming this parameter is optional. If not programmed, the dynamic blend is carried out according to machine parameter ICORNER as follows.

ICORNER	Type of dynamic blend
G5	According to the setting for the PRESENT value.
G50	According to the setting for the NULL value.
G7	According to the setting for the WAITINPOS value.

26.

PROGRAMMING STATEMENTS.  
Movements of independent axes





## Synchronization move (#FOLLOW ON)

The activation and cancellation of the different types of synchronization are programmed with the following instructions.

- #FOLLOW ON - Activates the synchronization movement (real coordinates).
- #TFOLLOW ON - Activates the synchronization movement (theoretical coordinates).
- #FOLLOW OFF - Cancels the synchronization movement.

The programming format for each of them is the following. Optional parameters are indicated between the <> characters.

- #FOLLOW ON [master, slave, Nratio, Dratio <,synctype>]
- #TFOLLOW ON [master, slave, Nratio, Dratio <,synctype>]
- #FOLLOW OFF [slave]

Executing the #FOLLOW OFF instruction involves eliminating the synchronization speed of the slave. The axis will take some time to brake and the instruction will stay in execution during that time. When the synchronization is activated from the program, it is necessary to program the instruction #FOLLOW OFF before M30, since the M30 does not override the synchronization.

### [ master ] Master axis

Name of the master axis.

To treat a rotary axis as an infinite axis making it possible to increase the feedback count of the axis indefinitely (without limits) regardless of the value of the module, program the master axis with the prefix ACCU. This way, the CNC does a follow-up of the axis through the variable V.A.ACCUDIST.xn.

### [ slave ] Slave axis

Name of the slave axis.

### [ Nratio ] Gear ratio (slave axis)

Numerator of the gear ratio. Turns of the slave axis.

### [ Dratio ] Gear ratio (master axis)

Denominator of the gear ratio. Rotations of the master axis.

**[ synctype ] Type of synchronization**

Optional parameter. Indicator that determines whether it is a velocity or position synchronization type.

[ synctype ]	Type of synchronism
POS	It is a position synchronization.
VEL	It is a velocity synchronization.

Programming it is an option. If not programmed, it executes a velocity synchronization.

```
#FOLLOW ON [X, Y, N1, D1]
#FOLLOW ON [A1, U, N2, D1, POS]
#FOLLOW OFF [Y]
#FOLLOW ON [ACCUX, Y, N1, D1]
```

**26.****PROGRAMMING STATEMENTS.**

Movements of independent axes



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## 26.16 Electronic cams.



*This function has a specific manual. This manual that you are reading now only offers some information about this function. Refer to the specific documentation to obtain further information regarding the requirements and operation of the electronic cams.*

The electronic cam mode may be used to move a slave axis based on a positions-table or on a cam profile. If while executing a cam profile, a second cam profile is executed, the second one gets ready and waiting for the current execution to be over. Once the current cam profile has ended, it starts executing the second cam blending both profiles like when blending (joining) two positioning blocks. Executing the cam synchronization ending instruction (`#CAM OFF`) will end the current cam, but not immediately, it will do so the next time it goes through the end of the cam profile.

After executing the synchronization of the cam, no positioning moves of the independent axis (MOVE) are possible. It makes no sense to add another movement to the cam synchronization movement that would brake the synchronization already achieved.

### Position - position cam

On this type of cam, it is possible to obtain non-linear electronic synchronization ratios between two axes. Thus, the position of the slave axis is synchronized with the position of the master axis using a cam profile.

### Position - time cam

On this type of cam, it is possible to obtain other movement profiles different from trapezoidal or S-shaped ones.

### Electronic-cam editor.

Before activating an electronic cam, it must have been defined at the cam editor which is accessed from the machine parameters. This editor offers a friendly assistance to analyze the behavior of the cam projected through graphically assisted data entry for speed, acceleration and jerk.

It is up to the user to select the parameters and the functions to design an electronic cam and must make sure that his design is coherent with the required specifications.

### Activating and canceling the file cam from the part-program.

The data of a cam may be defined in a file that may be loaded from the CNC or from the PLC. When executing a cam from a file, the CNC reads its data dynamically and, consequently, the number of points to define the cam is unlimited. After selecting a file cam, it stays available until the cam table of the machine parameters is validated or the CNC is turned off.

To select or cancel a file cam, use the following instructions. The following instructions only defines the location of the cam; to activate it, use the instruction `#CAM ON`.

`#CAM SELECT` - Select a file cam.  
`#CAM DESELECT` - Cancel the cam of a file.

The programming format for each of them is the following.

`#CAM SELECT [cam, file]`  
`#CAM DESELECT [cam]`

Parameter.	Meaning.
cam	Cam number.
path/file	Name and path of the file containing the cam data.

`#CAM SELECT [6, "C:\USERCAM\cam.txt"]`  
 (The CNC uses for cam ·6· the data defined in the cam.txt file)  
`#CAM DESELECT [6]`  
 (The CNC stops using for cam ·6· the data defined in the cam.txt file)

26.

PROGRAMMING STATEMENTS.  
 Electronic cams.

**FAGOR** 

FAGOR AUTOMATION

CNC 8070

(REF: 2008)

## Activating and canceling the electronic cam (#CAM).

The activation and cancellation of the electronic cam is programmed with the following instructions.

- #CAM ON - Activates the cam (real coordinates).
- #TCAM ON - Activates the cam (theoretical coordinates).
- #CAM OFF - Cancel the electronic cam.

The programming format for each of them is the following. Optional parameters are indicated between the <> characters.

```
#CAM ON [cam, master/"TIME", slave, master_off, slave_off, range_master,
range_slave <,type>]
#TCAM ON [cam, master/"TIME", slave, master_off, slave_off, range_master,
range_slave <,type>]
#CAM OFF [slave]
```

Executing the #CAM OFF instruction involves eliminating the synchronization of the cam. Once this instruction has been programmed, the cam ends when reaching the end of its profile.

### [cam] Cam number.

To activate a cam, it must have been previously defined at the cam editor, within the machine parameters.

### [master/"TIME"] Master axis.

Name of the master axis, when it is a position cam. When programming the "TIME" command instead of an axis name, the cam is interpreted as being a time cam.

In a position cam, treating a rotary axis as an infinite axis making it possible to increase the feedback count of the axis indefinitely (without limits) regardless of the value of the module, program the master axis with the prefix ACCU. This way, the CNC does a follow-up of the axis through the variable V.A.ACCUDIST.xn.

```
#CAM ON [1, X, Y, 30, 0, 100, 100]
#CAM ON [1, ACCUX, Y, 30, 0, 100, 100]
#CAM ON [1, TIME, A2, 0, 0, 6, 3, ONCE]
#CAM OFF [Y]
```

### [slave] Slave axis.

Name of the slave axis.

### [master\_off] Offset of the master axis or time offset.

In a position cam, this offset sets the position where the cam is activated. The offset is subtracted from the master axis position to calculate the entry position of the cam table.

In a time cam this offset may be used to set a time for the cam trigger.

### [slave\_off] Offset of the master axis.

The slave\_off and range\_slave values allow moving the slave axis positions outside the range of values set by the cam function.

### [Range\_master] Master axis activation scale or range.

The position cam is activated when the master axis is between the positions "master\_off" and "master\_off + range\_master". The cam only regulates the slave axis position within this range.

In a time cam, this parameter sets the time range or total duration of the cam.

### [Range\_slave] Slave axis activation scale or range.

The cam applies it to the slave axis when the slave axis is between "slave\_off" and "slave\_off + range\_slave".

26.

PROGRAMMING STATEMENTS.

Electronic cams.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)

**[type] Cam type.**

Depending on the execution mode, the time cams and the position cams may be of two different types; i.e. periodic or non-periodic. This selection is made using the following commands:

<b>[type]</b>	<b>Meaning.</b>
ONCE	Non-periodic cam. This mode maintains the synchronization for the range defined for the master axis. If the master axis moves backwards or if it is a module, the slave axis will keep on executing the cam profile until the cancellation is programmed.
CONT	Periodic cam. In this mode, when reaching the end of the range of the master axis, it calculates the offset to execute the cam again shifted in the amount of that range. In other words, identical cams are executed along the path of the master axis.

If the master axis is a rotary module and the cam definition range is that module, the two execution modes are equivalent. Either mode maintains synchronization until the #CAM OFF instruction is executed. When reaching that instruction, the execution of the cam will end the next time the end of the cam profile is reached.

## 26.17 On line modification of the machine configuration in HD graphics (xca files).

The CNC offers several xca files, one per model, containing the definition and configuration of the machine for HD graphics. On power-up, the CNC assumes the last file used. These files cover most configurations; therefore, new xca will have to be generated when the machine has some special requirement that affects graphics.

If the physical configuration of the machine changes during execution (for example, spindle change with different number of axes), the corresponding xca file must be loaded so the graphics show the changes. The xca files may be loaded either from the softkey menu or from the program using the instruction #DEFGRAPH.

When changing the machine configuration, the CNC saves the part displayed on the screen automatically as LastPiece.stl into the folder ../Users/Grafdata, and it recovers it after the new configuration.

### Programming.

This instruction must be programmed alone in the block. When programming this instruction, the name of the file must be defined and, optionally, its location.

#### Programming format.

The programming format is the following; the list of arguments appears between curly brackets and the optional ones between angle brackets.

```
#DEFGRAPH [{"<{path\}>{file.xca}"]
```

{path\}           Optional. File location.

{file.xca}        Name of the file.

```
#DEFGRAPH ["Machine.xca"]
#DEFGRAPH ["c:\FagorCnc\MTB\Grafdata\Machine.xca"]
```

#### Path definition.

Defining the path is optional. If defined, the CNC only looks for the file in that folder; if not defined, the CNC looks for the file in the default folder ../MTB\Grafdata. If the file is missing, the CNC issues the relevant error message.

### Remarks

The machine configuration files supplied by Fagor consist of a single file, the xca. When an OEM creates his own configuration files, for each xca file, he must create a ".def" file with the same name that completes the configuration of the axes involved in the kinematics. Both files must be copied when saving the configuration file in another folder.

# 26.

**PROGRAMMING STATEMENTS.**

On line modification of the machine configuration in HD graphics (xca files).



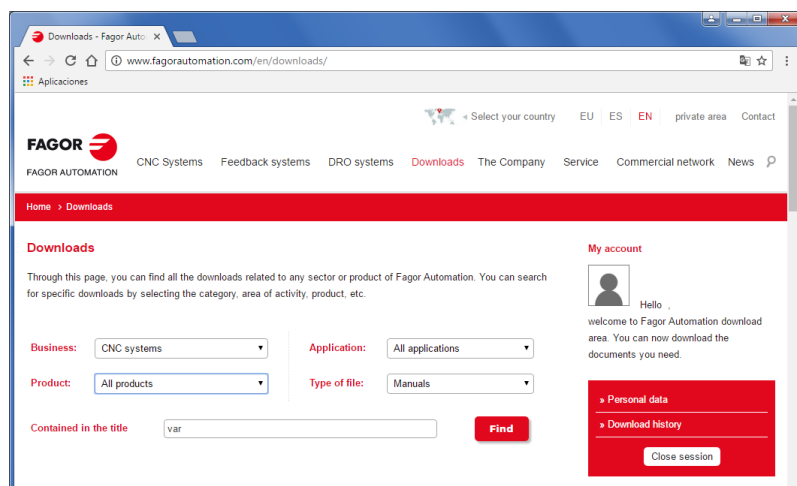
FAGOR AUTOMATION

CNC 8070

(REF: 2008)

All information regarding CNC variables can be found in the manual on "CNC Variables", which can be downloaded from the Fagor Automation corporate website. The electronic document is called man\_8070\_var.pdf.

<http://www.fagorautomation.com/en/downloads/>



# 27.

## CNC VARIABLES.



FAGOR AUTOMATION

CNC 8070

(REF: 2008)











FAGOR AUTOMATION

**Fagor Automation S. Coop.**

Bº San Andrés, 19 - Apdo. 144  
E-20500 Arrasate-Mondragón, Spain

Tel: +34 943 719 200

+34 943 039 800

Fax: +34 943 791 712

E-mail: [info@fagorautomation.es](mailto:info@fagorautomation.es)

[www.fagorautomation.com](http://www.fagorautomation.com)