Servo Motor Compensation Features

Galil motor controllers provide a compensation filter, which includes a PID (Proportional-Integral-Derivative) filter followed by a notch filter and a low-pass filter. The compensation also includes velocity and acceleration feedforward. All filter parameters are adjustable, allowing servo system tuning for best performance. Dual loop control is provided for reducing the effect of backlash.

The dual-loop (DV) feature enables the controller to compensate for mechanical backlash. Typically, dual-loop systems use a rotary encoder on the motor and a linear encoder on the load (most Galil controllers accept inputs from two encoders per axis as a standard feature). Dual-loop control changes the standard PID control and closes the position loop with the load encoder ("PI") and derives the damping terms ("D") from the motor encoder. This method provides smooth and accurate control along the motion path regardless of backlash.

Most Galil motion controllers also include a sinusoidal commutation feature that allows designers to use lower-cost servo drives. This feature assures smooth motion and reduces torque ripple when using brushless motors. Each axis of sinusoidal commutated motion requires two DAC outputs that are phase shifted by 120°. The servo amplifier generates the third commutation signal. The commutation can be initialized with or without hall sensors. Two controller axes are required for each brushless motor. For example, a two-axis controller is required to drive one brushless motor with sinusoidal commutation.

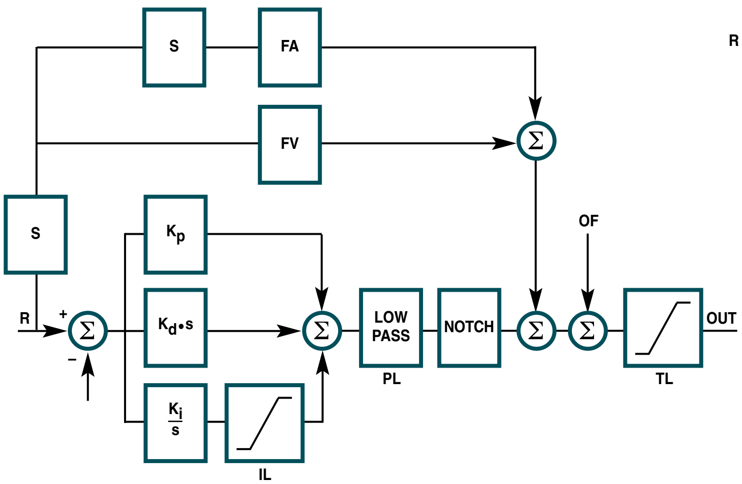
Command Language

Galil's Command Language is comprised of intuitive, two-letter, English-like ASCII commands that make programming quick and easy. For example, the "BG" command begins motion while the "SP 2000, 4000" command sets the speed of the X-axis as "2000" and the Y-axis to "4000". Commands are included for system set-up, tuning, prescribing motion, error handling and application programming. Custom commands can be created upon request.

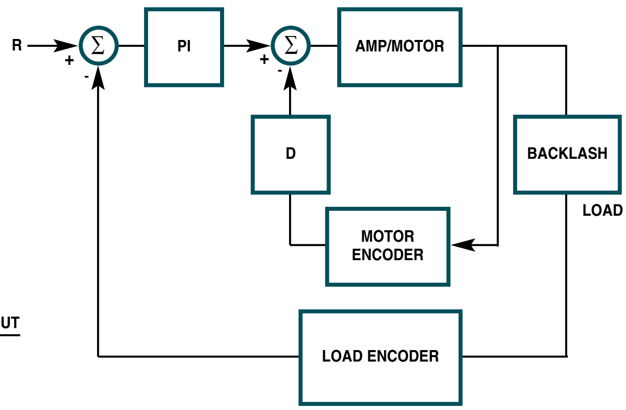
One of the more powerful features of all Galil controllers is their ability to store and execute complex application programs designed by the user. Application programs can be downloaded directly to the controller and executed without host intervention. The main benefit is that this frees the PC for system-level tasks. In fact, Galil controllers permit multitasking, which allows up to eight programs to execute simultaneously. Also, special commands are available for application programming including event triggers, IF/THEN/ELSE statements, conditional jumps, subroutines, symbolic variables and arrays.

A list of typical DMC commands is shown at the end of this section.

PID Block Diagram



Dual-loop Block Diagram



I/O

Error Handling

Dedicated I/O is provided for the following safety controls: forward and reverse limit inputs for each axis, home input for each axis, amplifier enable output for each axis, configurable abort inputs for each axis, master abort input, and error output. Most Galil drives have an electronic lockout input (ELO). Also, the controller provides the following safety functions in software: upper and lower software travel limits, position error limits, and automatic shut-off on excess position error. Program interrupts are provided for error and limit conditions and run-time program errors. The program interrupts cause the program sequencer to automatically branch to an error handling subroutine. In order to provide flexibility and system protection, the error handling subroutine can be customized by the user.

User I/O

In addition to dedicated inputs for home and limits, Galil controllers provide user I/O for synchronizing motion with external events such as switches and relays. The DMC-18x6 controller, for example, includes 8 analog inputs, 8 digital inputs and 8 digital outputs for 1 to 4-axis models; and 8 analog inputs, 24 digital inputs and 16 digital outputs for 5–8 axis models. All Galil controllers include many commands for handling I/O such as input interrupts, I/O triggers and timers. The combination of user I/O and application programming often eliminates the need for a PLC. When extra I/O is needed, Galil provides daughter boards and remote I/O units such as the RIO Pocket PLC to expand a controller's I/O capability.

As part of the user I/O, Galil controllers provide a high-speed position capture and position compare feature for each axis. The high-speed position capture latches the exact position within 0.1 microseconds (40 µsec with optoisolation) of the occurrence of an input. Position capture is crucial for applications requiring precise synchronization of position to external events such as coordinate measurement machines.

The high-speed position compare feature produces an output pulse at a precise position. The starting position for the initial pulse and incremental distance for subsequent pulses are programmable.

Modes of Motion

Point-to-Point Motion

Any combination of axes can be operated in the Point-to-Point Motion mode to allow the target position (PA or PR), slew speed (SP), acceleration (AC) and deceleration (DC) to be specified independently for each axis. Upon begin (BG), the controller generates a trapezoidal velocity profile where the speed and acceleration can be changed anytime during motion. For applications that require smooth motion without abrupt velocity transitions, a motion smoothing function (IT) is provided. The position (TP) and position error (TE) may be interrogated at any time.

Example 1—Point-to-Point Motion

PROGRAM	INTERPRETATION
AC 1000000;DC 1000000	*Specify acceleration and deceleration*
SP20000	*Specify slew speed*
PR40000	*Specify distance*
BG	*Begin motion*

Position Tracking

The Position Tracking mode allows an axis to precisely follow a dynamic position target. In this mode, a new absolute position may be specified even if the axis is in motion. The controlled axis is commanded to move to the new position following a trapezoidal velocity profile.

The (PT) command places the controller in the Position Tracking mode, which allows the host to issue absolute position commands on the fly. The axis moves to the new position and waits until a new position target is specified and given by the (PA) command. The (ST) Stop command is used to exit the Position Tracking mode.

Example 2—Change Speed on Input, Position Tracking

Move the x-axis forward a distance of 20,000 counts at an initial speed of 50,000 counts/sec and with an acceleration and deceleration of 1,000,000 counts/sec^2. Once the sensor connected to input 1 triggers, reduce the speed to 25,000 counts/sec. Upon motion complete, begin position tracking mode and follow the target as updated by a host PC. Activation of input 2 will end motion. Note: multiple commands can be issued on the same line to conserve program space and give command priority while multitasking.

PROGRAM	INTERPRETATION
#A	*Label*
PR20000;SP50000	*Relative Move, Speed*
AC1000000;DC1000000	*Accel and Decel*
BGX	*Begin motion*
AI1	*Trip point: Wait for sensor input*
SP25000	*Reduce speed*
AMX	*Wait for original distance to profile*
PT1	*Turn on position tracking mode*
target=_RPX	*Set target to current commanded position*
#LOOP	*Label*
PAX=target	*Track target updated by host*
JP#LOOP,@IN[2]=1	*Repeat unless input two is tripped*
STX;AMX;EN	*End position tracking mode and program*

Modes of Motion

Jogging

In the jog mode, each axis is given a jog speed and direction (JG), acceleration (AC), and deceleration (DC). Upon begin (BG), the controller ramps up to the jog speed at the prescribed acceleration following a trapezoidal profile. A smoothing function (IT) is provided to smooth abrupt velocity transitions. The stop command (ST) stops the motion at the prescribed deceleration rate. The jog speed and direction, acceleration and deceleration may be changed at any time during motion. The average speed can be interrogated at any time using the Tell Velocity (TV) command.

Example 3 — Joystick with Coarse/Fine Speed Control

To control the motor velocity by a potentiometer, connect it to analog input #1 and read its voltage. Set the motor speed in proportion to the analog input with a maximum speed of 100,000 counts/sec for a 10 Volt input. Also, limit the acceleration and deceleration to 500,000 counts/sec^2. The speed scale is selectable by input 1 for fine or coarse velocity.

PROGRAM	INTERPRETATION
#AUTO	*#AUTO label executed on powerup*
JG0	*Initial Speed*
AC500000;DC500000	*Accel and Decel*
BGX	*Begin Jog mode*
#LOOP	*Label*
scale=(9*@IN[1])+1	*Set scaling, 1 (fine) or 10 (coarse)*
JG@AN[1]*1000*scale	*Read pot and update speed*
JP#LOOP	*Repeat*
EN	*End Program*

2D Linear and Circular Interpolation (for controllers with two or more axes)

The Vector Mode (VM) is an extremely powerful mode where any two-dimensional path consisting of straight-line (VP) and arc segments (CR) can be prescribed. Up to 511 segments can be given prior to the start of motion and additional segments can be sent during motion allowing unlimited motion paths to be followed without stopping. The vector speed (VS), vector acceleration (VA), vector deceleration (VD), and motion smoothing (VT/IT) are also prescribed. The vector speed can be changed at any time during motion, permitting feedrate override, slow down around corners and assignment of different speeds to specific segments. Setting the vector speed ratio (VR) to zero and increasing the vector ratio to resume can easily accomplish a pause during motion.

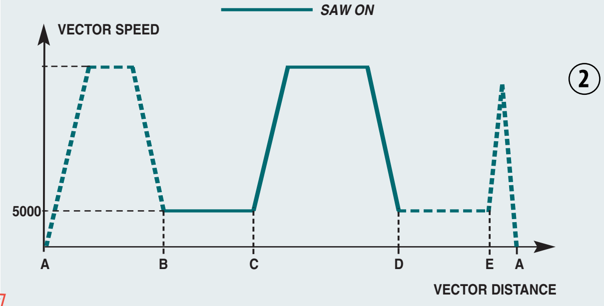
The vector mode can be operated on two sets of coordinated axes at the same time using the (CA) command, which specifies the plane of motion as S and T. By having dual sets of coordinated motion, users can accomplish completely separate coordinated motion tasks with a single controller. It can even handle more complex motion control functions such as collision avoidance.

Another feature of the vector mode is tangential following that allows a third axis to remain tangent to the trajectory, which is ideal for cutting tools. Helical motion is also possible by commanding the third axis to follow the coordinated path at the same rate.

Example 4 — Vector motion with tangential following and curve slowdown

Perform a move along the trajectory shown in figure 1 starting at the point A and move counter clockwise toward B. Due to accuracy requirements, the vector speed must be limited to 5,000 counts/sec on the circular segments BC and DE. On the linear segments, the motor speed is limited to 25,000 counts/sec. This operation is simplified given the controller's ability to associate two speeds with each segment—upper and lower limits. These limits are designated by the < and > symbols. The resulting vector speed is shown as a function of the path in figure 2 below. A saw is attached to Axes Z and is lowered externally by setting bit 2 and turned on by setting bit 1. The blade will stay tangent to the vector path through the tangential following mode.

PROGRAM	INTERPRETATION
#PATH	*Label*
CAS	*Set coordinate system*
VMXYZ	*Define XY plane, Z is tangent*
TN100,0	*Setup Tangential following*
VA500000;VD500000	*Vector mode accel and decel*
VP4000,0<25000>5000	*Segment AB, slows for curve*
CR500,-90,180	*Arc segment BC*
VP-1000,1000<25000>5000	*Segment CD*
CR500,90,180	*Arc segment DE*
VP0,0<25000	*Segment EA*
VE	*Indicate end of path*
PAZ=_TN;BGZ;AMZ	*Orient saw blade to tangent*
BGS	*Start motion sequence*
AV4000	*Wait 4000 vector distance (B)*
SB1;WT100;SB2	*Turn on and lower saw*
AV6571	*Wait 6571 vector distance (D)*
CB2;WT500;CB1	*Raise and turn off saw*
EN	*End program*



Modes of Motion

Electronic Gearing

The electronic gearing mode makes it easy for Galil controllers to simulate the motion of mechanical gears electronically. Any slave axis or set of slave axes can be geared to a master at a prescribed gear ratio defined by the (GR) command. The gear ratio can be changed on-the-fly and the controller permits multiple masters as defined by the (GA) command. A powerful feature of electronic gearing is that an axis can be geared and simultaneously be commanded to perform an independent or vector move. This is useful for the position correction required in packaging applications or when shapes must be cut on a moving conveyer belt. The electronic gearing mode is also useful for gantry applications where a special gantry mode (GM) command tightly couples two axes by ensuring that gearing cannot be disabled.

The gearing mode allows for a gradual ramp-to-gearing which results in smoother transitions when the gear ratio is changed. (GD) sets the distance of the master axis over which the slave will be engaged or changed to a new gear setting. The parameter (_GP) corrects for any accumulated errors in gearing during the ramp-to-gearing phase.

Example 5 — Electronic Gearing with Correction

Gear Axis X and Z to Y with gear ratios of 2 and -4 respectively. Output the absolute single turn position for X at regular intervals. Assume the resolution of the X axis is 4000 counts per revolution. Upon input 1, automatically issue a correction movement superimposed upon the concurrent gearing.

PROGRAM	INTERPRETATION
#GEAR	*Label*
GAY,,Y	*Specify Y axis as master for X and Z*
GR2,,-4	*Specify gear ratios for X and Z*
PRY=50000;SPY=10000	*Specify relative move and speed of Y*
ACY=1000000;DCY=1000000	*Specify Accel and Decel of Y*
II1	*Setup Input Interrupt on input 1*
BGY	*Begin motion on Y axis. X & Z gear*
#POS	*Label*
abposx=_TPX%4000	*Current position modulo encoder resolution % available on Accelera Class*
MGabposx	*Message current single turn position*
WT500	*Wait 500 ms*
JP#POS	*repeat*
EN	*End of program*
#CORRECT	*Label for #CORRECT*
IP-1000,,-1000	*X and Y move back 1000 counts, gearing is still engaged.*
EN	*End of correction program*
#ININT	*Automatically run on input 1*
XQ#CORRECT,1	*Run #CORRECT in separate thread*
AI1	*Wait for input 1 to clear*
RI	*Return from Interrupt*

Contouring

The contouring mode (CM) is extremely flexible as it allows any arbitrary profile on any set of axes to be prescribed. Here, the user bypasses the controller profiler and directly inputs the position versus time trajectory. The trajectory is described as the position increment (CD) over a defined time period (DT). Additionally, the controller performs linear interpolation between prescribed points. The contour mode is useful for following complex, computer-generated paths or for "teaching" position paths. An automatic data-recording feature allows the controller to "learn" a path and then follow it in the contour mode.

Example 6 — Contour circle with buffer monitoring

Fill arrays with contour data inscribing a circle with radius of 50000 counts. Contour the data at a time interval of 4 samples. The Accelera series incorporates a buffer of 512 contour segments to allow caching of contour data. The program will monitor this buffer to avoid overruns and fill it with more data when possible.

PROGRAM	INTERPRETATION
#CONTOUR	*Label*
radius=50000	*Set radius variable*
DMcdx[720];DMcdy[720]	*Dimension arrays for data*
i=0;d=0	*Index and degrees variables*
#CALC	*Label*
cdx[i]=radius*@COS[d]-radius	*Calculate shifted Cosine data*
cdy[i]=radius*@SIN[d]	*Calculate sin data*
d=d+0.5	*Increment degrees*
i=i+1	*Increment index*
JP#CALC,i<720	*Repeat until arrays are full*
i=0	*Reset index*
CMXY	*Start Contour mode*
DT2	*Setup time slice,* 2^2
curx=0;cury=0	*Set incremental reference*
#PLAY	*Label*
CD (cdx[i]-curx),(cdy[i]-cury)	*Contour with incremental calculation*
curx=cdx[i];cury=cdy[i]	*Update incremental reference*
i=i+1	*Increment index*
JP#PLAY,(_CM>0)&(i<720)	*Repeat while buffer not full & more data remains*
JP#END,i=720	*Jump to end if data done*
#WAIT;JP#WAIT,_CM<100	*Wait until buffer has plenty of space*
JP#PLAY	*Jump back to continue playback*
#END	*Label for exit*
CD0,0=0	*Stop contour mode*
EN	*End of program*

Modes of Motion

PVT

The PVT mode of motion allows arbitrary motion profiles to be defined by position, velocity and time individually on all 8 axes. This motion is designed for systems where the load must traverse a series of coordinates with no discontinuities in velocity. By specifying the target position, velocity and time to achieve those parameters the user has control over the velocity profile. Taking advantage of the built in buffering the user can create virtually any profile including those with infinite path lengths.

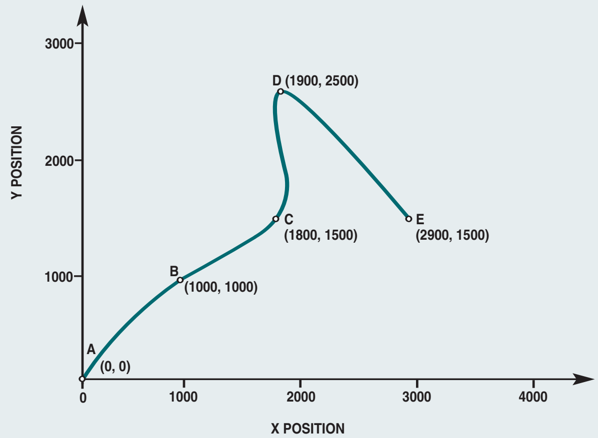
PVT segments must be entered one axis at a time using the PVn command. The PV command includes the target distance to be moved and target velocity to be obtained over the specified timeframe. Positions are entered as relative moves, similar to the standard PR command, in units of encoder counts and velocity is entered in counts/second. The controller will interpolate the motion profile between subsequent PV commands using a 3rd order polynomial equation.

Electronic Cam

Any slave axis or set of slave axes can be linked to a master axis to simulate the motion of a mechanical cam. Here, the master axis can be a motor-driven axis or a master encoder. The Cam functions are specified by a table that allows complex profiles with varying position ratios to be prescribed. Any follower axis may be engaged or disengaged independently at specific points along a Cam cycle. This allows the user to select engagement and disengagement points where the speed change of the follower is minimal. The electronic Cam is an ideal mode for periodic operation, especially those requiring a varying position ratio along the motion cycle. Applications include flying shears, rotating knives, and packaging systems. Galil's Cam-generating software can assist the user in defining the Cam table.

Example 7 — PVT

PROGRAM	INTERPRETATION
PVA=1000,500,1024	Horizontal axis segment AB. Ending velocity 500
PVA=800,100,1024	Horizontal axis segment BC. Ending velocity 100
PVA=100,1000,1024	Horizontal axis segment CD. Ending velocity 1000
PVA=1000,0,1024	Horizontal axis segment DE. Ending velocity 0
PVA=0,0,0	Exit PVT mode on the A axis
PVB=1000,500,1024	Vertical axis segment AB. Ending velocity 500
PVB=500,1000,1024	Vertical axis segment BC. Ending velocity 1000
PVB=1000,100,1024,	Vertical axis segment CD. Ending velocity 100
PVB=-1000,0,1024	Vertical axis segment DE. Ending velocity 0
PVB=0,0,0	Exit PVT mode on the B axis
BT	Begin PVT motion on both axes
EN	End program



TECHNICAL REF

Modes of Motion

Linear Interpolation (for controllers with two or more axes)

The linear interpolation mode (LM) allows any arbitrary path of up to 8 axes to be defined as a set of linear segments (LI). The vector speed (VS), vector acceleration (VA), vector deceleration (VD), and vector smoothing (VT) are also defined. Up to 511 LI segments can be given prior to the start of motion and additional segments can be sent during motion to allow paths of unlimited length to be followed.

Example 8 — Linear Interpolation with High Speed Latch

Move a 3D Cartesian robot through the following points with the coordinates indicated in inches. Assume that the resolutions of all the axes are 1,000 counts/inch, and set the required speed to 1.2 inches/sec (1,200 counts/sec) and the acceleration and deceleration to 100 in/sec^2 (100,000 counts/sec^2). Note that the LM mode requires defining the segments in incremental form. A sensor will trigger a high speed latch on each axis to indicate a desired or reference position. The latch will store the current position within 40µsec of the sensor trip and the robot will return to this "set" position after the initial move.

Point	*Coordinates (inches)*	*Coordinates (counts)*	*Incremental length (LI argument)*
P0	(0,0,0)	(0,0,0)	0,0,0
P1	(4,2,1)	(4000,2000,1000)	4000,2000,1000
P2	(6,6,2)	(6000,6000,2000)	2000,4000,1000
P3	(8,8,0)	(8000,8000,0)	2000,2000,-2000

PROGRAM	INTERPRETATION
#ROBOT	*Label*
CAS	*Set coordinate system*
LMXYZ	*Define XYZ space*
VS1200;VA100000;VD100000	*Vector speed, Accel, Decel*
LI4000,2000,1000	*Segment P0-P1*
LI2000,4000,1000	*Segment P1-P2*
LI2000,2000,-2000	*Segment P2-P3*
LE	*End of sequence*
ALXYZ	*Arm latches for axes XYZ*
BGS	*Begin motion*
AMS	*Wait for motion to profile*
IF _ALX|_ALY|_ALZ	*Ensure axes have latched*
MG"Not all axes have latched"	*Message to operator*
ELSE	*If not all axes tripped sensor*
MG"Tracking back to latch positions"	*Message to operator*
LMXYZ	*Define XYZ space*
LI(_RLX-_RPX),(_RLY-_RPY),(_RLZ-_RPZ)	*Incremental distance back to latch*
LE	*End of sequence*
BGS	*Begin move back to latches*
AMS	*Wait for motion to profile*
MG"Robot in position"	*Message to operator*
ENDIF	*End of IF*
EN	*End of program*

Note: **There are many homing and positioning algorithms available.**

Galil Instruction Set*

Ethernet

DH	DHCP Configuration
HS	Handle switch
IA	Set IP address
IH	Open IP handle
IK	Ethernet port blocking
MB	Modbus
MW	Modbus wait
SA	Send command
SM	Subnet mask

Servo Motor

AF	Analog feedback
AG	Set amplifier gain
AU	Set current loop gain
AW	Report AMP-43040 bandwidth
DV	Dual loop operation
FA	Acceleration feedforward
FV	Velocity feedforward
IL	Integrator limit
KD	Derivative constant
KI	Integrator constant
KP	Proportional constant
NB	Notch bandwidth
NF	Notch frequency
NZ	Notch zero
OF	Offset
PL	Pole
SH	Servo here
TK	Peak torque
TL	Torque limit
TM	Sample time

Stepper Motor

KS	Stepper motor smoothing
LC	Low current
QS	Error magnitude
YA	Step drive resolution
YB	Step motor resolution
YC	Encoder resolution
YR	Error correction
YS	Stepper position maintenance

Internal Sine Commutation

BA	Brushless axis
BB	Brushless phase
BC	Brushless calibration
BD	Brushless degrees
BI	Brushless inputs
BM	Brushless modulo
BO	Brushless offset
BS	Brushless setup
BZ	Brushless zero

I/O

AL	Arm latch
AQ	Analog configuration
CB	Clear bit
CO	Configure I/O points
II	Input interrupt
OB	Define output bit
OC	Output compare function
OP	Output port
SB	Set bit
@AN[x]	Value of analog input x
@IN[x]	State of digital input x
@OUT[x]	State of digital output x

System Configuration

BN	Burn parameters
BP	Burn program
BR	Brush motor enable
BV	Burn variables and arrays
BW	Brake Wait
CC	Configure communications port
CE	Configure encoder type
CF	Configure unsolicited messages handle
CI	Configure communication interrupt
CN	Configure switches
CW	Data adjustment bit
DE	Define dual encoder position
DP	Define position
DR	Data record update rate
EI	Event interrupts
EO	Echo
IT	Independent smoothing
LB	LCD Bias contrast
^L^K	Program protect (Lock)
LU	LCD Update
LZ	Leading zeros format
MO	Motor off
MT	Motor type
PF	Position format
PW	Password
QD	Download array
RS	Reset
^R^S	Master reset
UI	User interrupt
VF	Variable format

Math Functions

@ABS[x]	Absolute value of x
@ACOS[x]	Arc cosine of x
@ASIN[x]	Arc sine of x
@ATAN[x]	Arc tangent of x
@COM[x]	1's complement of x
@COS[x]	Cosine of x
@FRAC[x]	Fraction portion of x
@INT[x]	Integer portion of x
@RND[x]	Round of x
@SIN[x]	Sine of x
@SQR[x]	Square root of x
@TAN[x]	Tangent
%	Modulus operator

Interrogation

ID	AMP ID
LA	List arrays
LL	List labels
LS	List program
LV	List variables
MG	Message command
QH	Query hall state
QR	Data record
QU	Upload array
QZ	Return data record information
RL	Report latch
RP	Report command position
^R^V	Firmware revision information
SC	Stop code
TA	Tell amplifier status
TB	Tell status
TC	Tell error code

Interrogation (cont.)

TD	Tell dual encoder
TE	Tell error
TH	Tell handle
TI	Tell input
TP	Tell position
TR	Trace program
TS	Tell switches
TT	Tell torque
TV	Tell velocity
TZ	Tell I/O configuration
WH	Which handle

Programming

BK	Breakpoint
DA	Deallocate variables/arrays
DL	Download program
DM	Dimension arrays
ED	Edit program
ELSE	Conditional statement
ENDIF	End of cond. statement
EN	End program
HX	Halt execution
IF	If statement
IN	Input variable
JP	Jump
JS	Jump to subroutine
NO	No-operation—for comments
RA	Record array
RC	Record interval
RD	Record data
RE	Return from error routine
REM	Remark program
RI	Return from interrupt routine
SL	Single step
UL	Upload program
XQ	Execute program
ZA	Data record variables
ZS	Zero stack
'	Comment

Error Control

BL	Backward software limit
ER	Error limit
FL	Forward software limit
LD	Limit disable
OA	Encoder failure
OE	Off-on-error function
OT	Encoder failure period
OV	Encoder failure voltage
TW	Timeout for in-position

Trippoint

AD	After distance
AI	After input
AM	After motion profiler
AP	After absolute position
AR	After relative distance
AS	At speed
AT	After time
AV	After vector distance
MC	Motion complete
MF	After motion—forward
MR	After motion—reverse
WT	Wait for time

Independent Motion

AB	Abort motion
AC	Acceleration
BG	Begin motion
DC	Deceleration
FE	Find edge
FI	Find index
HM	Home
HV	Home speed
IP	Increment position
IT	Smoothing time constant
JG	Jog mode
PA	Position absolute
PR	Position relative
PT	Position tracking
SD	Switch deceleration
SP	Speed
ST	Stop

Contour Mode

CD	Contour data
CM	Contour mode
DT	Contour time interval

PVT Mode

PV	Position, velocity, time
BT	Coordinate start

ECAM/Gearing

EA	ECAM master
EB	Enable ECAM
EC	ECAM table index
EG	ECAM go
EM	ECAM modulus
EP	ECAM interval
EQ	Disengage ECAM
ET	ECAM table entry
EW	ECAM widen
EY	ECAM cycle counter
GA	Master axis for gearing
GD	Engagement distance for gearing
GM	Gantry mode
_GP	Correction for gearing
GR	Gear ratio for gearing

Vector/Linear Interpolation

CA	Define vector plane
CR	Circular interpolation move
CS	Clear motion sequence
ES	Elliptical scaling
IT	Smoothing time constant
LE	Linear interpolation end
LI	Linear interpolation segment
LM	Linear interpolation mode
ST	Stop motion
TN	Tangent
VA	Vector acceleration
VD	Vector deceleration
VE	Vector sequence end
VM	Coordinated motion mode
VP	Vector position
VR	Vector speed ratio
VS	Vector speed
VV	Vector Velocity

*Typical Instructions. This list is for DMC-40x0 Accelera. Other controllers have a slightly different instruction set.