General Description

Command Language

Galil's Command Language is comprised of intuitive, two-letter, English-like ASCII commands that make programming as quick and easy as possible. 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 applications 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.

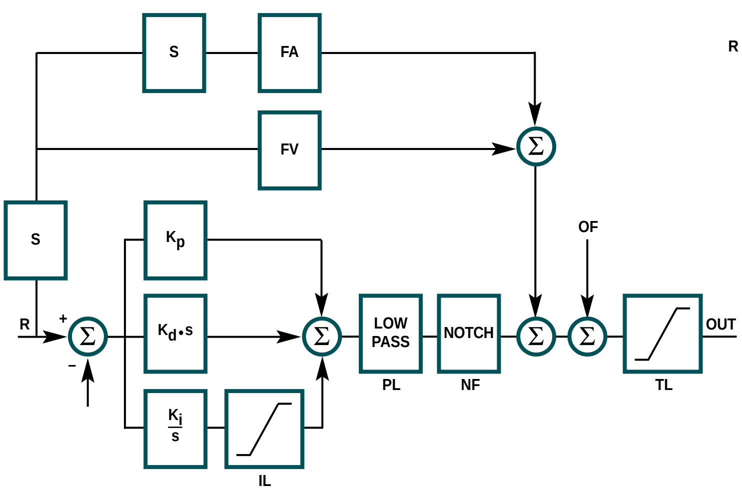
Servo Motor Compensation Features

Galil 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.

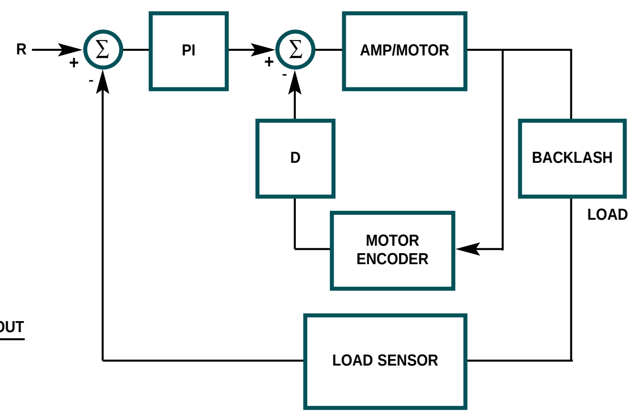
The dual-loop (DV) feature enables the controller to compensate for mechanical backlash. Typically, dual-loop systems use a rotary encoder on the motor and a linear encoder on the load (Galil Optima controllers accept inputs from two encoders per axis as a standard feature). Dual-loop control changes 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 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.

PID Block Diagram



Dual-loop Block Diagram



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. That way, motion on any axis can be started and stopped independently. 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 anytime.

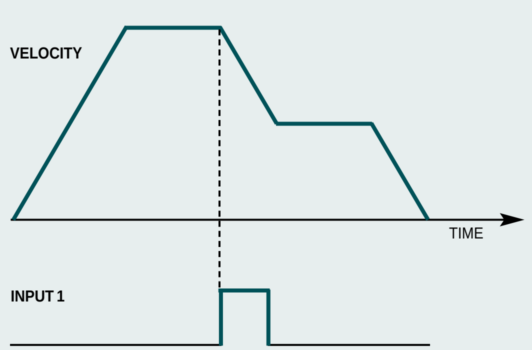
Example—Point-to-Point Motion

COMMAND	INTERPRETATION
AC 1000000	*Specify acceleration*
DC 1000000	*Specify deceleration*
SP 20000	*Specify slew speed*
PR 40000	*Specify distance*
BG	*Begin motion*

Example—Change Speed on Input

Move the x-axis forward a distance of 20000 counts at an initial speed of 50000 counts/sec and with an acceleration and deceleration of 1000000 counts/sec^2. As soon as the motor activates the sensor connected to input 1, reduce the speed to 25000 counts/sec.

PROGRAM	INTERPRETATION
#A	*Label*
PR 20000	*Distance*
SP 50000	*Initial speed*
AC 1000000	*Acceleration rate*
DC 1000000	*Deceleration rate*
BGX	*Start the motion*
AI 1	*Wait for the sensor input*
SP 25000	*Reduce the speed*
EN	*End program*



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 anytime during motion. The average speed can be interrogated at any time using the Tell Velocity (TV) command.

Example—Velocity Control by a Potentiometer

To control the motor velocity by a potentiometer, connect the potentiometer 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.

COMMAND	INTERPRETATION
#A	*Label*
JG 0	*Initial speed*
AC 500000	*Acceleration*
DC 500000	*Deceleration*
BGX	*Begin motion*
#LOOP	*Label*
JG@AN[1]*10000	*Read potentiometer 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) are also prescribed. The vector speed can be changed at anytime during motion, permitting feedrate override, slow down around corners and assignment of different speeds to specific segments. Setting the vector speed to zero and increasing the vector speed 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

General Description

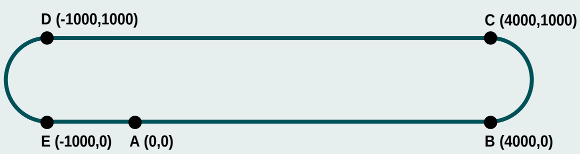
controller. It can even handle more complex motion control functions such as collision avoidance.

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—Coordinated Motion

Perform a move along the trajectory shown in the figure starting at the point A and moving counter clockwise toward B. Set the vector velocity to a uniform rate of 5000 counts/sec, and set the accel/decel rate along the vector to 500000 counts/sec^2.

COMMAND	INTERPRETATION
#PATH	*Label*
CAS	*Set coordinate system*
VMXY	*Define plane as XY*
VA 500000	*Acceleration*
VD 500000	*Deceleration*
VS 5000	*Vector speed*
VP 4000,0	*Move AB*
CR 500,-90,180	*Move BC*
VP -1000,1000	*Move CD*
CR 500,90,180	*Move DE*
VP 0,0	*Move EA*
VE	*Indicate end of path*
BGS	*Start motion sequence*
EN	*End program*

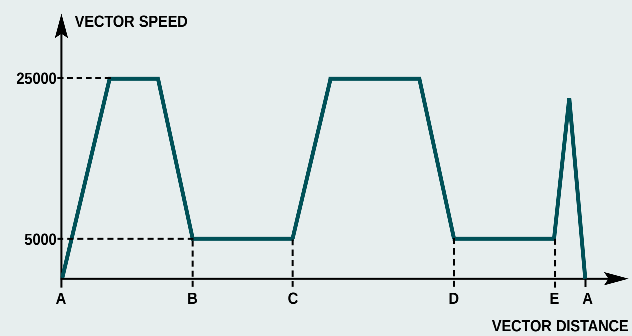


Example—Coordinated Motion with Maximum Feedrate

Repeat the previous example, but set the feedrate to the maximum values allowed. Due to accuracy requirements, the vector speed must be limited to 5000 counts/sec on the circular segments BC and DE. On the linear segment, 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 motion will accelerate the motors to the maximum allowed speed, but will assure that the speed at the end of the segment is the one indicated by the > symbol. The resulting vector speed is shown as a function of the path in the figure below.

COMMAND	INTERPRETATION
#B	*Label*
CAS	*Set coordinate system*
VMXY	*Define XY plane*
VA 500000	*Acceleration*
VD 500000	*Deceleration*
VP 4000,0 <25000>5000	*Segment AB*
CR 500,-90,180	*Segment BC*
VP -1000,1000 <25000>5000	*Segment CD*
CR 500,90,180	*Segment DE*
VP 0,0 <25000	*Segment EA*
VE	*Specify end of path*
BGS	*Begin motion sequence*
EN	*End program*



General Description

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—Linear Interpolation

Move a cartesian robot through the following points with the coordinates indicated in inches.

P0 (0,0,0)

P1 (4,2,1)

P2 (6,6,2)

P3 (8,8,0)

Assume that the resolutions of all the axes are 1000 counts/inch, and set the required speed to 1.2 inches/sec (1200 counts/sec) and the acceleration and deceleration to 100 in/sec^2 (100000 counts/sec^2). Note that the LM mode requires defining the segments in incremental form.

Point	Coordinate	Difference
P0	0,0,0	
P1	4000,2000,1000	4000,2000,1000
P2	6000,6000,2000	2000,4000,1000
P3	8000,8000,0	2000,2000,-2000

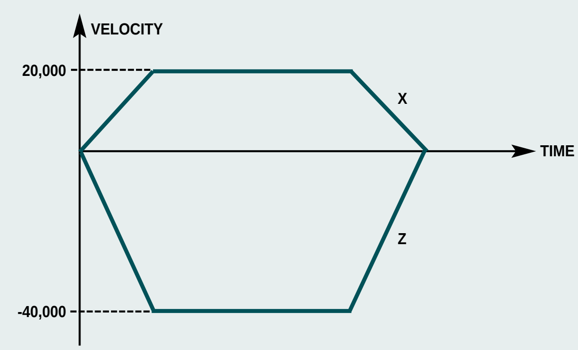
PROGRAM	INTERPRETATION
#C	*Label*
CAS	Set coordinate system
LM XYZ	*Define XYZ space*
VS 1200	*Vector speed*
VA 100000	*Vector acceleration*
VD 100000	*Vector deceleration*
LI 4000,2000,1000	*Segment P0-P1*
LI 2000,4000,1000	*Segment P1-P2*
LI 2000,2000,-2000	*Segment P2-P3*
LE	*End of move*
BGS	*Start the motion*
EN	*End program*

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.

Example—Electronic Gearing

PROGRAM	INTERPRETATION
GA Y,,Y	*Specify Y axis as master of X and Z*
GR 2,,-4	*Specify gear ratios for X and Z*
PRY=50000	*Specify end position of master*
ACY=1000000	*Specify acceleration of master*
DCY=1000000	*Specify deceleration of master*
SPY=10000	*Specify slew speed of master*
BG Y	*Begin motion*



General Description

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—Contouring

Generate an XY motion according to a file of points specifying the position increments every 16 msec. To simplify the example, it is assumed that only 4 segments are required. An array is used for each axis.

Segment	DX	DY
1	40	40
2	100	20
3	80	-40
4	40	-20

PROGRAM	INTERPRETATION
#D	
DM DX[4],DY[4]	*Define arrays*
DX [0]=40;DY[0]=40	
DX [1]=100;DY[1]=20	
DX [2]=80;DY[2]=-40	
DX [3]=40;DY[3]=-20	
CM XY	*Specify contour*
DT4; P=0	*Time interval*
#LOOP	*Loop*
CD DX[P],DY[P]	*Specify contour data*
P=P+1	*Increment counter*
WC	*Wait to complete*
JP#LOOP, P<4	*Loop until done*
CD 0,0	*Contour data for end*
DT0	*Set time interval to 0 to end mode*
EN	*End program*

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 gear ratios to be prescribed. Any follower axis may be engaged or disengaged independently at specific points along a Cam cycle. This allows the selecting of the engagement and disengagement points as those where the speed change of the follower is minimal. The electronic Cam is an ideal mode for periodic operation, especially those requiring a varying gear ratio along the motion cycle. Applications include flying shears, rotating knives, and packaging systems. Galil's Cam-generating software can assist the user to define the Cam table.

I/O

Error Handling

Dedicated I/O is provided for the following safety controls: forward and reverse limit inputs for each axis, home inputs for each axis, amplifier enable outputs for each axis, configurable abort inputs for each axis, master abort input, and error output. 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-18x0 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 inputs and 16 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.

As part of the user I/O, Galil controllers provide a high-speed position capture or position compare feature for each axis. The high-speed position capture latches the exact position within .1 microsecond 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. The accuracy allows for triggering external events to exact positions within .1 microseconds.