BLX Servo Drive Positioner
User Guide

Software Version: Issue 3.8 onwards

For engineering
assistance in Europe:
Parker Hannifin plc
Digiplan Division
21 Balena Close
Poole, Dorset
England, BH17 7DX
Telephone: 01202-699000
Fax: 01202-695750

For engineering
assistance in the U.S.:
Parker Hannifin Corporation
Digiplan Division
5500 Business Park Drive, Suite D
Rohnert Park, CA 94928
USA
Telephone: (800) 358-9070
Fax: (707) 584-8015

Part No: 1600.137.05   8th March 1995

IMPORTANT INFORMATION FOR USERS

Installation and Operation of Digiplan Equipment

It is important that Digiplan motion control equipment is installed and operated in such a way that all applicable safety requirements are met. It is your responsibility as a user to ensure that you identify the relevant safety standards and comply with them; failure to do so may result in damage to equipment and personal injury. In particular, you should study the contents of this user guide carefully before installing or operating the equipment.

Under no circumstances will the suppliers of the equipment be liable for any incidental, consequential or special damages of any kind whatsoever, including but not limited to lost profits arising from or in any way connected with the use of the equipment or this user guide.

⚠ SAFETY WARNING

High-performance motion control equipment is capable of producing rapid movement and very high forces. Unexpected motion may occur especially during the development of controller programs. **KEEP WELL CLEAR** of any machinery driven by stepper or servo motors. Never touch it while it is in operation.

This product is sold as a motion control component to be installed in a complete system using good engineering practice. Care must be taken to ensure that the product is installed and used in a safe manner according to local safety laws and regulations. In particular, the product must be enclosed such that no part is accessible while power may be applied.

The information in this user guide, including any apparatus, methods, techniques, and concepts described herein, are the proprietary property of Parker Digiplan or its licensors, and may not be copied, disclosed, or used for any purpose not expressly authorised by the owner thereof.

Since Digiplan constantly strives to improve all of its products, we reserve the right to modify equipment and user guides without prior notice. No part of this user guide may be reproduced in any form without the prior consent of Digiplan.

IBM PC AT XT are registered trademarks of International Business Machines Corporation
WINDOWS is a registered trademark of Microsoft Corporation

User Guide Change Summary

The following is a summary of the primary changes to this user guide since the last version was released. This user guide, version 1600.137.05, supersedes version 1600.137.04.

When a user guide is updated, the new or changed text is differentiated with a change bar in the outside margin (this paragraph is an example). If an entire chapter is changed, the change bar is located on the outside margin of the chapter title.

Changes introduced at revision 05 are:

Minor additions to Chapter 2.

Table 2-2 Switch Type Selection added.

Chapter 3, software scaling explanation changed.

Chapter 4, using Windows™ terminal emulator added.

Chapter 5, note added to SSB command and following commands HELP 13 added.

CONTENTS

List of Figures

List of Tables

How To Use This Manual

This manual is designed to help you install, develop and maintain your system. Each chapter begins with a list of specific objectives that should be met after you have read the chapter. This section is intended to help you find and use the information in this manual.

Assumptions

This user guide assumes that you have the skills or fundamental understanding of the following:

- Basic electronics concepts (voltage, switches, current, resistors, etc.)
- Basic motion control concepts (torque, velocity, distance, etc.)

With this basic level of understanding, you will be able to effectively use this manual to install, develop and maintain your system.

Contents of This Manual

This user guide contains the following information:

Chapter 1: Introduction

This chapter provides a description of the positioner and a brief account of its specific features.

Chapter 2: Interfacing Signals

This chapter details the input and output signal connections to the positioner. It also describes the signal characteristics and shows examples of interfacing circuit arrangements.

Chapter 3: Basic Motion Control Concepts

For those unfamiliar with motion control systems, this chapter explains the basic concepts It will help you to become familiar with the system and provide a basis for understanding the use of the command set.

Chapter 4: Communicating with the Positioner

This chapter will enable you to set up communications with the positioner.

Chapter 5: Programming

Chapter 5 lists the motion control commands of the positioner. It describes their use and explains the variable parameters associated with them. You should study this chapter before starting to program the system.

Developing Your Application

Before you attempt to develop and implement your application, you should consider the following:

- Recognize and clarify the requirements of your application. Clearly define what you expect the system to do.
- Follow the guidelines and instructions outlined in this user guide. **Do not skip any steps or procedures.** Proper implementation can be ensured only if all procedures are completed in the proper sequence.

Chapter 1. INTRODUCTION

Chapter Objectives

The information in this chapter will enable you to understand the basic functions and features of the positioner.

Product Description

The positioner performs position control and indexing functions using an industry standard RS232C interface. It is easily controlled from computers, terminals and most programmable controllers.

The programming language is based on Digiplan's X-code and the positioner is capable of storing and executing complex motion programs from its non-volatile memory (battery backed-up RAM).

Features

- High speed operation
- RS232C command interface
- 1 to 8 devices can be daisy-chained on a single RS232C port using zero delay daisy chaining (ZDDC)
- 6K program memory for storing up to 63 sequences and parameters
- Automatic load and execution of motion programs (sequences) at power up
- Sequence execution can be initiated by external switches, computer, or programmable controller
- Sequence upload, download, and memory verification from the computer, or programmable controller
- Encoder following - the axis follows the encoder of another axis
- Encoder superposition - the motion is the sum of the encoder following input and an internally generated index.
- Optically isolated inputs for Home position, End-of-travel limits, Emergency Stop and jog functions
- 7 optically isolated user-definable inputs
- Programmable resolution
- Analogue outputs for signal monitoring

- 3 optically-isolated programmable outputs:-
  One configurable as user-programmable, watchdog output or composite fault output
  One configurable as user-programmable or as an in position output
  One is user-programmable only
- A 24v supply for opto-I/O is available
- Single LED combined fault indicator
- Registration mode
- Servo self-tuning

Front Panel Indicator

A red LED fault indicator is the only front panel indicator. It signals a number of fault conditions according to the number of times it flashes. Table 1-1 lists the fault conditions against the corresponding number of flashes.

LED indicator	Fault
Off	No fault
Flashes once	drive de-energised by ST1 or OFF Command
Flashes three times	EPROM changed with different memory map
Flashes four times	excessive following error
Flashes five times	memory failure - failed checksum
Flashes seven times	prolonged maximum torque demand
Flashes eight times	emergency stop input seen

Table 1-1. Fault LED Indications

Positioner Specification

Parameter	Value
Command Input	RS232C
Type	3-wire (Tx, Rx, Gnd). Minimum voltage swing = ±3V
Parameters	9600 baud, 8 data bits, 1 stop bit, no parity
Connector	Removable screw terminals
Configuration	Up to 8 interfaces can be controlled from a single RS232C port using zero delay daisy chain. Device address set up by jumper links on the board.
Operating Ranges	
Position	±1 to 268,435,455 steps
Velocity	0.0001 to 100 revs/sec
Acceleration	0.06 to 999,999 revs/sec^2
Maximum Encoder Frequency	100 kHz (lines/sec before multiplication)
Resolution Ranges	
Feedback encoder	1 to 32,767 counts/rev
User-programmed	1 to 32,767 steps/rev
Co-ordinate System	Incremental or absolute
Operating Modes	Preset, preset with speed change, continuous
Positioner Loop Update	Every 2 milliseconds
Motion Program Storage	
Memory Type	Battery-backed RAM
Memory Capacity	6400 characters total
Number of Programs	63
Program Length	Variable up to memory limit
Program Selection	a) Via RS232C, b) Automatic execution at power up, c) Binary address on sequence select inputs
Digital Servo Loop	
Update Time	500 microseconds
Servo Tuning	RS232C. Values stored in battery-backed RAM. Servo self-tuning facility.
Tuning Parameters	PIVF or PID options with digital filter
Opto-isolated I/P's	Home, Limits, jog ±, emergency stop, 7 user-definable inputs (also used for program selection):- 12-24V on max at 4mA. 30V off max. Max. reverse voltage -5V.
Optically-isolated O/P's	3 user-definable: can also be assigned as watchdog, In Position and Fault outputs. NPN open-collector, common to isolated ground. 300 mA on max. 30V off max. 2.5V at 300 mA max voltage in the on state.
Analogue Monitor O/P's	Velocity and position error. ±2.5V relative to interface 0V
Power Requirements	+5V, derived from the drive module

Table 1-2. Positioner Specification

Chapter 2. INTERFACING SIGNALS

Chapter Objectives

This chapter defines the electrical and functional requirements for all of the signals connected to the positioner.

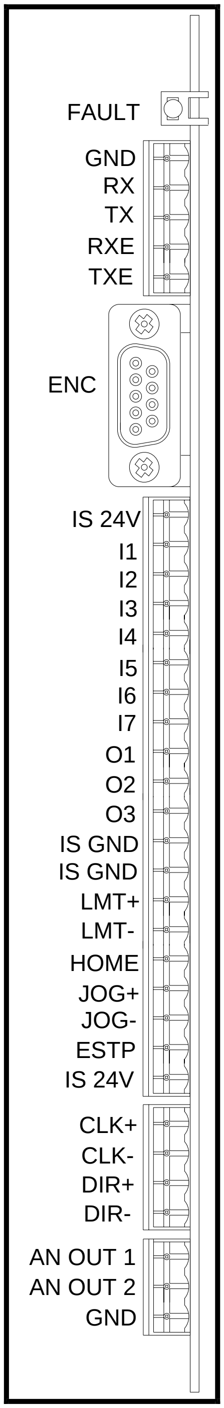


Figure 2-1. Signal Connections

Encoder Connector

Pin	Signal Name	Function	Signal Type
1	A+	ENC A CHANNEL	K
2	A-		
3	B+	ENC B CHANNEL	K
4	B-		
5	Z+	ENC Z CHANNEL	K
6	Z-		
7	5V ENC.	ENC SUPPLY	E
8	GND	ENC GND	E
9	GND	ENC GND	E

Main Connector

	GND	Signal ground	D
	RX	Receive input	C
	TX	Transmit output	C
	RXE	ZDDC receive line	C
	TXE	ZDDC transmit line	C

	IS 24VDC	Power supply (100mA max.)	
	I1	User input	A
	I2	User input	A
	I3	User input	A
	I4	User input	A
	I5	User input	A
	I6	User input	A
	I7	User input	A
	O1	Output 1	B
	O2	Output 2	B
	O3	Output 3	B
	ISOL GND	Power supply	F
	ISOL GND	Power supply	F
	LMT +	+Limit switch input	A
	LMT -	-Limit switch input	A
	HOME	Home switch input	A
	JOG +	+Jog switch input	A
	JOG -	-Jog switch input	A
	E stop	Emergency stop input	A
	IS 24VDC	Power supply (100mA max.)	

	CLK IN +	Clock input	I
	CLK IN -	Clock input	I
	DIR IN +	Direction input	I
	DIR IN -	Direction input	I

	AN OUT 1	Analogue output	G
	AN OUT 2	Analogue output	G
	AN GND	Analogue ground	H

Table 2-1. Positioner Signal Types

Key to Table 2-1

A Optically isolated inputs referenced to ISOL GND. 12-24V applied at input represents logic 1
B Open collector optically isolated outputs referenced to ISOL GND.The parameters of these outputs are:-
30V absolute maximum
300 mA absolute maximum
Maximum output voltage 2.5V at 300 mA
C RS232C data signals
D Interface 0v (RS232C)
E Encoder supply voltage (5v)
F Interface 0v
G Analogue monitoring output ±2.5v relative to GND(VE)
H Analogue monitoring ground
I Differential optically isolated inputs, TTL levels
K Differential optically isolated encoder inputs, TTL levels

Signal Descriptions

(Refer to Figure 2-1)

Encoder Input

Balanced differential line receivers are used on the A, B and Z inputs.

Clock

When clock and direction control is used, the clock signal controls the acceleration, deceleration and speed of the motor. It is a balanced input.

Direction

The two direction inputs + and - are balanced inputs controlling the direction of rotation when clock and direction control is used such that:

If the + input is positive with respect to the - input, the rotation is CW when viewing the shaft end of the motor.

If the positive input is negative with respect to the - input the rotation is CCW.

Inputs 1 to 7

These optically isolated connections can be read by the positioner for use by the controller. They can also be used for triggering sequences (see commands IS, TRE). Input I5 can be assigned as a controlled stop line using the command SSE. This input triggers an internal S command which interrupts all other indexing activity. Alternative functions can be assigned to inputs 4 to 7 as follows:-

Input 7 is a sequence select line if OSD is set to 1

Input 4 is a sequence select line if SSF is set to 1

Input 5 is stop input if SSE is set to 1

Input 6 clears a pause if SSB is set to 1

E stop

The ESTOP connection is a fail safe input which needs current to be sourced into it to keep the drive energised. It provides a hardware-only path to drive energisation. If current is momentarily stopped the resident positioner software will suppress re-energisation of the drive when the supply returns.

The ESTOP input must be physically connected to the +24V on I/O3, preferably through a normally closed emergency stop switch. This circuit arrangement is shown in Figure 2-9.

+ and - Limits

When activated, the + and - limit inputs trigger a controlled stop and prevent further movement in the same direction as the active limit (the convention is positive direction = CW rotation). These inputs need current sourced into them in order to allow motion in the specified direction. They can be disabled by the LD3 command. The system will not operate without limit switch connections or use of the LD3 command.

CAUTION

Damage may occur to the system mechanics due to excessive travel if the LD3 command is used. You should also ensure that clockwise rotation of the motor shaft (when viewing the motor from the shaft end) produces movement towards the + limit switch.

+Jog, -Jog

These inputs provide a switch activated method of moving in the positive or negative direction respectively at a constant velocity previously defined using the JV command.

The inputs can also be used as user definable inputs for input triggers etc.

Home

The home position is a reference position defined by the user and usually activated by a switch. The positioner can be programmed to search for (or datum to) this position.
See the GA, GHF, GH and PZ commands.

Output O1

If SSD is set to 0, O1 functions as a user defined output.
If SSD is set to 1, then this output becomes a composite fault indicator with a watchdog function. If the software for some reason

enters an illegal state, then the output will become high impedance and the drive will de-energise. The output sinks current during normal operation.

The Watchdog function is always operational. Even if SSD0 is selected, the output will still become high impedance and the drive will de-energise in the event of a watchdog timeout.

(See commands IO, O, SSD).

Output O2

If SSC is set to 0, O2 functions as a user-definable output.
If SSC is set to 1, O2 is configured to indicate "in-position".
The user defines the criteria for being in position using CIT and CEW commands.

(See commands IO, O, SSC, CIP and CEW).

Output O3

This is a user-definable output with no alternative fixed function.

AN-OUT 1

This is an analogue output for position monitoring. The output voltage is from +2.5V (representing a position error of 128 steps lagging the commanded position) to -2.5v (representing a position error of 128 steps in advance of the commanded position). The error corresponding to intermediate voltages is proportional to the full scale.

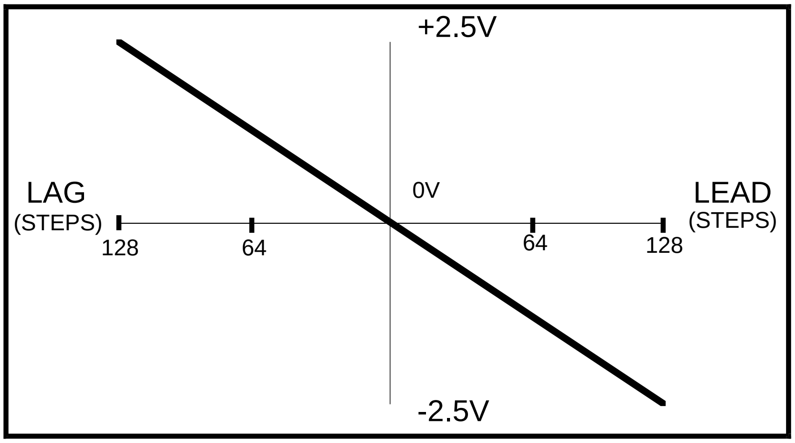


Figure 2-2. AN-OUT 1 Graph

AN-OUT2

This is an analogue output for velocity monitoring. The output is +2.5V (representing a CW speed of 256,000 steps/sec) to -2.5V (representing an CCW speed of 256,000 steps/sec). The speed corresponding to intermediate voltages is proportional to the full scale.

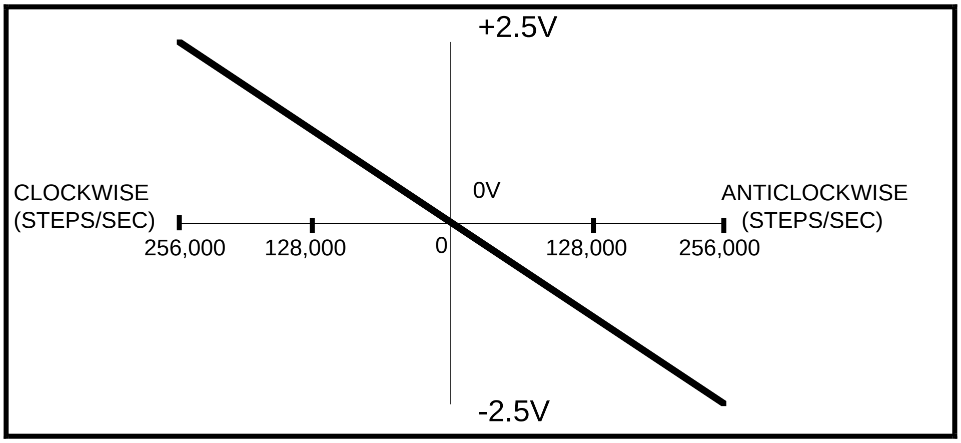


Figure 2-3. AN-OUT 2 Graph

Controller Connections

The RX input to accepts ASCII data from the controller at RS232C signal levels.

The TX output sends ASCII data from the positioner to the controller at RS232C signal levels.

The TXE and RXE connections are for connection to other positioners in the system using the Zero Delay Daisy Chain.

Encoder Connector

With jumper links 1 to 6 on the adaptor board (see Figure 4-3) in position B the positioner will respond to the encoder which is plugged into the Motor Feedback socket on the drive. With these jumper links in position A, the positioner will respond to a separate encoder which can be connected to this socket.

Interfacing Circuit Arrangements

This section describes some suitable circuits for interfacing to the positioner.

Figure 2-4 shows the general circuit arrangement of the inputs and outputs of the positioner. Both types of circuit are optically isolated to give immunity from noise and transients.

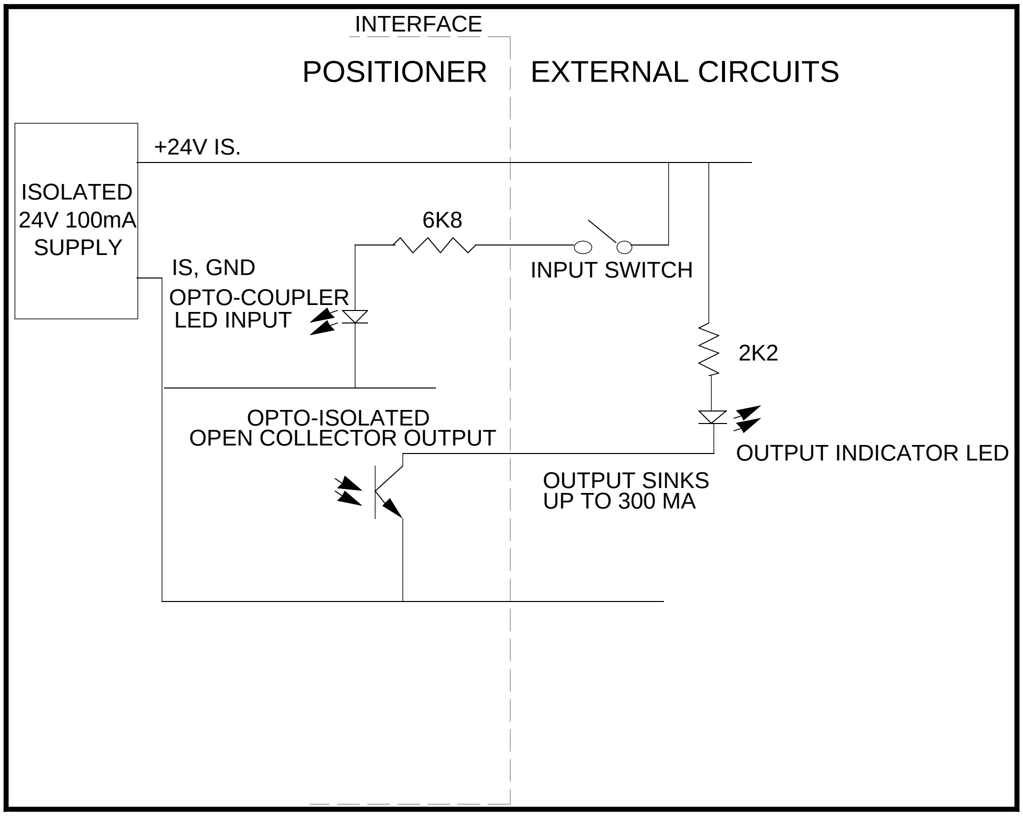


Figure 2-4. Inputs and Outputs - General Arrangement

Coupling Optically Isolated Outputs and Inputs

An output can be coupled to an input on another positioner as shown in Figure 2-5.This connection may be required when running sequences on two axes with a need to couple the two sequences. Using this arrangement, setting the output to 1 causes the open collector output transistor to turn on, diverting current from the input. With no input current, the input is at 0 level, so a data inversion occurs.

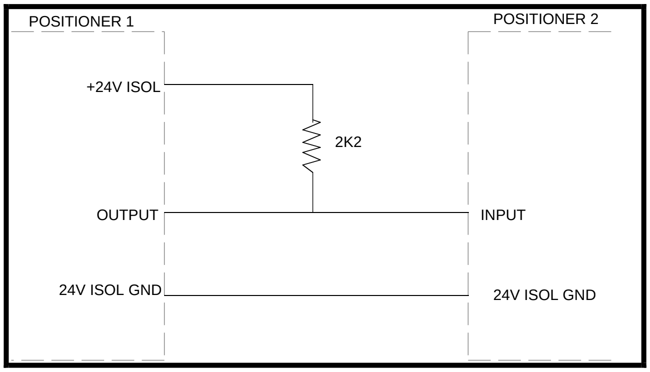


Figure 2-5. Input to Output Coupling

Clock and Direction Inputs

These balanced inputs may be driven from a single-ended output via a differential line driver such as the National Semiconductor DS8830, which accepts a TTL level input and provides a balanced output. Each of the two circuits in the DS8830 should have their four inputs connected together and to the input signal if this device is used (See Figure 2-6). Connecting the clock/direction source equipment ground to the positioner ground should be avoided since this would violate the isolation conditions.

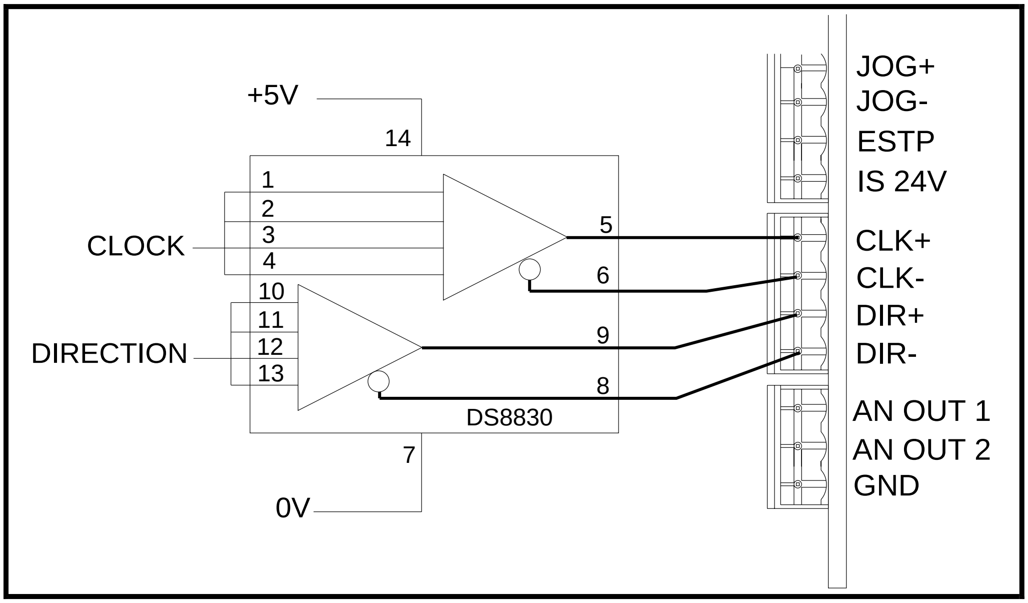


Figure 2-6. Clock and Direction Inputs

Limit and Home Inputs

Both limit inputs are optically-isolated, requiring to be connected to +24V when they are not operational. If disconnected or taken to 0V, they are operational and will prevent movement of the axis. Two methods of switching these inputs are shown in Figures 2-7 and 2-8. NPN proximity switches and mechanical switches are the examples shown. The limits can be disabled with the LD3 command.

The GO HOME function of the positioner is initiated by issuing the GO HOME (GH) command. When the command is issued, the direction in which it should search for home and the velocity must be included.

(See the GA, SS, SR, GHP and GHF commands).

A normally open, load activated switch to a current source is the most common way of detecting the home position (See Figure 2-7). When the positioner receives the command "GO HOME", it initiates a move in the direction and at the velocity specified, looking for the home limit input to change state.

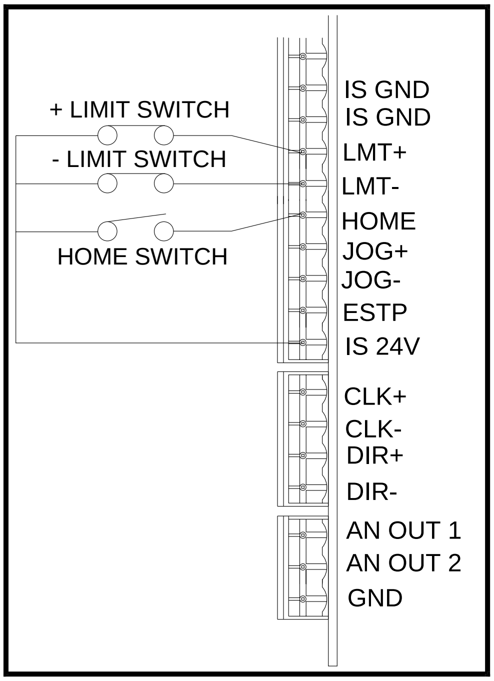


Figure 2-7. Limit and Home Switch Connections

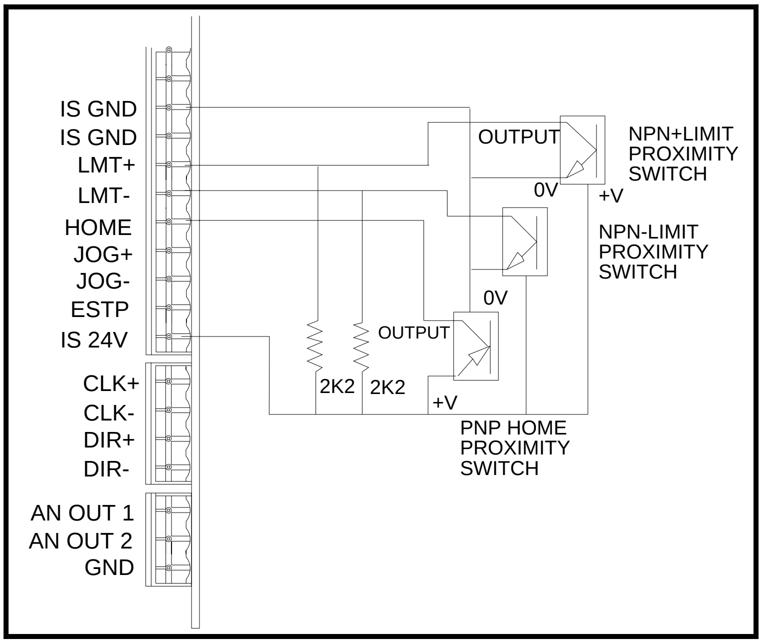


Figure 2-8. Limit and Home Proximity

Note the NPN limit switches shown in Figure 2-8 are normally open (the limit input is held inactive by the additional 2K2 resistor connected to the isolated +24V supply). Once a limit is reached the NPN transistor will turn ON, activating the limit input. Table 2-2 provides a summary of switch types for use as Home or EOT switching.

Usage	Logic	Switch type
HOME	PNP	N.O.
	NPN	N.C. use external resistor
END OF TRAVEL	PNP	N.C
	NPN	N.O. use external resistor

Table 2-2. Switch Type Selection

ESTOP Input

The ESTOP input must have current sourced to it to keep the drive energised. It is usually connected to the +24V supply via a normally closed push button as shown in Figure 2-9. If the push-button is pressed, the drive immediately de-energises.

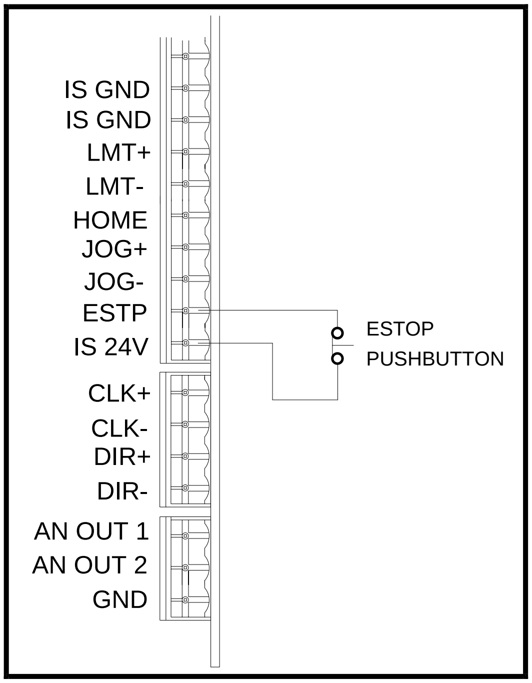


Figure 2-9. ESTOP Switch Connection

Trigger Inputs

The positioner has thirteen optically isolated inputs, any of which may be used as triggers with, for example, the TRE command.

The semi-dedicated inputs such as + and - jog must have their dedicated functions disabled when used as trigger inputs. They may be used either high true or low true but you must supply the inputs with 7 to 24 volts at 4mA per input to turn on the opto-isolators.The isolated 24V supply available at the edge connector may be used as shown in Figure 2-10.

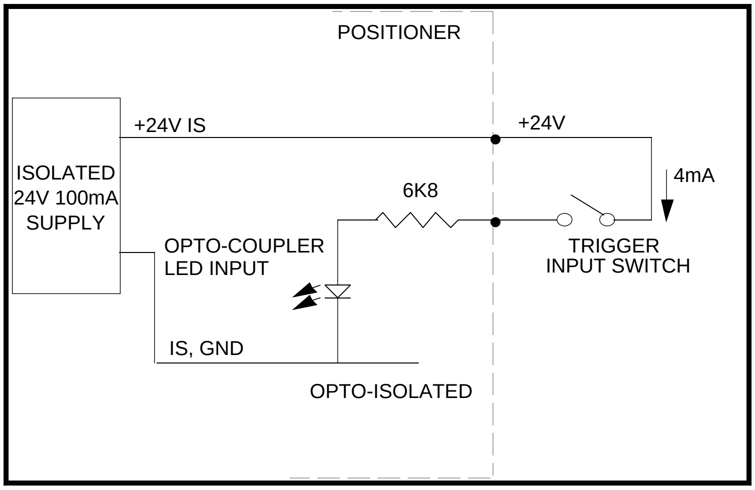


Figure 2-10. Trigger Input Connection

Jog Inputs

The jog switches provide manual control of the motor position when the jog function is enabled using the OSE command. The acceleration and speed of the motor when the jog switches are operated are set by the commands JA and JV respectively. Normally-open push buttons connected to the motherboard as shown below are frequently used for this function:

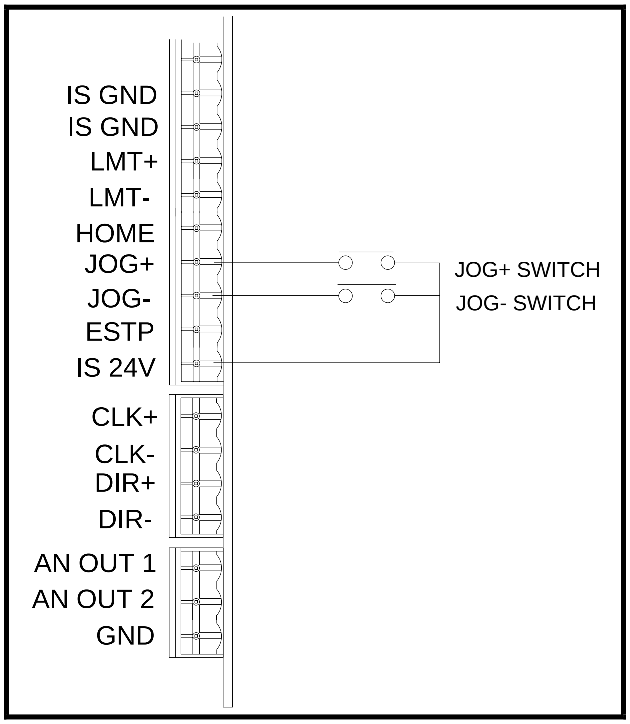


Figure 2-11. Jog Switch Connections

Composite Fault Indicator

The following example of a composite fault indicator requires Output 1 to be configured as a composite fault output. O1 sinks current while no fault exists and switches off if a fault occurs.

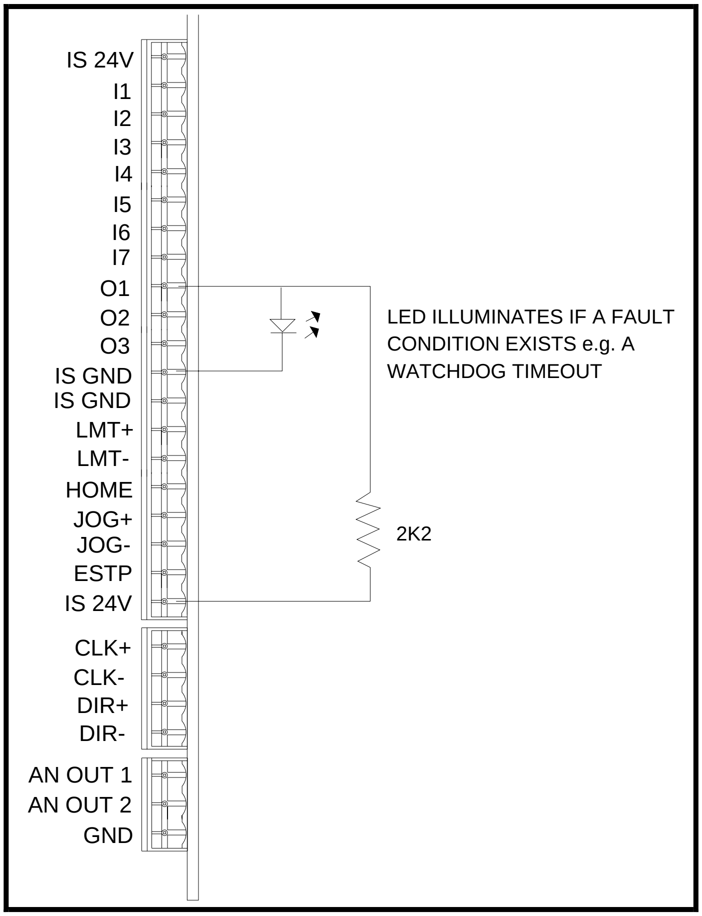


Figure 2-12. Composite Fault Indicator

In Position Indicator

The following example of an "in position" indicator requires O2 to be configured as an "in position" output by the SSC command. O2 sinks current while inside of the deadband and switches off when outside of it.

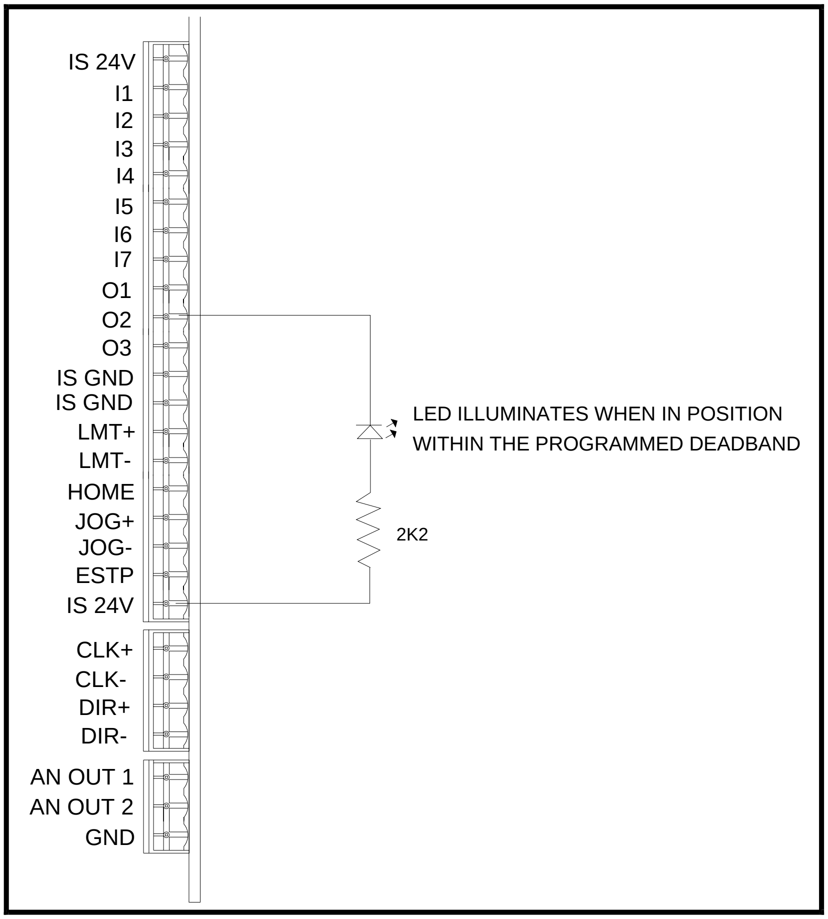


Figure 2-13. In Position Indicator

Chapter 3. BASIC MOTION CONTROL CONCEPTS

Chapter Objectives

This chapter describes some of the basic concepts of motion control systems.

Motion Profiles

In any motion control application the most important requirement is precise shaft rotation, whether it be with respect to position, time or velocity. The type of motion profile needed will depend upon the motion control requirement. The following sections describe the basic types of motion profiles.

Preset Moves

A preset move referred to in this manual is a move of a specified distance (in user steps). Preset moves allow the user to position in relation to the motor's previous stopped position (incremental moves) or in relation to a defined zero reference position (absolute moves). Preset moves are selected by putting the positioner into incremental mode using the MPI command or the MN command and absolute moves are made using the MPA command.

Incremental Preset Moves

If the positioner is in the incremental mode (MPI command), a preset move will move the shaft of the motor the specified distance (in user steps) from its starting position. For example, to move the motor shaft 1.5 revolutions, a preset move with a distance of +6000 steps would be specified, assuming a 4000 step per rev encoder resolution setup. Every time this move is executed, the motor shaft will move 1.5 revolutions positive from its current position. The direction of the move can be specified at the same time as the distance by using the optional sign (D+6000 or D-6000), or it can be defined separately with the H command
(H+ or H-).

Absolute Preset Moves

A preset move in absolute mode (MPA command) will move the shaft of the motor the specified distance (in user steps) from the absolute zero position. The absolute position can be set to zero with the PZ or SP commands, for instance at the end of a GO HOME move (GH command). The absolute zero position is initially the power-up position, and will remain that way until changed with a PZ command. Any preset move performed while in the absolute mode will position the motor shaft the defined distance (in user steps) from the absolute zero position. For example, with the positioner at the absolute zero position a move with a distance of +4000 will cause the motor shaft to turn 1 revolution in the positive direction. If a move with the same defined distance is executed immediately after this move, the motor shaft will not turn, since it is already +4000 steps from the absolute zero position.

The direction of an absolute preset move will depend upon the shaft position at the beginning of the move and the position that it is being commanded to move to. For example, if the motor shaft is at absolute position +12,800, and position commanded is +5000, the motor shaft will move in the negative direction a distance of 7800 steps to absolute position +5000.

The positioner saves the mode that it was in at power down and powers up again in the same mode. Issuing the MPA command will set the mode to absolute. Issuing the MPI command or MN command will switch the mode from absolute to incremental. The positioner retains the absolute position referenced, even while in incremental mode. However, the counter does have an upper limit just slightly more than 268,400,000 counts.

Continuous Moves

Continuous moves (MC command) cause the motor to accelerate and attain the specified velocity and then continue to move. To change velocity while the motor is moving use the V command. Issuing a stop (the S command) will cause the motor to decelerate to a stop at the last defined acceleration rate. The distance parameter is not used, although it is saved in case the mode is changed back to preset.

This mode is useful for applications which require constant spinning of the load, when the motor must stop after a period of time has elapsed rather than after a fixed distance, or when the motor must be synchronised to external events such as trigger input signals.

Registration Moves

A move may be programmed to end a specified distance after a registration pulse appears at Input 6. The mode is selected using the TRR command with the required registration distance in the MQ or MC modes. Its use is easiest to understand in the context of an example registration move:

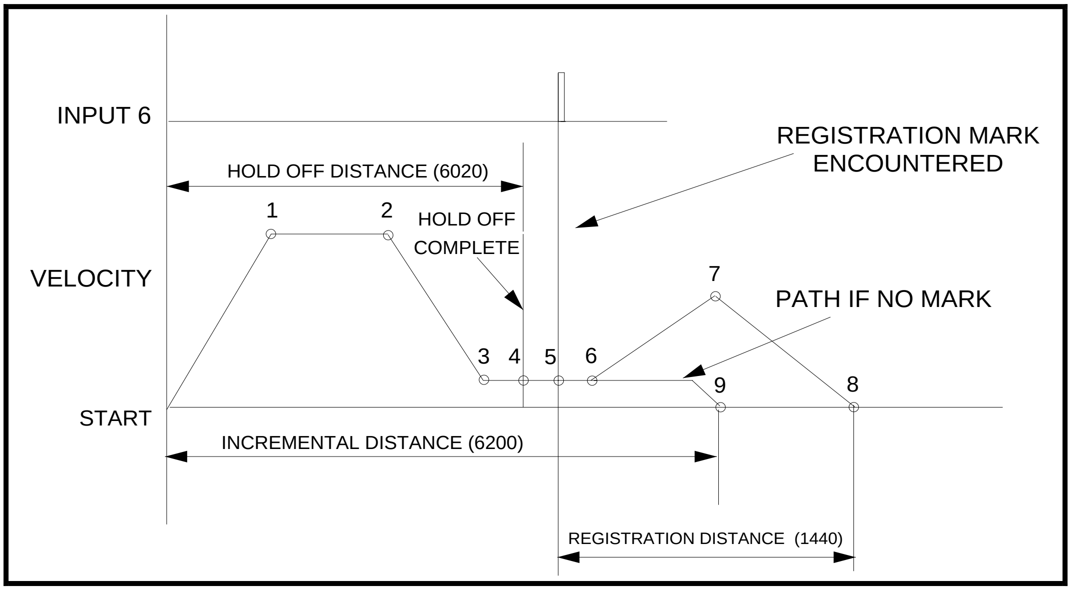


Figure 3-1. Example Registration Move

Example Program

MQ	Select MQ mode
D6200	Set move distance to 6200 steps
A100	Set all accelerations to 100 steps/rev^2
V10	Set velocity to 10 revs/sec
G	Go
TRD4000	At distance = 4000 steps
V1	Change velocity to 1rev/sec
TRD6020	At distance 6020 look for registration pulse
TRR1440	Travel 1440 steps from registration pulse
V50	At a velocity of 50 revs/sec
TRIP	Set Output 2 when in position

Program Operation

When executing this program, the axis will first accelerate at 100 steps/rev^2 to 10 revs/sec (1) and when it reaches a distance of 4000 steps (2), its speed will be reduced to 1 rev/sec (3) whilst it is waiting for the registration pulse. Shortly after the pulse is received (4), the axis will accelerate towards 50 revs/sec, but will not reach the speed before starting to decelerate (5) to stop at 1440 steps from where the pulse occurred (6).

If the pulse was not received after 6020 steps (the hold off distance) and before 6200 steps (the incremental distance), the axis will stop at 6200 steps. This sets a window of 180 steps during which the pulse is expected.

Program Criteria

The distance command (D6200)	This command sets the distance the axis will travel if no registration pulse is received and the maximum distance at which the pulse will be recognised.
Trigger distance command (TRD4000)	The distance at which the axis will start to decelerate to the slower speed of 1 rev/sec (V1) in readiness for the registration pulse is set by this command.
Trigger distance command (TRD6020)	This command sets the minimum distance at which the system will recognise the registration pulse.
The TRR command (TRR1440)	This command sets up the registration mode and the registration distance (the distance the axis will travel after the pulse is received)
Registration move velocity (V50)	To save time in travelling the registration distance, the axis then accelerates towards 50 revs/sec (V50), but it does not reach that speed before starting to decelerate to stop at 1440 steps from where the pulse occurred.

Motion Profiles

Velocity, acceleration and distance must be defined before any preset move can be executed. The value of these parameters determines the type of motion profile as either triangular or trapezoidal.

Triangular Profile

A triangular profile will result when the velocity and acceleration are set such that the designed velocity is not attained before half of the specified distance has been travelled. This results from either a very low acceleration or a very high velocity or both over a relatively short distance. For example, if the acceleration is set to 1 rev/sec/sec, velocity is set to 5 revs/sec and distance is set to 16000 steps (2 revs), a triangular motion profile will result. This is because by the time the motor shaft has reached a velocity of 2 revs/sec, it will also have travelled half of the defined distance due to the acceleration setting of 1 rev/sec/sec. The motion profile for this move would look like this.

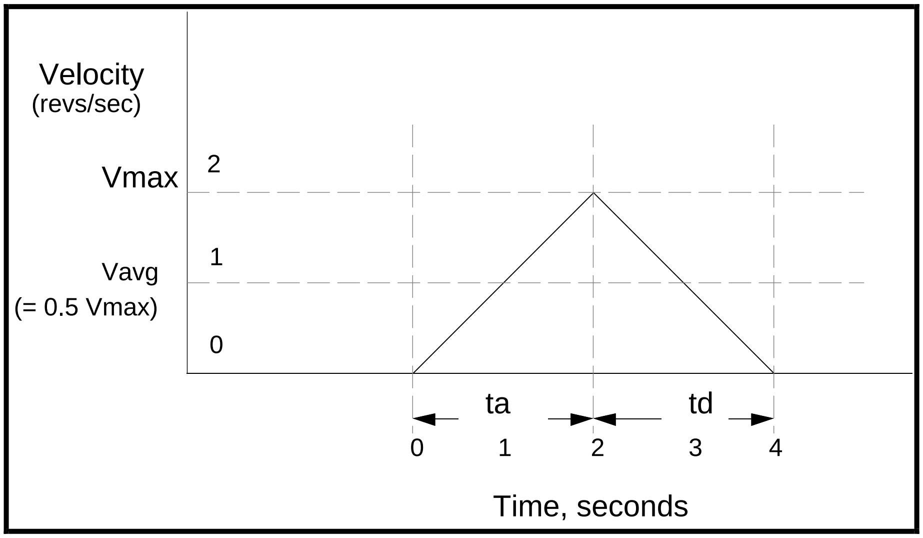
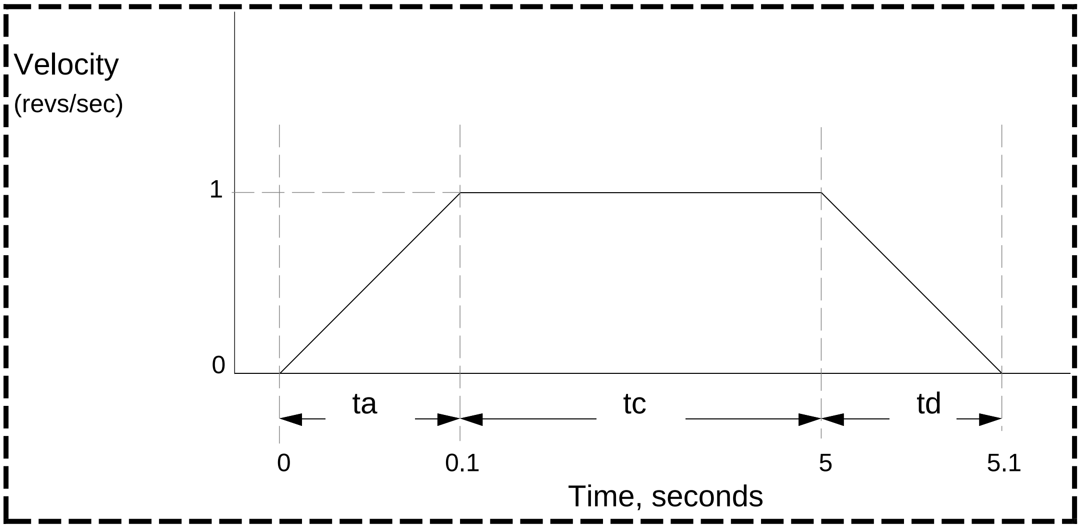


Figure 3-2. Triangular Profile

Trapezoidal Profile

A trapezoidal move profile results when the defined velocity is attained before the motor shaft has moved half of the specified distance. This is due to a defined velocity that is low, a defined acceleration that is high, a move distance that is long, or a combination of all three. For example, if the acceleration is set to 10 revs/sec/sec, velocity is set to 1 rev/sec, and distance is specified as 20000 steps (5 revs), the resulting motion profile would look like this:



ta = Accelerate
tc = Constant velocity
td = Decelerate

Figure 3-3. Trapezoidal Profile

User Profiles

The user may define a move profile using the MC command to establish the continuous mode. Velocity can be programmed on the fly by the V command. Sending commands to the positioner in rapid succession allows smooth profiling which will allow circles and arcs to be traced using a two-axis system, or allow smooth (low jerk) motion by virtue of S-curve acceleration rather than straight ramping as with triangular and trapezoidal moves. Once a sequence of commands is found to trace the correct profile, the profile will be very repeatable. This may require some trial and error to establish the correct sequence of V commands. The V commands can be combined with time delays in a sequence buffer to create a very complex move profile.

The positioner also has a mode MQ, which is like the preset move mode in that the move distance is pre-defined, but it is possible to change speed in the middle of the move as required based on a distance, input or time delay trigger.

Encoder Following

The commands SIM and CCS may be used to allow the motor of one axis to follow the encoder of another axis or an externally-generated clock and direction signal. When the control module is to be used in following mode, the input from the encoder to be followed should be connected to the terminals marked CLK+, CLK-, DIR+ and DIR- on the motherboard. These inputs can be configured using the CCS command as an encoder input (x1, x2 or x4) or as a clock and direction input. For encoder input, connect:

CHA+	to	CLK+
CHA-	to	CLK-
CHB+	to	DIR+
CHB-	to	DIR-

The SIM command may be used to select:

a) Normal control module operation (SIM0), used for reverting to normal operation by overiding previous SIM commands

b) Encoder following with control module motion commands inoperative (SIM1).

c) Encoder following with motion control commands operative (SIM 2), allowing the superimposing of indexer moves.

d) Software scaled encoder following (SIM3).

e) Software scaled encoder following with direction reversal (SIM4).

f) Preset following index mode (SIM 5).

For SIM1 and SIM2 operation the motor output rate follows the encoder input, using hardware scaling, at a ratio of 1 or less.

Hardware Scaling

This is the scaling of the second encoder input when using the SIM1 and SIM2 commands to achieve following at a ratio of 1 or less relative to the following input. Scaling is achieved using a hardware rate mutiplier, the division ratio of which is set by the RAT command, at a resolution of 16 bit or 24 bit operation determined by the OSJ command.

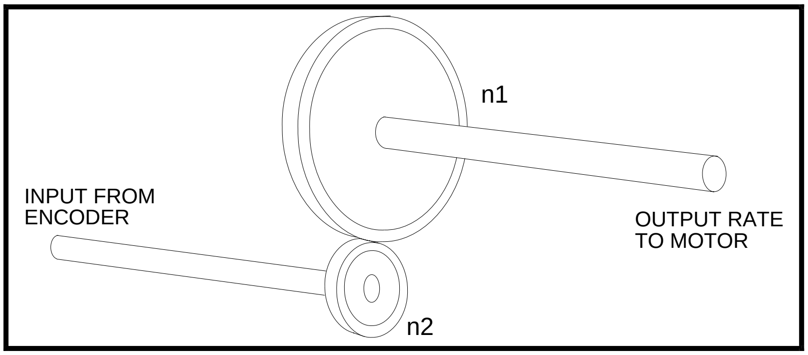


Figure 3-4. Simple Gearbox

The operation of the rate multiplier can be compared to that of a simple gearbox (see Figure 3-4), where the ratio of output revolutions to the input revolutions (always less than or equal to 1) is determined by the number of teeth on n2 divided by the number of teeth on n1. If the number of teeth on n1 is fixed, but the number of teeth on n2 is allowed to vary (up to a maximum of n1 teeth), the gearbox will provide a variable output rate which is always a fraction of the input rate. The resolution of the gearbox, that is how fine a gear ratio can be set, will be determined by the number of fixed teeth on n1. In the rate multiplier the RAT value sets the number of teeth on n2, and the resolution (n1 teeth) can have one of two fixed values determined by the OSJ command.

Using hardware scaling the output motor rate is determined by :

$$\text{Output Motor Rate} = \text{input rate} \times \frac{n}{65536} \quad \text{for OSJ} = 0$$

or

$$\text{Output Motor Rate} = \text{input rate} \times \frac{n}{16777216} \qquad \text{for OSJ} = 1$$

Where n = the RAT value

NOTE: RAT now controls the division ratio, and the direction will be determined by the sign of the RAT value.

As the resolution (set by OSJ) is a large number, very fine adjustment of the ratio can be set by choosing a suitable value for the RAT command.

A limitation of binary hardware scaling is the lack of certain exact following ratios that can be obtained. Since the division ratio can only be a binary fraction it is impossible to choose an exact ratio such as 1:3, although it could be closely approximated if you were to choose a RAT value of 5592405 with 24-bit resolution selected.

Summary of hardware scaling :

The scaling ratio is determined by the RAT value.
Resolution is fixed at 16-bit or 24-bit , determined by the OSJ command.
Non binary fraction exact division ratios cannot be obtained.

Software Scaling

This is the scaling of the encoder input when using the SIM3 or SIM4 commands to achieve following at a ratio greater or less than 1. Unlike hardware scaling, exact following ratios can be achieved by controlling both the numerator and denominator parts of the fraction used to set the scaling ratio, thus ratios such as 3:1 can be obtained.

The scaling ratio is set using the CMR command value divided by the CUR command value to give :

$$\text{Motor Output Rate} = \text{2nd input rate} \times \frac{\text{CMR}}{\text{CUR}}$$

Software scaling works by calculation, allowing the motor to run 255 times faster than the encoder or 255 times slower. But because the the processor is always busy calculating pulse rates, it is not able to superimpose indexer moves on the motion.

Both CMR and CUR can take any value up to 32,767 but you must be able to write the CMR/CUR ratio as a fraction using numbers less than 255. If you cannot achieve this, the indexer will not accept it as a valid ratio and the following error message will be output:

#70 “255 reduced ratio exceeded”

For example, a motor has 4000 steps/rev, and the following source is a 500-line encoder giving 2000 counts/rev. We want the motor to move one rev for every 5 revs of the encoder, in other words 10,000 encoder counts should produce 4000 motor steps. The speed ratio must be 4000/10,000.

So CMR = 4000, CUR = 10,000. We can reduce the CMR/CUR fraction to 4/10, so both numbers are within the 255 limit. This ratio would be OK.

Consider a second example where the motor is still 4000 steps/rev, but this time the encoder has a binary output with 4096 counts/rev. We want one rev of the encoder to produce 20 revs of the motor, so the ratio is 80,000/4096. Dividing each by 4 will give CMR = 20,000 and CUR = 1024, so both values are less than 32,767.

The ratio 20,000/1024 can be further reduced to 625/32, but no further using whole numbers. So this ratio will not work because 625 is outside the 255 limit.

In general it’s better to try and make the following encoder resolution the same as the motor resolution, or at least a simple ratio of it. This will give the widest choice of valid following ratios.

When using SIM3 or SIM 4 operation for short moves it is possible to predict the number of steps the motor will take using the formula :

Number of motor steps to move =

$$\mathrm{INT}\left(\mathrm{CMR}\ \text{x no. of pulses received} + \frac{\text{prev. remainder}}{\mathrm{CUR}}\right)$$

where INT means "take the integer part of", with the value rounded towards zero whether positive or negative. The controller can repeatedly apply this formula to establish an iterative calculation.
In situations where short moves have been programmed the ratio of CMR/CUR may only approximate the number of steps the motor is required to move, but since the remainder is carried forward no steps are lost. This allows the CMR/CUR value to better approximate the following ratio in subsequent moves.
In summary, short moves may only approximate the defined following ratio, but no positioning accuracy is lost in later moves.

Buffered Clock

SIM3 or SIM4 work in a buffered clock mode, which allows the

Mode

controller to buffer an unprofiled following-input pulse stream until the output velocity equals the following input velocity. The controller accelerates the output velocity to match the input velocity using an exponential acceleration profile, the time constant of which is set using the CAG command. The default time constant value of CAG is 1.00ms to maintain compatibility with earlier software issues, and to accept already profiled follower inputs.

The input pulses are buffered (or stored) at the input resolution, the actual number being stored at any particular instant being termed the following error. This can lead to input pulses being lost above a maximum speed, due to excessive following error.

The maximum number of input pulses that can be buffered is

+/-32767 which requires CAG to be less than $\frac{32767}{\text{input pulse rate}}$ to prevent following error overflow.

Comparison of following features

Hardware Scaling SIM1, SIM2	**Software Scaling SIM3, SIM4**
The motor always moves by the same or fewer pulses than received. The pulse stream is effectively divided.	The motor can move by more or less pulses than received. The input pulse stream can be multiplied or divided.
The ratio is programmed by the RAT command with 16 or 24 bit resolution as a binary fraction.	The RAT command has no effect.
CMR and CUR have no effect.	The ratio is programmed by CMR and CUR.
Superposition onto an additional internally generated indexer motion is available.	Superposition is not available.
The pulse stream is followed directly without additional profiling.	The pulse stream is profiled (filtered) by a programmable exponential characteristic (CAG command)

Preset Following Index Mode

The following mode SIM5 selects indexing at a speed determined by the external input. In this mode the controller sets the motor velocity to a speed in rps determined as a percentage of the following input speed in rps. The percentage following factor is set by the FOL command which can be varied between 0.0 and 5000.0%.

When using this mode the acceleration is fixed to whatever is defined by the A command, and the V command value has no effect since the FOL command percentage value will now control the velocity.

The velocity is dependent on the user resolution (CUR value) and the motor resolution (CMR value). CUR and CMR set the encoder / motor resolution ratio so that the input shaft speed will match the output shaft speed with FOL set to 100%.

Program Storage

The program memory is battery backed up RAM with memory retention of 1000 hours. The RAM has 6K characters available for sequence storage and a write protect facility is provided. Programs are stored in variable length buffers and the total length, including servo parameters, cannot exceed 6K.

Write Protection

A write protection facility is incorporated for the protection of stored sequences and parameters stored by the SV command. If LK1 on the positioner card (see Figure 4-3) is fitted, then the memory is not write protected. An attempted SV command when the jumper link is not made will result in an error message being displayed.

Motion Program Selection

The system may be set up so that no sequence is executed at power-up. In this case sequence execution would be initiated from the controller over the RS232C.

Alternatively, a single sequence to be automatically executed at power-up can be programmed. After that sequence is executed, control can pass to another sequence or to the RS232C interface.

Sequence selection may also be initiated via inputs. Up to 63 user defined sequences can be selected for execution in this way.

Parameter Ranges

Table 1-3 gives the internal arithmetic positioner limits based on 4000 encoder steps/rev. These parameters apply only to the positioner and the motor/drive combination will not necessarily be able to respond to the full ranges stated:

	Position	**Velocity**	**Acceleration**
Smallest	1 step	0.488 steps/sec 0.000122 RPS 0.00732 RPM	244 steps/sec/sec 0.061 RPS^2 3.66 RPM/sec
Largest	±268,435,455 steps ±500 revs at 4000 steps/rev	4 x 10^6 steps/sec 999 RPS	4 x 10^9 steps/sec/sec 999,999 RPS^2

Table 3-2. Parameter Ranges

NOTE: Longer distances than 268,435,455 steps can be achieved using a sequence of incremental moves or continuous motion mode.

Additional flexibility is possible using false motor resolutions (e.g. 2,000 steps/rev instead of 200).

Chapter 4. COMMUNICATING WITH THE POSITIONER

Chapter Objectives

The information contained in this chapter will enable you to set up communications with the positioner. If more than one positioner is present in the system, details are provided on the connection of multiple positioners using the Zero Delay Daisy Chain.

Command Interface

The interface is a three wire implementation (Tx, Rx, Ground) of RS232C. The Tx and Rx lines requires a minimum voltage swing of ±3 volts.

Hardware handshaking is not supported in any form. The computer or terminal sending characters to the positioner should have its handshaking disabled by either hardware or software.

Communication Parameters

The positioner communications protocol is fixed and as follows:-

9600 baud, 8 data bits, no parity, 1 start bit, 1 stop bit

Device address: 1-8 (set with jumper links on the positioner)

Echo function: all characters received are immediately re-transmitted by the last device to be addressed. This function can be enabled using the SSA0 command or disabled using the SSA1 command.

Once entered, a parameter will be remembered. It is not necessary to keep re-stating the same parameter value.

When you power up the positioner all parameters are assigned default values which the 1DR command will show you. You can change any of these and then make them the new default value by typing the SV (Save command).
(see also commands RFS, RIFS).

XON, XOFF software handshaking is supported.

Installing the RS232C

RS232C connections from the controller to the positioner are as shown in Figure 4-1. Note that the Tx and Rx lines are cross-connected so that transmit output is connected to receive input.

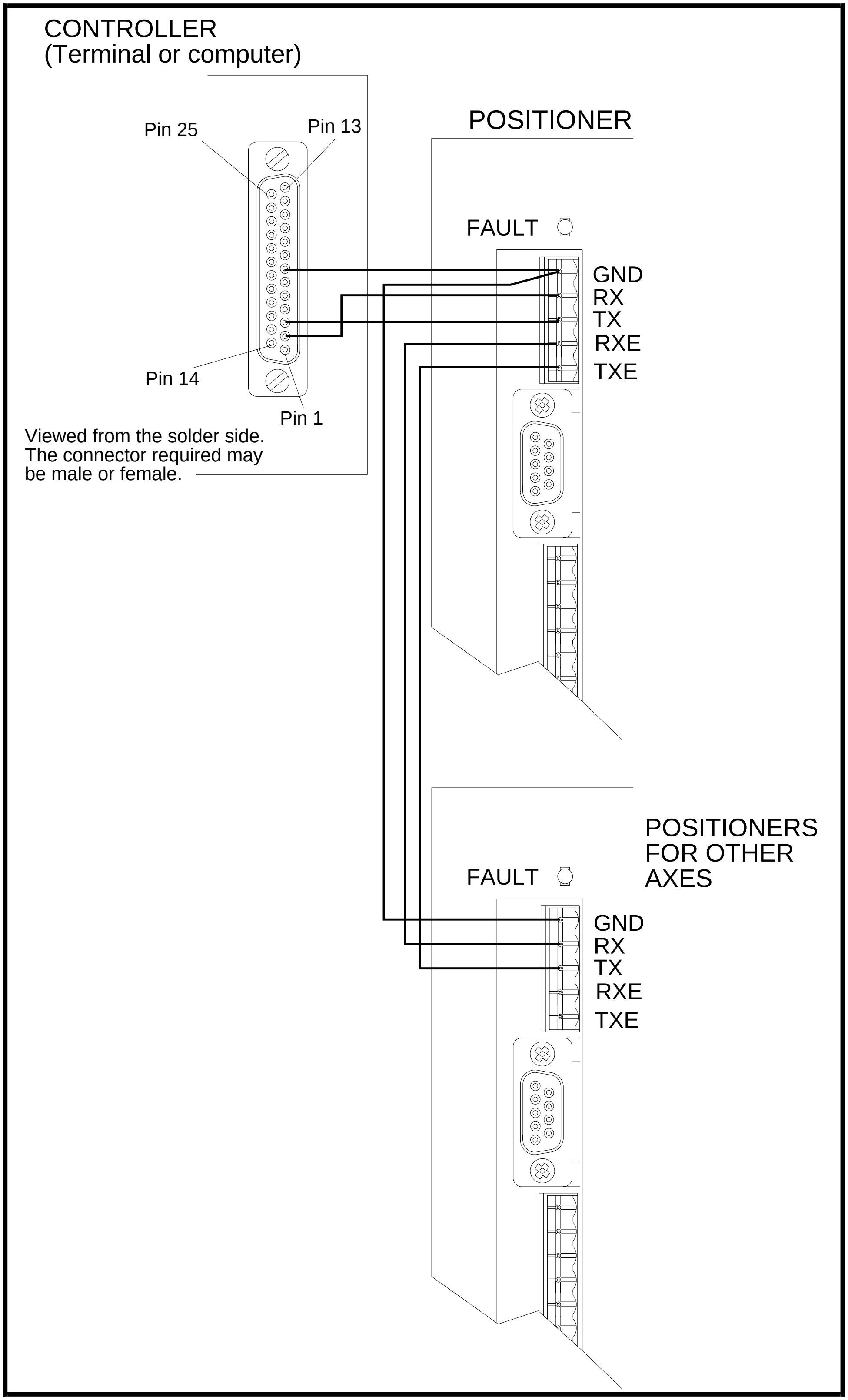


Figure 4-1. Controller to Positioner Connections

A cable suitable for this connection is available from Digiplan. Its part number is 7967.100. The connections RXE and TXE are for connection to other positioners in the system using the ZDDC as shown.

ZDDC (Zero Delay Daisy Chain)

The Zero Delay Daisy Chain is used to connect up to eight positioners to the controller and keep propagation delays to a minimum. Characters sent by the controller to the positioners are echoed back to the controller to verify that the character has been correctly received. In the ZDDC, for practical purposes, all devices are connected in parallel. To avoid confusion of data on the return line, only the addressed positioner performs the echo back.

Universal commands will be echoed back by the last positioner to be addressed; this function defaults to device no.1 at power-up.

Each positioner is equipped with an additional RS232C receiver and transmitter to enable successive positioners to be linked together.

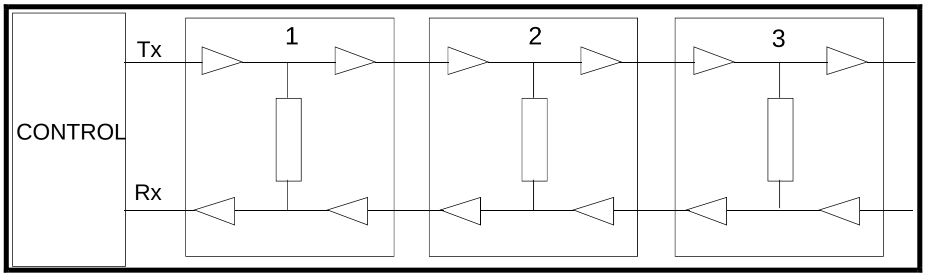
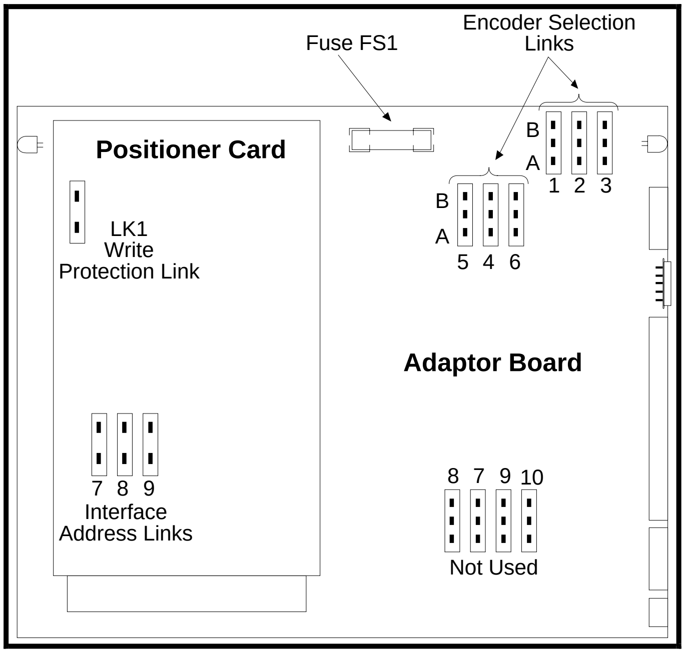


Figure 4-2. Zero Delay Daisy Chain

Positioner Address Jumper Linking

A unique address must be assigned to each positioner in the chain. Normally positioner 1 will be the first in the serial communication link. The positioner address is assigned by configuring jumper link 6, 7 and 8 on each positioner in a binary pattern as shown in Table 4-1.

The device address should ideally be set before the system is installed. Changing the address setting will cause the positioner to loose the drive definition, so it will need to be re-defined (see RFS command). This is done to minimise the risk of unexpected movement following an address change.



With jumper links 1 to 6 on the adaptor board in position B, the positioner accepts signals from the encoder wired to the Motor Feedback socket on the drive. With these jumper links in position A, the positioner accepts signals from a separate encoder connected to the 9-way Encoder socket on the adaptor board.

Links 7 to 10 on the adaptor board are not used and must be left in position A.
Links 2 to 6 on the positioner card are not used.

Figure 4-3. Positioner Jumper Link Locations

	Jumper Link 7	Jumper Link 8	Jumper Link 9
Interface 1	0	0	1
Interface 2	0	1	0
Interface 3	0	1	1
Interface 4	1	0	0
Interface 5	1	0	1
Interface 6	1	1	0
Interface 7	1	1	1
Interface 8	0	0	0

1 = link fitted 0 = no link

Table 4-1. Interface Address Jumper Links

Drive Configuration

The BL or BR drive should normally be configured as a torque amplifier when the positioner option is included. Please refer to the drive User Guide for information on setting up as a torque amplifier.

Configure the positioner to suit the drive by typing RFS4 (for a BL 16 or 23-size motor), followed by the SV command to save. Refer to the Command Listing for further information on these commands.

Using Windows™

Microsoft Windows™ comes with a terminal emulation program which can be used in place of XWARE. The terminal emulation program is configured as follows:

Run the terminal program.

From the FILE menu select NEW.

From the SETTINGS menu, setup the following options:

Terminal emulation sub-menu	DEC VT-100 (ANSI)
Terminal preferences sub-menu	Terminal modes Line wrap ON
	Local echo OFF
	CR_>CR/LF Inbound ON
Communications sub-menu	Baud rate 9600
	Data bits 8
	Stop bits 1
	Parity NONE
	Flow control NONE

Also choose the appropriate connector

These basic configurations will allow operation with the terminal only.

Chapter 5. PROGRAMMING

Chapter Objectives

This chapter provides complete information on programming the positioner to perform the motion control functions you require.

Communicating with the Positioner

The positioner is designed to work in one of two modes:

a) The human interactive mode

b) The computer host mode

The command EX1 enables the first of these and the command EX0 the second. In the human interactive mode, the positioner will respond to a dumb terminal in an intelligent, user friendly manner and provide the user with help facilities and error messages if errors occur. In the host computer mode all unsolicited output and help information is suppressed so that communication may be handled by software resident in the host computer.

If a computer is to be used for the terminal function, terminal emulation software will be required. The X-Ware program available from Digiplan for use with IBM PC's and compatibles is suitable for this purpose. X-Ware is supplied free of charge.

Individual Commands

An individual positioner command controls a single parameter, function, or action such as acceleration, velocity, position, time delay, pause, loop, go, etc. There are two classes of individual commands, Immediate and Buffered.

Individual commands are variable in length. They can consist of one or more letters with a delimiter (carriage return or space character) and one or more letters and numbers with a delimiter. Each command is entered as a character/delimiter combination. Some commands include a sign (±) to denote direction of motion. The number of characters used depends on the type of command entered.

Typical commands have the form:

```
1E
S
A10
V3
D46000
```

When two or more individual commands are entered on the same line, they are separated by spaces, and multiple command entries will be displayed on a single line of your terminal screen.

The example below shows a set of individual command entries with space delimiters on the same line:

```
MN A10 V2 D25000 L10 G N
```

If spaces are used as the delimiter and a large number of multiple command entries are made, you could exceed the ability of your terminal to display characters as a single line (80 characters per line is a typical value). When the carriage return is used as a delimiter, the cursor returns to the beginning of the next line.

NOTE: The Positioner will normally echo carriage return with carriage return line feed in terminal mode. If you get double line spacing you need to turn off the auto line feed facility on your terminal. In computer mode (EX0) only the characters received are echoed.

The example that follows shows the effect of using carriage return as the delimiter when the terminal is not in auto line feed.

```
MN
A10
V2
D25000
L10
G
```

Immediate Commands

Commands that are identified as "Immediate" are executed immediately on receipt and will take priority over whatever operation is in progress. These commands include various Stop commands that clear the command buffer, and various Status Request commands that have no effect on the command buffer.

Buffered Commands

Commands that are identified as "Buffered" are received by the positioner and stored (in the command buffer) if the positioner is not free to execute them. Stored commands are executed as soon as the positioner gets to them, in the order that they were received.

Buffer Capacity

Any combination of individual commands and command groups can be entered in the buffer until the total number of characters currently stored (including the delimiters) equals 2000. The positioner uses a first-in-first-out serial buffer. As the commands are read from the buffer, additional commands can be entered to replace them. Therefore, the possibility exists that a command set could actually consist of more than 2000 characters. Examples of command sets that might be used for different applications are presented later.

The sequence buffer is different from the command buffer in that it can be copied to battery backed up RAM using the SV command, while the command buffer is not battery backed up, so its contents are lost on power down.

Multiple Positioner Commands

When multiple positioners are on the communications line, commands like the previous example are executed by all positioners on the line. To send commands to a single positioner, the device address of that positioner should be put in front of the instruction. The device address is set using jumper links on the individual positioner.

Example:

Three positioners are on an RS232C chain. They are sent the following commands:

```
MN A10 V10 1D25000 2D50000 3T1 3D100000 G
```

Unit 1 moves 25000 steps, unit 2 moves 50000 steps, and unit 3 waits 1 second and moves 100000 steps. All three use the same rates.

NOTE: because multiple positioners receive characters in parallel, universal commands are always echoed by the last addressed axis (not by all positioners). An addressed axis continues to echo characters up to the next address i.e. the "2" of 2D50000 of the above example is echoed by axis 2 whereas 1D250000 and the space is echoed by axis 1. After power-up, axis 1 is the default axis.

During sequence download to one axis in a chain, all of the other axes must have their communication switched off by use of the F command. The alternative is to make each command in the sequence device specific by adding the device address.

Programming Modes

Normal Mode

The command MN selects normal mode. In this mode moves are set by the D command. Within normal mode there are two options, absolute mode and incremental mode:-

Absolute Mode

The command MPA puts the positioner into absolute position mode. Distances entered using the D command are interpreted as absolute positions relative to absolute zero

Example:

MPA	Sets absolute mode
PZ	Sets current position to be absolute zero
D4000	Sets move to absolute position 4000
G	Execute move
D2500	Sets move to absolute position 2500
G	Executes move of 1500 steps CCW

Incremental Mode

The command MPI puts the positioner into incremental positioning mode. D command distances are interpreted as the number of steps to move from the current position.

Example:

1MPI	Sets incremental mode
1DPA	Response=1500 (current position)
1D4000	Sets move to be 4000 steps CW
1G	Executes move to position 5500
1D-3000	Sets move to 3000 steps CCW
1G	Executes move to position 2500

Continuous Mode

The command MC selects continuous mode in which the motor runs continuously at the specified velocity until stopped or a new velocity is programmed. Distance data is ignored.

Example:

MC	Sets continuous mode
D4000	Ignored in continuous mode
A50	Sets acceleration to 50 revs/sec^2
V20	Sets velocity to 20 revs/sec
G	Motor runs at 20 revs/sec continuously
V10	Motor speed changes to 10 revs/sec
S	Motor stops

Speed Change Mode

The MQ command allows speed changes to be made during a preset move. The speed changes can be triggered at a set distance using the TRD command or by an input using the TRE command.

Example:

3MQ	Set MQ mode for axis 3
3A50	Set axis 3 acceleration to 50 revs/sec^2
V15	Set velocity to 15 revs/sec
D8000	Set move distance to 8000 steps
G	Start move
TRD3000	Go to next command at D = 3000 steps
V5	Change velocity to 5 revs/sec
TRE_X1	Go to next command when Input 2 goes high
V2	Change velocity to 2 revs/sec

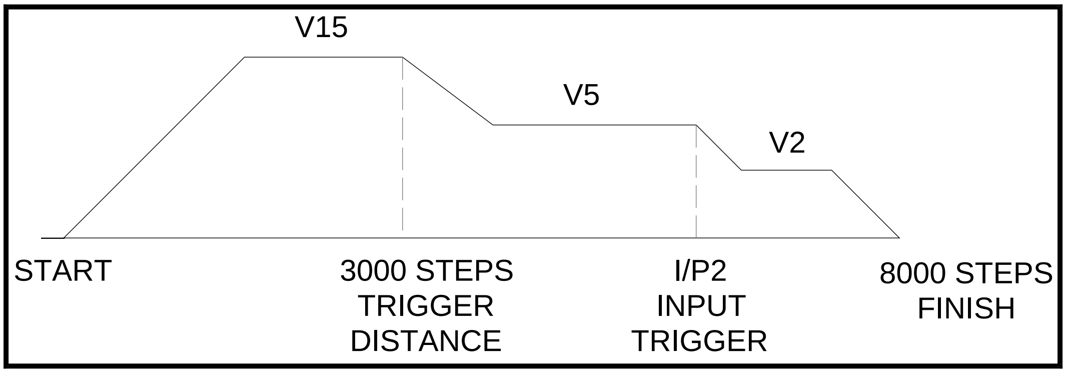


Figure 5-1. Speed Change Mode

Command Types

Start Move Commands

The positioner reads and stores individual commands in the order they are entered. Each command is read and executed before the next command is read. The GO command "G" is the principal buffered command that initiates shaft motion. When the buffer reads a GO command, the motor will then move in accordance with the move parameters that reside in the positioner at the time the Go command is read.

When the Go command is read, the positioner will incorporate any motion parameter command that was encountered prior to the Go command. For example, the "A " command would be combined with the Go command to change the acceleration value as follows:

A123 G

This command instructs the Positioner to change the acceleration value to 123 rev/sec/sec and then execute the move using the last mode, velocity, and distance values that were previously entered.

When the RS232C positioner is initially activated, a series of commands of the form " MN An Vn Dn G " is required. These command sets up the initial operating conditions for the positioner. After the positioner has been initialised, single move parameter commands can be used.

If only "G " is entered, the positioner will repeat the previous motion pattern. For example, a string of Go commands:

G G G G

would instruct the positioner to move the motor the same way four times.

There is no need to re-enter all the shaft motion commands to change one of the variables. If one or more of the motion parameters is changed by a command entry and the Go command is then entered, the prior pattern will be repeated using the new motion parameter(s) that were entered prior to the latest GO command.

For example:

A14 G V2.6 G D-27634 G

would change acceleration to 14 and move, then change velocity to 2.6 and move with 14 as the acceleration value, then change position and move again using 14 and 2.6 as the acceleration and velocity values.
The CONTINUE command "C " is used following execution of a PAUSE command, "PS " or "U ". CONTINUE will allow execution of the next command waiting in the command buffer (if any command is stored).

Loop Commands

The LOOP command allows a cycle to be repeated continuously "L" or a given number of times "Ln" up to 65535. The END-OF-LOOP command "N" indicates where the loop ends. The END-OF-LOOP command can be used to indicate that the positioner should proceed with further commands after the designated numbers of loops have been executed, or in combination with the "Y" command to indicate where execution is to stop. The "U" command may be used to temporarily halt Loop execution, the C command will then cause the loop to resume execution.

Stop Move Commands

The STOP MOVE commands are presented in order of severity of response. SOFTWARE RESET "Z" command is the most abrupt and severe stop command you can use. The STOP AT END OF CURRENT LOOP command "Y" is the least abrupt and severe stop command.

Stop at End of Current Loop Y

Also the 'stop sequence' command.

This command will not halt processing until command processing reaches the character N at the end of the command loop. At that time, the Positioner will execute the next command in the buffer after N, if any. The command loop cannot be restarted without re-entering the commands.

Controlled Stop command S

In the PRESET or CONTINUOUS mode this command will decelerate the motor to a stop at the last used acceleration rate.

An "S " command will always cause a deceleration to velocity zero at the last used acceleration.

The "S " command clears any remaining commands in the command buffer unless prevented from doing so via the SSH command.

NOTE: Normally, the motor is decelerated to a stop at the same rate as it was accelerated. A different deceleration rate may be programmed in CONTINUOUS mode only by calling for a velocity of zero rev/sec with a new acceleration and executing a V0 rather than an "S " command.

Example:

" MC A1 V10 G .... A100 V0 G "

Controlled Stop command LS

The LS or 'Limit Stop' command will decelerate the motor to a stop at the deceleration rate defined by the LA or 'Limit Acceleration' command. This will guarantee a quick stop for any move.

Kill Command K

This command stops positioner commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a command sequence download in progress (XD command). The command buffer is also cleared.

Software Power-on Reset Z

The "Z " command is equivalent to cycling the AC power to the Positioner; that is, it disables the communications interface and returns all internal settings to their power-on values. The command buffer is cleared. Like the "K " command, "Z " causes an immediate cessation of output pulses to the motor. Z also de-energises and resets the drive ready for re-initialisation at power up.

NOTE: When the "Z " command is used, the Positioner is busy for up to 2 sec and will ignore any commands.

Pause Commands

There are four types of pause commands available as follows:-

Pause and wait for continue command C to continue. A common use of the Buffered Pause PS is to hold execution of individual commands in a Command Loop until the entire loop has been loaded.

Tnnn

Pause in processing commands for a given number of seconds and then continue.

U

Hold (pause) now and wait for continue. This is an immediate command.

TRE

Pause in processing commands until the designated TRIGGER input pattern is present.

Triggers are used to synchronise positioner operations with external events. They can be used to implement a "handshaking" function with other devices. See command details for exact syntax.

Status Request Commands

All status request commands result in data being returned to the controller from the positioner. To prevent multiple positioners from all responding at once, the status request commands are given the classification "Device Specific" meaning that the device address of the responding positioner must be placed in front of the command. No positioner will execute a Device Specific command without its device number.

The use of status request commands must be conducted in an orderly fashion. Commands should only be issued when the host is ready to read the response. New commands should not be sent until the response is received. In particular, after a buffered status command, an immediate status command should not be sent until the response has been received by the host. If this procedure is not followed, the command responses will be intertwined, rendering the information useless.

There are two status commands that can be used to request position.

The DPA Command will report how many steps the motor has moved relative to the absolute zero position. This position is calculated by summing the total number of moves commanded since the positioner was at the absolute zero position. If position reports are needed that are relative to the beginning of each individual move, the PR command should be used.

Drive Energise/De-energise Function

Energising and de-energising the drive is subject to the commands given and error status as follows:

The drive will energise if:-

The ST0 or ON command is given when the ESTOP, WATCHDOG, FOLLOWING ERROR and EXCESS TORQUE DEMAND signals are inactive.

The drive will de-energise if :-

The ST1 or OFF command is given.

The ESTOP or WATCHDOG signals become operative.

A FOLLOWING ERROR or an EXCESS TORQUE demand occurs.

The de-energising signal must be restored, then the ST0 or ON command must be given to re-energise the drive.

Homing Function

The GH command causes the motor to rotate in the direction and at the speed specified until the home position is reached. In practice, it continues at little beyond this point but you can locate the home position accurately by then using the commands:

```
MPA D0 G
```

(The GH command always makes the home position zero)

The SP command can be used after homing to change the accurate home position from zero if required.

If the home position is not found in the direction given in the GH command, the limit switch position will be reached and the switch will be operated. The motor will then reverse and seek the home position in the opposite direction. For recognised home detection, the home switch must always be activated from the direction specified in the command (shown by the arrow in Figure 5-2), so as illustrated in this figure, a limit switch reversal will take place if the direction given in the GH command is away from the home position (GH+ in the illustration).

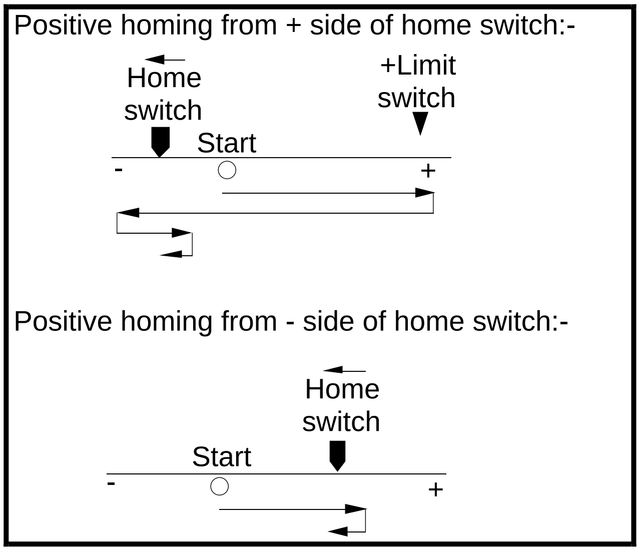

Positive homing from + side of home switch:-
Home switch
+Limit switch
Start
-
+
Positive homing from - side of home switch:-
Home switch
Start
-
+

Figure 5-2. Positive Homing

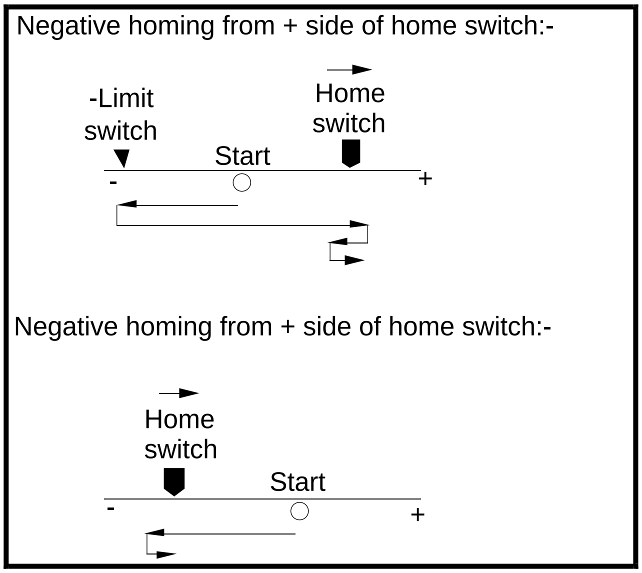

Negative homing from + side of home switch:-
-Limit switch
Home switch
Start
-
+
Negative homing from + side of home switch:-
Home switch
Start
-
+

Figure 5-3. Negative Homing

For servo systems fitted with an incremental encoder with an index track, option OSA enables homing to an encoder index mark within the home proximity range, as opposed to the home switch edge. The mark is always approached from the specified direction. Using an index mark to determine the stopping point results in the home position remaining fixed even if there are small variations in the operating point of the home switch. The final home position will be the leading edge of the first index pulse following detection of the home switch edge.

Basic Programming Guide

This section uses examples of simple programs as an introduction to programming. If you are unfamiliar with the system, after reading Chapter 1 a study of this section will give you an insight to the usage of the command language.

The system configuration is a brushless servo motor with a 500 line/rev encoder. 4X decoding is used and the user resolution has been set to 200 steps/rev. Tuning the servo loop is dealt with in the section headed "Servo Tuning", so we will assume that it has been completed. The positioner jumper links have been set for unit 1 addressing. The motor starting position is 400 steps CW from the home position.

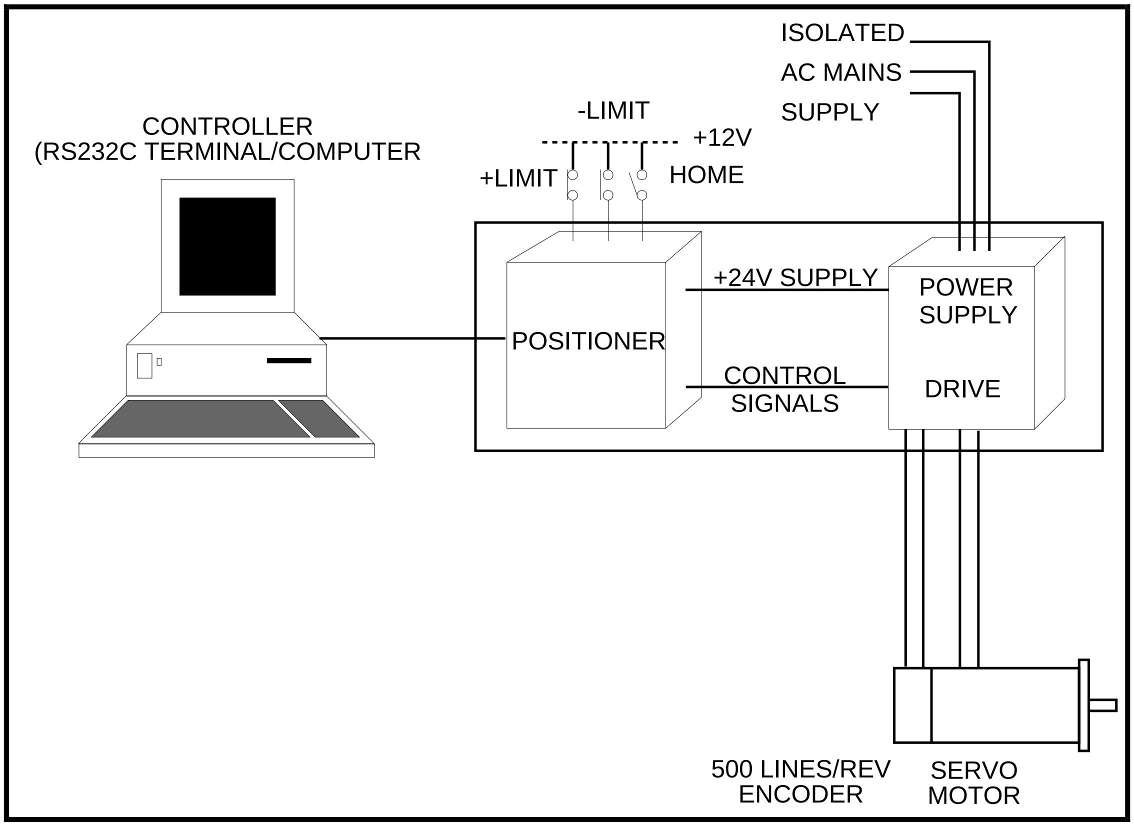


Figure 5-4. System Used in Example Programs

Example 1

In this simple example the requirement is to cause the motor to perform two revolutions in a CW direction and stop.

The following commands in the order shown will fulfil the requirement:-

Command	Purpose
MN	Set move mode to normal
1A10	Set acceleration motor 1 = 10
1V5	Set velocity motor 1 = 5
1D400	Set move distance motor 1 = CW 2 revs
1G	Start move - motor 1 moves 2 revs CW

Description

Commands 2 - 5 cause motor 1 to accelerate at 10 revs/sec^2 to a velocity of 5 RPM, travelling CW for 400 steps (2 revs) at the G command.

Example 2

This example causes axis 1 to go to the home position, then turn 5 revolutions CCW at 10 revs/sec and set the resulting position as absolute zero. The motor finally turns 8 revolutions CW to finish at absolute position 1600.

Command	Purpose
ST1	Energise the drive
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
GH-2	Send axis 1 to go home (2 revs/sec CCW)
1D-1000	Set axis 1 to go 5 revs CCW
G	The axis moves to the required absolute zero position
MPA	Set absolute position mode
PZ	Set position to absolute zero
1D1600	Set move to abs. position 1600 (8 revs CW)
G	Move to position 1600 (1 rev CW from starting position)

Description

Commands 2 - 6 cause the motor to go to the home position and then turn 5 revolutions CCW. Command 7 sets the absolute positioning mode and Command 8 sets the resulting position as absolute zero. Commands 9 and 10 cause the motor to turn 8 revolutions CW to absolute position 1600 which is 1 revolution CW from its starting position.

Sequences

Introduction

A sequence is a series of commands that is executed in order whenever the sequence is run. Immediate commands cannot be stored in a sequence since they cannot be stored in the command buffer. The only "sequence legal" commands are buffered commands.

The positioner has a section of non-volatile RAM totalling 6400 characters allocated for storing up to 63 sequences. The sequence buffers are of variable length, so they can store long sequences of up to 6400 characters or several shorter ones, provided that the total length of all sequences does not exceed the allocated 6400 character space.

During normal operation a command can specify the execution of any sequence stored in memory.

Sequence identifiers: number from 1 - 63.

When a sequence is programmed, the positioner automatically calculates a checksum for the sequence and stores the length of the sequence and its checksum in an internal directory.

The command XSS can be used to verify the existence of a sequence.

If a power-on sequence is used, the sequence and its sequence number are stored in the non-volatile RAM. A power-on sequence execution failure will be indicated by flashing the LED.

Programming Sequences

Sequences are programmed by first sending a special start command XD, which clears the command buffer. All subsequent commands are stored in the command buffer (but not executed) until an end of sequence command (XT) is received. Any condition which prevents proper recording of the sequence is saved and can be accessed with the status command XSD. The programmed sequence can be tested with the run sequence command XR, which executes any specified sequence.

The XU command can be used to read back the entered sequence.

To begin the programming of a sequence, the XD command immediately followed by sequence identifier number (1 to 63) and a delimiter must be entered. An XT command ends the sequence. The save command SV should be used to store the sequence into the positioner's non-volatile memory.

The commands entered after the XD command and before the XT command will be executed in the order in which they were entered when the sequence is run.

Example:

NOTE: The ">" prompt is replaced by the "$" prompt when in sequence edit mode.

XD1	Begin definition of sequence No.1
A10	Set acceleration to 10 revs/sec/sec
V10	Set velocity to 10 revs/sec
D4000	Set move distance to 4000 steps
G	Go
H	Reverse next move
G	Go
XT	End of sequence

* The definition of sequence No.1 is terminated by XT but the sequence is NOT stored in non-volatile memory until
the save command SV is given.

NOTE: Remove LK1 on the positioner card (see Figure 4-3) to write-protect sequences.

Whenever the above sequence is run, the motor will turn 1 revolution CW and then 1 revolution CCW.

Running Sequences

By Command Input

A sequence can be run by entering the XR command immediately followed by a sequence identifier number (1 - 16) and a delimiter.

Example:

XR3[space] runs sequence No.3

Standalone Operation

Stored sequences may be automatically executed when you power up the system or executed by remote switches.

By Hardware Inputs

A sequence selected as a binary pattern set by input switches can be run in response to a trigger on Input 4 if SSF1 is set.

A sequence selected as a binary pattern, set by input switches, can be run in response to a trigger on Input 4 if SSF1 is set.

Selection is by Inputs 1-3 if OSD0 is set.	(1-7)
Selection is by Inputs 1-3 and Input 7 if OSD1 is set .	(1-15)
Selection is by Inputs 1-3 and 5-7 if SSI1 is set.	(1-63)

Table 5-1 shows how the binary sequence number can be setup using inputs I1 to I7 (I4 being used as an input strobe line). D5 is the most significant binary bit and D0 the least. Sequences 1-7, 1-15, or 1-63 can be selected depending upon what options are set using SSI, OSD and SSF.

Options set			I7	I6	I5	I4	I3	I2	I1
SSI0	OSD0	SSF0	User	User	User	User	User	User	User
SSI0	OSD0	SSF1	User	User	User	Strobe	D2	D1	D0
SSI0	OSD1	SSF1	D3	User	User	Strobe	D2	D1	D0
SSI1	OSD0/1	SSF1	D5	D4	D3	Strobe	D2	D1	D0

Table 5-1. Sequence Selection

Example:

Set SSF1 and set Inputs 1-3 and Input 7 as follows:

Input 1 = 1
Input 2 = 0
Input 3 = 0
Input 7 = 1

On an Input 4 transition from low to high level, sequence number 9 will be executed if OSD1 and SSF1 have been set. The Input 4 trigger will not be accepted if the axis is busy. When a PLC is used, it is recommended that an output (O3 for example)is set at the beginning of a sequence to indicate to the PLC that the axis is busy. A command could then be added at the end of the sequence to reset the output and indicate to the PLC that the axis is ready to receive a new trigger.

Power On Sequence

One of the stored sequences can be designated to be executed when the device is powered up. The sequence to be executed can be selected by prior RS232C command (XP1-16). The single sequence specified is executed once; control then passes to the normal command processor loop. Alternatively, power-on execution can be disabled with either the XP0 or XZ commands.

It is possible to have the positioner execute a sequence on power up by defining the commands within a sequence using the XD, XT and XP commands. To define sequence No.1 the operator enters:

```
XP1 XD1 MN A10  V5 D40000 G XT
```

In this example the operator would not power the unit on until he was ready for the sequence to begin. The XP1 command enables a single run of sequence No.1 after power on. The XD1 command signifies the start of the definition of sequence No.1 and the XT command signifies the end of the sequence No.1 definition. When the operator applies power to the positioner, sequence number 1 will be executed (the motor will turn 10 revolutions in the positive direction). The sequence can be stopped using the S command or, if necessary, by using the emergency stop (K command).

Changing a Programmed Sequence

Once a sequence is defined it cannot be redefined until it has been deleted. A new sequence cannot be downloaded over an existing one.

To change a sequence, it must be be deleted from memory and then redefined as a new sequence using the same number.

First, the sequence must be deleted from memory using the XE command. It will not be deleted from back-up RAM unless the SV command is used after the XE.

Example:

XE1 erases sequence No.1

The changed sequence should then be entered. It will not be saved to non-volatile memory until the SV command is issued.

Examining the Contents of a Sequence

If the operator wishes to check the contents of the sequence, he would enter:

```
1XU1
```

This would cause the positioner to send the contents of sequence number 1 to the terminal's screen. The 1 preceding the XU command is the device address which must be present since the XU command is a "device specific" command. After issuing this command the following will be displayed on the terminal's screen:

```
MN A10 V5 D40000 G
```

This is the command string that is performed when sequence number 1 is executed.

If the operator does not wish to have the sequence execute on power up, it can be cancelled by issuing either the XP0 or XZ command.

Sequence no.1 can also be run manually using the XR command.

Sequence Status Requests

Sequence status commands include XC, XSD, XSP, XSR, XSS. The XC command can be used to verify that the memory in the back-up

RAM has not been corrupted in any way since last power down. The XC command will cause the positioner to send a checksum in the form of three decimal digits (000 - 255) to the terminal. The checksum may be recorded when sequences have been defined, so that it may be used for comparison later, such as each time the positioner is powered up.

Interactive operation with a PC/PLC

To create a reliable, trouble free interactive RS232 communications control program between a PC/PLC and the serial interface you should note the following points :

- Use the X-WARE package for terminal emulation on an IBM PC or equivalent computer system.

- The interface should only be operated in computer mode to suppress error messages and extended text output.

- The program should operate the interface with a response check during critical processes, such as sequence download. The response check will need to detect :

- No response at all
- Incorrect characters
- More characters than expected
- Less characters than expected
- Control characters such as EOF caused by errors
- Correct response

The response expected should be either echoback, or character returned by interrogation commands, or the sequence of the two.

- If there is no check for echoback on each character, you should include a delay of up to 1ms between each character sent . This may require characters to be sent individually rather than as strings.

- You should not use the simple BASIC INPUT statement as this 'hangs' the link without time out if an error causes no carriage-return character to be received from the interface.

- Interpreted BASIC should not be used, as it is too slow for most applications.

- Interrupt error handling, such as the BASIC 'ON ERROR" statement, should be enabled.

- Commands such as SV, Z, RFS and RIFS take several seconds to execute. This also applies to status commands like DR and HELP. It is recommended you insert a suitable delay after these commands to allow them to fully execute their intended function without corrupting immediately following commands, although the use of these commands in an interactive situation is unlikely.

- If problems arise the communications error responses actioned by the R command will help diagnosis. In some cases a recovery can be attempted by sending the command again (such as the R command itself). But this may not always be appropriate, for example LD3 altered to L3 would not be fully corrected by simply sending LD3 again.
- A routine that breaks some of these rules may appear to work satisfactorily, but is not necessarily safe. The worst case communications response condition may involve a number of coincidences, such as sending a status request just as the interface hits a limit, just as it starts decelerating or just as the servo is responding to a load change. Any motion control requirement explicitly takes priority over communications under these circumstances, and can therefore cause a communications problem unless you wait for echoback. Also different issues of software may have different execution time characteristics which can make a routine based on delays work at one software issue, and not another.

Accurate defensive programming is required if safety constraints are to be met and possible damage to hardware prevented.

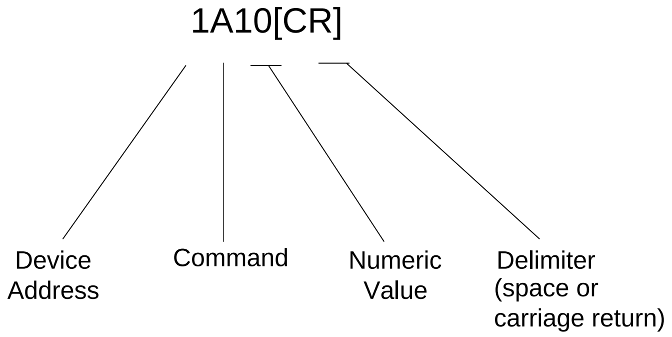
Command Structure

This section defines the basic format for positioner commands.

Basic Format

All positioner commands are sequences of upper case characters of the form:

[device address] [command] [numeric value] [delimiter]



where:

device address - is a single digit from 1 to 8. It determines which device will receive the command. Its value will be between 1 and 8 since up to 8 positioners may be present in a system. If the device address is omitted, the command applies to all positioners.

command - up to four characters beginning with an uppercase ASCII letter

numeric value - a number defining the value to be programmed, such as the number of steps to move

delimiter - is a SPACE or CARRIAGE RETURN

All individual commands end in a delimiter that signifies that the command is complete. A delimiter serves the same function as the space between words in a sentence. The delimiter, which is part of the command, is a space character (entered with the keyboard space bar) or a carriage return [CR].

The format for each individual command is given by the **syntax**, for example:

<a>GH<s>n

<a>	is the device address
GH	is the command
<s>	is the sign, indicating direction of travel
n	is the numeric value, in this case the speed in revs/sec

Characters inside <> are optional.The direction is taken to be positive if no sign is included.

Sign <s>

s = sign (+ or -)

The sign placed before the numeric value indicates the required direction of travel for a move. If the sign is omitted, a positive value is assumed.

Numeric Value n

The value of n may be expressed as:

an Integer (a whole number such as 40)

a decimal number such as 10.25

a scientific E notation number such as +12.3840E-04 etc.

NOTE: All commands which take a value, such as a velocity command like 1V5, will return the current programmed value if none is sent. So sending 1V[CR] will result in the positioner responding with a message such as *V = 5 revs/sec.

Defaults

Command parameters have default or factory setting values which are listed for each command. The value given is the general product default (corresponding to the commands RFS0 followed by RIFS). See commands RFS and RIFS for further information.

Command Attributes

All commands are either Immediate or Buffered and Universal or Device Specific, as explained below.

Immediate (I): Executed immediately upon receipt.

Buffered (B): Stored in the command buffer and executed in sequential order.

Universal (U): Commands that are intended for all positioners in the system. They do not require a device address. Any universal command can be made device specific by including a device address.

Device Specific (D): Commands that are intended for one positioner only and therefore require a device address for execution. All commands that request data to be transmitted back to the host are device specific. This includes position report backs and status requests. All 'Universal' commands can be made 'device specific' by including a device address, except the 'Z' (reset) command.

Buffered/Immediate Commands

All commands annotated with * can be prefixed by the character B, in which case they become buffered rather than immediate commands.

WARNING: Although every effort is made to ensure that the default values of parameters are harmless and conservative, there can be problems when the unit is connected to non-standard servo drives or loads. Under these circumstances follow the tuning procedures starting with low servo gains as described.

DETAILED COMMAND LIST

A Acceleration Rate

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>An	Revs/sec^2	n = 0.030 - 999,999	10	Buffered, Universal Savable in sequence

EXECUTION TIME

SEE ALSO V, D

Description

Set the acceleration rate. The acceleration remains set until it is changed or the system is reset.

If n = 0 there is no acceleration period. The maximum possible acceleration is used (start/stop move).

Example

Command	Description
1A10	Sets acceleration of device 1 at 10 revs/sec^2
A5	Sets acceleration of all devices to 5 revs/sec^2

B Buffer Status Request

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
aB	N/A	N/A	N/A	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO BS

RESPONSE TO aB **IS** *B or *R

Description

The command requests the status of the command buffer. The response, *R[CR] or *B[CR], indicates if more or less than 10% of the command buffer is free.

*R = more than 10% of the buffer is free
*B = less than 10% of the buffer is free

The command is normally sent after a long series of commands to ensure that room exists for more commands. The communications buffer is 2000 bytes long.

Example

Command	Response
1B	*B (less than 10% of command buffer is free)

BS Buffer Size Request

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aBS	Bytes	0 - 2000	N/A	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO B

RESPONSE TO aBS **IS** *n

Description

Requests the remaining bytes available in the command buffer.The response, 4 decimal digits (0-2000) followed by a [CR], indicates the number of bytes remaining in the command buffer. When entering long strings of commands, check the buffer status to ensure that there is enough room in the buffer, otherwise commands will be lost. Each character (including delimiters) uses one byte.

Example

Command	Response
1BS	Response *122[CR] (122 bytes remaining in the buffer)

C Continue

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>C	N/A	N/A	N/A	Immediate, Universal, Never Saved

EXECUTION TIME

SEE ALSO PS, U

Description

The continue command ends a pause state. It enables the interface to continue executing buffered commands after a pause has been initiated with the PS command or the U command. This command is useful when you want to transmit a string of commands before you actually execute them.

Example

Command	Description
MC	Set continuous mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
PS	Pause
G	G will not be recognised until PS is cleared with C
T10	Delay for 10 secs
V0	Stop
C	Continue

CAG Configure Acceleration Gain

VALID FROM Version 3.2

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CAGn	Milliseconds	1.0 to 32768	1.00	Immediate, Universal

EXECUTION TIME

SEE ALSO SIMM, CCS

Description

During following in scaling mode (SIM3, 4) the acceleration profile to the input stream velocity can be adjusted using the CAG command. CAG sets the time constant of a digital filter which profiles or smoothes the input pulse stream. In response to a step change in input frquency, the output will accelerate or decelerate to the new rate following an exponential profile. The CAG value represents the time in milliseconds to reach 67% of the final speed; 95% of the final speed is reached after 3 times the CAG value.

Example

Command	Description
ST0	Motor energised
SIM3	Normal unreversed scaled following
CCS3	Clock and direction decode
CUR200	The input pulse stream resolution
CMR4000	The system 7 motor resolution
CAG100	100ms input filter controls acceleration
1SV	Save, and power up in the following mode

CCP Configure Command Peak

VALID FROM Version 3.0

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aCCPn	N/A	n = 0 - 1023	1023	Immediate, Universal

Description This command can be used to limit or clamp the maximum value of the output demand to a value less than the full range of -1023 to +1023. Since the drive is configured as a torque amplifier, CCP serves as a torque-limiting function.

Example

Command	Description
1CCP102	Limit the interface output to -102 to +102 i.e. 10% of full scale

This clamping action is the last to take place before the control signal is output, therefore a command such as CCP0 would block any output signal to the drive.

CCS Configure Command Source

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aCCSn	N/A	n = 0 - 3	0	Buffered, Device Specific

EXECUTION TIME **SEE ALSO** SIM

Description This command is used to configure the decode of the second axis encoder input when using following mode of operation as follows:-

n = 0	Normal x4 decode
n = 1	x2 decode
n = 2	x1 decode
n = 3	Clock and direction decode

Example

Command	Description
1CCS1	Set axis 1 to x2 decode
1SIM1	Select encoder following mode, axis 1

If an encoder to be tracked by axis 1 has 1000 lines per rev, it is decoded as 2000 steps per rev (x2 decode). The motor on axis 1 with 4000 steps per rev would move 1/2 revolution in response to a 2000 step move (1 revolution) of the tracked axis.

*CEW Configure In-Position Window

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CEWn	Steps	n = 0 - 32,767	50	Immediate, Universal

Description This command, together with the CIT command, can be used to configure an in-position window, which in its turn can be used to indicate that the preceding index has terminated. Output 02 can be configured with the SSC command to reflect the state of the in-position detector, thus allowing the user to trigger external hardware from the in-position condition. The in-position condition is met when:-

a) The indexer algorithm has finished (no input position command)
b) The CEW condition is met i.e. the following error is less than that specified by CEW
c) The above condition has been true for the length of time specified by the CIT command

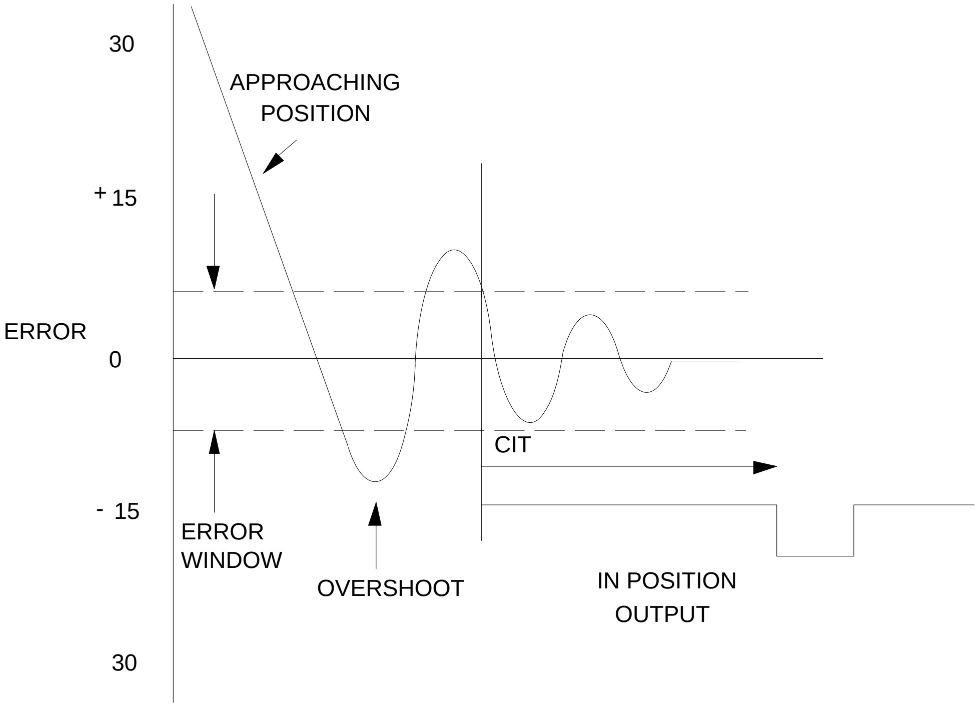


Figure 5-5. Final Positioning

*CDG Configure Derivative Gain

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CDGn	N/A	n = 0 - 32,767	57	Immediate, Universal

Description Configure the derivative gain when using PID tuning. This term represents the derivative of position error, in other words the rate at which position error is changing. It produces a damping action in a similar way to velocity feedback and sets the velocity feedback and feedforward gains to equal values.

*CFG Configure Feedforward Gain

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CFGn	N/A	n = 0 - 32,767	57	Immediate, Universal

Description Used to set the velocity feedforward gain. The opposing action of proportional and velocity feedback results in a position error which depends on speed. This is called 'following error'. Velocity feedforward can be used to offset the following error and improve tracking accuracy. This is important in contouring applications. For true PID, this can be set to the same value as the velocity feedback, or the CDG command may be used instead.

*CIG Configure Integral Gain

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CIGn	N/A	n = 0 - 32,767	0	Immediate, Universal

Description Used to set the integral gain. If proportional feedback is insufficient to overcome static position errors due to friction, integral action accumulates a steady-state position error until sufficient torque is applied to move the load to reduce the error. It improves overall positioning accuracy, but low frequency oscillation may occur around the commanded position.

*CIT Configure In-Position Time

VALID FROM

Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>CITn	Milliseconds	n = 0 - 32,767	20	Immediate, Universal

EXECUTION TIME

SEE ALSO CEW, SSC

Description

This command is used to specify the time period that the servo is to be within the in position window before the 'in position' signal is generated. The range of n is 0 to 32,767 and it represents the number of milliseconds to be used as the testing time frame. The shortest time for which the motor must be stopped and within the window before it is considered 'in position' is 2 milliseconds. If at any point during that 2 milliseconds the motor is outside the window, then the 'in position' output will remain inactive.

Example

Command	Description
1SSC1	Set output 2 as 'in position'
1CIT30	Set 'in position' time to 30ms
1CEW20	Set error window to 20 steps

*CIW Configure Integral Action Window

VALID FROM

Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>CIWn	User steps	n = 0 - 32,767	40	Immediate, Universal

EXECUTION TIME

SEE ALSO OSB

Description

Sets the width of the of the position window within which integral action is active when the indexer demanded motion is complete.

Example

Command	Description
1CIW5	Figure 5-6 shows the integral action window set to 5 user steps by this command and it initiates final positioning under integral gain.

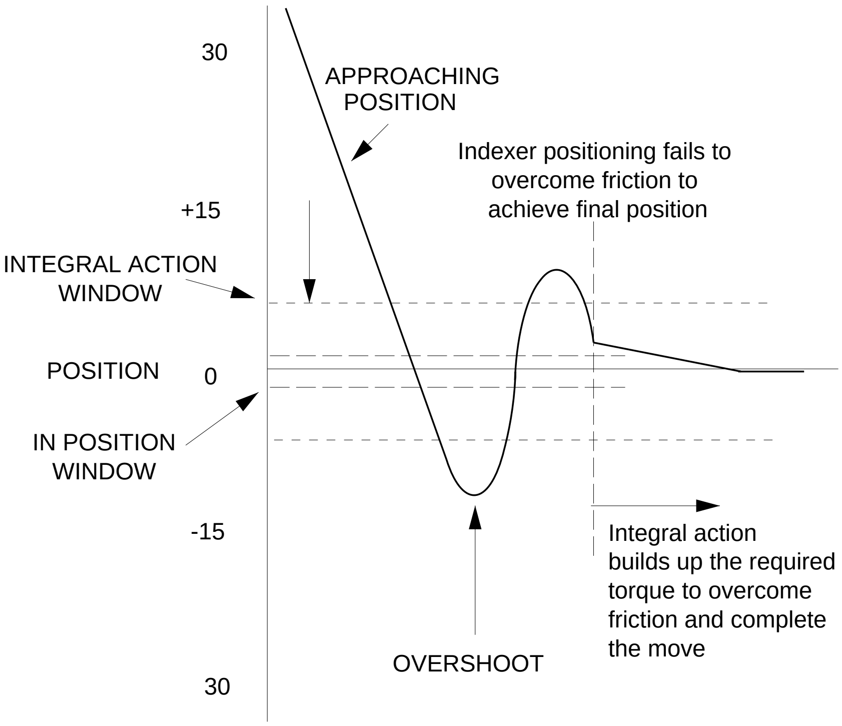


Figure 5-6. Integral Action

*CIX Configure Index Mark Resolution

VALID FROM

Version 3.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CIX	Steps/mark	1 - 32,767	4000	Immediate, Universal

EXECUTION TIME

SEE ALSO CUR, CMR, OSK

Description

For servo systems fitted with an incremental encoder with an index mark, this command programs the RS232C control module with the encoder resolution in terms of the number of steps between index marks. This number is used for encoder error checking (see option OSK). It is usually the number of steps per rev of the motor and it will normally be 2000 or 4000 for a brushless system. This parameter does not need to be changed unless the motor is changed to one having a different encoder resolution or number of steps per revolution.

Example

Command	Description
1CIX2000	Set the index mark resolution for device 1 for a 500 line encoder

Note that the control module uses a x4 decoding circuit giving an effective resolution of 4 times the number of encoder lines.

*CJL Enter Combined Motor & Load Inertia

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aCJLn	Kg-cm^2	N/A	1.6	Immediate, Device Specific

EXECUTION TIME

SEE ALSO HELP11, CTQ

Description

This command has no effect on the behaviour of the RS232C Control Module. It allows entry of the combined motor and load inertia in calculations when using HELP11 as an aid to setting up the servo system parameters.

Example

Command	Description
1CJL1.8	Defines a motor and load inertia of 1.8 Kg-cm^2

*CMR Configure Motor Resolution

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CMRn	Steps/rev	n = 1 - 32,767	4000	Immediate, Universal

EXECUTION TIME

SEE ALSO CUR, CIX

Description

Programs the RS232C Control Module to match the motor resolution for the servo motor or the open loop stepper motor. The number n is determined by the encoder fitted to the motor or the number of steps of an open loop stepper relative to your required speed and velocity distance units (motor revs, metres, table revs etc.). It will normally be 2000 or 4000 for a brushless sytem to be programmed in rps units at the motor.

Example

Command	Description
1CMR2000	Set the resolution for device 1 to be programmed in motor revs per second for a 500 line encoder

Note that the control module uses a x4 decoding circuit giving an effective resolution of four times the number of encoder lines.

*COFF Configure Amplifier

VALID FROM

Offset

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>COFFn	N/A	n = +1023 - -1023	0	Immediate, Universal

Description

This command can be used to cancel the effect of an offset in the torque amplifier. An excessive offset error can sometimes be the cause of movement at power up, but it is unlikely to affect the closed loop operation unless it is very large. The value should be chosen such that at standstill, position demand equals actual position. It can be set by opening the loop (make CPG, CVG, CIG all zero) and adjusting COFF for zero drift, or reading back the position using the DPE command and setting COFF for the smallest error.

*CPE Configure Position Error

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CPEn	Steps	n = 0 - 32,767	4000	Immediate, Universal

Description

Defines maximum allowed following error. If the absolute position error is greater than this number, the RS232C Control Module will de-energise the drive. If a valid number in steps is entered, it will become the new maximum following error, otherwise the current setting is reported. Exceeding the maximum following error is a fault condition that will cause the amplifier to be shutdown. If the maximum following error is defined as 32,768 the shutdown function is disabled and no amount of following error will generate an error condition or shutdown the motor.

NOTE: In the event that the gain used during setting up is too great, a small position error setting will prevent oscillation and potential mechanical damage to the connected system.

Example

Command	Description
1CPE1000	Set the following error limit of device 1 to 1000 steps

*CPG Configure Proportional Gain

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CPGn	N/A	n = 0 - 32,767	92	Immediate, Universal

Description

This command is used to set the proportional gain. The proportional gain determines the amount of torque produced in response to a given position error. It sets the stiffness of the system and also affects the following error. A high proportional gain gives a stiff, responsive system but results in overshoot and oscillation which require damping.

If no value is supplied with the command, the previous setting is reported.

*CTG Configure Filter Time Constant

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CTGn	Milliseconds	n = 0 - 32,767	3	Immediate, Device Specific

Description

Sets the time constant of the filter used to reduce the effect of transients at the input of the drive amplifier and to smooth the response of the incremental velocity feedback loop.

*CTQ Enter Motor Torque

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aCTQn	Nm	N/A	3.6	Immediate, Device Specific

EXECUTION TIME

SEE ALSO HELP11, CJL

Description

This command has no effect on the behaviour of the RS232C Control Module. It allows entry of the available motor torque in calculations when using HELP11 as an aid to setting up the servo system parameters.

Example

Command	Description
2CTQ2.4	Defines the maximum motor torque as 2.4Nm

*CUR	Configure User Resolution	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CURn	User steps	n = 1 - 32,767	4000	Immediate, Universal

EXECUTION TIME **SEE ALSO** CMR

Description This command is used to define the number of steps per revolution required by the user. Previous D commands are reconfigured by the CUR command so that the move distance remains the same.

Example 1

Command	Description
1CUR2000	Set user resolution as 2000 steps/rev
D4000	Set distance to 4000 user steps (2 revs)
G	Go - move 2 revolutions

Example 2

Command	Description
1CUR4000	Set user resolution to 4000 steps/rev
1D4000	Set the move distance to 4000 steps (1 rev)
1CUR2000	Change the resolution to 2000 step/rev, but the distance programmed by the D command changes to 2000 so the move distance remains 1 revolution.

*CVG	Configure Velocity Gain	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>CVGn	N/A	n = 0 - 32,767	57	Immediate, Universal

Description Used to set the gain of the velocity feedback loop. The velocity feedback is a signal which increases with shaft speed. It acts in opposition to proportional feedback to stabilise the motion. This setting is generally used to damp vibrations in the servo response, allowing a higher proportional gain to be used. See 'Setting Up Servo Parameters' in Section H for further details

D Distance

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>D<s>n	Steps	-268,435,455 to +268,435,455	0	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO MN, MPA, MPI, A, V, G, MQ

Description

This command is used in MN and MQ to move the motor a number of steps (n) in the direction specified by <s>. The direction is assumed to be positive if no sign is given. The D command overrides a previous H command in terms of motor direction.

Example

Command	Description
MN	Normal mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
D-100000	A 4000 steps/rev motor will turn 25 revolutions in the negative direction
G	Go

DFX Display Flags Indexer

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDFX	N/A	N/A	N/A	Immediate, Device Specific, Never Saved

RESPONSE TO aDFX **IS** *bbbb_.....

Description

Reports all indexer status flags as 32 bits where the response is:

bbbb_.............bbb[CR]

The order of the bits is *31,30,29,28_....._3,2,1,0*

Response Bits:-

31-27 are all reserved for future use

26	1= Paused by PSY command
25	1= Paused waiting for SYNC input
24	1= Paused waiting for registration trigger
23	1= Run sequence on power up
22	1= Executing a sequence
21	1= Paused, waiting on in-position
20	1= Paused, waiting on distance trigger bits
19	1= Paused, waiting on trigger bits
18	1= Paused by U command flag, waiting on a C (continue)
17	1= Paused by PS command, waiting on C (continue)
16	1= Performing a wait
15	1= Homing 2nd leg, low speed move back to home limit
14	1= Homing 1st leg of home move, high speed to home limit
13	1= Go home move to encoder position
12	1= Home limit switch has been hit
11	1= +limit switch has been hit
10	1= +limit switch has been hit
9	1= Jogging is enabled, we are jogging
8	1= Skip next buffered command
7	1= Set if continuous move direction is negative
6	1= Set if current move direction is negative
5	1= Set if we want to change velocity
4	1= Set if in continuous mode,clear if in preset mode
3	1= Set if in absolute mode, clear if in incremental mode
2	1= Performing a variable speed move
1	1= Performing a preset move
0	1= Performing a continuous move

DIC

Display Indexer Counter

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDIC	User steps	-2,097,152 to +2,097,151	N/A	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO D

RESPONSE TO aDPa **IS** *n

Description

Requests a single display of the contents of the indexer counter as a single value in steps at the user resolution. This is the programmed D value.

Example

Command	Response
1DIC	*1000 (CR). The counter is programmed for a D value of 1000 user steps.

DPA Display Position Actual

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDPA	User steps	-32,767 to +32,768	N/A	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO OSC, PZ, SP

RESPONSE TO aDPA **IS** *n

Description Continuous display of actual position. The response is the position in user steps which should have resulted from the number of clock pulses sent to the drive from the RS232C Control Module since the drive was enabled or a PZ or SP command was issued, provided that the motor did not de-synchronise.

DPE Display Position Error

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDPE	User steps	-32,767 to +32,768	N/A	Immediate, Device Specific, Never Saved

Description Continuous display of position error. The response is the difference between the setpoint and the actual position in user steps. It is used by the position control algorithm to control motor current. The difference between command setpoint and actual position is also used to determine if the motor is within the deadband specified by the CEW command. The response is a single instantaneous value reported at 150ms intervals until the return key is pressed. See also OSC command.

DPS Display Position Setpoint

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
**aDPS	User steps	+268,435,455 to -268,435,455	N/A	Immediate, Device Specific, Never Saved

Description Continuous display of the commanded position. The response is repeatedly updated until the return key is pressed. It is the absolute number of pulses sent to the drive from the interface since the drive was enabled (or reset). See also OSC command.

DR Display Report

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDR<n>	N/A	1 or 2	N/A	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO PS,U

Description

This command reports the setup of the various parameters of the control module:

If n = 1, the general status of the control module parameters is displayed.

If n = 2, the status of the selectable facilities is displayed.

If n is omitted, both the control module parameters and the selectable facilities are shown on consecutive displays.

DS Display Signal

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDSn	N/A	N/A	N/A	Immediate, Device Specific, Never Saved

Description

Requests the continuous display of variable parameters. The PIC screen shown below is a representation of the control algorithm and any parameter in this diagram can be displayed. If n is omitted, a list of the signal displays is produced. See also OSC command.

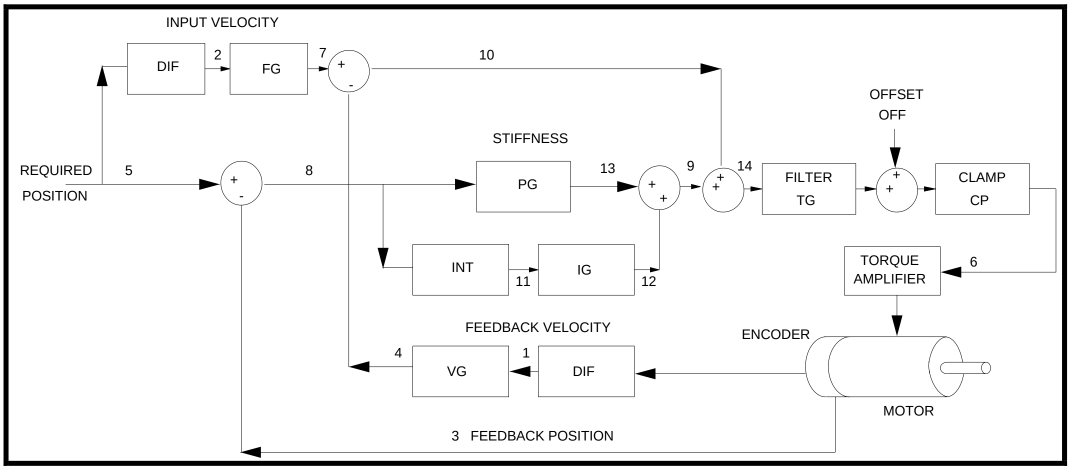


Figure 5-7. Servo Control Loop

The individual parameters are:-

DS1	Motor velocity	DS8	Position error
DS2	Input velocity	DS9	Demand less filtered velocity
DS3	Motor position	DS10	Velocity action difference
DS4	Feedback velocity	DS11	Integral of position error
DS5	Input position	DS12	Integral action
DS6	Filtered torque demand	DS13	Error times gain (= torque)
DS7	Input velocity action	DS14	Torque demand

Example

Command	Description
1DS8	Shows the position error for RS232C Control Module 1

The significance of the numbers displayed will vary with the signal point chosen. Input and motor velocities are expressed in encoder edges per 500µS period, with a maximum value of +/-255. All other parameters have values between +32767 and -32768, except the torque demand which covers the range +1023 to -1024 for full torque. A continuous display of the maximum value indicates saturation.

*DTA Set Dither Amplitude

Not used in System 7, PS7X, BLX

*DTF Set Dither Frequency

Not used in System 7, PS7X, BLX

DVA Display Velocity Actual

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDVA	Steps/500µs	±255	N/A	Immediate, Device Specific, Never Saved

Description This command returns a continuous display of the actual velocity. The number is reported in motor steps per 500µs and is repeatedly updated until the return key is pressed. This value is the shaft velocity being read from the encoder measured over a 500µs period. The DVA command has the same effect as the DS1 command. See also OSC command

DVS Display Velocity Setpoint

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aDVS	Steps/500µs	±255	N/A	Immediate, Device Specific, Never Saved

Description Requests a continuous display of the velocity setpoint. The displayed value is the velocity being sent to the velocity part of the servo loop by the servo algorithm. It is repeatedly updated until the return key is pressed. The DVS command has the same effect as the DS2 command. See also OSC command.

E	Enable Communications	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>E	N/A	N/A	N/A	Immediate, Universal

Description

This command allows the RS232C Control Module to accept commands over the serial communications interface. You can re-enable the the communications interface with this command if you had previously disabled the RS232C interface with the F command. If several units are using the same communications interface, use of the E and F commands can help to streamline programming.

Example

Command	Description
F	Disable communications all axes
1E	Enable communications axis 1
4E	Enable communications axis 4
A10	Set acceleration to 10 revs/sec^2
V5	Set velocity to 5 revs/sec
D5000	Set distance to 5000 steps
G	Only axes 1 and 4 move

EX	Set Communication Style	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aEXn	N/A	n = 0 or 1	User messages sent (EX1)	Immediate, Device Specific,

Description

Sets the style of communication between the RS232C Control Module and the terminal/computer.

n = 1 sets terminal mode i.e. user-friendly messages.
n = 0 sets computer mode i.e. no user-friendly messages are sent.

The ">" prompt is not returned when n = 0

F	Disable Communications	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>F	N/A	N/A	N/A	Immediate, Universal,

Description The F command disables command execution by devices connected to the RS232C Control Module. The disabled device will still echo back commands sent to it however. It is useful when you are programming multiple devices on a single interface. Devices that are not intended to respond to universal commands should be disabled using device specific F commands. This enables you to program other devices without specifying a device indentifier on every command.

Example

Command	Description
1F	Disable axis 1
3F	Disable axis 3
G	All axes except 1 & 3 will move

FDA	Acceleration, Linear Interpolation Moves	Not used in PS7X, BLX

FDV Velocity, Linear Interpolation Moves

Not used in PS7X, BLX

FOL Following Percent

VALID FROM
Version 3.2

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aFOLn	Percent	0.0 - 5000.0	100	Buffered, Universal,

EXECUTION TIME

SEE ALSO SIM, CCS

Description

During preset following indexing mode (SIM5) the move velocity in rps is set as a percentage of the following input velocity in rps by the FOL command value.

Example

Command	Description
CUR800	Pulse source resolution in pulses / rev
1CCS3	With clock and direction decode
CMR4000	The interface motor resolution
MN	Start linear interpolation move
MPI	D values are incremental
1SIM5	Set indexer mode to preset following
FOL50.0	Follow at 50% of the input encoder rate
A100	Acceleration fixed at 100 rps/s
D16000	Will move 4 revs
G	The motor will accelerate at 100 rps absolute to 1/2 the input encoder speed in rps

The CMR / CUR ratio matches the input/output pulse rate, allowing FOL to control the output velocity in rps.

G Go

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>G	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO S, MN, MC, MQ, A, V, D

Description Go - make a move using the previously entered parameters. It is not necessary to re-enter A, V and D.

Example

Command	Description
MN	Select normal mode
A10	Set acceleration to 10 revs/sec/sec
V10	Set velocity to 10 revs/sec
D100000	Set distance to 100,000 steps
G	Go

GA Go Home Acceleration

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>GAn	Revs/sec^2	n = 0.001 - 999,999	10	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO GH, SS, GHF, GHP

Description This command is used to set the acceleration rate to be used in performing the GH command. The value can be saved in non-volatile RAM. The default is changed by the user if the GA command is issued and then a save is performed (SV command).

GH Go Home

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>GH<s>n	Revs/sec	n = 0.0001 - 100	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO RG

Description Go Home + or -. This command causes the RS232C Control Module to rotate the motor in the direction and at the speed specified until its home limit input is activated.

The GH+2 command causes the controller to seek the home position at 2 revs/sec. The sign is optional (a "+" or positive is assumed if omitted). The controller will reverse direction if a limit is activated and it will cease the attempt to go home if the second limit is also activated.

Note that the GH command will reset the absolute position counter to zero. If you want to set the counter to another value, use the SP command after the GH is complete.

Example

Command	Description
GH-2	Motor will turn negative at 2 revolutions per second and look for Home limit input to go active. It will then stop, reverse and finally stop completely when the home switch is hit for the second time.

GHF Go Home Final

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>GHFn	Revs/sec	n = 0.01 - 100	0.1	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO GH, SS, GA

Description

This command is used to set the velocity for the final move in the go home sequence.

The value can be saved in battery backed up RAM.

GL Go, Linear Mode

Not used in PS7X, BLX

GLD Go Linear Define Not used in PS7X, BLX

GSY Go, Synchronous Mode **Not used in PS7X, BLX**

H Change Direction

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>H	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME **SEE ALSO** H+, H-, D

Description This command reverses direction of the next move.
The D command could subsequently be used to reset the direction.

Example

Command	Description
D8000	Set distance to 8000 steps
G	Go - move 8000 steps in the + direction
H	Reverse direction
G	Go - move 8000 steps in the - direction

^H Backspace

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
^H	N/A	N/A	N/A	Immediate, Never Saved

Description This produces a backspace during command input, deleting the last character. It will not prevent the execution of an immediate command.

The ^H command (^H indicates that the CONTROL or CTRL key is held down when the H key is pressed) backspaces one character provided a delimiter has not been sent. A new character may be entered at that position to replace the existing character. The effect of this command character is to cause the RS232C Control Module to back up one character in the command buffer regardless of what appears on the terminal. On some terminals pressing the BACKSPACE key will produce the same character.

Because the RS232C Control Module processes each command upon receipt of the delimiter, it is not possible to backspace once the delimiter is encountered. If you type a device address wrongly, you must re-type the whole command.

H+ & H- Set Direction

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>Hs	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO H, D

Description

Sets the direction of all moves according to s = "+" (positive) or "-" (negative). A subsequent D command will reset the direction for following moves.

Example

Command	Description
H-	All moves are made negative until otherwise specified in the command string

HELP Produce Help Screens

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>HELP<n>	N/A	n = 1 - 13	N/A	Immediate, Universal, Never Saved

Description

Displays the help screens. It provides a list of commands and a brief description of each command. The command on its own will produce the base menu followed by all other help screens. Individual screens can be accessed by typing HELP followed by the appropriate screen number. HELP 11 provides a worked example on servo set up.

This is available in terminal mode only (see EX command).

The CR key will take you out of the help menu.

With the exception of HELP 11, all interfaces return the same help information, so no device address is needed. The gain values returned in HELP 11 relate to the torque and inertia figures programmed by the CJL and CTQ commands.

HS — Display Hardware Configured Switches

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aHS	N/A	N/A	N/A	Immediate, Device Specific, Never Saved

Description

Display hardware configured switches i.e. the way in which the interface has been configured for its environment. The command displays a binary representation of this status. The response is *bbbb_bbbb where the order of the bits is:

ABCD_XXXX

Therefore bit 1 represents the status of HSA, bit 2 the status of HSB etc. Bits 5-8 are not used.

HSA: 1 = Resolver 0 = Other (not available)
HSB: 1 = Brush 0 = Brushless initialisation
HSC: 1 = Servo 0 = Open loop
HSD: 0 = No daughter board 1 = Output expansion board fitted 2 = Analogue feedback board fitted

Example

Command	Response
1HS	*0010_0000 This indicates that the interface is configured for brushless servo operation

HSB Configure Motor Type

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aHSBn	N/A	n = 0 , 1	0	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO HSC

Description

Sets the configuration for drive commutation;

n = 0: non self-commutating drive (System 7)
n = 1: self-commutating drive (BL & BR)

	HSB	HSC
Stepper	1	0
Hybrid Servo	0	1
Brushless Servo (System 7)	0	1
Brushless Servo (BL)	1	1
DC Brush Servo	1	1

HSB & HSC Combined Commands

HSC Configure Motor Type

VALID FROM
Version 2.0

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aHSCn	N/A	n = 1, 0	0	Immediate, Device Specific, Never Saved

EXECUTION TIME

SEE ALSO HSB

Description

Sets the configuration for a stepper motor;

n = 0: open loop stepper
n = 1: other

	HSB	HSC
Stepper	1	0
Hybrid Servo	0	1
Brushless Servo (System 7)	0	1
Brushless Servo (BL)	1	1
DC Brush Servo	1	1

HSB & HSC Combined Commands

HSD Daughter Board Outputs

NOT USED IN BLX

ID Immediate Distance

VALID FROM
Version 3.1

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>ID<s>n	Steps	-268,435,455 to +268,435,455	4000	Immediate, Universal,

SEE ALSO D

Description

During motion, the distance the motor is travelling can be altered to a new value set by the ID command. An attempt to change the motion direction or to set the target within the current stopping distance causes the unit to stop immediately at the currently programmed acceleration rate - if you are in terminal mode a warning message is returned. In incremental mode the sign is ignored and the distance is relative to the original start point of the current move, and in the same direction of motion. When stopped the ID command is simply an immediate version of the D command.

Example

Command	Description
MN	Normal preset movement
MPA	Absolute programming
PZ	Force current position 0
D-140000	Target is -140000
G	Start moving
ID-40000	Change target
IPR	Reports -40000 at end of move

IO Immediate Output

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>IObbb	N/A	b = 0, 1, X	N/A	Immediate, Universal, Never Saved

Description This command sets the output bits as specified in the pattern on an immediate basis. b = 0, 1 or X (X leaves the output unchanged). If the optional output daughter board is fitted (HSD=1), the syntax becomes <a>IObbbbbb.

Example

Command	Description
2IO10X	This command sets the outputs of device 2 : O1 to 1 and O2 to 0. It leaves O3 unchanged.

IS Input Status

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aIS	N/A	N/A	N/A	Immediate, Device Specific, Never Saved

Description Report Input status. The response is:

*bbbbbbb_bbbbbbb[CR]

The pattern of the response is, from left to right:

HOME, LMT-, LMT+, +JOG, -JOG, EIN (Drive energised), INDEX TRACK _ INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

This is an immediate command that will report the status of all the inputs no matter what the configuration.

Open circuit inputs are indicated by 0's.

Example

Command	Response
1IS	*0000000_1000000[CR]. This indicates that only Input 1 is active

IV Immediate Velocity

VALID FROM
Version 3.1

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
aIVn	Revs/sec	n = 0.0001 to 100.00	1	Immediate, Universal

SEE ALSO V

Description

This is an immediate version of the V command. During motion, the speed the motor is travelling can be altered to a new value set by the IV command. In response to the IV command the motor will accelerate or decelerate to the new velocity, but will still travel the programmed distance.

The IV command uses the RS232C communication link to alter the motor velocity. If the motor is already stopping at the end of a move a warning message is returned (terminal mode only).

Example

Command	Response
MN	Normal preset movement
MPA	Absolute programming
V5	
D-140000	Target is -140000
G	Start moving
IV20	Speed up during move
1PR	Shows end point is unchanged

JA Jog Acceleration

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>JAn	Revs/sec^2	n = 0.01 - 999,999	99	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO JV

Description

Sets the jog acceleration. The acceleration rate used in jog operations is set by this command.

Example

Command	Description
JV1.5	Set jog speed to 1.5 revs/sec
JA20	Set jog acceleration to 20 revs/sec/sec

JV Jog Velocity

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>JVn	Revs/sec	n = 0.0001 - 100	1	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO JA

Description Sets the jog velocity. The velocity used in jog operations is set using this command. The factory default can be changed by saving a new JV value to non-volatile RAM using the SV command.

K Kill

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>K	N/A	N/A	N/A	Immediate, Universal, Never Saved

EXECUTION TIME

SEE ALSO S, LS

Description Kill - This command stops RS232C Control Module commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a command sequence download in progress (XD command). The command buffer is also cleared.

KILL Kill

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>KILL	N/A	N/A	N/A	Immediate, Universal, Never Saved

EXECUTION TIME

SEE ALSO S, LS

Description This is an alternative expression for the K command. It stops RS232C Control Module commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a command sequence download in progress (XD command). The command buffer is also cleared.

L Loop

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES

<a>Ln	Times	n = 0 - 65545	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME **SEE ALSO** Y, N

Description

When combined with the N command, the L command will cause all of the commands between L and N to be repeated the number of times indicated by n. If L is entered with no number following it or if n = 0, the commands will be repeated continuously.

The END-OF-LOOP command (N) can be used to indicate that the RS232C Control Module should proceed with further commands after the designated numbers of loops have been executed, or in combination with the "Y" command, to indicate where execution is to stop. The "U" command may be used to temporarily halt loop execution, the C command will then cause the loop to resume execution.

There should be a balanced number of loops and loop terminators inside a sequence. Starting a loop in one sequence and terminating it in another sequence is not allowed. Nested loops require complete closure before execution will begin.

Example

Command	Description
PS	Pause
A10	Set acceleration to 10 rps^2
V10	Set velocity to 10 revs/sec
D1000	Set distance to 1,000 steps
L5	Loop 5 times
G	Go
N	End of loop
C	Continue

Example (nested loop)

Command	Description
L10	Loop 10 times
D4000	Set distance to 4,000 steps
G	Go
L5	Loop 5 times
D10	Set distance to 10 steps
G	Go
N	End of the 5 x 10 step
N	End of the overall loop

The commands L5 D10 G N form a nested loop.

LA Limit Deceleration

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>LAn	Revs/sec^2	n = 0.001 - 999,999	900	Buffered, Universal, Savable in Sequence

Description Define or report the deceleration rate after limit switch operation. This allows a rapid deceleration in response to a limit switch regardless of the rate programmed by the "A" command.

LD Limit Disable

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>LDn	N/A	n = 0 - 3	0	Buffered, Universal, Savable in Sequence

Description This command is used to disable the limit switch functions.

n = 0: Enable all limits (default)
n = 1: Disable limit +
n = 2: Disable - limit
n = 3: Disable + & - limit

Example

Command	Description
1LD3	The detection of both + and - limit switch operation is disabled for axis 1

LS Limit Switch Fast Deceleration Stop

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>LS	N/A	N/A	N/A	Immediate, Universal, Never Saved

Description Decelerate and stop at the limit switch deceleration rate set by LA (usually set fast compared to normal acceleration rate).

MAS Mode Asynchronous

Not used in PS7X, BLX

MC Mode Continuous

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>MC	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO MN, A, V

Description

MC sets the move mode to continuous. It causes subsequent moves to ignore any distance parameter and move continuously at the programmed velocity until stopped by an S, LS or K command.

Example

Command	Description
MC	Set continuous mode
H-	Set direction to negative
A10	Set acceleration to 10 rev/sec/sec
V10	Set velocity to 10 rev/sec
G	Go - run continuously at 10 rev/sec

MN Mode Normal

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES

<a>MN	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO MQ, MC, MPA, MPI, A, V, D

Description

MN sets the move mode to normal preset distance. It causes last issued distance parameter to be used as the distance for the current move. The MN command will change the mode of operation from continuous back to preset.

Example

Command	Description
MN	Set mode to normal
A10	Set acceleration to 10 rev/sec/sec
V10	Set velocity to 10 rev/sec
D8000	Set distance to 8000 steps
G	Go - run for 8000 steps

MPA

Mode Position Absolute

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>MPA	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO MN, MQ, MC, MPI, D, PZ

Description

Sets the position mode to absolute. In this mode all move distances are referenced to absolute zero. Units are scaled by the CUR command. You must be in the preset mode (MN or MQ command) before the MPA command will take effect.

Example

Command	Description
MN	Set normal mode
MPA	Set absolute mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
D10000	Set new absolute position to 10,000 steps
G	Go - move to position 10,000
D2000	Set new absolute position to 2,000 steps
G	Go - move 8,000 steps negative to position 2,000

Note that in the absolute mode, giving two G (go) commands in succession will cause the motor to move only once as the motor will have achieved the desired absolute position at the end of the first move.

MPI — Mode Position Incremental

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>MPI	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO MPA, D

Description

Sets the position mode to incremental. In this mode all move distances are referenced to the starting position of each move. You must be in the Preset mode (MN or MQ command) before the MPI command will take effect. MPI is the power up default on the MN command.

Example

Command	Description
MN	Set normal mode
MPI	Set incremental mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
D8000	Set distance to 8000 steps
G	Go - move 8000 steps positive
D-4000	Set distance to 4000 steps negative
G	Go - move to position +4000 steps from the starting position

MQ Speed Change Mode

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>MQ	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description This command allows velocity to be changed using buffered commands during a preset move.

Example

Command	Description
MQ	Set speed change mode
A50	Set acceleration to 50 revs/sec^2
V20	Set speed to 20 revs/sec
D8000	Set total distance to 8000 steps
G	Go - start move
TRD3000	Change speed at distance of 3000 steps
V10	New speed set to 10 revs/sec
TRE_X1	Change speed when input goes to 1
V4	New speed set to 4 revs/sec

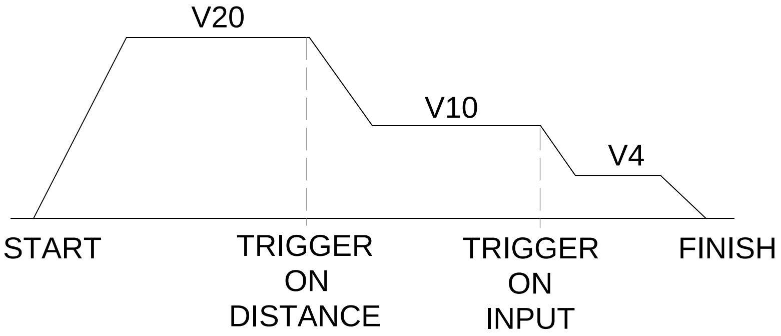


Figure 5-8. Complex Velocity Profile Using MQ Mode

The motor accelerates at 50 revs/sec^2 to a velocity of 20 revs/sec. At a distance of 3000 steps the velocity changes to 10 revs/sec. When Input 2 is energised the velocity changes to 4 revs/sec until the programmed distance is reached at 8000 steps.

MSY Mode Synchronous Not used in PS7X, BLX

N End Loop

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>N	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME **SEE ALSO** L, Y

Description Marks the end of a loop. When used in conjunction with the L command, it causes the buffered commands between the L and the N to be executed as many times as the number following L.

Example

Command	Description
L5	Set to loop 5 times
A10	Set acceleration to 10 revs/sec^2
V10	Set speed to 10 revs/sec
D8000	Set distance to 8000 steps
G	Go - move 8000 steps (repeated 5 times)
N	End of loop

O Programmable Output

VALID FROM

Version 2.1

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>Obbb<bbb>	N/A	b = 0, 1, X	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME **SEE ALSO** SS, OS

Description

Turns outputs on and off. When HSD0 is set, no daughter board is fitted and the three outputs are set by the first three digits 'bbb' where the first 'b' sets output 1, the second 'b' output 2 and the third 'b' output 3. When HSD1 is set, the three additional outputs on the daughter board are set by the last three digits <bbb>. The fourth 'b' sets output 4, the fifth 'b' sets output 5 and the sixth 'b' sets output 6.

b = 1 sets the output to sink current, i.e. output on
b = 0 turns the output current off
b = X leaves output unchanged

Examples

Command	Description
1HSD1	Optional motherboard fitted
1O111111	Outputs 1-6 on device 1 to sink current from an externally supplied source
O000XXX	Current stops flowing in outputs 1-3
OX1XX0X	Leaves outputs one, three, four and six unchanged while turning output 2 on and output 5 off

Using the configuration switches it is possible to dedicate output 1 as the "Composite fault" line and/or output 2 as an "in position" line. The O command will not control these outputs if they have been configured in this way.

OFF De-Energise Drive

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>OFF	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME **SEE ALSO** ST, ON

Description

De-energises the drive immediately. This command may be used to shut down the drive quickly in an emergency. Issuing an SV command after the OFF command will cause the drive to power up in the de-energised state.

ON Energise Drive

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>ON	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME **SEE ALSO** ST, OFF

Description

Starts up the drive immediately. This command may be used to re-energise the drive after a shutdown. Issuing an SV command after the ON command will cause the drive to power up in the energised state.

OS Other Switches

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOS	N/A	N/A	N/A	Buffered, Device Specific

Description

The command OS will report the state of the OS switches with a response *bbbb_bbbb_bbbb_bbbb where the order of the bits is:

ABCD_EFGH_IJK(X)_M(X)O(X)

(X) = not used. Therefore bit 1 indicates the status of OSA, bit 2 the status of OSB etc.

OSA Home at Index Pulse

VALID FROM
Version 3.1

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSAn	N/A	n = 0 or 1	0	Buffered, Device Specific

Description

This command enables homing to an index pulse for servo systems fitted with an incremental encoder with an index track. On completion of the homing routine, the motor will stop on the first index pulse after the edge of the home switch is detected. This results in a highly-repeatable home position which is not affected by small variations in the operation of the home switch.

n = 1: Home to index pulse within home switch range
n = 0: Home at home switch edge

OSB Integral Action Selection

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSBn	N/A	n = 0 or 1	0	Buffered, Device Specific

Description This command selects whether integral action will occur all the time or only whilst in position

n = 1: Integral action will only occur whilst 'in-position'
n = 0: Integral action occurs all the time

OSC Monitor Command Reporting

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSCn	N/A	n = 0 or 1	0	Buffered, Device Specific

Description The OSC command selects continuous reporting of monitor commands (DS, DPA etc.) or reporting of one value only.

n = 1: Signal monitor commands report only one value
n = 0: Monitor commands report continuously

OSD Input 7 Sequence Select

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSDn	N/A	n = 0 or 1	0	Buffered, Device Specific

Description This command configures Input 7 as a sequence select line enabling up to 16 sequence selections.

n = 1: Input 7 defined as sequence select
n = 0: Input 7 not sequence select

OSE Jog Enable

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSEn	N/A	n = 0 or 1	0	Buffered, Device Specific

Description The OSE command is used to enable or disable the jog function.

n = 1: Jog enabled
n = 0: Jog disabled

OSF	**Initialisation On Limit**	**NOT USED IN BLX**

OSG	**Dither Enable**	**Not used in System 7, PS7X, BLX**

OSH	**Select Triangular/ Square Dither**	**Not used in System 7, PS7X, BLX**

OSI Initialisation Movement

NOT USED IN BLX

OSJ Select 24/16 Bit Gearbox

VALID FROM
Version 2.3

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSJn	N/A	n = 0 or 1	0	Buffered, Device Specific

EXECUTION TIME:

SEE ALSO: RAT

Description

The OSJ command selects 24 or 16 bit gearbox used with the RAT command for external pulse stream following.

n = 1: 24 bit gearbox operation -
RAT command range = +/-16777216
n = 0: 16 bit gearbox operation
RAT command range = +/-65535

The 24 bit range has been included in this software version for higher resolution, but the 16 bit range has been retained for compatibility with earlier software verions.

OSK Integrity Check of Optical Encoder

VALID FROM
Version 3.1

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSKn	N/A	n = 0 or 1	0	Buffered, Device Specific

SEE ALSO CIX

Description

This command enables an encoder integrity check for servo systems fitted with an incremental encoder.

n = 1 The encoder count will be checked at the index track
n = 0 The encoder count will not be checked

If a count error occurs at the index track with the OSK option enabled, the interface will de-energise the drive, and an RSE command will produce the message:

#90 Shutdown by <X> counts encoder error at index track

Bit 9 in the de-energise data will be set and the LED flash code will be set to 1.

The RE command will show the drive to be de-energised.
This option requires the encoder to have an index pulse that is one pulse wide (when gated with the A and B encoder channels). The option also requires only one index pulse every time the actuator moves the number of pulses defined in the CMR command, i.e. generally, only one pulse per rev.

OSM Integral Action Sensitivity

VALID FROM Version 3.1

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSMn	N/A	n = 0 or 1	0	Buffered, Device Specific
				SEE ALSO CIG

Description

This command enables a fast, wide range version of the integral action capability.

n = 0 Accumulate at 20ms sampling to standard range
n = 1 Accumulate at 2ms sampling to a 256 times larger range

The default setting OSM0 is suitable for the majority of situations in which integral action would be needed, i.e. systems with a significant frictional load. Lightly-loaded systems using smaller motors may benefit from the OSM1 setting.

OSO Suppress Units

VALID FROM
Version 3.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aOSOn	N/A	n = 0 or 1	0	Buffered, Device Specific

SEE ALSO CMR, CUR, CIX

Description This command suppresses any reference to units such as RPS in the message prompts. This avoids confusion on linear and rotary systems where the velocity is not expressed in revolutions at a motor shaft.

n = 0 rps and rps/s units appear in messages
n = 1 rps and rps/s units do not appear in messages

P Position

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aP	User steps	±2,097,151	N/A	Immediate, Device Specific

Description Displays the position relative to the start of the last index. The position counter is cleared and restarted by next G command. This is a single response in user steps, and can be used during a move.

PIC Picture

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aPIC	N/A	N/A	N/A	Immediate, Universal

Description Displays a picture of the servo loop with the signal monitor numbers. A device address is not necessary since any control module will return the same display.

PR Position Report

VALID FROM
Version 1.5

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aPR	User steps	-268,435,455 to +268,435,455	N/A	Buffered, Device Specific

EXECUTION TIME

SEE ALSO MPA, MPI,

D, PZ, SP

Description

Requests the current absolute position. The report is a number preceded by a sign and followed by a carriage return *(s)n[CR]. The number represents the cumulative position in user steps (n) with respect to the position at power up or the last point at which an SP or PZ command was issued resetting absolute zero. The sign indicates which side of the absolute zero position the motor is on.

Example

Command	Response
1PR	*-25600[CR] (Motor is at absolute position -25600)

PS Pause

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>PS	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO C, SSB

Description

This command pauses the execution of the current command string or sequence. Execution will then be resumed after a C command (continue) is received. The command is used to allow the entering of a complete command string before the commands are executed.

Example

Command	Description
PS	Pause until C command is received
D5000	Set distance to 5000 steps
G	Go - start move
H-	Set direction to negative
G	Go - start negative direction move
C	Release pause - both moves now carried out

PSY Pause for Synchronisation

Not used in PS7X, BLX

PZ Position Zero

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>PZ	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO MN, MPA, MPI, PR, D, SP

Description Sets the current position to be absolute zero.

Example

Command	Description
MPA	Set absolute mode
D2500	Set new position to 2500 steps
G	Go - move to absolute position +2500
1DPA	Report the new position (+2500 returned)
1PZ	Set the new position as absolute zero
1DPA	Report the position (zero is returned)

Where a servo is used, due to offsets in the loop the demanded position and the feedback position may not be exactly equal. This command sets the physical or feedback position zero, so the demand position indicated by DPS may show a non-zero value

QS Transmit An Identifier

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>QS	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description

Responds with the identity of the echoing device. The command can be used to confirm which device is echoing after a series of universal commands.

The response consists of an asterisk, a two digit device address, an exclamation mark and a carriage return.

Example

Command	Description
MN	Select normal mode
D4000	Set distance to 4000 steps
A100	Set acceleration to 100 revs/sec^2
V50	Set velocity to 50 revs/sec
G	Go
QS	Transmit identifier

If this is executed as a sequence by axis 1, the response *01! will be received over the RS232C line when the sequence is complete.

R Report Serial Interface Status

VALID FROM

Version 2.6

XSR,XSS,DFS,DFX

Description

Requests the status of the RS232C Control Module. The response is *<char>[CR] :

*R [CR] - ready for a command with no errors.
*S [CR] - ready for a command with function errors.
*T [CR] - ready for a command with previous comms error.
*U [CR] - ready for a command with function error and previous comms error.
*B [CR] - busy performing a move with no errors.
*C [CR] - busy performing a move with function error.
*D [CR] - busy performing a move with previous comms error.
*E [CR] - busy performing a move with function error and previous
comms error.

In terminal mode a message apperars as well. The ready response means a buffered command will be executed immediately on receipt.

The RS232 communications is considered to have an error if:

A framing error occured (start / stop bits incorrect)
An overrun error occured (new character received whilst previous one unread)

Reading the status clears an outstanding communications error.

Example

Command	Response
1R	*R[CR] RS232C Control Module ready to accept a command. If it's a buffered command, it will be executed immediately.

RA — Report A - Limit Status Request

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRA	N/A	N/A	N/A	Immediate, Device Specific

Description

Requests the status of the limits. The response, in accordance with the following table, is the current and last move limit status represented by a single character from @[CR] to O[CR].

Example

Command	Response
1RA	*I [CR] The current move is limited by the negative limit and the last move was terminated by the positive limit.

Response Character	**Last Move Terminated By**		**Current Move Limited By**	
	Positive Limit	**Negative Limit**	**Positive Limit**	**Negative Limit**
*@	NO	NO	NO	NO
*A	YES	NO	NO	NO
*B	NO	YES	NO	NO
*C	YES	YES	NO	NO
*D	NO	NO	YES	NO
*E	YES	NO	YES	NO
*F	NO	YES	YES	NO
*G	YES	YES	YES	NO
*H	NO	NO	NO	YES
*I	YES	NO	NO	YES
*J	NO	YES	NO	YES
*K	YES	YES	NO	YES
*L	NO	NO	YES	YES
*M	YES	NO	YES	YES
*N	NO	YES	YES	YES
*O	YES	YES	YES	YES

The RA command is useful when the motor will not move in either or both directions. The report back will indicate whether or not the last move was terminated by a limit switch activation and if the current move is disabled by an active limit.

RAT Set Rate Multiplier Value

VALID FROM

Version 2.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRATn	Rate multiplier value	See description	16384	Buffered, Device Specific

EXECUTION TIME

SEE ALSO CCS, SIM, OSJ

Description

This command is used in servo mode (HSC1) to configure a rate multiplier on the second encoder input when using the following mode of operation. The range of the command is determined by the setting of the OSJ command as follows:

OSJ = 0: RAT range -65536 to +65536
OSJ = 1: RAT range -16777216 to +16777216

The effect is that of a gearbox on the second encoder input that obeys the following formula:

Output rate to the motor (OSJ = 0) = $\frac{n}{65536}$ x input rate from the second encoder.

Output rate to the motor (OSJ = 1) = $\frac{n}{16777216}$ x input rate from the second encoder.

For a limited number of pulses:

$$\text{Pulses to motor (OSJ = 0)} = \text{INT}\ \frac{\text{n x pulses in}}{65536}$$

$$\text{Pulses to motor (OSJ = 1)} = \text{INT}\ \frac{\text{n x pulses in}}{16777216}$$

where INT means 'the integer part of'. The value is rounded down towards zero whether the value is positive or negative.

Negative values of RAT allow the direction of following to be reversed.

NOTE: The number of second encoder input pulses is also affected by the CCS command.

Example

A 4000 step/rev motor is required to follow a 1000 line/rev encoder at half the encoder speed in the opposite direction. 1 revolution of the encoder produces 1000 x 4 (CCS0 set) = 4000 pulses; -2000 pulses are required to produce a half revolution of the motor shaft in the opposite direction.

$$\text{So } n = \frac{65536 \text{ x } (-2000)}{4000} = -32768$$

Command	Description
1CCS0	Select x 4 decode
1OSJ0	Set OSJ for 65536 RAT range
1SIM1	Select encoder following mode
1RAT-32768	If the tracked encoder turns 1 revolution CW, the number of pulses sent to the motor is:

$$\frac{-32768 \text{ x } 4000}{65536} = -2000$$

The motor of axis 1 therefore turns a half revolution CCW.

RB Report B - Misc. Status Request

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRB	N/A	N/A	N/A	Immediate, Device Specific

EXECUTION TIME

SEE ALSO TR, PS, L, ST

Description

Requests the status of a loop, a pause, a shutdown or an input . The response is @ [CR] to O [CR] according to the following table.

Response Character	Loop Active	Pause Active	Shutdown Active	Input Active
*@	NO	NO	NO	NO
*A	YES	NO	NO	NO
*B	NO	YES	NO	NO
*C	YES	YES	NO	NO
*D	NO	NO	YES	NO
*E	YES	NO	YES	NO
*F	NO	YES	YES	NO
*G	YES	YES	YES	NO
*H	NO	NO	NO	YES
*I	YES	NO	NO	YES
*J	NO	YES	NO	YES
*K	YES	YES	NO	YES
*L	NO	NO	YES	YES
*M	YES	NO	YES	YES
*N	NO	YES	YES	YES
*O	YES	YES	YES	YES

Loop active means that a loop is in progress.

Pause active means that buffered commands are not being executed and the RS232C Control Module is waiting for a C command.

Shutdown active means that the motor is shutdown.

Input active means that at least one input is active.

Example

Command	Response
1RB	*J[CR] There is no loop active, pause is active, the motor is not shutdown and there is an input active.

RE Drive Status Request

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRE	N/A	N/A	N/A	Immediate, Device Specific

Description

Requests the energised/de-energised status of the drive. The response is *@ if the drive is energised or *B if the drive is de-energised.

# RFS	## Return to Factory Settings	**VALID FROM** Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>RFSn	N/A	0-5	N/A	Immediate, Universal

EXECUTION TIME **SEE ALSO** Z, RIFS

Description

Return to standard default settings appropriate to the product in use. This command will configure the motor type, set normal resolution values and define appropriate servo gains for the product you specify. The Z command will restore the settings to their values prior to entry of the RFS command if they have not been saved using SV.

Available product numbers are as follows:

1. System 7 34-size brushless or hybrid servo
2. System 7 microstepper
3. System 7 23-size brushless servo
4. BL Series 16 or 23-size brushless servo
5. BL Series 34-size brushless servo

Sending the command RFS with no address or value will result in a list of the product numbers being returned. Preceding RFS by the axis address will return the existing product number for that axis.

Example

Command	Description
1RFS3	Set axis 1 as a System 7 23-size brushless servo
2RFS	Return the current product number for axis 2

The command RFS0 will give a generalised setup; you will then need to send the appropriate HSB and HSC values before the drive can be energised. (This setup is equivalent to that obtained by sending the RFS command alone using Issue 2 software). You can also set appropriate indexer default values by sending RIFS after the correct RFS command.

RG	Report Go Home Status	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRG	N/A	N/A	N/A	Immediate, Device Specific

EXECUTION TIME

SEE ALSO GH, GA, GHP, GHF

Description

Requests the status of the last Go Home attempt. The response is @ [CR] or A [CR] , indicating the success or failure of last go home attempt as follows:

Response	Go Home Successful
*@	NO
*A	YES

RIFS	Return Indexer to Factory Settings	VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>RIFS	N/A	N/A	N/A	Immediate Universal

EXECUTION TIME

SEE ALSO Z, RFS

Description

This command sets indexer default values appropriate to the product number entered using the RFS command.

Example

Command	Description
1RFS2	Set axis 1 as a System 7 microstepper
1RIFS	Set indexer default values suitable for System 7 microstepper

RPO Report Power-On Time

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRPO	N/A	N/A	N/A	Immediate, Device Specific

Description

Requests the RS232C Control Module power on time (hours) in decimal, for example 33.7 hours. A continuous record of usage is maintained by the interface.

RS Report Sequence Status

VALID FROM

Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRS	N/A	N/A	N/A	Immediate, Device Specific

EXECUTION TIME

SEE ALSO XR, XP

Description

Request the status of the last sequence execution. The response will be @ [CR] to D [CR]) according to the following table:-

Response Character	Sequence Started	Sequence Ended
*@	NO	NO
*A	YES	NO
*B	NO	YES
*C	YES	YES
*D	NO	YES

Whenever a sequence is started, the sequence start bit is set and the sequence end bit is cleared (this only occurs if the sequence is valid and is actually run). Whenever a sequence is ended, the start bit is cleared and the end bit is set. Any abrupt move termination (e.g. limit activation), or a K or S command clears both bits.

*D is reported when there is an unbalanced number of loops and loop terminators inside a sequence. Starting a loop in one sequence and terminating it in another sequence is not allowed. Nested loops require complete closure before execution will begin.

Example

Command	Response
1RS	*A Sequence in progress

RSE Report Servo Errors

VALID FROM
Version 3.0

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRSE	N/A	N/A	N/A	Immediate, Device Specific

Description Returns all servo error flags as a set of messages to the terminal and as a bit pattern. The messages correspond with the drive Fault LED indications. The bit pattern sent is:-
*xxxx_xxxx_000x_xxxx_xxxx where the rightmost "x" bit is bit 0.
Possible messages include:

Message	Bit No.	Flash Code	Fault
#20	0	1	De-energised by ST1 or OFF command
#22	2	3	EPROM changed with different memory map
#23	3	4	Excessive position error
#24	4	5	Memory failure - failed checksum
#25	5	1	Undefined drive
#26	6	7	Prolonged max torque demand
#27	7	8	Output to drive is zero torque
#28	8	8	Emergency stop input seen
#29	1	1	Shutdown by limit switch during initialisation
#90	9	1	Shutdown by <n> counts encoder error at index track
#80	12	2	Drive disabled by composite drive fault
#81	13	2	Impending power loss
#82	14	2	Drive disabled by motor over temperature
#83	15	2	Drive disabled by drive over temperature
#84	16	2	Drive disabled by HT over voltage
#85	17	2	Drive disabled by drive over current
#86	18	2	Drive disabled by drive specific fault (see manual)
#87	19	2	Drive disabled by drive logic supply fault

RV Revision Level

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aRV	N/A	N/A	N/A	Immediate, Device Specific

Description Request software version number. An example response is:

* ISSUE: 1.1 DATE: 27-Jun-89 17:29 [CR]

This is the software revision and the date and time of the revision.

S Stop

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>S	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME **SEE ALSO** K

Description Stop. If SSH 0 (Don't save command buffer on stop) is set, the command buffer is cleared (at the end of a move if one is in progress).

A command sequence download is aborted (XD command) and a time delay is terminated.

The motor is decelerated to stop using the current acceleration value ('A') but the drive stays energised.

If SSH1 (Save command buffer on stop) is set, the command buffer is not cleared and only the move that is in progress is terminated.

Example

Command	Description
MC	Set continuous mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
G	Go - run continuously
S	The motor will decelerate to a stop at a rate of 10 revs/sec/sec as soon as S is entered

SAVE Save

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>SAVE	N/A	N/A	N/A	Immediate, Device Specific

Description This is an alternative expression for the SV command. It causes the current servo, set up and index parameters to be saved in non-volatile RAM.

Note that in order to successfully save data, link 1 must be in place. Remove the link to write protect the RAM.

SB Stop Buffered

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>SB	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description Stop. The motor is decelerated to a stop using the current acceleration value (A). The command can be used in the MC or MQ modes.

Example

Command	Description
2A10	Set acceleration for axis 2 to 10 revs/sec^2
2SB	Axis 2 is decelerated to a stop at the rate of 10 revs/sec^2

SIM Set Indexer/Following Mode

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSIMn	n = Mode number	0 - 5	0	Buffered, Device Specific

EXECUTION TIME

SEE ALSO CCS, RAT, FOL

Description This command is used to configure the serial control module for following a second encoder. The mode of operation is selected by n as follows:-

n = 0 Normal control module operation. In this mode any pulses arriving at the second encoder input are ignored.

n = 1 Encoder following operation. In this mode the position of the motor follows the pulse stream on the second encoder input and the control module motion commands are inactive.

n = 2 Super position operation. In this mode both the incoming pulses from the second encoder and control module motion commands are active. The position of the motor is the sum of both motions.

n = 3 Positive software scaled encoder following operation.

n = 4 Negative software scaled encoder following operation.

n = 5 Preset following index mode

Note: SIM1 and SIM2 can only be used in stepper mode with a 1:1 following ratio. The RAT command is not valid for stepper mode.

SIM3 and SIM4 provide a software scaled following capability which allows the input pulse stream to be multiplied or divided with a sign reversal if required. The scaling ratio is set by the CMR and CUR commands which define the motor and user resolutions. More information on encoder following can be found in Section H.

SIM5 selects indexing at a speed determined by the external input. The percentage following factor is set by the FOL command. More information on preset following index mode can be found in Section H.

NOTE: The stop command does not affect the second encoder input. Energise/de-energise commands are effective.

Example

Command	Description
1SIM0	Set normal control mode
GH2	Go home
1SIM1	Set control by second encoder

SKE Skip On 'Equals'

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>SKEn	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description

Skip the following command on an input pattern equal to n. This command is used in sequences to control execution flow. The format for n, the input pattern, is specified by:

bbbbbbb_bbbbbbb

The pattern is from left to right :

HOME, LMT-, LMT+, +JOG, -JOG, EIN (DRIVE ENERGISED), INDEX TRACK_INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care). You can omit trailing X specifiers within each group of 7.

For example the following two pattern are equivalent:

001XXXX_XXX11XX and

001_XXX11

The next two are also equivalent:

XXXXXXX_X1XXXXX and

_X1

Example

Command	Description
1SKEXX1_	The next command will be skipped if the positive limit input is active

SKN Skip On 'Not Equal'

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>SKNn	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description

Skip the next command on input pattern not equal to n. The command is used in sequences to control execution flow. The format for n, the input pattern, is specified by:

bbbbbbb_bbbbbbb

This pattern represents from left to right :

HOME, LMT-, LMT+, +JOG, -JOG, EIN (DRIVE ENERGISED), INDEX TRACK_INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care). You can omit trailing X specifiers within each group of 7 specifiers.

For example the following two specifiers are equivalent:

001XXXX_XXX11XX and

001_XXX11

The next two are also equivalent:

XXXXXXX_X1XXXXX and

_X1

Example

Command	Description
1SKN1_	The next command will be skipped if the HOME input is not active

SP Set Position

VALID FROM
Version 2.2

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSP<s>n	Motor steps	-268,435,455 to +268,435,455	0	Buffered, Device Specific, Savable in Sequence

EXECUTION TIME

SEE ALSO MPA, MPI, PR, D, PZ

Description

Adds the value of n to the absolute position counter. The counter will be set to n if a PZ command is given prior to the SP command. This command is useful for setting the absolute zero point to some location other than that of the physical hardware home. If you have a cut-off saw for example, you may not be able to mount the home switch at the cut point. However, by mounting the home a known distance away and resetting the reference point with the SP command, the system may be made to function as if the home switch were at the cut point.

Note that the units of the SP command are scaled by the CMR command. You can get a position report in the same units by using the DPA command.

Example

Command	Description
GH2	Go home
PZ	Set position to zero
SP4000	The absolute position will be set to 4000 so that the absolute zero will be one motor revolution away from the switch location.

SS Set Switches

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSS	N/A	N/A	N/A	Buffered, Device Specific, Savable in Sequence

Description

The command SS will report the state of the SS switches with a response *bbbb_bbbb_bbbb_bbbb where the order of the bits is:

ABCD_EFGH_I(X)(X)(X)_(X)(X)(X)(X)

(X) = not used. Therefore bit 1 indicates the status of SSA bit 2 the status of SSB etc.

SSA RS232C Echo Control

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSSAn	N/A	n = 0 or 1	0	Buffered, Savable in Sequence

Description

This command turns the RS232C echo on and off

SSA0 = Echo on
SSA1 = Echo off

In the Echo On mode characters that are received by the RS232C Control Module are echoed automatically. In the Echo Off mode, characters are not echoed. This command is useful if your computer cannot handle echoes.

Example

Command	Description
SSA1	Turns the echo off (Characters sent to the RS232C Control Module are not echoed back to the host).

SSB Set Input 6 as a Clear Pause Input

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSSBb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

EXECUTION TIME

SEE ALSO U, PS

Description

This command can be used to allow input 6 to clear a pause. The input will function in the same way as issuing a 'C' command.

b =0: Normal
b =1: Input 6 clears a pause

Note that input 6 should be low (i.e. off) when the PS command is issued, otherwise the command will be ignored and there will be no pause.

Example

Command	Description
1SSB1	Set Input 6 as clear pause for axis 1
1PS	This causes axis 1 to pause. Input 6 can be used to clear the pause.

SSC — Set Output 2 as an In-Position Output

VALID FROM Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
aSSCb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

EXECUTION TIME

SEE ALSO CEW, CIT

Description

b = 0: Normal
b = 1: Output 2 is configured as an 'in position' output

With SSC set to 1, output 2 will turn on when the motor is within the error window for the specified time defined by CEW & CIT commands.

SSD — Set Output 1 as Composite Fault Signal

VALID FROM Version 2.2

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
aSSDb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

Description

b = 0: Normal
b = 1: Output 1 is configured as a composite fault indicator

When functioning as a composite fault indicator, the output is normally on and any fault that de-energises the drive generated by software, watchdog timeout or an incoming emergency stop signal will cause the output to turn off.

This does not include the effect of ST1 or OFF, which deliberately de-energises the drive. If ST1 or OFF have been sent, the composite fault output will not then indicate the occurrence of an ESTOP input or of any other fault conditions such as excessive position error.

SSE — Set Input 5 as a Controlled Stop Input

VALID FROM Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
aSSEb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

EXECUTION TIME

SEE ALSO LA

Description

b = 0: Normal
b = 1: Input 5 is configured as a controlled stop input

If the command SSE1 is given, a controlled stop at the limit deceleration rate (LA) will occur on an Input 5 transition to high level.

Example

Command	Description
2LA10	Set limit deceleration to 10 revs/sec^2
2A100	Set acceleration to 100 revs/sec^2
2V5	Set velocity to 5 revs/sec
2SSE1	Set input 5 as controlled stop
2G	With axis 2 running at the programmed 5 revs/sec, Input 5 going to high level causes a stop at the programmed limit deceleration, 10 rev/sec^2.

SSF

Set Input 4 as Sequence Strobe

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSSFb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

Description

b = 0: Normal
b = 1: Input 4 defined as a sequence strobe. When active, Input 4 causes execution of the sequence number found at the value on Inputs 1-3, or inputs 1-3 and input 7 if OSD1 is selected.

See 'Sequence Programming'.

SSG

Save Command Buffer On Limit

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSSGb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

Description

b = 0: Normal
b = 1: Command buffer saved on limit

Normally, when a limit is hit, the current command buffer or sequence is scrapped thus preventing further execution. Setting this bit to 1 prevents this activity.

SSH Save Command Buffer On Stop

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSSHb	N/A	n = 0 or 1	0	Buffered, Device Specific, Savable in Sequence

Description

b = 0: Normal
b = 1: Save command buffer on stop ('S' command).

This command operates in a similar way to SSG, but relates to the stop command.

SSI Set All Inputs As Sequence Select

VALID FROM
Version 3.1

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aSSIn	N/A	n = 0 or 1	0	Buffered, Device Specific,

Description

This command configures inputs 5,6, and 7 as sequence select lines enabling up to 63 sequence selections.

n = 0 Inputs 5,6,7 not configured as sequence select lines
n = 1 Inputs 5,6,7 are configured as sequence select lines

See Standalone operation at the beginning of this Section.

ST Shutdown

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>STn	N/A	n = 1 or 0	N/A	Buffered, Universal, Savable in Sequence

Description

Motor shutdown.

0: The drive is enabled and any brushless or stepper motor initialised if required.
1: The drive is disabled.

Any move commands given during motor shutdown will not be executed. ST0 ramps the currents up to normal level (at the position read upon execution of the ST1 command) and re-enables all move commands. This function is normally used to allow manual positioning of the load.

Example

Command	Description
ST1	The drive is disabled

STOP Stop

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>STOP	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME **SEE ALSO** K

Description

This is an alternative expression for the S command. If SSH 0 (Don't save command buffer on stop)is set, the command buffer is cleared (at the end of a move if one is in progress).

A command sequence download is aborted (XD command) and a time delay is terminated.

The motor is decelerated to stop using the current acceleration value ('A') but the drive stays energised.

If SSH1 (Save command buffer on stop)is set, the command buffer is not cleared and only the move that is in progress is terminated.

Example

Command	Description
MC	Set continuous mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
G	Go - run continuously
S	The motor will decelerate to a stop at a rate of 10 revs/sec/sec as soon as S is entered

SV Save

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>SV	N/A	N/A	N/A	Immediate, Device Specific

Description

This command causes the current servo, set up and index parameters to be saved in non-volatile RAM.

Note that in order to successfully save data, link 1 must be in place. Remove the link to write protect the RAM.

T — Time Delay

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>Tn	Seconds	n = 0.01 - 999.99	N/A	Buffered, Universal, Savable in Sequence

Description This command causes the RS232C Control Module to wait the number of seconds specified before it executes the next command in the buffer.

Example

Command	Description
MN	Set normal mode
A5	Set acceleration to 5 revs/sec^2
V5	Set velocity to 5 revs/sec
D25000	Set distance to 25000 steps
T2	Pause for 2 seconds
G	Go
T5	5 second time delay
G	After a further pause of 5 seconds, the move is again executed

TRD — Trigger on Input Distance

VALID FROM
Version 3.0

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TRDn	User steps	±2,097,151	N/A	Buffered, Universal, Savable in Sequence

Description Pauses command execution until a specified distance has been reached. It is used in the speed-change mode MQ to cause velocity to change at a specific point, but can also be used for example to turn on an output. The distance specified can be incremental, or absolute in MPA mode.

Example

Command	Description
TRD3000	Set speed change distance to 3000 steps
V10	Change speed to 10 revs/sec

TRE Trigger on Input Equal

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TREn	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME **SEE ALSO** SKE

Description

Pause until input status matches the pattern n.

The format for n is bbbbbbb_bbbbbbb, and it is equivalent to:

HOME, LMT-, LMT+, +JOG, -JOG, EIN (Drive energised), INDEX TRACK - INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care).

You can omit trailing X specifiers within each group of 7 specifiers. For example the specifiers:

001XXXX_XXX11XX and
001_XXX11 are equivalent.

The next two specifiers are also equivalent:

XXXXXXX_X1XXXXX and
_X1.

TRIP Trigger on In Position

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TRIP	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME **SEE ALSO** SSC

Description

After use of this command, further buffered commands are not executed until the motor has stopped within the deadband region for the specified time. The TRIP command can be used in the MC or MQ modes.

Example

Command	Description
MQ	Set speed change mode
1SSC1	Set output 2 as 'in position'
V35	Set velocity to 35 revs/sec
A200	Set acceleration to 200 revs/sec^2
D80000	Set distance to 80000 steps
L	Set to loop once
G	Go - start the move
TRIP	Wait in position before repeating move
N	End of loop

TRN — Trigger on Input Not Equal

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TRNn	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description

Pause until the input status is not equal to n.

The format for n is bbbbbbb_bbbbbbb and it is equivalent to:

HOME, LMT-, LMT+, +JOG, -JOG, EIN (Drive energised), INDEX TRACK - INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care).

You can omit trailing X specifiers within each group of 7 specifiers. For example the specifiers:

001XXXX_XXX11XX and
001_XXX11 are equivalent.

The next two specifiers are also equivalent:

XXXXXXX_X1XXXXX and
_X1.

TRR Registration Mode

VALID FROM
Version 2.3

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TRRn	User steps	-2,097,152 to +2,097,151	0	Buffered, Universal Saveable in sequence

Description Selects Registration move, a move ending a specified distance in user steps after a mark signal appears at Input 6. The move is used in modes MC or MQ. Registration moves are described in detail in Section H.

TUNE Display Tuning Settings

VALID FROM
Version 2.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aTUNE<n>	N/A	1-11	N/A	Buffered, Device Specific, Savable in Sequence

EXECUTION TIME

SEE ALSO TUNET, TUNEV, DR

Description The TUNE command with no parameter causes the control module to report tuning information calculated or measured during the previous move. The response format is:

1. Measured largest error - steps.
2. Measured largest torque demand.
3. Measured settling time within window (ms).
4. Measured total indexing + settling time (ms).
5. Calculated stiffness (mNm/step).
6. Calculated error at maximum torque (steps).
7. Estimated load inertia (Kg/cm^2).
8. Estimated maximum available acceleration (revs/sec^2).
9. Estimated double time constant (ms).
10. Estimated 1/2 settling time (ms).
11. Estimated closed loop bandwidth (Hz).

If a numeric parameter <n> is entered, any one of the above values as defined by the number will be reported.

If either CPG or CVG is set to zero, the range of values returned will be restricted to the first 4 (i.e. TUNE5 and above are not available).

Example

Command	Description
1TUNET	Self-tune servo
CEW2	Set narrow 'in position' window
D400	Set distance to 400 steps
A100	Set acceleration to 100 revs/sec^2
V50	Set velocity to 50 revs/sec
G	Go - start the move
1TUNE3	Report data item 3 (settling time) only

TUNET Servo Self-Tuning

(Torque amplifier settings)

VALID FROM Version 2.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TUNET	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO TUNE, TUNEV,

Description

This command self-tunes the servo as a torque amplifier. The following parameters are set:

CPG	Maximised to the point where the standing digital vibration is reasonable.
CVG	Set for stability - little overshoot beyond the end point of a move and no oscillations.
COFF	Set so that the average error is zero. This cancels the effect of amplifier and gravitational offsets on positioning accuracy.
CTG	Set to cut off the velocity feedback action at high frequency.
CIG	Set to zero. Integral action is always de-stabilising and a specialised feature associated with a requirement for extreme accuracy.
CFG	Set to zero, assuming a point to point application. You should set this to the value of CVG (PID equivalent) if you have a following/contouring type application.

Example

Command	Description
1TUNET	Self-tune servo
1SV	Save parameters for future use

SAFETY

The tuning process moves the load a distance of 178 steps backwards and forwards using step changes of position demand that result in about 50% of the maximum available motor torque being suddenly applied to the load. This process continues for up to 30 seconds depending on the load conditions. During this time buffered commands typed at the keyboard or pending in a sequence will not be executed. However, the unit will respond to immediate commands such as K, S or OFF, to enabled end of travel inputs, to an enabled Input 5 stop input and to the emergency stop input. If the tuning is interrupted in this way, the servo parameters will then be in an intermediate state and it may be necessary to cycle the power or use the Z command to restore them to a viable state.

TUNEV Servo Self-Tuning (Velocity amplifier settings)

VALID FROM Version 2.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>TUNEV	N/A	N/A	N/A	Buffered, Universal, Saveable in sequence

EXECUTION TIME

SEE ALSO TUNE, TUNET,

Description

This command self-tunes the servo as a velocity amplifier. The process may also be used for pneumatic and hydraulic actuators. It is a limited process, maximising CPG starting from the value you enter so that there is little overshoot and no oscillations occur. The final value of CPG is dependent on the velocity loop gain which you have set. The following parameters are set:

CPG Maximised, starting from the entered value and taken to the point where there is little overshoot beyond the end of move position and no oscillations occur.

CVG Not altered - normally zero since damping is provided by the velocity amplifier.

COFF Set so that the average error is zero. This cancels the effect of amplifier and gravitational offsets on positioning accuracy.

CTG Not altered.

CIG Set to zero. Integral action is always de-stabilising and a specialised feature associated with a requirement for extreme accuracy.

CFG Set to zero.

Example

Command	Description
1CPG50	CPG set for a low gain
1TUNEV	Self-tune servo
1SV	Save parameters for future use

SAFETY

The tuning process moves the load a distance of 178 steps backwards and forwards using step changes of position demand that result in about 50% of the maximum available motor torque being suddenly applied to the load. This process continues for up to 30 seconds depending on the load conditions. During this time buffered commands typed at the keyboard or pending in a sequence will not be executed. However, the unit will respond to immediate commands such as K, S or OFF, to enabled end of travel inputs, to an enabled Input 5 stop input and to the emergency stop input. If the tuning is interrupted in this way, the servo parameters will then be in an intermediate state and it may be necessary to cycle the power or use the Z command to restore them to a viable state.

U Pause

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>U	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME

SEE ALSO PS, C

Description

Pause immediately and wait for continue.

Execution of a command string will be paused at the point where a U command is entered. Command C will cause execution to be resumed at the point where it was paused.

Example

Command	Description
MN	Set normal mode
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
L	Loop continuously
D8000	Set distance to 8000 steps
G	Go - start move
H-	Set direction to negative
G	Go - start negative move
N	End of loop

In this example the motor will turn 2 revolutions in the positive direction, then two revolutions in the negative direction in a continuous loop. If a U command is entered during the execution of the loop, the motor will stop at the end of its current move. On receipt of a C command, execution of the loop will be resumed from the point where it was paused.

V Velocity

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>Vn	Revs/sec	n = 0.0001 - 100.00	1	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO A, D

Description

Sets the velocity. Note that the actual top speed of the motor is limited by the motor type and drive. If the velocity is set too high, the following error will exceed the maximum position error (set by CPE) and the drive will be shut down. If the command V0 is given when moving in MQ or MC the motor will stop as in the S command. If the motor is not moving, the fastest possible velocity is used for the move.

Example

Command	Description
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
D8000	Set distance to 8000 steps
G	Go - Motor will turn 2 revolutions in the positive direction at 10 revolutions per second

VSY Velocity Change - Synchronous Mode

Not used in PS7X, BLX

XC Checksum

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
aXC	N/A	N/A	N/A	Buffered, Device Specific, Savable in Sequence

EXECUTION TIME

SEE ALSO LA

Description

This command causes the checksums for the volatile and non-volatile RAM to be computed and reported .

XD Sequence Download

VALID FROM
Version 1.4

SYNTAX	**UNITS**	**RANGE**	**DEFAULT**	**ATTRIBUTES**
<a>XDn	N/A	1 - 63	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO XT, XE, XR, XRP

Description

This is the 'Start sequence' indicator for downloading a sequence. It clears the command buffer and all subsequent commands will be stored in the buffer until receipt of an XT command. A sequence will not be stored if a sequence with the same number identifier already exists, or if there is an error in writing the sequence. THE SEQUENCE IS NOT STORED IN NON-VOLATILE BACKUP RAM UNTIL THE SV COMMAND IS GIVEN.

Example

Command	Description
XD1	Start sequence 1 download
A10	Set acceleration to 10 steps/sec^2
V10	Set velocity to 10 revs/sec
D8000	Set distance to 8000 steps
G	Go - start move
D4000	Set distance to 4000 steps
H	Reverse direction
G	Go - start move
XT	End of download

The sequence is defined as sequence No.1. Each time XR1 is entered the motor will turn 2 revolutions positive and then 1 revolution negative.

XE Sequence Delete

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XEn	n = sequence number	n = 1 - 63	N/A	Buffered, Universal

EXECUTION TIME

SEE ALSO XD, XT, XR, XRP

Description Delete sequence n from RAM.

Example

Command	Description
XE1 SV	Sequence No.1 is deleted from RAM

XP Power-On Sequence Number

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XPn	n = sequence number	n = 0 - 63	N/A	Buffered, Universal

EXECUTION TIME

SEE ALSO XQ, XSP, XSR, XZ, IM, SS

Description Set the power-on sequence mode and sequence number to be executed on power up. The command must be saved before power off to enable the sequence to be executed on power up. If the command XP0 is sent, no power-on sequence will be executed. This facility is normally used for stand alone operation.

Example

Command	Description
XP1	Sequence No.1 will be executed on power-up
SV	Save current setup

XR Run Sequence

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XRn	n = sequence number	n = 1 - 63	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO XT, XD, XE, XRP

Description

Run a sequence.

An XR command can be used within one sequence to start execution of another sequence; all commands in the first sequence following the XR will be executed after the sequence called by the XR command is complete (in this respect an XR acts like a subroutine call).

You must be certain that the number of loops and loop terminators are balanced within a single sequence.

You cannot start a loop in one sequence and end it in another called by the XR command.

Example

Command	Description
XR1	Sequence No.1 will be executed

XRP Run/Pause Sequence

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XRPn	n = sequence number	n = 1 - 63	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO XR, XD, XT, XE

Description

Run a sequence with a pause. The XRP command operates in the same way as XR except that a pause condition, which must be cleared before the command buffer is executed, is automatically generated . The pause condition is executed only if the sequence is valid and actually run

Example

Command	Description
XRP1	Sequence No.1 is loaded into the command buffer and is paused until a continue (command C) is entered

XRT Return From Sequence

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XRT	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

EXECUTION TIME

SEE ALSO XT

Description

This command is used in a sequence to terminate execution of that sequence. It is typically used in branched sequences where the RS232C Control Module must exit the sequence before the XT command. It is used in conjunction with 'SKIP' commands.

XSD Sequence Download Status Report

VALID FROM Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aXSD	N/A	N/A	N/A	Buffered, Device Specific

EXECUTION TIME

SEE ALSO XD

Description

Requests report back of the status of a previous sequence sent from the controller to the RS232C Control Module as a single ASCII character followed by [CR] as follows:

0[CR]: Received OK

1[CR]: Cannot overwrite existing sequence (sequence with this number already exists - erase previous one first)

2[CR]: Sequence buffer full

Note that in order to retain the sequence it is necessary to save it using the SV command.

Example

Command	Response
XSD	0[CR] (Sequence received OK)

XSR Sequence Run Status Report

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aXSR	N/A	N/A	N/A	Immediate, Device Specific

EXECUTION TIME

SEE ALSO XR, XRP

Description

Return sequence run status as a single ASCII character followed by [CR].

0[CR] = Running (or ran successfully)
2[CR] = Invalid sequence - requested sequence doesn't exist

Example

Command	Response
1XSR	0 (sequence OK)

XSS X Sequence Status

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aXSSn	Sequence number	n = 1 - 63	N/A	Buffered, Device Specific

EXECUTION TIME

SEE ALSO XD, XT, XE

Description: Return status of sequence n as a single ASCII character followed by [CR].

0[CR]: Empty - sequence doesn't exist
2[CR]: OK - sequence does exist

The XSS command will inform the user if the sequence indicated by the number immediately following (no space) the XSS command does in fact exist in memory.

Example

Command	Response
1XSS1	0 [CR] (Sequence not defined)

XT Sequence Terminator

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XT	N/A	N/A	N/A	Buffered, Universal, Savable in Sequence

Description The XT command is used to terminate a sequence download, thus returning the system to command mode. The executable end of sequence is also marked at this point.

XU Sequence Upload

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
aXU<n>	n = sequence number	n = 1 - 63	N/A	Immediate, Device Specific

Description The sequence upload command displays the sequences stored in RAM. The value n is the sequence number and if it is omitted a display of all stored sequences will occur. Note the RS232C Control Module is supplied with default sequences to get you started (see RIFS command).

XZ Reset Power-Up Sequence Mode

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>XZ	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME

SEE ALSO Z, XP

Description Sets the power-on sequence mode to zero (thereby disabling sequence activation on power-on).

This is the same function as XP0 but it also calls the SV command to save this mode.

*Y Terminate Loop

VALID FROM
Version 2.3

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
<a>Y	N/A	N/A	N/A	Immediate, Universal

EXECUTION TIME

SEE ALSO L, N

Description Terminates a loop at end of the current pass.

Example

Command	Description
L	Loop continuously
A10	Set acceleration to 10 revs/sec^2
V10	Set velocity to 10 revs/sec
D8000	Set distance to 8000 steps
T2	Wait for 2 seconds
G	Go - start move
N	End of loop
Y	Terminate loop at end of pass

Loop will continuously cause the motor (4000 steps/rev) to move 2 revolutions positive and then wait for 2 seconds. The loop will be terminated at the end of the pass during which the Y command is entered.

Z Reset

VALID FROM
Version 1.4

SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES
Z	N/A	N/A	N/A	Immediate, Universal

Description Software reset. This command applies a reset as if the power had been switched off and then on again. The command is universal and it returns all axes to their last saved settings - it cannot be made device specific.

Command summary

Help Command (HELP1)

HELP	Produce all help displays
HELP1	HELP list
HELP2	Indexer Commands
HELP3	Start/Stop Commands
HELP4	Sequence Commands
HELP5	Homing, Jog, Limit
HELP6	Option Set Commands
HELP7	Information Commands
HELP8	Inputs and Outputs
HELP9	Control and Memory Commands
HELP10	Using the Indexer
HELP11	Recommended Servo Settings
HELP12	Servo Setup Procedure

Motion Commands (HELP2)

A(n)	Set acceleration
C	Continue
D(n)	Set move distance
FDA	Set feed acceleration
FDV	Set feed velocity
G	Go, make a move
GL	Go linear
GLD	Go linear define
GSY	Go synchronous
H	Reverse motor direction
H+	Set positive direction
H-	Set negative direction
ID	Immediate distance
IV	Immediate velocity
L(n)	Loop instructions
MAS	Mode asynchronous
MC	Move continuous
MN	Normal preset moves
MPA	Absolute position mode
MPI	Incremental position mode
MQ	Velocity change preset moves
MSY	Mode synchronous
N	End of loop indicator
PS	Wait for continue
PSY	Pause synchronous
SIM	Set indexer/following mode
T(n)	Delay for n secs
TRD	Pause until distance reached
TRE	Pause until inputs equal
TRIP	Pause until in position

TRN	Pause until inputs not equal
TRR	Pause for I6 registration
U	Pause, wait for continue
V(n)	Set velocity
VSY	Synchronise velocity change

Start/Stop Commands (HELP3)

K	Kill motion
LS	Stop now at LA deceleration
OFF	Immediate stop - de-energise
ON	Immediate start - re-energise
S	Stop motion, decelerate now
SB	Stop motion, buffered
ST0	Energise drive
ST1	De-energise drive
Y	Terminate loop or sequence
Z	Power up reset

Sequence Commands (HELP4)

XC	Report checksum
XD(n)	Begin download sequence
XE(n)	Delete sequence
XP(n)	Select power up sequence
XR(n)	Execute sequence
XRP(n)	Execute sequence with pause
XRT	Return from sequence
XSD	Report status of download
XSR	Report run sequence status
XSS(n)	Report sequence status
XT	Sequence terminator
XU(n)	Sequence upload
XZ	Turn off power up sequence mode

Homing, Jog, Limit Commands (HELP5)

GA(n)	Set go home acceleration
GH+/-(n)	Go home with direction + velocity
GHF(n)	Final home move velocity
JA(n)	Set jog acceleration
JV(n)	Set jog velocity
LA(n)	Set limit accn
LD(n)	Set limit function
PZ	Set current position as absolute zero
SP	Set current position to value

Option Set Commands (HELP6)

E	Enable communications
EX	1=terminal 0=computer mode
F	Disable communications
HS	Display H/W switches
HSA	1=resolver feedback (not available)
HSB	0=closed loop brushless
HSC	0=open loop stepper 1=other
HSD	0=no daughter board 1=output expansion board fitted 2=analogue feedback board fittes
OS	Display 'OS' switches
OSA	1=Home to index pulse
OSB	1=switched integral action on
OSC	1=single value returned by DS
OSD	1=input 7 is sequence select line
OSE	1=enable jog
OSF	1=allow initialise on limit
OSG	1=allow dither
OSH	1=square dither 0=triangular
OSI	1=reduced movement initialise
OSJ	1=24 bit gearbox 0=16 bit
OSK	1=encoder count checked
OSM	1=accumulate at 2ms sampling
OSO	1=suppress units
SS	Display SS switches
SSA	0=echo on 1=echo off
SSB	1=input 6 clears and store pause
SSC	1=output 2 is in position
SSD	1=output 1 is composite fault
SSE	1=input 5 is stop line
SSF	1=input 4 is sequence select
SSG	1=save comm buffer on limit
SSH	1=save command buffer on stop
SSI	1=inputs 5,6 & 7 select sequences

Information Commands (HELP7)

B	Command buffer status request
BS	Command buffer size request
DFX	Returns index status (32 bits)
DIC	Report Control Module position
DPA	Display position-actual
DPE	Continuous error display
DPS	Display position setpoint
DR	Display report
DVA	Display velocity-actual
DVS	Display velocity setpoint
OSC	1=report one signal value only

P	Request position this move
PIC	Display servo loop diagram
PR	Request absolute position
QS	Send a response at move end
R	Status request
RA	Limit switch status request
RB	Pause status request
RE	Report drive status
RG	Go home status request
RPO	Report power on time
RS	Sequence status request
RSE	Report servo error conditions
RV	Report software revision

Inputs and Outputs (HELP8)

HSD	0=no daughter board 1=output expansion board fitted 2=analogue board fitted
IO	Sets outputs immediately
IS	Reports input status
O	Turns outputs on and off
OSD	1=input 7 is sequence select line
SKE	Skip on inputs equal
SKN	Skip on inputs not equal
SSB	1=input 6 clears pause
SSC	1=output 2 is in position
SSD	1=output 1 is composite fault
SSE	1=input 5 is controlled stop line
SSF	1=input 4 is sequence select strobe
SSG	1=save command buffer on limit
SSH	1=save command buffer on stop
SSI	1=inputs 5,6,7 are sequence select
TRE	Pause until inputs equal
TRN	Pause until inputs not equal

Control and Memory Commands (HELP9)

CAG	Configure acceleration gain
CCP	Configure command peak
CCS	Configure command source
CDG	Set D gain in PID
CEW	Set in position window
CFG	Set feed forward gain
CIG	Set integral action gain
CIT	In position wait time (2ms ints)
CIW	Set integral action window
CJL	Set motor + load inertia (Kg-cms^2)
CMR	Set motor resolution
COFF	Set amplifier offset
CPE	Set position error limit

CPG	Set forward path gain
CTG	Set filter time constant
CTQ	Set available motor torque (Nm)
CUR	Set user resolution
CVG	Set velocity feedback gain
DS	Signal display by number
DTA	Set dither amplitude
DTF	Set dither frequency
FOL	Set following percent
OFF	Immediate de-energise
ON	Immediate energise
RAT	Set rate multiplier value
RFS	Servo to factory settings
RIFS	Indexer to factory settings
RSE	Reports servo error conditions
SIM	Select indexer operation
ST0	Energise drive
ST1	De-energise drive
SV	Save current settings in backup RAM
TUNE	Report tuning information
TUNET	Automatic torque amp tuning
TUNEV	Automatic velocity amp tuning

Following Commands (HELP13)

CAG	Set following acceleration gain
CCS0	x4 following decode
CCS1	x2 following decode
CCS2	x1 following decode
CCS3	Clock and Direction Decode
CUR	Set user resolution
CMR	Set feedback (motor) resolution
FOL	Set following (SIM5 mode)
RAT	Set gearbox scaling factor
SIM0	Indexer, not following, mode
SIM1	Encoder following mode
SIM2	Superposition following mode
SIM3	Positive scaled following
SIM4	Negative scaled following
SIM5	Preset following indexer mode

ALPHABETICAL COMMAND LISTING

A	Acceleration Rate
B	Buffer Status Request
BS	Buffer Size Request
C	Continue
CAG	Configure Acceleration Gain
CCP	Configure Command Peak
CCS	Configure Command Source
CEW	Configure In-Position Window
CDG	Configure Derivative Gain
CFG	Configure Feedforward Gain
CIG	Configure Integral Gain
CIT	Configure In-Position Time
CIW	Configure Integral Action Window
CIX	Configure Index Mark Resolution
CJL	Enter Motor + Load Inertia
CMR	Configure Motor Resolution
COFF	Configure Amplifier Offset
CPE	Configure Position Error
CPG	Configure Proportional Gain
CTG	Configure Filter Time Constant
CTQ	Enter Motor Torque
CUR	Configure User Resolution
CVG	Configure Velocity Gain
D	Distance
DFX	Display Flags Indexer
DIC	Display Indexer Counter
DPA	Display Position Actual
DPE	Display Position Error
DPS	Display Position Setpoint
DR	Display Report
DS	Display Signal
DTA	Set Dither Amplitude
DTF	Set Dither Frequency
DVA	Display Velocity Actual
DVS	Display Velocity Setpoint
E	Enable Communications
EX	Set Communication Style
F	Disable Communications
FDA	Acceleration, Linear Interpolation Moves
FDV	Velocity, Linear interpolation Moves
FOL	Following Percent
G	Go
GA	Go Home Acceleration
GH	Go Home
GHF	Go Home Final
GL	Go Linear
GLD	Go Linear Define
GSY	Go Synchronous
H	Change Direction
^H	Backspace
H+	Set Direction
H-	Set Direction
HELP	Produce Help Screens
HS	Display H/W Configured Switches
HSB	Configure Motor Type
HSC	Configure Motor Type
HSD	Daughter Board Select
ID	Immediate Distance
IO	Immediate Output
IS	Input Status
IV	Immediate Velocity
JA	Jog Acceleration
JV	Jog Velocity
K	Kill
KILL	Kill
L	Loop
LA	Limit Deceleration
LD	Limit Disable
LS	Limit Switch Fast Stop
MAS	Mode Asynchronous
MC	Mode Continuous
MN	Mode Normal
MPA	Mode Position Absolute
MPI	Mode Position Incremental
MQ	Speed Change Mode
MSY	Mode Synchronous
N	End Loop
O	Programmable Output
OFF	De-Energise Drive
ON	Energise Drive
OS	Other Switches
OSA	Home to an Index Pulse
OSB	Integral Action Selection
OSC	Monitor Command Reporting
OSD	Input 7 Sequence Select
OSE	Jog Enable
OSF	Initialisation on Limit
OSG	Dither Enable/Disable
OSH	Dither Square/Triangular Select
OSI	Select Standard/Reduced Movement on Initialising
OSJ	RAT 16/24 Bit select
OSK	Integrity Check of Encoder

OSM	Integral Action Sensitivity
OSO	Suppress Units
P	Position
PIC	Picture
PR	Position Report
PS	Pause
PSY	Pause for Synchronisation
PZ	Position Zero
QS	Transmit An Identifier
R	Report Control Module Status
RA	Report A - Limit Status Request
RAT	Set Rate Multiplier Value
RB	Report B - Miscellaneous Status Request
RE	Drive Status Request
RFS	Return Servo to Factory Settings
RG	Report Go Home Status
RIFS	Return Indexer to Factory Settings
RPO	Report Power-On Time
RS	Report Sequence Status
RSE	Report Servo Errors
RV	Revision
S	Stop
SAVE	Save
SB	Stop Buffered
SIM	Set Indexer/Following Mode
SKE	Skip On 'Equals'
SKN	Skip On 'Not Equal'
SP	Set current position to value
SS	Set Switches
SSA	RS232C Echo Control
SSB	Set Input 6 = Clear Pause Input
SSC	Set Output 2 = In-Position Output
SSD	Set Output 1 as Composite Fault Signal
SSE	Set Input 5 = Controlled Stop Input
SSF	Set Input 4 as Sequence Strobe
SSG	Save Command Buffer On Limit
SSH	Save Command Buffer On Stop
SSI	All Inputs Select Sequence
ST	Energise/De-Energise Drive
STOP	Stop
SV	Save
T	Time Delay
TRD	Trigger On Input Distance
TRE	Trigger On Input Equal
TRIP	Trigger On In Position
TRN	Trigger On Input Not Equal
TRR	Registration Mode
TUNE	Show Tuning Settings
TUNET	Self-Tune Servo (Torque Amplifier)
TUNEV	Self-Tune Servo (Velocity Amplifier)
U	Pause
V	Velocity
VSY	Change velocity, Synchronous Mode
XC	Checksum
XD	Sequence Download
XE	Sequence Delete
XP	Power-On Sequence Number
XR	Run Sequence
XRP	Run/Pause Sequence
XRT	Return From Sequence
XSD	Sequence Download Status Report
XSR	Sequence Run Status Report
XSS	X Sequence Status
XT	Sequence Terminator
XU	Sequence Upload
XZ	Reset Power-Up Sequence Mode
Y	Terminate Loop
Z	Reset

Chapter 6. SERVO TUNING

Tuning the Drive with a Positioner

The positioner is designed to operate with the drive configured as a torque amp, and this is the usual mode of operation described here. In a few applications, it is desirable to configure the drive as a volocity amp, in which case the drive Time Constant and Damping pots are set up as described in the drive manual and the only gain which needs to be set on the positioner is CPG (small values of CVG and CFG can be tried, if desired).

Tuning Positioner with Drive in Torque Amp

To obtain smooth rotation at optimum speed and accurate positioning of the motor without overshoot or oscillation, with the motor connected to your system, you must tune the gain and damping characteristics of the servo loop. The TUNET command provides a self-tuning feature but for some applications the best results will be obtained by manual tuning using the commands listed at the end of this appendix, together with those from the Detailed Command List in Chapter 5. Refer to the servo loop block diagram shown in Figure 1-12. The parameters affecting tuning are as follows.

Tuning Parameters

Proportional gain (CPG command)

Proportional gain determines the amount of torque produced in response to a given position error. It sets the stiffness of the system and affects the following error. A high proportional gain gives a stiff, responsive system but results in overshoot and oscillation which require damping.

Velocity gain (CVG command)

Velocity feedback is a signal which increases with shaft speed. It acts in a negative sense opposing the proportional action and helping to stabilise the motion. The damping action of velocity feedback allows a higher proportional gain to be used.

Integral gain (CIG command)

Proportional action may be insufficient to overcome static position errors due to friction. Integral action accumulates a steady state error until sufficient torque is produced to move the load. It improves overall positioning accuracy but may produce low frequency oscillation around the commanded position.

Feedforward gain (CFG command)

The opposing action of proportional and velocity gains result in a position error which depends on speed. This is called "following error". Feedforward gain can be used to offset the following error and improve tracking accuracy. This is important in contouring applications.

Filter time constant(CTG command)

Fast positioning systems need high proportional and velocity gains. By limiting the bandwidth, the digital filter prevents a high gain system from becoming too lively. The filter also serves to average the effects of the digital control loop, reducing the jitter at standstill and the audible noise. The value of CTG should be kept as low as possible.

You can examine these parameters by using the DR command.

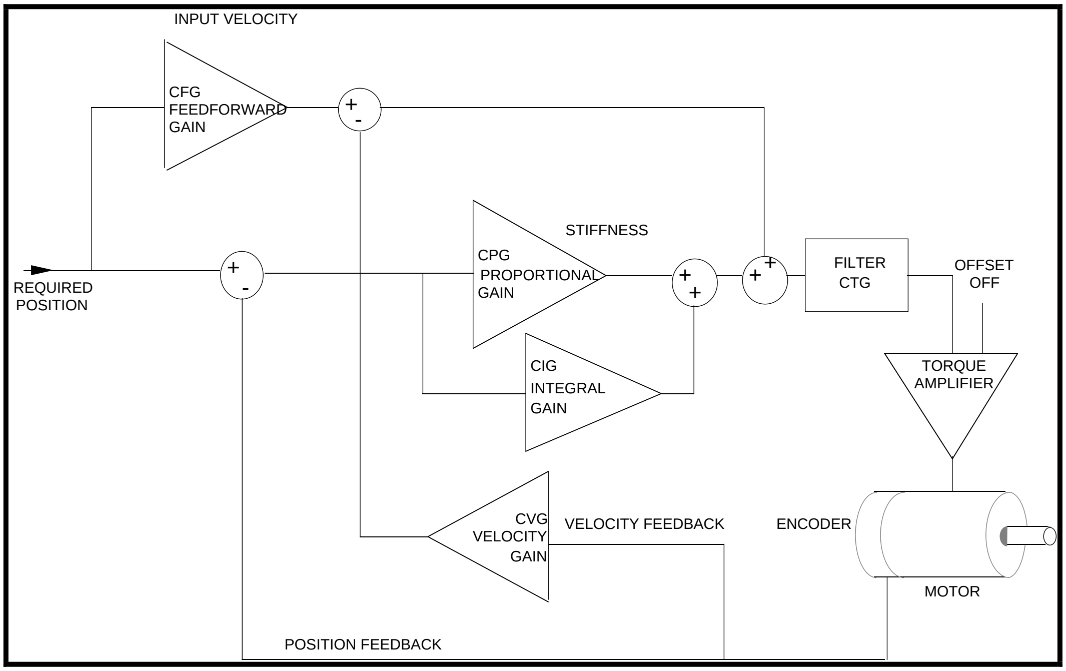


Figure 6-1. Servo Control Loop

Terminal/ Computer

The terminal/computer communication should be set to to 9600 baud, no parity, full duplex (not local), 1 stop bit.

After power up, the sign on message "DON'T PANIC Type 'HELP'" and the > prompt will appear on the screen unless you have previously set the positioner into the terse computer communications mode using EX0. In any case using the command 1DR will display the status of the positioner.

Connecting the Loop

Safety

It is advisable to have the ESTOP input wired to a stop switch located in an easily accessible position so that the system can be halted quickly in the event of instability causing oscillation. An alternative is to cut the power using the command OFF[CR] but this takes longer.

Unless you have saved data in the past while the motor was energised, the positioner will power on without attempting to move or energise the motor.

If any of the motor or encoder connections have been re-made or changed, the first step is to check the integrity and polarity of the motor connections. Proceed as follows:-

Make sure the axis is de-energized, type OFF[CR], if in doubt. Use the command :

```
aDS3[CR]
```

where a = the device address
to display the feedback position continuously (approximately 0).

Turn the motor shaft CW and check that the displayed value increases, turn the shaft CCW and it should decrease. If this does not happen you must not close the loop with ON at any time since the feedback will be positive.

Press [CR] to restore the > prompt.

Manual Tuning Procedure

1. Initial Setting Up

Type the commands CFG0 CPG0 CVG0 COFF0 CIG0 CTG0 to software disable the servo. Check the result of these commands with command 1DR.

When the axis is energised, it will be open loop with a slight torque demand due to amplifier offsets. This may cause motion of the motor. You may have to restrain the axis and be prepared to switch off, or type OFF[CR]. In any case there is an automatic software limit set by the command CPE on the distance it will drift.

2. Position Error Window Setting

The drive is disabled to protect the system if the position error goes outside of the position error window. The window can be set to a low number of steps to avoid damage due to overshoot or oscillation.

The value of CPE defines the size of the window in steps. It can be displayed by using the CPE command.

During tuning, CPE is normally set to 4000 steps i.e. 1 rev either way at the usual 4000 steps/rev. If in doubt set a lower value. The effect of going beyond the set error limit will be to automatically de-energise the motor.

Type the command RSE (report servo errors). This may show quite a list of errors. "Drive disabled by software command", "error limit exceeded" and "failed checksum of backup ram" are all acceptable at this point but "Drive fault" should not come up. If it does, check the LED on the drive front panel and try powering down the system and then powering it up again.

DO NOT PROCEED IF A DRIVE FAULT PERSISTS

3. Energising the Axis

Energise the axis by typing ON [CR]. You may find it convenient to restrain the axis to stop it drifting out of the error limit and de-energising. If it does de-energise, it can be restarted by typing ON.

4. Cancelling Drift

Cancel any offset using the command aCOFF<value>[CR]. Negative values stop CW drift; positive values stop CCW drift. Values up to a few thousand are acceptable (the maximum is 32,000 corresponding to full torque demand). Only small values are normally required, for example 1COFF10.

Check that making COFF more positive tends to turn the motor CW and making it more negative tends to turn it CCW. If this does not happen, do not proceed, as the loop will have a net positive feedback and will oscillate (although probably slowly) when the loop is closed.

5. Closing the Loop

Set a small position error limit (2000 steps for example) so that if oscillation occurs during tuning, the resulting overshoot producing large position errors will exceed the limit set by CPE and shut down the drive.

If necessary first move the axis to a good starting point, then type ON[CR] to energise the motor and CPG1[CR] to close the loop. If a slight vibration appears at this point, it indicates positive feedback. This should only occur if you have changed the ,motor or encoder connections. Taking care, displace the axis slightly and then let go. You should experience a slight spring effect and the axis should return to its original position approximately. The closed loop is now working.

Setting Up Servo Parameters

NOTE: When setting up servo parameters, do not allow the motor to go unstable for more than a second or two.

Tracking Applications

For tracking applications (PID tuning), where minimum following error is required, velocity feedforward should be used to improve the tracking accuracy. This can be set using the CFG command or the CDG command can be used to set feedforward gain and velocity feedback gain to the same value.

Point to Point

For fast point to point positioning applications (PIVF tuning), a high

Positioning Applications

proportional gain (CPG) damped by velocity gain (CVG) should be used.

Help 11

A guide to servo loop parameter settings for a fully damped non-oscillating response is provided in HELP 11. This uses the maximum low speed motor torque entered by the CTQ command and the total moment of inertia entered by the CJL command in calculating suggested settings for the loop parameters.

First enter the values of CTQ and CJL (Motor + load inertia) for your system, then use HELP 11 to display recommended servo loop parameter settings calculated from the data you have entered. For example, if 3.6 is entered for the CTQ command to indicate a maximum torque of 3.6 Nm available and 1.6 is entered for the CJL command indicating a total inertia of 1.6 Kg/cm^2 , the following table will be displayed:

Gain CPG	Velocity Gain CVG/CFG/ CDG	Stiffness mNm/ step	Error @ maximum torque (fb steps)	Doubled T Constant (msecs)	1/2 moved time (msecs)	Closed loop bandwidth (3db) (Hz)
92	57	10	360	5	8	20
46	40	5	720	7	12	14
23	29	2	1400	10	17	10
11	20	1	2900	14	24	7
5	14	0	5700	20	34	5

Table 6-1. HELP 11 - Servo Setup Table

Each successive line of the table corresponds to a less highly tuned servo system with reduced stiffness and speed of response. These reduced values may be used in matching the servo to lower specification mechanics.

The following describes in more detail the terms used in the table:

Velocity Gain

The settings of CVG, CDG and CFG depend upon the application as described above.

For the figures in the rest of the line of the table to apply, the CDG or CVG setting should be as in column 2.

For PID applications, setting CDG to the figure quoted in the table will automatically set CVG and CFG to the same value.

Stiffness

Stiffness is the restoring force per unit of displacement. The force is dependent on the number of steps displaced from the demanded position. For example, a displacement of 100 steps with a stiffness of 10 mNm/step would produce a restoring force of 1 Nm.

Error at Maximum

As the axis is increasingly displaced from the demanded position,

Torque the restoring torque increases at the quoted stiffness until the maximum available torque is applied. The Error @ Max Torque is the number of steps of displacement required to produce a restoring force of the maximum available torque.

Doubled T Constant When the damping is critical, the response of the servo loop is that of two simple filters in series, each having a time constant T. The value of T is quoted in the table.

1/2 Moved Time The following is an example move using servo values taken from the top line of the HELP 11 table to show the 1/2 moved time. The move is at maximum acceleration and velocity and it is initiated by the command string: D100 A0 V0 G

Figure 6-2 represents instantaneous position plotted against time for the resulting servo movement .

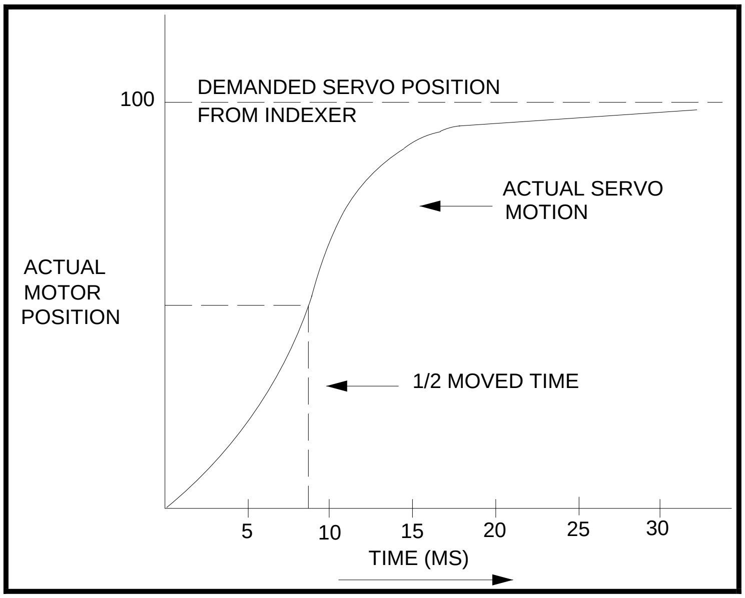


Figure 6-2. Example Servo Response Curve

The 1/2 moved time is the time taken to complete half of the move, 50 steps in this example.

The move will be half complete in the time 1.7T (column 6 of the table) or 1.7 x 5 = 8 milliseconds approx.

The move will be 96% complete at time 5T. In the example, the motor will have moved 96 steps in 25ms.

Bandwidth The bandwidth of the servo system is the range of frequencies at which a sine wave move pattern input will be followed within 3db by the actual motion of the motor. Figure 6-3 shows the sine wave position demand input and the graphical representation of the motor movement.

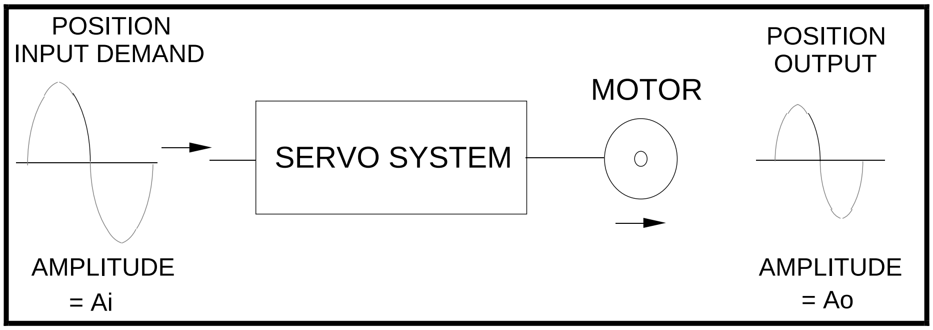


Figure 6-3. Servo System Bandwidth

As the frequency of the sine wave of demanded positions at the input (Ai) increases, the ability of the servo system to follow is reduced and the amplitude of the sine wave of the resulting output positions (Ao) decreases. The frequency at which it is reduced by 3db is the upper limit of the bandwidth.

Coarse Tuning the Loop

Attach a nylon cable tie to the shaft of the motor. The loose end can be used as a coarse indicator for overshoot and oscillation.

Use the command string A0 V0 D200 G to produce a short, sharp rotation of the shaft. Observe the movement, watching the cable tie for overshoot and oscillation. Repeat the movement, increasing CPG using the settings suggested in HELP11 until overshoot just occurs.

Balance the increasing gain with damping by setting CVG to 10, 20, 40 ... Increase it until the oscillation and overshoot indicated by the cable tie ceases.

Fine Tuning

After coarse tuning the loop continue balancing the gain and damping, monitoring the overshoot at the terminal by typing:

```
aDPE [space] G [space] [CR]
```

where a=the device address.

The CR should be entered quickly after typing the space following the G command - this will stop the position error display with the final positioning readings on screen to allow you to analyse them. Overshoot is indicated by the position error reading passing through zero at the end of the movement, then returning back through zero in the opposite direction, oscillating with decreasing error readings about the zero position before finally settling within the position error window. The damping should be increased to the point where overshoot just disappears. Too much damping will result in a sluggish movement.

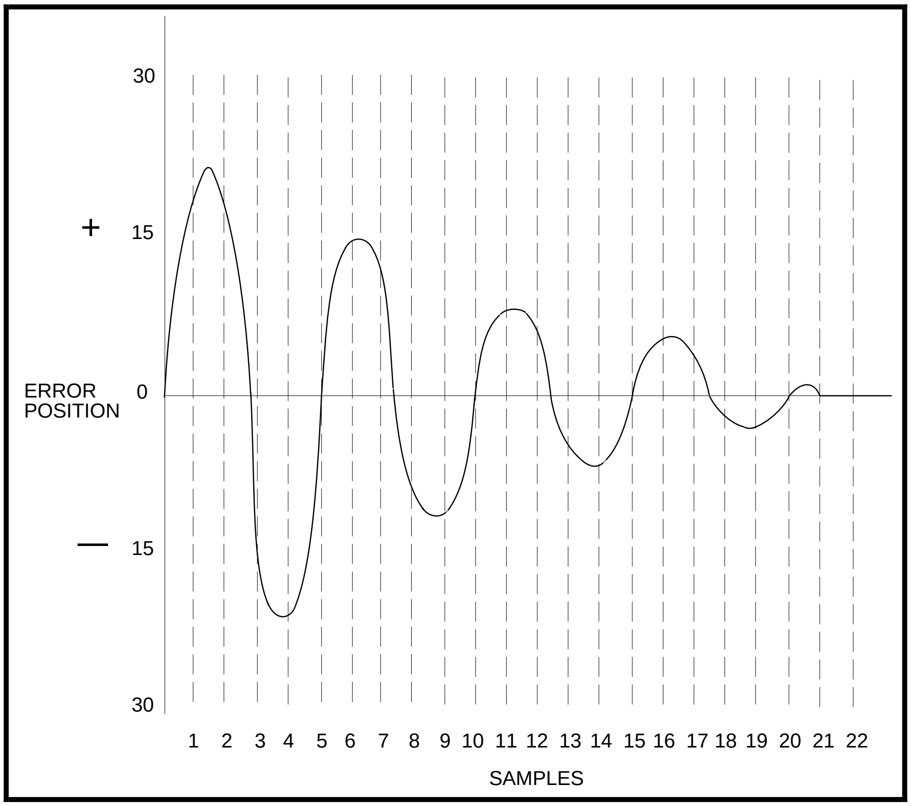


Figure 6-4. Overshoot During Positioning

The following table shows the DPE readings for the final positioning graph of Figure 6-4. This is only an example to illustrate the general principles involved in tuning - in practice, the resulting figures may differ considerably from those given but they will follow the general pattern shown. The overshoot shown here is rather more severe than might be generally expected.

Samples	Readings					
1 - 7	23	22	-17	-20	-5	14	12
8 - 14	-7	-10	0	6	4	-3	-4
15 - 21	-2	4	3	-2	-3	-1	0

Table 6-2. DPE Readings During Final Positioning

Notes on Tuning

The TUNE command will show how much torque was used, with 1023 corresponding to saturation. If the maximum torque demand is significantly less than 1023 you should try increasing CPG. You will normally need to increase CVG as well in order to prevent overshoot. Make sure that the maximum torque never exceeds about 1000 to guarantee that the servo loop does not saturate.

The TUNE command also shows the time taken for the complete move as well as the settling time. This information is also useful in optimising the response.

To summarise:

1. Use DPE to look for overshoot and oscillation.
2. Use TUNE to check maximum torque demand, move time and settling time.

CFG, the feedforward gain should never be set very much larger than the velocity feedback gain (CVG). A setting of 20% more than CVG is the largest practical value.

An increase in gain is always best matched by an increase in damping; when this is no longer true you have reached the optimum values. A useful guideline is that an increase of times X in gain is best matched by an increase of times square root X in velocity feedback (think of this as an X% increase in gain matches an X/2% increase in damping e.g. 30% more gain matches 15% more damping). This will keep the shape of the response the same and just increase its frequency (you are holding a quantity called the damping ratio constant).

Servo Self-tuning

A facility for self-tuning the servo is included in the positioner software, using the command TUNET. The servo parameters are set at the time that they are executed and there is no continuous on-going adjustment (known as adaptive control).

Tuning is for the following characteristics:

- Stability - does not oscillate as it moves or settles
- Maximised stiffness and accuracy - deflects as little as possible in response to external forces
- Maximised bandwidth and speed of response - settles to the commanded position as quickly as possible at the end of a move
- Does not overshoot the end position in a move
- Minimal sound during motion or when holding position

- Tracks a profile of constantly changing position or velocity with as little error as possible

- Resistant to change as friction and other load characteristics vary from time to time, machine to machine, application to application and with production variations

- Operates for smaller loads than that used for tuning down to no load.

There are a number of other move related requirements:

- Make full use of the torque and power available so that the longest, fastest move possible can be made

- Perform equally well for fast or slow motions and for short or long moves

- Minimise heating effects in the motor and drive

It is not correct to adjust the servo loop parameters to meet these requirements as the general servo performance would be adversely affected. They are best met by adjusting the move profile (D, V and A values) and possibly the feedforward CFG value whilst watching for saturation effects.

If the application is purely velocity control with no requirement for positioning, then the tuning is best carried out manually. The self-tuning settings will work, but they will not produce the best achievable results.

Index