ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Parker PMC-6270* at our website: ***Click HERE***

Compumotor

6270

2-Axis Motion Controller

User Guide

Compumotor Division
Parker Hannifin Corporation
p/n 88-013913-01B January, 1995

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Important User Information

To ensure that the equipment described in this user guide, as well as all the equipment connected to and used with it, operates satisfactorily and safely, all applicable local and national codes that apply to installing and operating the equipment must be followed. Since codes can vary geographically and can change with time, it is the user's responsibility to identify and comply with the applicable standards and codes. **WARNING:** ***Failure to comply with applicable codes and standards can result in damage to equipment and/or serious injury to personnel.***

Personnel who are to install and operate the equipment should study this user guide and all referenced documentation prior to installation and/or operation of the equipment.

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

© Compumotor Division of Parker Hannifin Corporation, 1994, 1995
— All Rights Reserved —

Motion Architect is a registered trademark of Parker Hannifin Corporation.
CompuCAM and Servo Tuner are trademarks of Parker Hannifin Corporation.
Microsoft and MS-DOS are registered trademarks, and Windows is a trademark of Microsoft Coporation.

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

Since Parker Compumotor constantly strives to improve all of its products, we reserve the right to change this user guide and equipment mentioned therein at any time without notice.

Technical Assistance

Contact your local automation technology center, or ...

North America:
Compumotor Division of Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070
Fax: (707) 584-8015
BBS: (707) 584-4059

Europe:
Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
Telephone: 0202-690911
Fax: 0202-600820

Parker **Compumotor**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6270 Motion Controller User Guide

Revision B Change Summary

The following is a summary of the primary technical changes to this user guide since the last version was released. Most of the changes are due to product revision 3.0. This user guide, p/n 88-013913-01**B** (released in January 1995), supersedes 88-013913-01**A**.

Topic	Description	See Also
ANI Analog Input Voltage and Position Capture	**Change**: When the ANI analog input is captured, you can use either the voltage data or the position data. • Voltage: The captured ANI voltage is not affected by scaling, PSET, ANIPOL, or CMDDIR. The captured data (in units of volts) is available through the use of the TCA and CA commands. • Position: The captured ANI position is affected by scaling, PSET, ANIPOL, and CMDDIR. The captured position (counts or scaled units) is available through the use of the TPCA and PCA commands.	Pg. 74
Commanded Direction: *Polarity Reversal*	**New Command**: The CMDDIR command allows you to change the sign of the commanded direction. Changing the sign of the commanded direction also reverses the polarity of the feedback devices. This allows you to reverse the direction of motion without the need to (a) change the connections to the drive/valve and reverse the wiring or mounting of the feedback device, or (b) change the sign of all the motion-related commands in your program. ☞ HINT: Before changing commanded direction polarity, make sure there is a direct correlation between the commanded direction and the direction of the feedback source counts (i.e., a positive commanded direction from the 6270 must result in positive counts from the feedback device). This is required for servo stability. The CMDDIR command may not be executed while motion is in progress or while the drive/valve is enabled. The CMDDIR command is automatically saved in non-volatile memory.	Pg. 114
DAC Limits	**New Command**: In addition to the defining the maximum DAC voltage limit with the DACLIM command, you may now define a minimum DAC voltage limit with the DACMIN command. In so doing, you limit the DAC output to the range between DACMIN and DACLIM. You may not set the DACLIM value less than the DACMIN value. For example, if on both axes you wish to prevent motion in the negative direction and limit the maximum voltage to 8.5V, use the DACMINØ,Ø and DACLIM8.5,8.5 commands. The DACMINØ,Ø command sets the minimum DAC output to zero volts, which effectively prevents the 6270 from commanding motion in the negative direction. The DACLIM8.5,8.5 command sets the maximum DAC output to 8.5V. The DACMIN value tracks (but with a reverse sign) the value of DACLIM, until a DACMIN command is executed with a value other than the current DACLIM command value. After the DACMIN command is executed, DACMIN and DACLIM values must be entered separately. To re-establish the default mode where DACMIN tracks DACLIM, issue the DACMIN command with a minus sign (-) in the command field for the affected axis (e.g., DACMIN,- restores axis 2 to the default mode).	Pg. 25
End-of-Travel Hardware Limits: *Default Setting*	**Change**: The default before 6270 revision 3.0 was LHØ,Ø, which disables both end-of-travel hardware limits on both axes. In revision 3.0, the default setting is now changed to LH3,3, which enables the limits. To disable the limits, issue the LHØ,Ø command or connect the **CW** and **CCW** limit terminals to the **GND** terminal on the **LIMITS** connector.	Pg. 60

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

<table>
<tr><td>Feedback Source Selection: Inclusive Elements</td><td>Enhancement: Some additional commands have been made specific to the current feedback source (encoder, LDT or ANI) selected with the SFB command. When these commands are executed, they will apply only to the current feedback source. Then, when you switch feedback sources with the SFB command, the values of the items listed below will change to match the selected feedback source. (☞ denotes new inclusive elements)
SGP.....Proportional Gain
☞ SGPN.....Proportional Gain Negative
SGV.....Velocity Gain
☞ SGVN.....Velocity Gain Negative
SGI.....Integral Gain
☞ SGIN.....Integral Gain Negative
SGVF.....Velocity Feedforward Gain
☞ SGVFN.....Velocity Feedforward Gain Negative
SGAF.....Acceleration Feedforward Gain Negative
☞ SGAFN.....Acceleration Feedforward Gain Negative
☞ SGILIM.....Integral Windup limit
PSET.....Position Offset
☞ SMPER.....Maximum Position Error
☞ SOFFS.....Servo Control Signal Offset
☞ SOFFSN.....Servo Control Signal Offset Negative</td><td>Pg. 25</td></tr>
<tr><td>Feedback Source: Polarity Reversal</td><td>New Commands: Servo stability requires a direct correlation between the commanded direction and the direction of the feedback source counts (i.e., a positive commanded direction from the 6270 must result in positive counts from the feedback device).
If the feedback device is counting in the wrong direction, you may reverse the polarity with the command specific to the feedback source—ENCPOL for encoder feedback, LDTPOL for LDT feedback, ANIPOL for ANI feedback (see table). This allows you to reverse the counting direction without having to change the actual wiring or mounting of the feedback device. For example, if the LDT on axis 2 was installed incorrectly and counted in the wrong direction, you could issue the LDTPOLx1 command to reverse the polarity.
<table>
<tr><th>Source</th><th>Command</th><th>Syntax</th><th>Reverse Polarity</th><th>Restore to Default</th></tr>
<tr><td>Encoder</td><td>ENCPOL</td><td>ENCPOL</td><td>ENCPOL1</td><td>ENCPOLØ</td></tr>
<tr><td>LDT</td><td>LDTPOL</td><td>LDTPOL</td><td>LDTPOL11</td><td>LDTPOLØØ</td></tr>
<tr><td>ANI</td><td>ANIPOL</td><td>ANIPOL</td><td>ANIPOL11</td><td>ANIPOLØØ</td></tr>
</table>
When you change the feedback polarity, the affect on the position report differs (i.e., reverses the sign of the LDT and ANI report; reverses the direction of subsequent encoder counts). Changing the feedback polarity effectively invalidates the existing absolute zero position (PSET) setting; therefore, you will have to re-establish the PSET position.
The feedback polarity commands are automatically saved in non-volatile memory.</td><td>Pg. 44, 114</td></tr>
<tr><td>Following Capability</td><td>New Features: With release 3.0, the 6270 supports ratio following, position following, and advance & retard. User information for Following features is provided in the 6000 Series Following User Guide.</td><td>6000 Series Following User Guide</td></tr>
<tr><td>Gain Set Elements</td><td>Enhancement: When you issue the SGSET command to save a gain set, it not only saves the general gains (SGP, SGV, SGI, SGVF, and SGAF), but also saves these elements as well:
SGPN..... Proportional Gain Negative
SGVN..... Velocity Gain Negative
SGIN..... Integral Gain Negative
SGVFN ..... Velocity Feedforward Gain Negative
SGAFN ..... Acceleration Feedforward Gain Negative
SGILIM..... Integral Windup Limit
SOFFS ..... Servo Control Signal Offset
SOFFSN..... Servo Control Signal Offset Negative
SMPER ..... Maximum Allowable Position Error
The gain set saved with the SGSET command can be activated later with the SGENB command or the SSWG command.
In addition, the TGAIN and TSGSET transfer commands will report all these elements noted above.</td><td>Pg. 39</td></tr>
</table>

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LDT Gradient	**Clarification**: The LDT gradient may be changed with the LDTGRD command only of the drive is disabled with the DRIVE command.	Pg. 26
Maximum Velocity	**Change**: The maximum velocity is now 1,600,000 counts/sec.	Pg. 59, 105
Negative Control Signal Offset	**New Command**: The SOFFSN command allows you to set an offset voltage to be added to the DAC output when the DAC output is negative. The negative offset (SOFFSN) value changes with (tracks) the corresponding positive offset (SOFFS) value until the SOFFSN command is executed with a value other than the current SOFFS command value. After the SOFFSN command is executed, separate positive and negative offset commands must be used. To re-establish the default mode where SOFFSN tracks SOFFS, issue the SOFFSN command with a minus sign (-) in the command field for the affected axis (e.g., SOFFSN,- restores axis 2 to the default mode).	Pg. 41
Negative Gains	**New Commands**: The new negative gain commands (see list below) allow you to establish gains to be used when the position error is negative. SGPN.....Proportional Gain Negative SGVN.....Velocity Gain Negative SGIN.....Integral Gain Negative SGVFN.....Velocity Feedforward Gain Negative SGAFN.....Acceleration Feedforward Gain Negative Each negative gain changes with (tracks) the corresponding positive gain value (SGP, SGV, SGI, etc.) until a negative gain command is executed. For example, the SGPN value automatically tracks the SGP value until an SGPN command is executed with a value other than the current SGP command value. After the negative gain command is executed, separate positive and negative gain commands must be used. To re-establish the default mode where the negative gain tracks the positive gain, issue the negative gain command with a minus sign (-) in the command field for the affected axis (e.g., SGPN,- restores axis 2 to the default mode).	Pg. 39
RP240 Menus	**Enhancements**: Additional RP240 menus have been added: • Jog in open-loop mode • Display position error • View and edit tuning gains and gain sets • View numeric and string variables, and edit numeric variables	Pg. 84
Servo Tuner Instructions	**Change**: All instructions for tuning the 6270 using the Servo Tuner option for Motion Architect have been removed and are found in the ***Servo Tuner User Guide*** which is included in the Servo Tuner box.	
Servo Update Rates	**Change**: Some of the servo update rates have been changed.	Pg. 47
Setpoint Window Gains	**New Commands**: You may now activate a specified set of gains to be used when with a defined region either side of the position setpoint ("setpoint window"). A hysteresis of 25% is enabled when the window is entered. The new commands are: SSWD.....Setpoint Window Distance SSWG.....Setpoint Window Gain Set The SSWD command is used to specify the distance on both sides of the position setpoint ("setpoint window") in which the gain set specified with the SSWG command is used. The gain set is defined with the SGSET command (see *Gain Set Elements* note above).	Pg. 40

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

T A B L E O F C O N T E N T S

Overview

Assumptions i
Contents of This User Guide i
Installation Process Overview ii
Conventions ii
- Linear and Rotary Motion ii
- Units of Measure iii
- 6000 Series Commands iii
- Active High/Active Low Convention iii

Related Publications iii

Chapter 1. Introduction

6270 Description 1
- System Hardware Block Diagram 2

6270 Features 2

Chapter 2. Getting Started

Inspect The Shipment 3
- Pre-Wired Connections 3

Bench Test 4
- ① RS-232C Communications 4
- ② Connect Power Cable 5
- ③ Test Procedure 5

Chapter 3. Installation

Installation Precautions 7
- Heat & Humidity 7
- Electrical Noise 7
- Airborne Contaminants 7
- Follow Installation Procedure 8

① Mount the 6270 8
- Panel Layout 8

② System Connections 9
- Servo Valve and Servo Drive Connections 9
- End-of-Travel Limit Connections 14
- Home Limit Connections 14
- LDT Connections 15
- Encoder Connections 15
- Auxiliary +5V Output Connection 16
- Enable Input Connection 16
- Programmable Inputs & Outputs Connections 17
- Trigger Input Connections 18
- RP240 Remote Operator Panel Connections 19
- Joystick and Analog Input Connections 19
- ANI Analog Input Connections (6270-ANI Option Only) 21
- Extending 6270 System Cables 21

③ Installation Verification 22
④ What's Next? 24

Chapter 4. Servo Tuning

In a Hurry? 29
Servo System Terminology 30
- Servo Tuning Terminology 30
- Position Variable Terminology 31
- Servo Response Terminology 32

6000 Series Servo Commands 34
Servo Control Techniques 36
- Proportional Feedback Control (SGP) 36
- Integral Feedback Control (SGI) 37
- Velocity Feedback Control (SGV) 38
- Negative Gains 39

Gain Sets 39
- Setpoint Window Gains 40

Open Loop Operation 42
Tuning Setup Procedure 43
Drive Tuning Procedure (Velocity Motor Drives Only) 45
Controller Tuning Procedure 46
Tuning Scenario 51
Target Zone (Move Completion Criteria) 53

Chapter 5. Feature Implementation

Before You Proceed With This Chapter 55
- 6000 Series Software Reference Guide 56

Support Software 56
- 6000 DOS Support Disk 56
- Motion Architect 56

6270 Safety Features 57
Scaling 58
- Acceleration & Deceleration Scaling (SCLA/PSCLA) 58
- Velocity Scaling (SCLV/PSCLV) 59
- Distance Scaling (SCLD) 59
- Scaling Example 60

End-of-Travel Limits 60
Homing 61
Positioning Modes 64
- Preset Mode 64
- Continuous Mode 65

Dithering Hydraulic Valves 66
User Interface Options 67
Programmable Inputs and Outputs 68
- Output Functions 68
- Input Functions 71
- Thumbwheel Interface 76
- I/O Device Interface 78

Joystick Interface 79
6270-ANI Option Offers14-Bit Analog Input 82
RP240 Remote Operator Panel Interface 82
- Operator Interface Features 83
- Using the Default Mode 84

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 5 *(continued)*

Host Computer Operation 90
Variables 91
Converting Between Binary and Numeric Variables . 91
Performing Operations with Numeric Variables 91
Performing Operations with Binary Variables 93
Teach Mode 94
Teach Mode Basics 94
Summary of Related 6000 Series Commands 96
Teach Mode Application Example 96
S-Curve Profiling 99
X-Y Linear Interpolation 101
RS-232C Daisy-Chaining 102
Daisy-Chaining from a Computer or Terminal 104
Daisy-Chaining from a Master 6270 104
Daisy-Chaining and RP240s 104

Chapter 6. Hardware Reference

General Specifications 105
I/O Pin Outs & Circuit Drawings 106
Drive Connectors 106
Linear Displacement Transducer (LDT) Connectors 107
Encoder Connector (For Use With Incremental Encoder Only) 107
Programmable I/O Connectors 107
Auxiliary (AUX) Connector 108
Limits Connector 108
Joystick Connector 108
RP240 Connector 109
Optional DIP Switch & Jumper Settings 109
Accessing the DIP Switch Package & Jumpers 109

Chapter 7. Troubleshooting

Troubleshooting Basics 111
Reducing Electrical Noise 112
Error Messages and Debug Tools 112
Common Problems & Solutions 112
RS-232C Troubleshooting 114

Appendix A

Reducing Electrical Noise 115

Appendix B

Alphabetical Command List 117

Index 121

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

O V E R V I E W

This user guide is designed to help you install, develop, and maintain your system. This section is intended to help you find and use the information in this user guide.

Assumptions

To effectively use this user guide to install, develop, and maintain your system, you should have a fundamental understanding of the following:

- Hydraulic system components such as valves, cylinders, LDTs, etc.
- Electronics concepts such as voltage, switches, current, etc.
- Motion control concepts such as torque, velocity, distance, force, etc.
- Programming skills in a high-level language such as BASIC, Fortran, or Pascal

Contents of This User Guide

Chapter	Purpose
① *Introduction*	Describes the 6270 and provides a brief account of its features.
② *Getting Started*	Lists and describes the items you should have received with your 6270 shipment. A *bench test* procedure is provided to verify the system's basic functionality.
③ *Installation*	Provides instructions for mounting, wiring, and testing the 6270 system. Complete all instructions in Chapter 3 before tuning the 6270 in Chapter 4.
④ *Servo Tuning*	Provides an empirical method for tuning the 6270. Complete all tuning instructions before implementing motion features. If you are using Servo Tuner™, refer to the ***Servo Tuner User Guide*** for instructions.
⑤ *Feature Implementation*	Describes the 6270's user features and instructs you on how to implement them in your application. Sample applications are provided. The ***6000 Series Software Reference Guide*** provides programming guidelines and detailed descriptions of all the 6000 Series commands.
⑥ *Hardware Reference*	Use the *Hardware Reference* as a quick-reference tool for 6270 electrical specifications, optional DIP switch settings (address & baud rate), optional command output jumper settings, and I/O signal descriptions and circuit drawings.
⑦ *Troubleshooting*	Describes methods for isolating and resolving hardware and software problems.
Appendices	A: Reducing electrical noise in your application B: Alphabetical listing of the 6270's commands
Index	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installation Process Overview

Step 1 Review this entire user guide. Become familiar with the user guide's contents so that you can quickly find the information you need. At times you may want to refer to the ***6000 Series Software Reference Guide*** for programming tips and detailed descriptions of the 6000 Series commands used in this user guide.

Step 2 Read Chapter 1, *Introduction*, **and the user documentation for all peripheral system components** to develop a basic understanding of all system components, their functions, and interrelationships.

Step 3 Read Chapter 2, *Bench Test*, and verify that you have received all the proper components for your system, and that all the items in your shipment have arrived without damage. Follow the step-by-step bench test procedures to verify the 6270's basic operability, as well as the functionality of the host computer (or terminal).

Step 4 Complete the system configuration, mounting, wiring, and testing instructions provided in Chapter 3, *Installation*. **Do not deviate from the sequence or installation methods provided.**

Step 5 After you successfully complete all procedures in Chapter 3, you will be ready to proceed to Chapter 4, *Servo Tuning*, to tune the 6270 (and the drive if using one) for your application. The tuning procedures in Chapter 4 are based on using an empirical method to tune your system; if you are using Servo Tuner™, refer to the ***Servo Tuner User Guide*** for instructions.

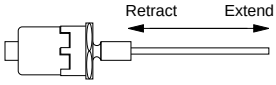
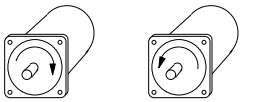
Step 6 After successfully completing all procedures in Chapter 4, you may proceed to Chapter 5 to implement the 6270's user features in your application. If you will be using the 6270's Following features, refer to the ***6000 Series Following User Guide***.

Conventions

Linear and Rotary Motion

The 6270 can be used in linear and rotary servo applications. The position feedback source can be an LDT (linear displacement transducer) or an incremental encoder. If you have the 6270-ANI option (equipped with two 14-bit, ±10V analog inputs), you can position to a voltage using the analog feedback.

The table below will help to clarify the conventions when referring to the direction of motion, depending on the application type and feedback source.

Application Type	Negative (-) Direction *	Positive (+) Direction *	Perspective
Linear (e.g., hydraulic valve & cylinder using LDT feedback)	Retract	Extend	
Rotary ** (e.g., rotary drive & motor using incremental encoder feedback)	Counter-clockwise (CCW)	Clockwise (CW)	

* For 6270-ANI owners using analog feedback, negative direction refers to negative voltage and positive direction refers to positive voltage.

** This convention is accurate only if you connect the incremental encoder as described in Chapter 3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When programming the 6270, the negative sign refers to retract or CCW or negative voltage. The positive sign refers to the extend or CW or positive voltage (e.g., the D+8 distance command indicates an 8-unit move in the extension/CW/positive voltage direction).

When referencing the 6270's software commands in the ***6000 Series Software Reference Guide***, you will find all direction references in rotary terms—*clockwise* (CW) and *counter-clockwise* (CCW).

Units of Measure

The table below provides the units of measure for distance, acceleration/deceleration, and velocity, based on the feedback source selected. These conventions hold true unless you use the scaling feature, in which case the units of measure are determined by the scaling factors (see *Scaling* section in Chapter 5 for details).

Feedback Device	Accel/Decel Units	Velocity Units	Distance Units
LDT	inches/sec^2	inches/sec	steps (counts)
Incremental Encoder	revolutions/sec^2	revolutions/sec	steps (counts)
ANI Input	volts/sec^2	volts/sec	steps (counts)

6000 Series Commands

The command language conventions are provided in the ***6000 Series Software Reference Guide***. Because some 6000 Series products have four-axis capability, the syntax of the example commands in the ***Reference Guide*** shows data fields for all four axes; ignore the third and fourth data fields when entering commands or reading status commands for the 6270.

Active High/Active Low Convention

Many people refer to a voltage level when referencing the state of inputs and outputs. The 6270 offers the ability to configure the active level of its inputs and outputs. The active state refers to the voltage level as set by the appropriate level command (see table below). The product defaults to an input/output level of zero volts as its active level (referred to as "active low"). Thus, a "1" will appear in a status command (e.g., TIN) referencing an input/output state when the voltage level is zero volts.

Input/Output	Active Level Command
Home inputs	HOMLVL
End-of-travel limit inputs	LHLVL
Programmable inputs (including TRG-A and TRG-B)	INLVL
Programmable outputs (including OUT-A and OUT-B)	OUTLVL

Related Publications

- ***6000 Series Software Reference* Guide,** Parker Hannifin Corporation, Compumotor Division; part number 88-012966-01
- ***6000 Series Following User Guide,*** Parker Hannifin Corporation, Compumotor Division; part number 88-014217-01
- ***Motion Architect User Guide,*** Parker Hannifin Corporation, Compumotor Division; part number 88-013056-01
- ***Servo Tuner User Guide,*** Parker Hannifin Corporation, Compumotor Division; part number 88-014249-01
- ***CompuCAM User Guide,*** Parker Hannifin Corporation, Compumotor Division; part number 88-013921-01
- Schram, Peter (editor). *The National Electric Code Handbook (Third Edition).* Quincy, MA: National Fire Protection Association

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ①

Introduction

6270 Description

The 6270 is a stand-alone, two-axis Motion Controller. The 6270 provides sophisticated two-axis control of any servo system driven by a voltage (±10V) or a current (±20mA, ±50mA, ±60mA, ±80mA, ±100mA, or ±150mA).

The 6270 implements a dual processor approach, comprising a microprocessor for executing high-level motion programs and a digital signal processor (DSP) for high-speed, sophisticated servo control.

The 6270 offers the flexibility of closing the positioning loop with three different types of feedback devices: linear displacement transducers (LDTs), an incremental encoder, or optional ±10V analog inputs (6270-ANI option).

Using the 6000 Series Programming Language, you can program the 6270 via a PC or a dumb terminal. User programs are stored in the 6270's battery-backed RAM.

The 6270 also provides operator interface capabilities when used with the Compumotor RP240 Remote Operator Panel.

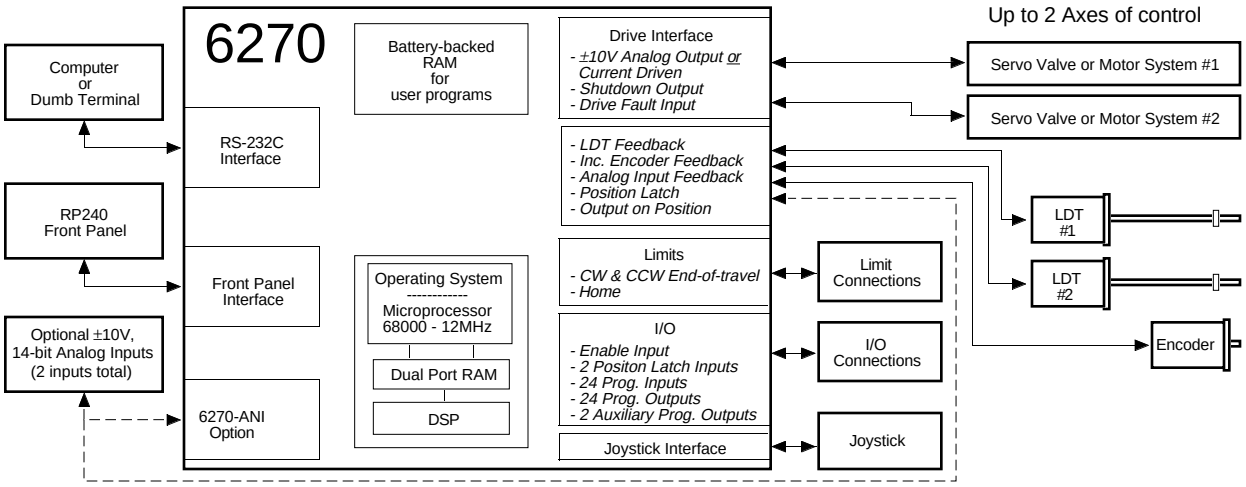
Support software for the Microsoft® Windows™ and DOS® operating environments comes standard with the 6270:

- *Motion Architect*® is an innovative, easy-to-use Microsoft® Windows™ based programming aide to help you easily create and implement complex motion programs. For more information, refer to the ***Motion Architect User Guide***. Special add-on modules are available:
  - Servo Tuner™ tuning aide allows you to visually gather data and tune your controller/drive system.
  - CompuCAM™ CAD-to-Motion (CAM) software allows you to translate DXF, HP-GL, and G-Code files into 6000 Series Language motion programs.
- The *6000 DOS Support Disk* contains a DOS-based program editor and terminal emulator package. Also included are sample 6000 Command Language programs.

Additional 6270 features are listed below in the *6270 Features* section.

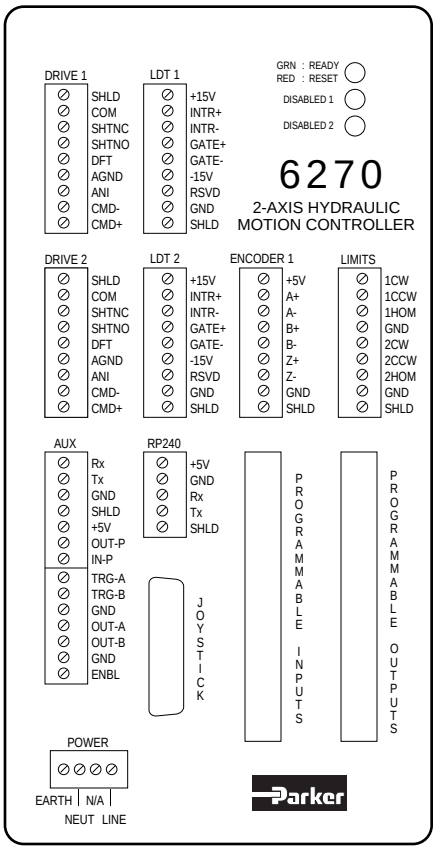
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

System Hardware Block Diagram



6270 Features

- 1 or 2 axes of control for current-driven or voltage-driven servo systems; feedback from linear displacement transducer (LDT) or incremental encoder feedback (or voltage feedback with ANI option)
- Controls electric servo drives in the velocity or torque mode
- Digital signal processor (DSP) for servo control (digital Proportional, Integral, and Velocity feedback, plus acceleration and velocity Feedforward—PIV&F)
- DOS Support Disk provided
- Motion Architect® for Windows™ provided
- Windows™-based visual data gathering and tuning aide available when using the Motion Architect® Servo Tuner™ option
- 40,000 bytes of non-volatile memory for storing programs; 150,000 bytes are available with the -M option
- Capability to interrupt program execution on error conditions
- Multi-axis teach mode
- S-curve motion profiling
- 2-axis linear interpolation
- Ratio following, position following, advance & retard (see ***6000 Series Following User Guide*** for details)
- Variable storage, conditional branching, and math capability
- Program debug tools — *single-step* and *trace* modes, breakpoints, and simulation of I/O
- Internal universal power supply
- Direct interface to RP240 Remote Operator Panel
- Operates stand-alone or interfaces to PCs & PLCs
- 3-wire, RS-232C interface to PC or dumb terminal
- I/O capabilities (all I/O are optically isolated):
  - ±10V analog control output (both axes)
  - Shutdown output for N.O. or N.C. relays (both axes)
  - Drive Fault input (both axes)
  - LDT input (both axes)
  - Incremental encoder input (axis 1 only)
  - CW & CCW end-of-travel limit inputs (both axes)
  - Home limit input (both axes)
  - 3 8-bit analog inputs for joystick control and variable input (0.0V - 2.5V)
  - 2 (trigger) inputs — use for hardware position latch
  - 24 programmable inputs (Opto-22™ compatible)
  - 24 programmable outputs (Opto-22™ compatible)
  - 2 auxiliary programmable outputs that can be configured for accurate output on position
- 6270-ANI Option offers two ±10V, 14-bit analog inputs; can be used for position feedback



6270 Front Panel

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ②

Getting Started

The information in this chapter will enable you to:

- Verify that each component of your 6270 system has been delivered safely and configured properly
- Bench test the 6270's power and RS-232C interface to the host computer/terminal

Inspect The Shipment

You should inspect your 6270 shipment upon receipt for obvious damage to its shipping container. Report any damage to the shipping company as soon as possible. The items listed below should be present and in good condition.

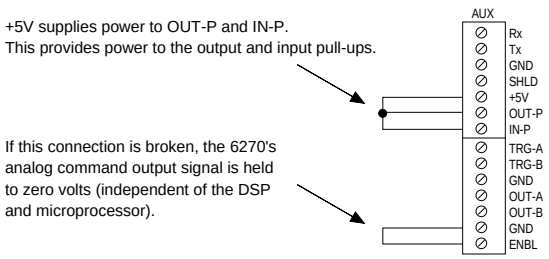
Part	
6270 main unit (w/ship kit)	
Ship kit:	*This user guide*
	6000 Series Software Reference Guide
	Motion Architect™ diskettes
	Motion Architect Servo Tuner diskette (optional)
	Motion Architect User Guide
	DOS Support Disk
	DOS Support Disk Quick Reference
	8-foot AC power cord

NOTE: The 6270-ANI is an optional version of the 6270 which provides two ±10V, 14-bit analog inputs.

If you ordered a -ANI option, check the serial tag on the side of the 6270's chassis; it should say **6270-ANI**.

Pre-Wired Connections

You should receive your 6270 with the following connections on the **AUX** connector prewired (from the factory):



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bench Test

This section leads you through step-by-step instructions to bench test your 6270 system. *This is a temporary (*bench top*) configuration; the permanent installation will be performed in Chapter 3,* Installation. In this section, you will complete the following tasks:

① RS-232C Communications
② Connect Power Cable
③ Test Procedure

① RS-232C Communications

To communicate with the 6270, your computer or terminal must have an RS-232C serial port.

The 6270 uses a three-wire implementation of standard EIA RS-232C signals.

Computer-to-Terminal Conversion

If you are using an IBM/compatible computer, you must use a terminal emulator software package to communicate with the 6270. The 6270 comes standard with Motion Architect™ for Windows™ and the 6000 DOS Support Disk; both provide a terminal emulator and program editor (refer to the ***Motion Architect User Guide*** or the ***6000 DOS Support Disk Quick Reference*** for installation and other user information). You may also use communication programs such as Crosstalk™, PC-Talk™, and Procomm™.

Set Communication Parameters

Make sure your computer or terminal is set to the following communication parameters. You can configure these parameters by using one of the terminal emulation software packages listed above in *Computer-to-Terminal Conversion*. If you are using Motion Architect™ or the 6000 DOS Support Disk, verify that the baud rate, data bit, parity, and stop bit parameters are set as follows:

- Baud Rate: 9600*
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Full Duplex
- XON/XOFF: Enabled

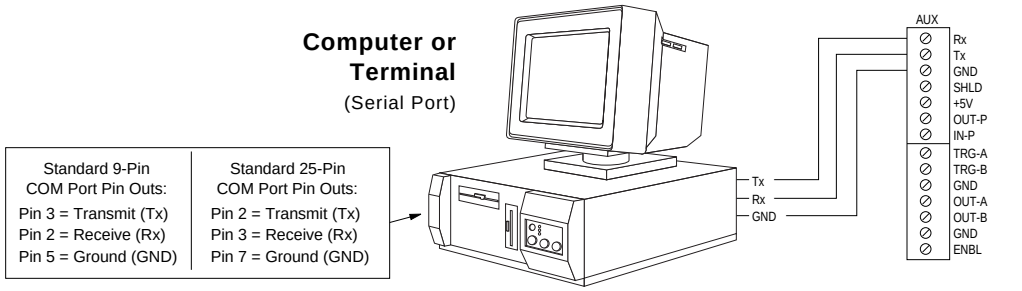
* If your terminal is not capable of 9600 baud, use the 6270's *auto-baud* function to automatically set the 6270's baud rate equal to the terminal's baud rate. Refer to *Optional DIP Switch Settings* in Chapter 6 for instructions.

Terminal Connections

The Receive Data (**Rx**), Transmit Data (**Tx**), and Ground (**GND**) signals are on the 6270's **AUX** connector (shown below). *The ground (***GND***) connection on the connector is signal ground or common as opposed to earth ground (***SHLD***).*

NOTE

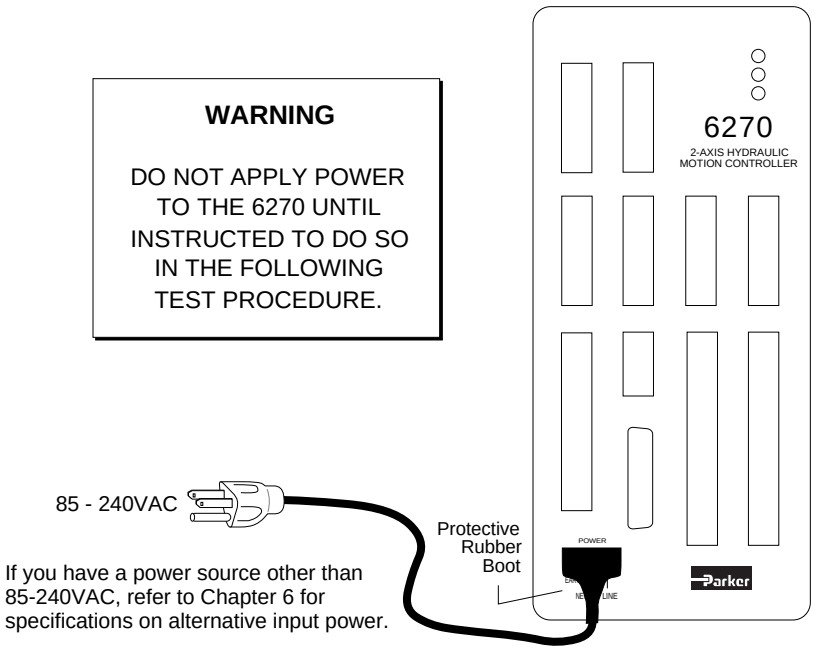
If you intend to daisy chain multiple 6270 Motion Controllers, **do not attempt the daisy-chain connections now**. Daisy-chain instructions are provided in Chapter 5, *Feature Implementation*.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

② Connect Power Cable

The 6270 is shipped with an 8-foot power cable that is prewired and keyed. Attach the power cable to the 6270's **POWER** connector as illustrated below.



③ Test Procedure

Use the following procedure to test the 6270's power and RS-232C connections. In Chapter 3, *Installation*, you will test the analog output, end-of-travel and home limits, LDTs, encoder, RP240, joystick, and programmable I/O.

Step 1 Apply power to the 6270 by plugging the power cable into a grounded power source.

> **CAUTION**
>
> The Earth ground connection must be made by plugging into a grounded receptacle or by physically connecting the green wire to earth ground.

Step 2 Watch the LEDs on the 6270. The **STATUS** LED should be green, indicating the 6270 is ready for operation. The other two LEDs (**DISABLED 1/2**) should be red because the valves or drives are not yet enabled with the `DRIVE11` command.

If the **STATUS** LED is red, or if none of the LEDs illuminate, check your power source and cable connections. If these connections seem correct, disconnect power and consult Chapter 7, *Troubleshooting*.

Step 3 If you are using the 6000 DOS Support Disk, go to the Set-up menu and move the cursor down to `CHECK OUT` and press `ENTER` to automatically verify the communication interface to the 6270.

If the interface is not successful (`Device not Ready` message will flash on the screen), refer to the RS-232C troubleshooting procedures in Chapter 7, *Troubleshooting*.

Step 4 Initiate the terminal emulator in Motion Architect or in the 6000 DOS Support Disk (refer to the ***Motion Architect User Guide*** or the ***6000 DOS Support Disk Quick Reference*** if necessary). You could also use your own terminal emulator package.

Press the RETURN key. The cursor should move down one or two lines each time you press the RETURN key. If the cursor does not move as described, refer to the RS-232C troubleshooting procedures in Chapter 7, *Troubleshooting*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ③

Installation

The information in this chapter will enable you to:

- Mount the 6270 properly
- Connect all inputs and outputs properly
- Verify that the complete system is installed properly

To ensure proper installation, you should perform all the bench test procedures in Chapter 2, Getting Started, ***before*** *proceeding with the permanent installation process in this chapter.*

Installation Precautions

To help ensure personal safety and long life of system components, pay special attention to the following installation precautions.

WARNING

Always remove power to the 6270 before performing wiring installation or changing DIP switch settings.

Heat & Humidity

Operate the 6270 system at an ambient temperature between 32° and 122°F (0° to 50°C). Keep the relative humidity below 95%.

Electrical Noise

Minimize the potential for electrical noise before installing the 6270, rather than attempting to solve such problems after installation. You can prevent electrical noise by observing the following installation precautions:

For more information on electrical noise, refer to Appendix A.

- Do not route high-voltage wires and low-level signals in the same conduit.
- Ensure that all components are properly grounded.
- Ensure that all wiring is properly shielded.

Airborne Contaminants

Contaminants that may come in contact with the 6270 should be carefully controlled. Particulate contaminants, especially electrically conductive material such as metal shavings, can damage the 6270.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

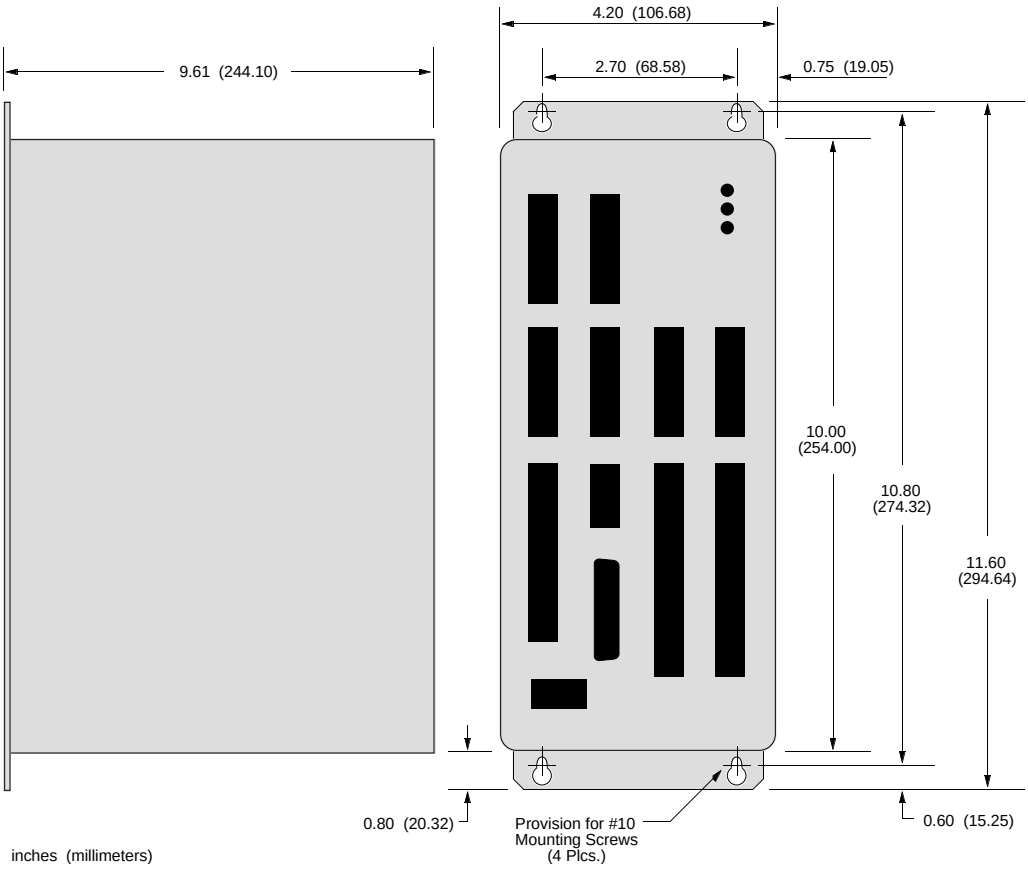
Follow Installation Procedure

To ensure proper installation of the 6270 system, this chapter is organized in logical, linear steps. *Deviating from this prescribed format may result in system problems.*

① Mount the 6270 Motion Controller
② Perform system connections
③ Perform the system test

① Mount the 6270

The 6270 should be installed in an enclosure that will protect it from atmospheric contaminants such as oil, metal, moisture, and dirt. Refer to the National Electrical Manufacturers Association (NEMA) specifications that pertain to your particular operating environment. The drawing below illustrates the 6270's dimensions.



Panel Layout

If you mount the 6270 in an enclosure with other equipment, be sure to maintain at least 2 inches of unrestricted air-flow space around the chassis. The maximum allowable ambient temperature directly below the 6270 is 122°F (50°C). Fan cooling may be necessary if adequate air flow is not provided.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

② System Connections

This section describes procedures for the following 6270 system connections:

- Servo valves and drives
- End-of-travel and home limits
- Linear displacement transducers (LDTs)
- Encoder
- Auxiliary +5VDC output
- Enable input (**ENBL**)
- Programmable inputs and outputs (including auxiliary outputs **OUT-A** and **OUT-B**)
- Output pull-up (**OUT-P**)
- Trigger inputs (**TRG-A** and **TRG-B**)
- RP240 Remote Operator Panel
- Joystick and analog inputs
- ANI analog inputs (6270-ANI option only)
- Extending cables

Refer to the bench test procedures in Chapter 2 for the following connections:

- Power
- RS-232C communications

Refer to Chapter 5 for connection procedures on the following:

- PLC
- Thumbwheels
- RS-232C daisy-chain

NOTE

Refer to Chapter 6, *Hardware Reference*, for system specifications and detailed I/O circuit drawings and signal descriptions. Active High and Active Low conventions are described in the *Overview,* page iii.

Servo Valve and Servo Drive Connections

CAUTION

Before connecting to your motor/drive or valve system, be sure that power is **not** applied to the 6270.

Pin Outs

The following table describes the pin outs for the **DRIVE** connectors. The **DRIVE** connectors are used for drive connections and for valve connections.

Pin #	Name	In/Out	**Description** (I/O circuit drawings are provided in Chapter 6)
1	SHLD	-----	Shield—internally connect to chassis (earth) ground.
2	COM	-----	Signal common for shutdown. Not connected to any ground or other COM.
3	SHTNC	OUT	Shutdown relay output to drives that require a closed contact to disable the drive. The shutdown relay is active (disabling the drive) when no power is applied to the 6270. When the 6270 is powered up, the shutdown relay remains active until you issue the `DRIVE11` command. Shutdown active (`DRIVEØØ`): this output is internally connected to COM. Shutdown inactive (`DRIVE11`): this output is disconnected from COM.
4	SHTNO	OUT	Shutdown relay output to drives that require an open contact to disable the drive. The shutdown relay is active (disabling the drive) when no power is applied to the 6270. When the 6270 is powered up, the shutdown relay remains active until you issue the `DRIVE11` command. Shutdown active (`DRIVEØØ`): this output is disconnected from COM. Shutdown inactive (`DRIVE11`): this output is internally connected to COM.
5	DFT	IN	Drive fault input. Set active level with the `DRFLVL` command.
6	AGND	-----	Analog ground.
7	ANI	IN	±10V, 14-Bit analog input **(available only with the 6270-ANI option)**.
8	CMD-	OUT	Command signal return.
9	CMD+	OUT	Command output signal (±10V is default; for optional current settings, see Ch. 6).

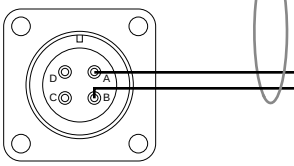
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

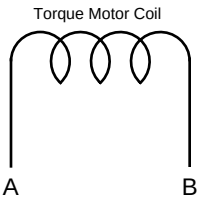
If you are controlling a valve...

The 6270 can control servo valves operating from a voltage of ±10V or from a current of ±20mA, ±50mA, ±60mA, ±80mA, ±100mA or ±150mA. The default setting is for ±10V operation. If you need to select one of the current settings refer to the instructions found in the *Optional DIP Switch and Jumper Settings* section of Chapter 6, *Hardware Reference*.

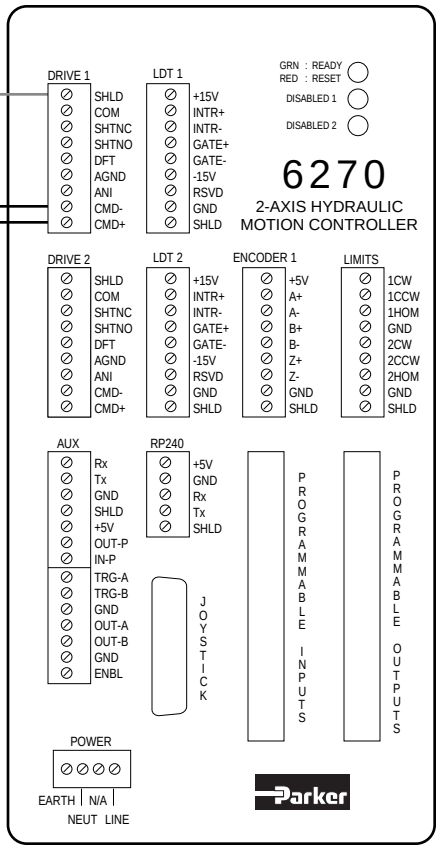
Connect the valve to the 6270's **DRIVE** connector. The illustration below shows connections to Parker Fluidpower's BD15 and BD30 valves.

Electrical Connector for Parker Fluidpower's BD15 and BD30 Valves





If you wish to reverse the valve's operation, swap the connections between pin A and pin B.



If you are using a servo drive...

Before you connect the drives to the 6270, configure your drives and connect the motors according to the user documentation for your drives.

The 6270 provides a standard ±10V analog control signal for use with any servo drive. Connect the drive to the 6270's 9-pin screw terminal connector (**DRIVE 1** or **DRIVE 2**) as illustrated below.

<<WARNING>> **SAFETY FIRST** **<<WARNING>>**

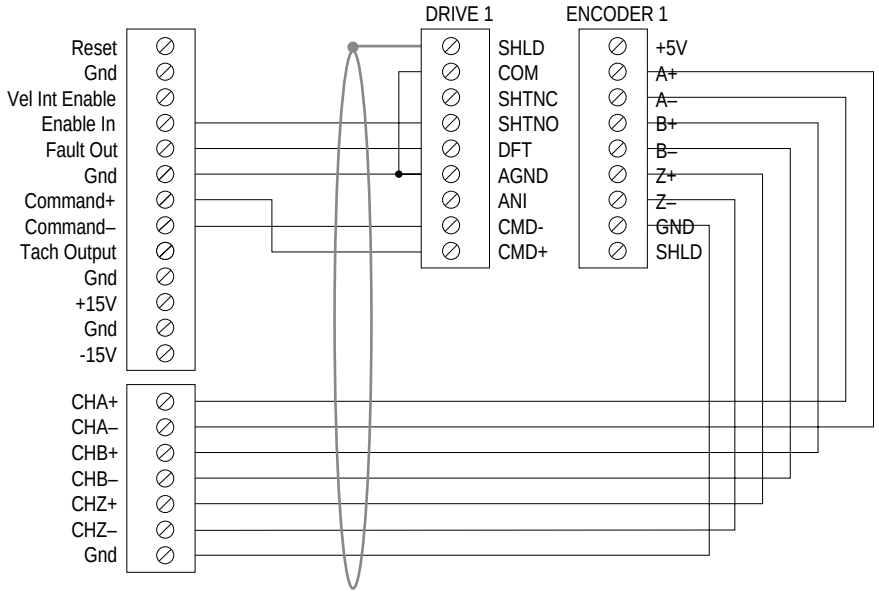
If your drive does not have a shutdown input, install a manual emergency-stop switch for the drive's power supply.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APEX Series Drive — 6270

APEX Series Drive		6270
Enable In	↔	SHTNO
Fault Out	↔	DFT
Gnd	↔	AGND
Command+	↔	CMD+
Command–	↔	CMD–
CHA+	↔	A–
CHA–	↔	A+
CHB+	↔	B+
CHB–	↔	B–
CHZ+	↔	Z+
CHZ–	↔	Z–
Gnd	↔	GND

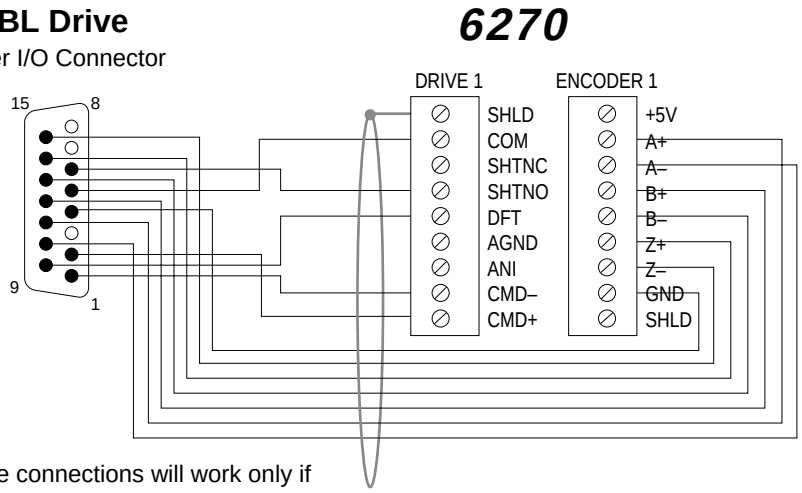
NOTE:
Apex Series CHA+ connected to 6270's A–
Apex Series CHA– connected to 6270's A+



BL Drive — 6270

User I/O Connector

BL Drive		6270
V2 (pin 1)	↔	CMD–
V1 (pin 2)	↔	CMD+
GND (pin 4)	↔	GND
$\overline{\text{RST}}$ (pin 5)	↔	COM
+15V (pin 6)	↔	SHTNO
$\overline{\text{FT}}$ (pin 9)	↔	DFT
AOP (pin 10)	↔	A–
$\overline{\text{AOP}}$ (pin 11)	↔	A+
BOP (pin 12)	↔	B+
$\overline{\text{BOP}}$ (pin 13)	↔	B–
ZOP (pin 14)	↔	Z+
$\overline{\text{ZOP}}$ (pin 15)	↔	Z–



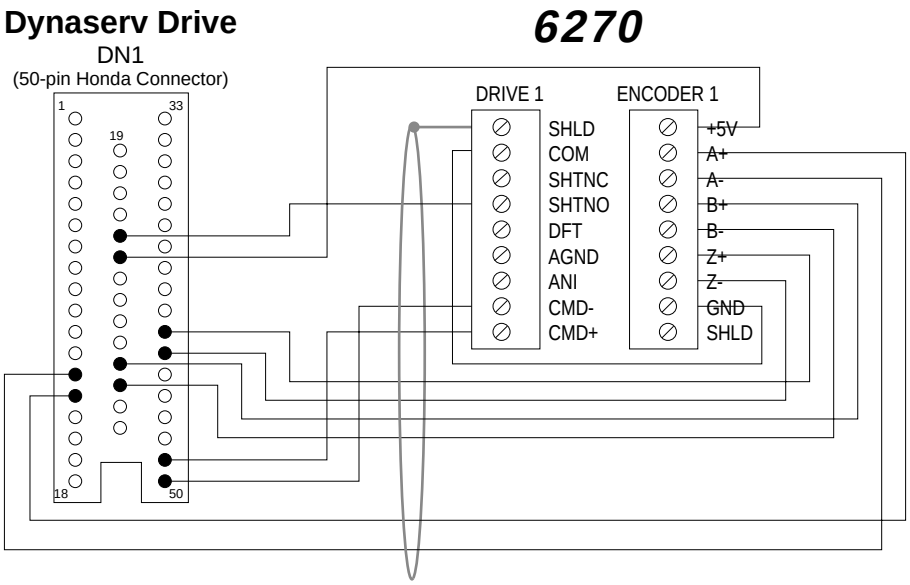
NOTE: These connections will work only if BL jumper LK2 is set to position B (not the factory default position).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Dynaserv Drive

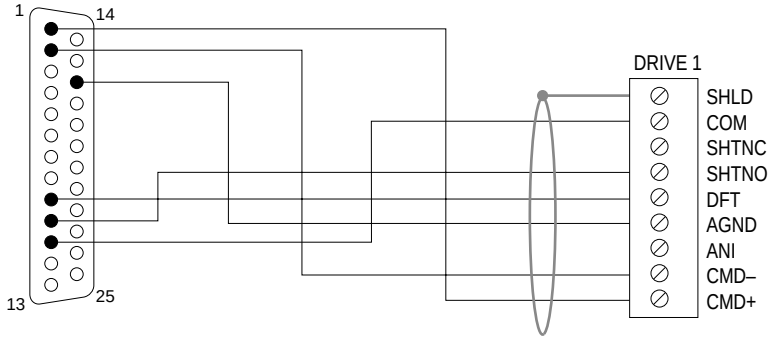
Dynaserv Drive		6270
A+ (pin 13)	↔	A–
A– (pin 14)	↔	A+
SRVON (pin 23)	↔	SHTNO
Vcc (pin 24)	↔	+5V
B+ (pin 29)	↔	B+
B– (pin 30)	↔	B–
Z+ (pin 43)	↔	Z+
Z– (pin 44)	↔	Z–
VIN (pin 49)	↔	CMD+
AGND (pin 50)	↔	CMD–

NOTE:
Dynaserv A+ connected to 6270's A–
Dynaserv A– connected to 6270's A+
6270 GND connected to 6270 COM



OEM670 Drive

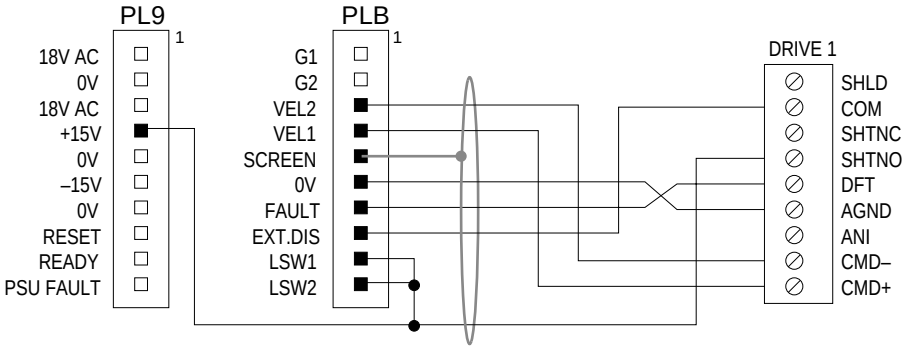
OEM670 Drive		6270
CMD+ (pin 1)	↔	CMD+
CMD– (pin 2)	↔	CMD–
FAULT (pin 9)	↔	DFT
ENABLE (pin 10)	↔	SHTNO
GND (pin 11)	↔	COM
GND (pin 16)	↔	AGND



UD2 & UD5 Drives
(UR3, UR4 or UR8 Rack)

UD2 & UD5 Drives		6270
+15V, LSW1 & LSW2	↔	SHTNO
VEL2	↔	CMD–
VEL1	↔	CMD+
0V	↔	AGND
FAULT	↔	DFT
EXT.DIS	↔	COM

NOTE: These connections will work only if UD2/5 jumper LK1 is set to the 0V position (not the factory default position).



If a drive fault occurs, you must cycle power to the drives, unless you control RESET (PL9 pin 8 on UR4 & UR8 racks, PL4 pin 8 on UR3 rack) with one of the 6270's general-purpose outputs. For additional instructions on detecting and reacting to UD rack faults, contact the Compumotor or Digiplan Applications Department.

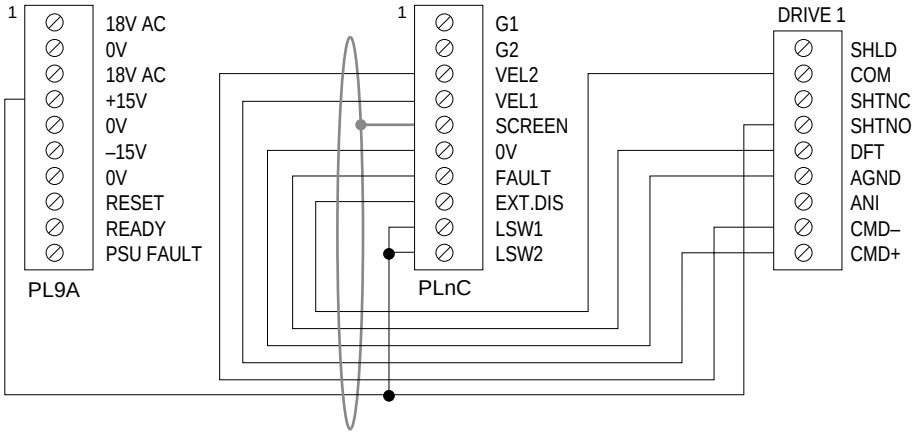
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

UD12 Drive

6270

UD12 Drive		6270
+15V, LSW & LSW2	↔	SHTNO
VEL2	↔	CMD–
VEL1	↔	CMD+
0V	↔	AGND
FAULT	↔	DFT
EXT.DIS	↔	SHLD

NOTE: These connections will work only if UD12 jumper LK3 is set to position A (not the factory default position).



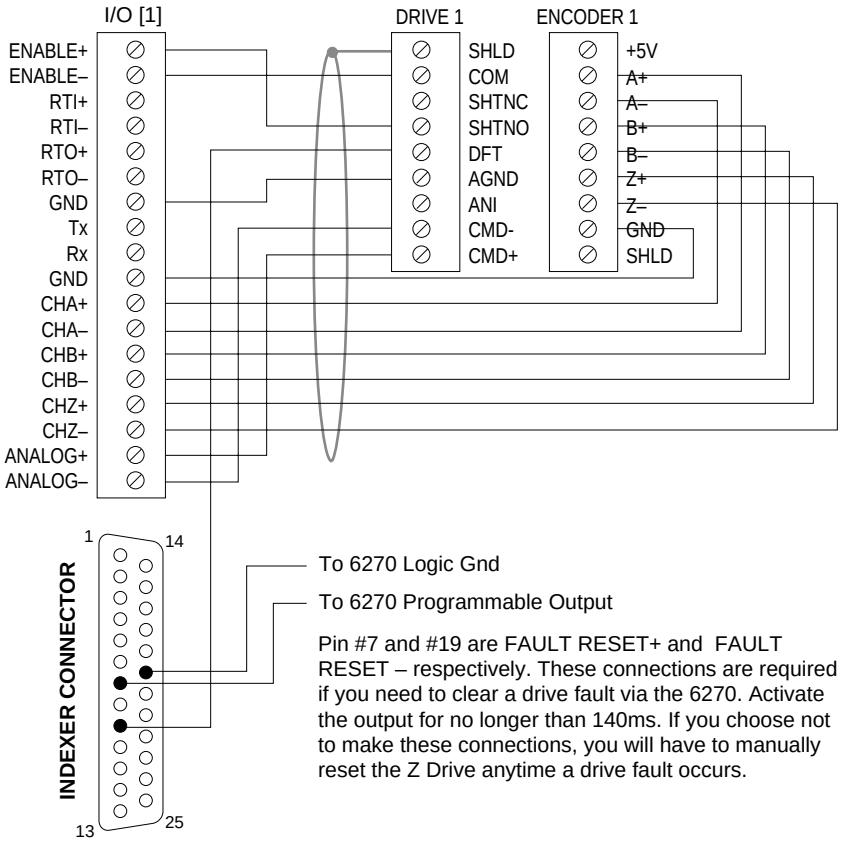
If a drive fault occurs, you must cycle power to the drives, unless you control RESET (pin 8 on the PL9 connector) with one of the 6270's general-purpose outputs.
For additional instructions on detecting and reacting to UD rack faults, contact the Compumotor or Digiplan Applications Department.

Z Drive

6270

Z Drive		6270
ENABLE+	↔	SHTNO
ENABLE–	↔	COM
GND	↔	AGND
GND	↔	GND
CHA+	↔	A–
CHA–	↔	A+
CHB+	↔	B+
CHB–	↔	B–
CHZ+	↔	Z+
CHZ–	↔	Z–
ANALOG+	↔	CMD+
ANALOG–	↔	CMD–
Indexer Connector		
DRIVE FAULT (pin 9)	↔	DFT

NOTE:
Z Drive CHA+ connected to 6270's A–
Z Drive CHA– connected to 6270's A+



Pin #7 and #19 are FAULT RESET+ and FAULT RESET – respectively. These connections are required if you need to clear a drive fault via the 6270. Activate the output for no longer than 140ms. If you choose not to make these connections, you will have to manually reset the Z Drive anytime a drive fault occurs.

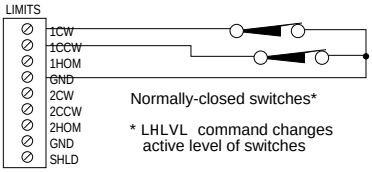
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

End-of-Travel Limit Connections

The 6270 provides end-of-travel limit inputs for both axes via the **LIMITS** connector. End-of-travel inputs serve as safety stops that prevent the load from crashing into mechanical stops and damaging equipment or injuring personnel. The drawing below illustrates typical end-of-travel limit switch connections.

In linear applications, the position direction (CW) is assumed to correspond to extension, and the negative direction (CCW) to retraction. In rotary applications, the CW or CCW direction is determined either by the direction the motor shaft, or by the sign (+ or -) of the commanded position (e.g., the D+8ØØØ distance command indicates a 8,000-unit move in the clockwise direction). *This convention is accurate only if you connect the drive or valve as described earlier in this chapter.*

End-of-Travel Limits



NOTE

Motion will not occur until you do one of the following:

- Install limit switches
- Disable the limits with the LH command
- Change the active level of the limits with the LHLVL command

Use of hardware (and software) end-of-travel limits is discussed in detail in the End-of-Travel Limits *section in Chapter 5.*

Mount normally-closed switches such that the load forces them to open before it reaches the physical travel limit **(leave enough room for the load to stop)**. When the load opens the limit switch, the servo mechanism comes to a halt. The actual stopping distance depends on load's speed and the Hard Limit Deceleration (LHADA and/or LHAD) setting. The servo mechanism will not be able to move in that same direction until you clear the limit (close the switch) **and** execute a move in the opposite direction (or you can disable the limits with the LH command, but this is recommended only if the load is not attached to the servo mechanism). Use the TLIM or TAS commands to check the status of the limit switches.

<< CAUTION >> **RUNAWAY** **<< CAUTION >>**

If a *runaway* occurs (servo mechanism starts moving, usually at the fastest possible velocity, due to servo instability), the 6270 will shut down the drive if the maximum position error (set with the SMPER command) is exceeded **before** an end-of-travel limit (either hardware of software) is encountered. However, if the maximum position error is **not** exceeded by the time the limit is encountered, the 6270 **may not be able to stop the servo mechanism**.

Home Limit Connections

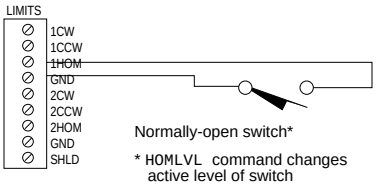
Use the Home input to establish a *home* position or zero position reference point. The home input (TTL compatible) is used for homing the servo mechanism. If you are using encoder feedback, the encoder's Z channel pulse can be used in conjunction with the home switch to determine the home position. To use the encoder's Z channel, the HOMZ command must be enabled.

Homing is discussed in detail in the Homing *section in Chapter 5.*

The 6270 is shipped configured for use with normally-open home switches. You can, if you wish, reverse the home input polarity (to use normally-closed switches) with the HOMLVL command. The most common way to use the *home* switch is to mount it at a *home reference position*. The drawing below illustrates typical home limit switch connections to the 6270.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Home Limit



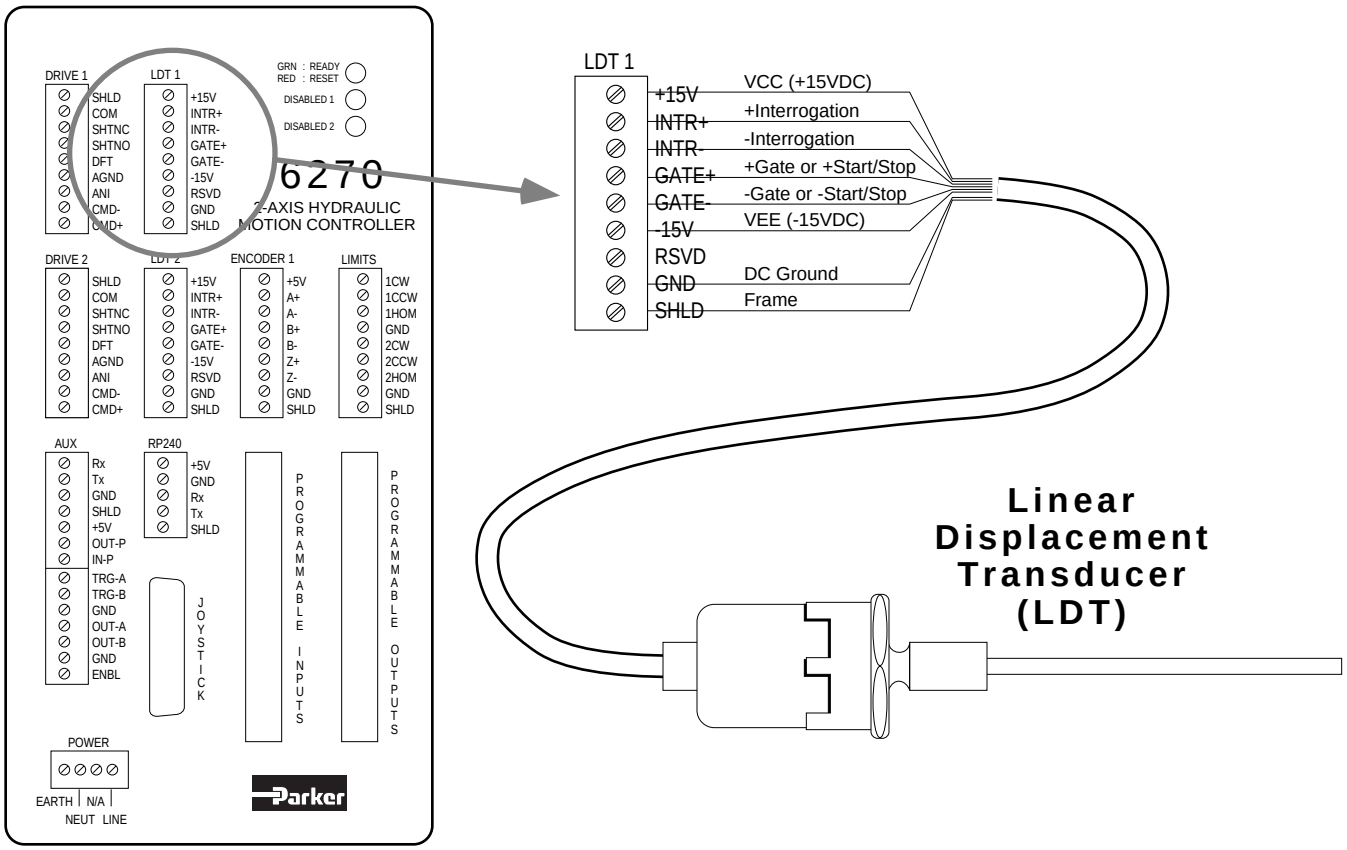
CAUTION

Proper homing performance cannot be guaranteed when the home and end-of-travel limit inputs are tied together.

LDT Connections

The 6270 supports pulse-feedback duration LDTs and start-stop LDTs. A jumper inside the 6270 determines which type of LDT feedback is supported (default is pulse-feedback duration LDTs). If you will be using one or more start-stop LDTs, change the jumper settings as instructed in the *Optional DIP Switch and Jumper Settings* section of Chapter 6, *Hardware Reference*.

Connect the LDT(s) to the 6270 as illustrated below.



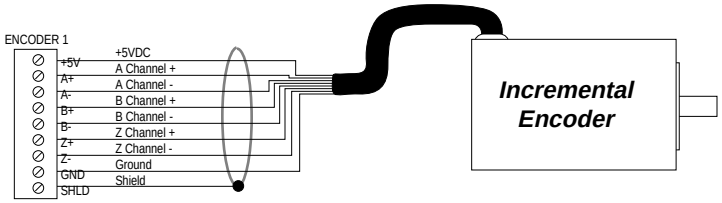
Encoder Connections

The 6270 supports one incremental encoder **and it can be used as a feedback device only for axis 1**. The incremental encoder must meet the following requirements:

- Use an incremental encoder with two-phase quadrature output. An index or *Z channel* output is optional. **Differential outputs are recommended.**
- It must be a 5V encoder to use the 6270's **+5V** output. Otherwise, it must be separately powered, with TTL-compatible or open-collector outputs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The illustration below shows the wiring techniques that you must use to connect an encoder to the 6270. Refer to Chapter 6 for the 6270's encoder input circuit drawing. *If you are using an encoder with a BL or Dynaserv drive, refer to the connection illustrations earlier in the* Servo Drive Connections *section.*



Note for Using Single-Ended Encoders

If you are using a single-ended encoder leave the 6270's A-, B-, and Z- terminals not connected.

Encoder Pin Outs

Pin out descriptions for the **ENCODER** connector (9-pin phoenix) are as follows.

Pin	In/Out	Name	Compumotor E Series Encoder Cable Colors	Description
9	OUT	+5V	Red	+5VDC output to power the encoder
8	IN	A Channel +	Brown	A+ channel quadrature signal from encoder
7	IN	A Channel -	Brown/White	A- channel quadrature signal from encoder
6	IN	B Channel +	Green	B+ channel quadrature signal from encoder
5	IN	B Channel -	Green/White	B- channel quadrature signal from encoder
4	IN	Z Channel +	Orange	Z+ channel quadrature signal from encoder
3	IN	Z Channel -	Orange/White	Z- channel quadrature signal from encoder
2	-----	Ground	Black	Isolated logic ground
1	-----	Shield	Shield	Internally connected to chassis ground (earth)

Auxiliary +5V Output Connection

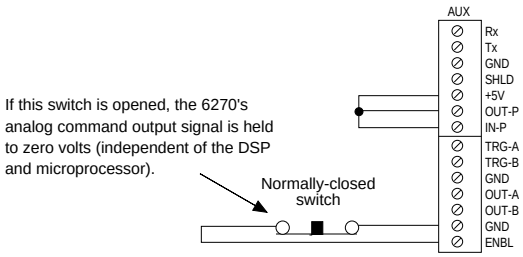
The 6270 provides +5VDC output on the **AUX**, **ENCODER**, and **RP240** connectors. As much as 1.5A is available. 1.5A is sufficient power for the total load on all the I/O connectors. For example, using an encoder drawing 250mA and one RP240 drawing 100mA, 1.15A would be left for other purposes. The drawing below illustrates example connections for powering the output pull-up.

Enable Input Connection

The **ENBL** (enable) input is located on the **AUX** connector. The 6270 is shipped with **ENBL** wired to **GND** to allow motion.

See the illustration below for an example connection using a normally-closed switch. Opening the switch changes the ±10V analog command output to zero volts and activates the shutdown outputs; this is done independent of microprocessor and DSP control, and, if error bit #9 of the ERROR command is enabled, the error program (ERRORP) will be executed. If motion is commanded while **ENBL** in not grounded the error message "WARNING: ENABLE INPUT INACTIVE" will be displayed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

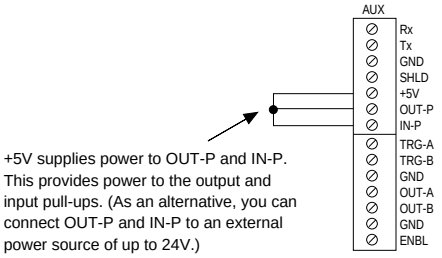


The encoder's position is retained when the **ENBL** input is activated. If you enable ERROR bit #9, you can check the status of the **ENBL** input with the TINO, INO, TER and ER commands.

Programmable Inputs & Outputs Connections

The **PROGRAMMABLE INPUTS** connector provides 24 general-purpose programmable inputs and the **PROGRAMMABLE OUTPUTS** connector provides 24 general-purpose programmable outputs. Two additional (and functionally identical) programmable outputs, **OUT-A** and **OUT-B**, are available on the **AUX** connector. Two additional trigger (*position latch*) inputs are available on the **AUX** connector, but due to their functional differences they are discussed later in the *Triggers* section. All these inputs and outputs are optically isolated and TTL compatible.

When the 6270 is shipped from the factory, all 26 programmable outputs and the 24 general-purpose programmable inputs are pulled up to +5VDC using the pull-up (**OUT-P** and **IN-P**) terminals on the **AUX** connector (see illustration below). If +5VDC logic is not to be used, disconnect the pull-up terminal(s) from the **+5V** terminal and use an external power source of up to 24VDC. The trigger inputs (**TRG-A** & **TRG-B**) are internally tied to 5V, but can have up to 24V connected to them. *Even if you use a 24V supply, the switching thresholds remain at the TTL level (≤0.4V = Low, ≥2.4V = High).*



Changing inputs from sourcing to sinking.

If you wish to have the 24 general-purpose programmable inputs sink current instead of source current, you can connect **IN-P** to **GND**.

These I/O are typically used with normally-open or normally-closed switches; however, they can also be used with I/O module racks, PLCs, and thumbwheels (including the Compumotor TM8).

If you are using PLCs or thumbwheels, refer to the connection instructions and application considerations provided in the *Programmable Inputs and Outputs* section of Chapter 5.

Also provided in the *Programmable Inputs and Outputs* section are instructions for defining and controlling programmable inputs and outputs via programs written with the 6000 Series programming language.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Programmable I/O Pin Outs

The following table lists the pin outs on the two 50-pin flat cable headers labeled **PROGRAMMABLE INPUTS** and **PROGRAMMABLE OUTPUTS**. Refer to Chapter 6, *Hardware Reference*, for internal I/O schematics.

PROGRAMMABLE INPUTS Connector				PROGRAMMABLE OUTPUTS Connector			
Pin #	**Function**	**Pin #**	**Function**	**Pin #**	**Function**	**Pin #**	**Function**
49	+5 VDC	23	Input #13	49	+5 VDC	23	Output #13
47	Input #1 (LSB)	21	Input #14	47	Output #1 (LSB)	21	Output #14
45	Input #2	19	Input #15	45	Output #2	19	Output #15
43	Input #3	17	Input #16	43	Output #3	17	Output #16
41	Input #4	15	Input #17	41	Output #4	15	Output #17
39	Input #5	13	Input #18	39	Output #5	13	Output #18
37	Input #6	11	Input #19	37	Output #6	11	Output #19
35	Input #7	09	Input #20	35	Output #7	09	Output #20
33	Input #8	07	Input #21	33	Output #8	07	Output #21
31	Input #9	05	Input #22	31	Output #9	05	Output #22
29	Input #10	03	Input #23	29	Output #10	03	Output #23
27	Input #11	01	Input #24 (MSB)	27	Output #11	01	Output #24 (MSB)
25	Input #12			25	Output #12		

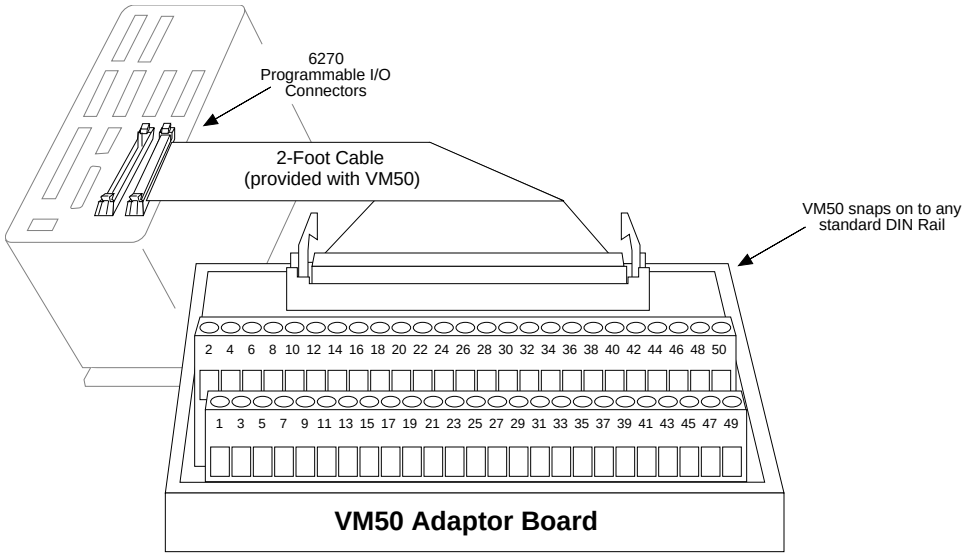
NOTE: All even-numbered pins are connected to logic ground (DC ground).

Optional VM50 Adaptor

If you wish to use screw terminal connections for the 24 programmable I/O, Parker Compumotor offers the VM50 adaptor (p/n VM50). If you wish to use screw terminal connections for both the 24 inputs **and** the 24 outputs, you will need two VM50 adaptors.

The pin numbers on the VM50's screw terminals correspond to the same pin outs on the **PROGRAMMABLE INPUTS** and **PROGRAMMABLE OUTPUTS** connectors. The VM50 simply attaches to the 6270 via the 2-foot, 50-pin ribbon cable that comes with the VM50 (see drawing below).

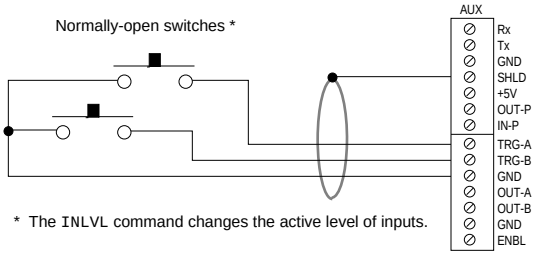
To order the VM50, contact your distributor or ATC.



Trigger Input Connections

The 6270 provides two trigger (*position latch*) inputs. Like the programmable inputs described earlier, trigger inputs can be connected to PLC outputs, discrete switches, or electronic sensors, and are monitored under program control. The status of triggers A and B is represented respectively by bits 25 and 26 in the `IN`, `INFNC`, `INLVL`, `ONIN`, and `TIN` commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Programming Tip:

Using the WAIT command, the 6270 can be programmed to wait until one or more trigger inputs switch to a desired state before executing the next command.

Position Latch Feature

The trigger inputs function identically to the regular 24 programmable inputs, except when they are programmed with the Trigger Interrupt Function (INFNCi-H) command to function as position latch inputs.

When configured as position latch inputs, the input enable/disable (INEN) command has no effect on the trigger inputs. The position latch feature is discussed in the *Programmable Inputs and Outputs* section in Chapter 5.

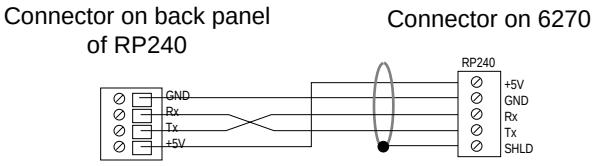
RP240 Remote Operator Panel Connections (RP240 is optional)

Refer to the RP240 Remote Operator Panel Interface *section in Chapter 5 for additional user information.*

Using a four-wire shielded cable, connect the RP240 to the 6270's **RP240** connector (see below). For cable lengths up to 50 feet, use 20 AWG wire (*cable lengths longer than 50 feet are not recommended*). Refer to the ***RP240 User Guide*** for mounting instructions.

NOTE

For the 6270 to recognize the RP240, the RP240 connection must be made prior to powering up (or resetting) the 6270. If you **connect the RP240 to the 6270 before powering up the 6270**, the 6270 will recognize the RP240 and send the *RP24Ø CONNECTED message to the RS-232C terminal. If the 6270 does not detect a RP240 upon power up or reset, then the following message will be sent to the RS-232C terminal: *NO REMOTE PANEL.



HINT: If you are not using an RP240, you can send text strings to another RS-232C device (using the DWRITE command).

Joystick and Analog Input Connections

You can use the three analog inputs on the **JOYSTICK** connector for 2-axis joystick control of the axes, and/or as a low-resolution analog input (8-bit A/D, 1mV/bit) for process control.

The Daedal JS6000 joystick is compatible with the 6270. To order the JS6000, contact Daedal at (800) 245-6903 or contact your local distributor.

Refer to Chapter 5 for a detailed discussion of joystick control.

The input range of the analog input is 0V to 2.5V. A joystick with a linear taper 5KΩ potentiometer (pot) with 60° of travel is recommended *(the pot has 300° of travel, but typically only 60° is usable with a joystick).* The pot should be adjusted so that its resistance is close to 0Ω when the joystick is all the way to one side, and about 1KΩ when the joystick is all the way to the other side. Also, connect a 1KΩ resistor between the analog input and +5V.

Joystick Connector Pin Outs

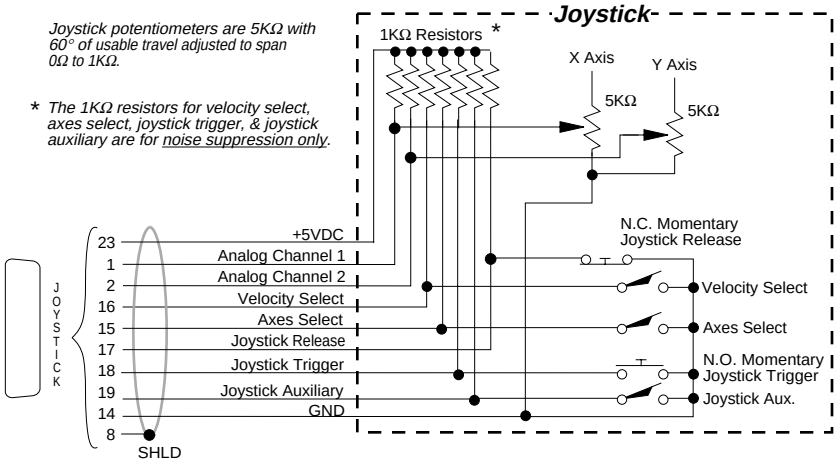
The **JOYSTICK** connector is a 25-pin D connector. The pin-out descriptions are provided in the table below. The 6270's internal analog input circuit diagram is provided in Chapter 6, *Hardware Reference*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pin	In/Out	Name	Description
1	IN	Analog Channel 1	8-bit analog input for joystick control of axis (can override with the ANVOEN and ANVO commands)
2	IN	Analog Channel 2	8-bit analog input for joystick control of axis (can override with the ANVOEN and ANVO commands)
3	IN	Analog Channel 3	8-bit analog input for joystick control of axis (can override with the ANVOEN and ANVO commands)
4	—	Unused	----------
8	—	Shield	Shield
14	—	Ground	Ground
15	IN	Axes Select	If only using one analog input, you can use this input to alternately control axes 1 or 2
16	IN	Velocity Select	Input to select high or low velocity range (as defined with JOYVH or JOYVL command)
17	IN	Joystick Release	Input to release the 6270 from joystick mode (JOY). Same as issuing the !JOYØØ command. Program execution will continue with the first statement after the joystick enable (JOY1) command.
18	IN	Joystick Trigger	Status of this active-low input can be read by a program (using the INO or TINO commands) to control program flow, or to enter the 6270 into joystick mode.
19	IN	Joystick Auxiliary	Status of this active-low input can be read by a program (using the INO or TINO commands) to control program flow, or to teach positions to a program.
23	OUT	+5VDC (out)	+5VDC power output

Analog Inputs

You can use the analog inputs for joystick control of the axes. An analog input can command an axis velocity from full CW to full CCW. The following drawing illustrates a typical joystick connection example.



Axes Select Input

You can define two configurations (JOYAXH and JOYAXL) that define which axes are controlled by which channels. The axes select input allows you to select the current configuration. An axes select input *high* references the JOYAXH command. An axes select input *low* references the JOYAXL command.

One possible configuration is as follows: With axes select input high, analog channel #1 controls axis one and analog channel #2 controls axis two (JOYAXH1,2). With axes select input low, analog channel #3 controls both axes (JOYAXL3,3).

Velocity Select Input

This input may be used to select either the high (high level-on input) or low (low level-on input) velocity range as defined with the JOYVH and JOYVL commands, respectively. The high range could be used to quickly move to a location while the low range could be used for accurate positioning. Refer to the illustration above. *When this input is not connected, the low velocity range is selected.*

Joystick Release Input

The joystick release input allows you to indicate to the 6270 that you have finished using the joystick and program execution may continue with the next statement. When a program enables joystick control of motion, program execution will stop and then resume when the user is finished with joystick mode (assuming the Continuous Command Execution Mode is disabled with the COMEXCØ command).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The joystick release input has an internal pull-up resistor to +5V. When the joystick release input is not grounded, joystick enable statements (JOY1) will be disabled upon execution. To enable the joystick mode, the joystick release input must be inactive (connected to ground). Refer to the illustration above.

Joystick Trigger Input

The status of this input can be read by a program and may be used to control program flow (see INO and TINO command). Refer to the illustration above.

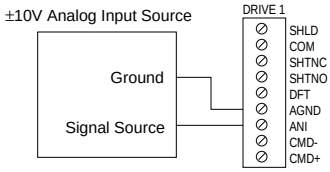
Joystick Auxiliary Input

The status of this input can be read by a program and may be used to control program flow (see INO and TINO command). Refer to the illustration above.

ANI Analog Input Connections (6270-ANI Option Only)

Application considerations are discussed in Chapter 5.

The 6270-ANI option offers two ±10V, 14-bit analog inputs (one **ANI** terminal found on each of the **DRIVE** connectors). These inputs are sampled at the servo sample rate (set with the SSFR command). The ANI input values are reported with the TANI and ANI commands (voltage) and the TPANI and PANI commands (position).



Polarity Reversal

If the ANI analog input counts in the wrong direction (opposite from the commanded direction), instead of reversing the mounting of the ANI input you can use the ANIPOL command to reverse the polarity. For example, if the ANI input on axis 2 only is reversed, issue the ANIPOLx1 command (or if the ANI on axis 1 only is reversed, use ANIPOL1x).

Extending 6270 System Cables

This section describes options for extending 6270 system LDT/encoder and I/O cables.

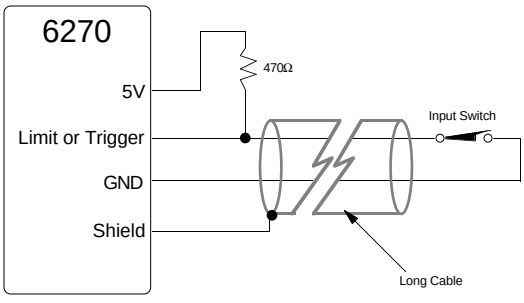
Encoder/LDT-to-6270 Cables

The maximum cable length between the LDTs or encoder and the 6270 is 100 feet. If you wish to lengthen these cables yourself, use 24 AWG wire. Be sure to shield the cable with the shield connected only to **SHLD** (pin 1 on the **ENCODER** connector or **LDT** connector).

I/O Cables

To avoid interference from external noise, shield all I/O cables, regardless of the length. The maximum length of cables is determined by the environment in which the equipment will be used. For cables longer than 50 feet or in electrically noisy environments, you should follow the guidelines below.

- Use a minimum wire size of 22 AWG wire.
- Use twisted pair shielded cables and connect the shield to the **SHLD** terminal on the 6270 connector. Leave the other end of the shield disconnected.
- Do not route these signals in the same conduit or wiring trays as high-voltage AC.
- Limit and trigger inputs are internally pulled up to +5VDC and are TTL compatible. In electrically noisy environments or when using long cable lengths, use an external pull-up resistor with a value of 330Ω to 2.2KΩ between the input and +5V. The external resistor lowers the input impedance makes the input less susceptible to electrical noise.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

③ Installation Verification

WARNING

This installation verification section is intended to be executed with the drives (or valves) not connected to the 6270. **Do not proceed until you are sure they are not connected.**

Step 1 Return to the *Test Procedure* in Chapter 2 to test the power interface and the RS-232C interface.

Step 2 Use the information in the following table to test the features appropriate to your application. If you receive responses other than those expected, check your system wiring and refer to the command descriptions in the ***6000 Series Software Reference Guide*** for assistance.

NOTE

The following table is based on the assumption that you have **not** changed the active levels of the 6270's inputs and outputs. Verify these settings with the following *status* commands:

Command Entered	Response Should Be
INLVL	*INLVL0000_0000_0000_0000_0000_0000_00
HOMLVL	*HOMLVL00
LHLVL	*LHLVL0000
OUTLVL	*OUTLVL0000_0000_0000_0000_0000_0000_00

Connections	Test Procedure	Response Format (left to right)
Hardware End-of-travel and Home limits	**NOTE**: If you are not using hardware end-of-travel limits, issue the Disable Limits (LH0,0) command and ignore the first two bits in each response field. 1. Enable hard limits using the LH3,3 2. Close the end-of-travel switches and open the home switches. 3. Enter the TLIM command. The response should be *TLIM110_110. 4. Open the end-of-travel switches and close the home switches. 5. Enter the TLIM command. The response should be *TLIM001_001. 6. Close the CW end-of-travel switch on axis 1 and open the home switch on axis 2. 7. Enter the TLIM command. The response should be *TLIM101_000.	TLIM response: bit 1 = axis 1 CW limit bit 2 = axis 1 CCW limit bit 3 = axis 1 home limit bit 4 = axis 2 CW limit bit 5 = axis 2 CCW limit bit 6 = axis 2 home limit
Analog Output Signal	1. If the servo devices are connected to the 6270's DRIVE connectors, disconnect them now. 2. Set all the gains to zero by entering the following: SGP0,0, SGI0,0, SGV0,0, SGAF0,0, SGVF0,0 3. Enable the 6270 to send out the analog command by entering the DRIVE11 command. 4. Set the DAC output limit to 10 volts by entering the DACLIM10,10 command. 5. Drive the analog output to the maximum positive range by entering the SOFFS10,10 command. 6. Enter the TDAC command to check the analog output value. The response should be *TDAC+10,+10. 7. Using a Digital Volt Meter (DVM), measure the actual analog output voltage between the CMD+ (analog command) and CMD- (analog command return) terminals. Compare the DVM reading to the entry for the SOFFS command (see step 5). If the reading deviates more than 0.1V from +10V, then there is either a problem with the system's grounding connection or the 6270's DAC is not functioning properly. 8. Repeat steps 5 through 7, using these servo output offset values: SOFFS-10,-10 SOFFS0,0 SOFFS5,5 SOFFS5,5	TDAC response (output voltage): ±axis 1, ±axis 2 *Current/Voltage Conversion* If you have configured the 6270's jumpers for current control, convert the voltage readings to current values and use an ammeter to measure current (mA range ÷ 10 volts = mA/volt). For example, if using ±20mA output, the conversion is2mA/volt.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LDT Feedback	1. Using and LDT, with a free magnet, place the magnet within the usable portion of the LDT. 2. Enter the `TLDT` command to determine the position of the LDTs. The response should not be zero; if it is, check for proper connections and placement of the magnet within the usable portion of the LDT. 3. When a valid reading is obtained, enter these commands: `L<cr>`, `TLDT<cr>`, `T.3<cr>`, and then `LN<cr>`. This will begin a continuous display of the LDT's position. Press the <return> key to move the display to the next line and save the current value. 4. Move the magnet along the LDT and verify that the position changes as the magnet is moved. If you connected the LDT(s) as instructed earlier in this chapter, moving the magnet in the extension direction should increase the position reading. *This can also serve as a test to find the usable region of the LDT.* 5. When finished, enter the `^K` (ctrl-K) command to stop the continuous report-back.	`TLDT` response (LDT counts): ±LDT 1, ±LDT 2
Encoder Feedback (Rotary Incremental Encoder)	1. Enter these commands: `L<cr>`, `TPE<cr>`, `T.3<cr>`, and then `LN<cr>`. This will begin a continuous display of the encoder's position. Press the <return> key to move the display to the next line and save the current value. 2. Manually rotate the encoder shaft and verify that the position changes as you rotate the encoder shaft. If you connected the encoder as instructed earlier in this chapter, moving the shaft clockwise should increase the position reading. If the reading does not change, or if the direction is reversed, check the connections. If the direction is reversed, swap the A+ and A- connections. 3. When finished, enter the `^K` (ctrl-K) command to stop the continuous report-back.	`TPE` response (encoder counts): ±encoder 1
ANI Analog Input Feedback **(6270-ANI option only)**	1. Enter these commands: `L<cr>`, `TANI<cr>`, `T.3<cr>`, and then `LN<cr>`. This will begin a continuous display of the voltage level at the ANI inputs (pin 3 on the **DRIVE** connectors). Press the <return> key to move the display to the next line and save the current value. 2. Change the voltage output from your voltage source and verify that the `TANI` report changes accordingly. If the reading does not change, check the connections. 3. When finished, enter the `^K` (ctrl-K) command to stop the continuous report-back.	`TANI` response (volts): ±ANI input #1, ±ANI input #2 *NOTE* ANI feedback is measured in volts
Programmable Inputs (incl. triggers)	1. Open the input switches or turn off the device driving the inputs. 2. Enter the `TIN` command. The response should be `*TIN0000_0000_0000_0000_0000_0000_00`. 3. Close the input switches or turn on the device driving the inputs. 4. Enter the `TIN` command. The response should be `*TIN1111_1111_1111_1111_1111_1111_11`.	`TIN` response: bits 1-24 = prog. inputs 1 - 24 bits 25 & 26 = TRG-A & TRG-B
Programmable Outputs	1. CAUTION: Disconnect all programmable outputs before proceeding to step 2. 2. Enter the `OUTALL1,26,1` command to turn on (sink current on) all outputs. 3. Enter the `TOUT` command. The response should be `*TOUT1111_1111_1111_1111_1111_1111_11`. 4. Enter the `OUTALL1,26,0` command to turn off all outputs. 5. Enter the `TOUT` command. The response should be `*TOUT0000_0000_0000_0000_0000_0000_00`.	`TOUT` response: bits 1-24 = prog. outputs 1 - 24 bits 25 & 26 = OUT-A & OUT-B

RP240	1. Cycle power to the 6270.
	2. If the RP240 is connected properly, the RP240's status LED should be green and one of the messages on the computer or terminal display should read `*RP240 CONNECTED`.
	If the RP240's status LED is off, check to make sure the +5V connection is secure.
	If the RP240's status LED is green, but the message on the terminal reads `*NO REMOTE PANEL`, the RP240 Rx and Tx lines are probably switched. Remove power and correct.
	3. Assuming you have not written a program to manipulate the RP240 display, the RP240 screen should display the following:
	`6270 MOTION CONTROLLER`
	`RUN JOG STATUS DRIVE DISPLAY ETC`	

Enable and Joystick inputs	1. Open the enable input (**ENBL**) switch, and open the joystick input switches or turn off the device driving the joystick inputs. 2. Enter the `TINO` command. The response should be `*TINO0000_0100`. 3. Close the input switches or turn on the device driving the inputs. 4. Enter the `TINO` command. The response should be `*TINO1111_1100`.	`TINO` response: bit 1 = joystick auxiliary bit 2 = joystick trigger bit 3 = joystick axes select bit 4 = joystick velocity select bit 5 = joystick release bit 6 = Enable input bits 7 & 8 are not used

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

④ What's Next?

At this point you should have successfully completed this chapter's mounting, connection, and test procedures for your 6270 system. If you intend to use thumbwheels, PLCs or other I/O devices, or if you intend to daisy-chain multiple 6270s, refer to the connection instructions and application considerations in Chapter 5.

The following steps are recommended to prepare you for applying the 6270 in your application.

Step 1 **Attach the Load**

Attach the load to the cylinder or motor, and attach the feedback device as appropriate to your application.

Step 2 **Perform the Basic System Configuration**

> **NOTE**
> The set-up commands referred to in this step are not saved in the 6270's battery-backed RAM. Therefore, you may want to add them to the startup (STARTP) program. For information on defining the startup program, refer to the *Programming Guide* section in the ***6000 Series Software Reference Guide.***

Startup Program Example:

	Command	Description
>	DEF PWRUP	Begin definition of PWRUP program
–	INDAX2	Place both axes in use
–	DRFLVL11	Set drive fault level to active high
–	KDRIVE11	Enable the DISABLE ON KILL feature for both axes
–	DRIVEØØ	Disable both valves/drives
–	SFB3,3	Select LDT feedback for both axes (subsequent scaling, gains, servo offset, PSET, and SMPER parameters are specific to LDT feedback)
–	LDTGRD9,9	Set LDT gradient to 9µs/inch for both LDTs
–	LDTRES432,432	Set LDT resolution to 432 counts/inch
–	LDTUPD1,1	Set LDT position update rate equal to the system update rate (see table in SSFR command description)
–	SCLA432,432	Set scaling for programming accel/decel in inches/sec^2
–	SCLV432,432	Set scaling for programming velocity in inches/sec
–	SCLD432,432	Set scaling for programming distances in inches
–	SGP5Ø,5Ø	Set proportional feedback gain
–	SGI.2,.2	Set integral feedback gain
–	SGV3Ø,3Ø	Set velocity feedback gain
–	PSET1Ø,1Ø	Set current position as absolute position 10
–	SFB1	Select encoder feedback for axis 1 (subsequent scaling, gains, servo offset, PSET, and SMPER parameters are specific to encoder feedback)
–	ERES4ØØØ	Set encoder resolution to 4,000 counts/rev
–	SCLA4ØØØ	Set scaling for programming accel/decel in revs/sec^2
–	SCLV4ØØØ	Set scaling for programming velocity in revs/sec
–	SCLD4ØØØ	Set scaling for programming distances in revs
–	SGP.5	Set proportional feedback gain
–	SGI1	Set integral feedback gain
–	SGV1	Set velocity feedback gain
–	PSETØ	Set current position as absolute position zero
–	SFB2,2	Select ANI feedback for both axes (subsequent scaling, gains, servo offset, PSET, and SMPER parameters are specific to ANI feedback)
–	SCLA819,819	Set scaling for programming accel/decel in volts/sec^2
–	SCLV819,819	Set scaling for programming velocity in volts/sec
–	SCLD819,819	Set scaling for programming distances in volts
–	SGP1,1	Set proportional feedback gain
–	SGIØ,Ø	Set integral feedback gain
–	SGV.5,.5	Set velocity feedback gain
–	PSET5,5	Set current position as absolute position 5
–	SFB3,3	Select LDT feedback
–	END	End definition of PWRUP program
>	STARTP PWRUP	Assign PWRUP as the startup program

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Analog Output — Voltage or Current Operation:

If will not be using ±10V output to the drive/valve, you may select a current range by configuring the internal jumpers (options are ±20mA, ±50mA, ±60mA, ±80mA, ±100mA, or ±150mA). Instructions for setting the jumpers are provided in the *Optional DIP switch and Jumper Settings* section at the end of Chapter 6.

DAC Output Limits (DACLIM and DACMIN):

If will not be using the entire -10V to +10V range of the 6270's analog output, you can set up maximum and minimum limits.

For example, if are using a 4-20mA control loop, set the analog output jumpers to operate at ±20mA (instructions provided in Chapter 6). Then issue these setup commands listed below. (Note that when using ±20mA output, you need to use the 2mA/volt equation to ascertain the proper *voltage* value to enter in the DACLIM, DACMIN, and SOFFS commands).

Command	Description
> DACLIM1Ø	Set DAC maximum limit to +20mA (20mA ÷ 2mA/V = 10V)
> DACMIN2	Set DAC minimum limit to +4mA (4mA ÷ 2mA/V = 2V)
> SOFFS6	Set offset analog output to the mid-range value of +12mA (12mA ÷ 2mA/V = 6V)

Number of Axes (INDAX):

By configuring the number of axes in use, you limit the number of axes you can control. This may be desired if you are only using one of the two axes available. The INDAX command configures the number of axes, and affects the servo sampling rate (see SSFR command description). INDAX2 (the default setting) enables both command fields to be entered (e.g., A1,1). If you enter INDAX1, instead of entering A1,1 you should enter A1, and all responses from the 6270 will also only show the one field; if you enter the command A, the response will be *A1.

Drive Fault Level (DRFLVL) – ***for Drives only***:

The drive fault level (DRFLVL) should be set to active high or active low for each axis (default is active low—DRFLVLØ). The DRFLVL command description provides recommended settings for known drives. When using a drive without a drive fault output, set the drive fault level to active low.

> **NOTE**
> Once the drive fault level has been configured, you must enable the drive fault input with the INFEN1 command before the input is usable.

Feedback Device Selection (SFB):

Use the SFB command to select the feedback devices to be used by each axis. The syntax for the SFB command is SFB<i>,<i>, where <i> could be 1 (encoder), 2 (ANI input—**6270-ANI** option only), or 3 (LDT). The default setting is SFB3,3, which selects LDT feedback for both axes. *NOTE: Encoder feedback can be used only on axis 1.* For example, if you wish to use encoder feedback on axis 1 and LDT feedback on axis 2, you would issue the SFB1,3 command.

> **NOTE**
> Parameters for scaling (SCLA, SCLD, etc.), tuning gains (SGP, SGV, etc.), servo offset (SOFFS and SOFFSN), maximum position error (SMPER), and position offset (PSET) are specific to the currently selected feedback device.
>
> If your application requires switching between feedback sources on the same axis, then for each feedback source, you must issue the appropriate SFB command and enter the scaling, gains, servo offset, SMPER, and PSET values specific to operating with that feedback source. (Refer to the startup program example above.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Feedback Device Set Up:

LDT Set Up

LDT Gradient (LDTGRD):

The LDTGRD value may be changed only when the drive is disabled with the DRIVE command.

The *gradient* is a measure of how quickly the LDT can respond to feedback requests. It is unique to each LDT and should be printed on the unit. The 6270's default gradient setting is 9.0000 µs/inch. The gradient is used to correct for positional differences created by different LDTs; this allows programs to be easily transported between 6270s or used with a new LDT. The LDT value is entered with the LDTGRD command. The 6270 uses this value to scale LDT positions so that it always provides 432 counts/inch (or a multiple of 432 if using recirculation, or 17 counts/mm).

LDT Resolution (LDTRES):

The *LDT resolution* is determined by the crystal frequency of the 6270 counters and the number of recirculations of the LDT. (A *recirculation* is a single request for position information from the LDT. Multiple recirculations provide greater resolution by increasing the length of the feedback pulse.) The 6270 counter frequency of 48 MHz provides a resolution of 432 counts/inch (assuming you use the LDT gradient default setting of 9 µs/inch). This number is multiplied by the number of recirculations. For example, if you are using LDTs with 4 recirculations, you should set the resolution of each LDT to 1728 (4 * 432 = 1728) with the LDTRES1728,1728 command.

LDT Position Update Rate (LDTUPD):

The LDTUPD command value is multiplied by the *system update rate* to determine the LDT position update rate. The system update rate is affected by the INDAX and SSFR command settings.

As the LDTUPD command value is decreased (update rate is increased), the quality of the dynamic response improves. However, if the update rate is too fast, the LDT will not have enough time to read the position and LDT read errors will occur. If a read error occurs, axis status bit #27 (reported with the TAS and AS commands) will be set. In addition, if ERROR bit #15 is enabled (ERROR.15-1), error status bit #15 (reported with the TER and ER commands) will also be set.

Formula to determine minimum allowable LDTUPD update rate:

((max. length of travel in inches * LDT gradient) * # of recirculations) + 140µs

Encoder Set Up (does not apply to LDT or ANI input)

Encoder Resolution:

The *encoder resolution* is determined by the resolution of the encoder used with the servo drive/motor system. The encoder resolution is essentially the number of steps, or *counts* (post quadrature), per unit of travel. For example, Compumotor E Series encoders are 1,000-line encoders, and therefore have a 4,000 count/rev post-quadrature resolution.

If the encoder is mounted directly to the motor, then to ensure that the motor will move according to the programmed distance and velocity, the 6270's resolution must match the encoder's resolution. Use the ERES command to set the 6270's resolution (default setting is 4,000 counts/rev, selectable range is 200 to 1,024,000).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Kill's Effect on the Drive or Valve (KDRIVE):

Normally, when you issue a Kill command (K, !K, or <ctrl>K) or activate a general-purpose input configured as a kill input (see INFNCi-C command), motion is stopped at the hard limit (LHAD/LHADA) deceleration setting and the valve or drive is left in the enabled state (DRIVE11).

However, your application may require you to *disable* (*shut down* or *de-energize*) the valve or drive in a Kill situation to, for example, prevent damage to mechanical components. If so, set the 6270 to the *Disable Drive on Kill* mode with the KDRIVE11 command. In this mode, a kill command or kill input will shut down the valve or drive immediately. In valve/cylinder systems, the valve will then return to the null position. In drive/motor systems, the motor will *free wheel* (without control from the drive) to a stop. When the valve or drive is disabled (DRIVEØØ), the SHTNC relay output is connected to **COM** and the SHTNO relay output is disconnected from **COM**. To re-enable the valve or drive, issue the DRIVE11 command.

Step 3 **Determine Your Application's Motion Control Requirements**

Applications can vary greatly from one to another. Consequently, the 6270 is equipped with many motion control features to satisfy a wide variety of application requirements—*but not all features are appropriate for every application*. Therefore, you must first determine the necessary motion features you need for your application. Once you have done that, you can proceed to tune the 6270 and then implement the necessary motion control features in your application.

Step 4 **Tune the Servo System**

To effectively tune the 6270 (and any velocity drives you may be using), use the interactive tuning features in the *Servo Tuner*™ add-on module for *Motion Architect*®. It greatly improves your efficiency and gives you powerful graphical tools to measure the performance of the system. Instructions for using Servo Tuner to tune the 6270 are provided in the ***Servo Tuner User Guide***.

NOTE

The Servo Tuner option is an add-on module and does not automatically come with the basic Motion Architect software package. To order your copy of Servo Tuner, which is provided on a separate disk, contact your local Automation Technology Center.

If are not using Servo Tuner, refer to the empirical-type tuning procedures provided in Chapter 4.

Step 5 **Implement the Necessary 6270 Features**

Chapter 5 describes how to implement the 6270's features in your application. (If you are using Following, refer to the ***6000 Series Following User Guide***.)

You will develop your application by creating and refining motion programs using the 6000 Series Command Language. We recommend you use Motion Architect® or the 6000 DOS Support Disk to aide in your programming efforts. Motion Architect and the 6000 DOS Support Disk are discussed briefly in Chapter 5, but for detailed user information refer to the ***Motion Architect User Guide*** or the ***6000 DOS Support Disk Quick Reference***.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ④

Servo Tuning

In a Hurry?

You should tune the 6270 before attempting to execute any motion functions. At a minimum, complete this chapter's *Tuning Setup Procedure* and *Controller Tuning Procedures* until you have found a proportional feedback gain that can give a stable response for your system. (The *Drive Tuning Procedure* below is for use with velocity drive systems only, not for servo valve systems.) Then you can proceed to execute your motion functions. To gain a full understanding of tuning, you should read through this entire chapter and follow its procedures to ensure your system is properly tuned.

Servo Tuning Software Available

To effectively tune the 6270 (and any velocity drives you may be using), use the interactive tuning features in the *Servo Tuner* add-on module for Motion Architect. It greatly improves your efficiency and gives you powerful graphical tools to measure the performance of the system. Instructions for using Servo Tuner to tune the 6270 are provided in the ***Servo Tuner User Guide***.

The Servo Tuner option is an add-on module and does not automatically come with the basic Motion Architect software package. To order your copy of Servo Tuner, which is provided on a separate disk, contact your local Automation Technology Center.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Servo System Terminology

This section gives you with an overall understanding of the principles and the terminology used in tuning the 6270.

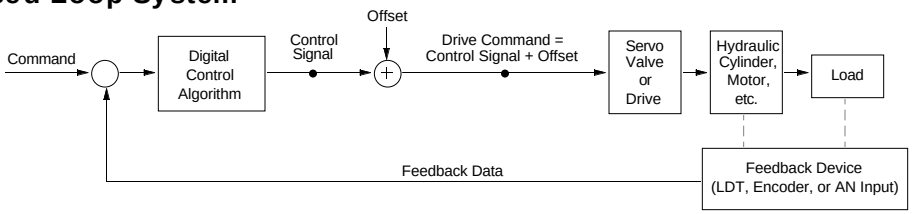
Servo Tuning Terminology

The 6270 uses a digital control algorithm to control and maintain the position and velocity. The digital control algorithm consists of a set of numerical equations used to periodically (once every **servo sampling period**) calculate the value of the **control signal** output. The numerical terms of the equations consist of the current commanded and actual position values (plus a few from the past sampling period) and a set of control parameters. Each control parameter, commonly called a **gain**, has a specific function (see *Servo Control Techniques* later in this chapter). Groups of gains may be saved to specified **gain sets** that can be invoked to affect motion under varied conditions at different times. **Tuning** is the process of selecting and adjusting these gains and gain sets to achieve optimal servo performance.

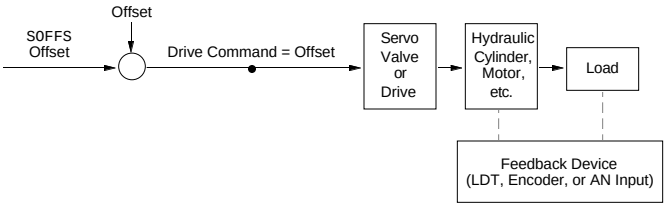
When this control algorithm is used, the whole servo system is a **closed-loop** system (see diagram below). It is called closed loop because the control algorithm accounts for both the **command** (position, velocity, tension, etc.) and the **feedback data** (from the LDT, encoder, or ANI input); therefore, it forms a *closed loop* of information flow.

When all gains are set to zero, the digital control algorithm is essentially disabled and the system becomes an **open loop** system (see diagram below). During system setup or troubleshooting, it is desirable to run the system in open loop so that you can independently test the drive/motor or valve operation (further details are provided in the *Open Loop Operation* section later in this chapter).

Closed Loop System



Open Loop System



The 6270 has the capability of providing an analog voltage output of ±10V or a current of ±20mA, ±50mA, ±60mA, ±80mA, ±100mA, or ±150mA for commanding the valve or drive. After the digital control algorithm has calculated the digital control signal, this digital value is sent out from the DSP (digital signal processor) to the Digital-to-Analog converter (DAC). The DAC has an analog output range of -10V to +10V or maximum current. It is often possible that the digital control signal calculated by the control algorithm can exceed this limit. When this happens, the analog output would just stay, or *saturate*, at the maximum limit until the position error changes such that the control algorithm would calculate a control signal less than the limit. This phenomenon of reaching the output limit is called **controller output saturation**. When saturation occurs, increasing the gains does not help improve performance since the DAC is already operating at its maximum level.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

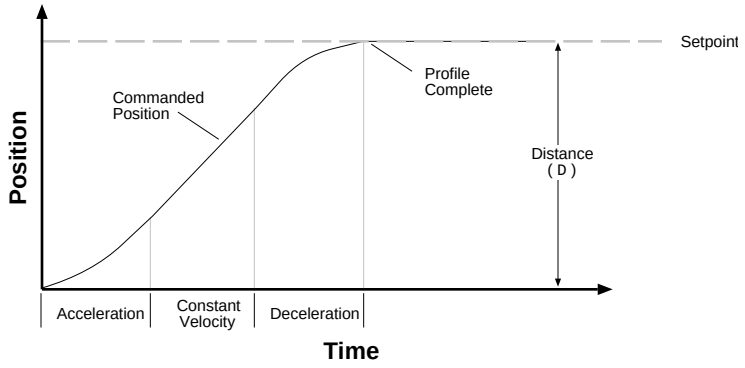
Position Variable Terminology

In a servo system, there are two types of **time-varying** (value changes with time) position information used by the controller for control purposes: commanded position and actual position. You can use this information to determine if the system is positioning as you expect.

Commanded Position

The **commanded position** is calculated by the motion profile routine based on the acceleration (A, AA), deceleration (AD, ADA), velocity (V) and distance (D) command values and it is updated every servo sampling period. Therefore, the commanded position is the intended position at any given point of time. To view the commanded position, use the TPC (Transfer Commanded Position) command; the response represents the commanded position at the instant the command is received.

When this user guide refers to the *commanded position*, it means this calculated time-varying commanded position, not the distance (D) command. Conversely, when this user guide refers to the **position setpoint**, it means the final intended distance specified with the distance (D) command. The following plot is a typical profile of the commanded position in preset (MCØ) mode.



Actual Position

The other type of time-varying position information is the **actual position**; that is, the actual position of the motor (or load or cylinder, etc.) measured with the feedback device (LDT, encoder, or ANI input). Since this is the position achieved when the drive/valve responds to the commanded position, we call the overall picture of the actual position over time the **position response** (see further discussion under *Servo Response Terminology*).

To view the actual position, use the TFB (Transfer Position of Feedback Device) command; the response represents the actual position at the instant the command is received. The goal of tuning the servo system is to get the actual position to track the commanded position as closely as possible.

The difference between the commanded position and actual position is the **position error**. To view the position error, use the TPER (Transfer Position Error) command; the response represents the position error at the instant the command is received. When the motor/valve is not moving, the position error at that time is called the **steady-state position error** (see definition of steady-state under *Servo Response Terminology*). If a position error occurs when the motor/valve is moving, it is called the **position tracking error**.

In some cases, even when the system is properly tuned, the position error can still be quite significant due to a combination of factors such as the desired profile, the servo mechanism's limitation, the dynamic characteristics of the system, etc. For example, if the value of the velocity (V) command is higher than the maximum velocity the hydraulic cylinder (or motor, etc.) can physically achieve, then when it is commanded to travel at this velocity, the actual position will always lag behind the commanded position and a position error will accumulate, no matter how high the gains are.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Servo Response Terminology

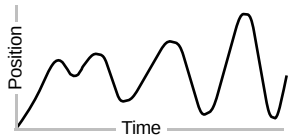
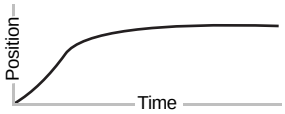
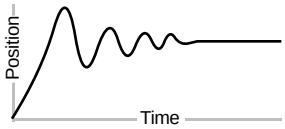
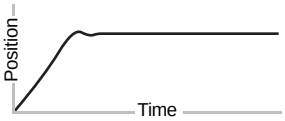
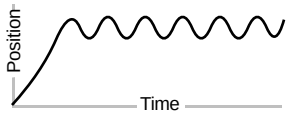
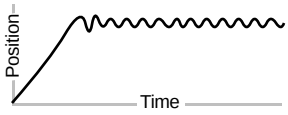
Stability

The first objective of tuning is to stabilize the system. The formal definition of **system stability** is that when a bounded input is introduced to the system, the output of the system is also bounded. What this means to a motion control system is that if the system is **stable**, then when the position setpoint is a finite value, the final actual position of the system is also a finite value.

On the other hand, if the system is **unstable**, then no matter how small the position setpoint or how little a disturbance (motor torque variation, load change, noise from the feedback device, etc.) the system receives, the position error will increase continuously, and exponentially in almost all cases. In practice, when the system experiences instability, the actual position will oscillate in an exponentially diverging fashion as shown in the drawing below. The definition here might contradict what some might perceive. One common perception shared by many is that whenever there is oscillation, the system is unstable. However, if the oscillation finally diminishes (damps out), even if it takes a long time, the system is still considered stable. The reason for this clarification is to avoid misinterpretation of what this user guide describes in the following sections.

Position Response Types

The following table lists, describes, and illustrates the six basic types of position responses. The primary difference among these responses is due to **damping**, which is the suppression (or cancellation) of oscillation.

Response	Description	Profile (position/time)
Unstable	Instability causes the position to oscillate in an exponentially diverging fashion.	
Over-damped	A highly damped, or *over-damped*, system gives a smooth but slower response.	
Under-damped	A slightly damped, or *under-damped*, system gives a slightly oscillatory response.	
Critically damped	A critically-damped response is the most desirable because it optimizes the trade-off between damping and speed of response.	
Oscillatory	An oscillatory response is characterized by a sustained position oscillation of equal amplitude.	
Chattering	Chattering is a high-frequency, low-amplitude oscillation which is usually audible.	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

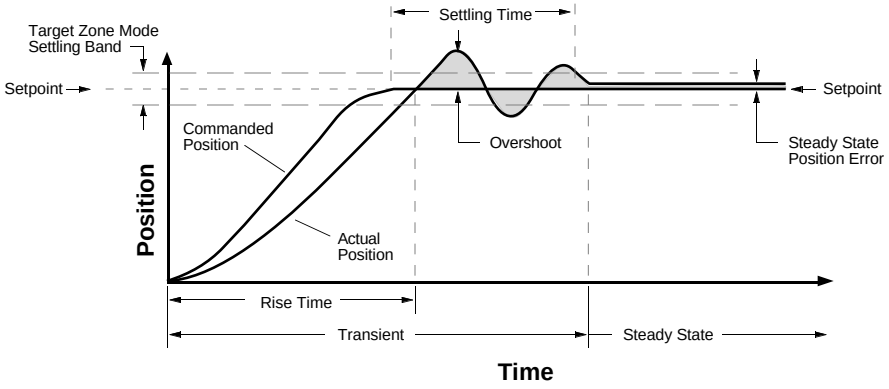
Performance Measurements

When we investigate the plot of the position response versus time, there are a few measurements that you can make to quantitatively assess the performance of the servo:

- **Overshoot**—the measurement of the maximum magnitude that the actual position exceeds the position setpoint. It is usually measured in terms of the percentage of the setpoint value.
- **Rise Time**—the time it takes the actual position to pass the setpoint.
- **Settling Time**—the time between when the commanded position reaches the setpoint and the actual position settles within a certain percentage of the position setpoint. (Note the settling time definition here is different from that of a control engineering text book, but the goal of the performance measurement is still intact.)

These three measurements are made before or shortly after the hydraulic cylinder (or motor, etc.) stops moving. When it is moving to reach and settle to the setpoint, we call such period of time the **transient**. When it is not moving, it is defined as in **steady-state**.

A typical stable position response plot in preset mode (MCØ) is shown below.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE

The following list briefly describes each servo-related 6000 Series command. More detailed information can be found in the rest of this chapter and within each command's description in the ***6000 Series Software Reference Guide***.

Command	Title	Brief Description (detailed descriptions in ***6000 Series Software Reference Guide***)
LDTUPD	*LDT Position Update Rate*	Use LDTUPD to select the rate at which the LDT position is sampled. Decreasing the LDTUPD command value (speeding up the update rate)will improve the quality of the dynamic response. However, if the update rate is too fast, the LDT will not have enough time to read the position, resulting in read errors. The occurrence of read errors can be monitored with the TER and [ER] commands if you enable ERROR bit #15, and you can check the LDT status with bit #27 of TAS and AS.
SFB	*Select Servo Feedback Source*	Selects the servo feedback transducer (options: LDT, encoder, or ANI analog input). The SFBØ command sets all the gains to zero so that the controller runs in open loop mode, and disables the Setpoint Window feature (equivalent to SSWGØ)
SGAF & SGAFN *	*Acceleration Feedforward Gain*	Sets the acceleration feedforward gain in the PIV&**F_a** servo algorithm.
SGENB	*Servo Gain Set Enable*	Enables a previously-saved set of PIV&F gains. A set of gains (specific to the current feedback source selected with the SFB command) is saved using the SGSET command.
SGI & SGIN *	*Set Integral Feedback Gain*	Sets the integral gain in the P**I**V&F servo algorithm.
SGILIM	*Set Integral Windup Limit*	Sets a limit on the correctional control signal that results from the integral gain action trying to compensate for a position error that persists too long.
SGP & SGPN *	*Proportional Feedback Gain*	Sets the proportional gain in the **P**IV&F servo algorithm.
SGSET	*Save a Set of Servo Gains*	Saves the presently-defined set of PIV&F gains as a particular *gain set* (specific to the current feedback source on each axis). Up to 5 gain sets can be saved and enabled at any point in a move profile, allowing different gains at different points in the profile.
SGV & SGVN *	*Set Velocity Feedback Gain*	Sets the velocity gain in the PI**V**&F servo algorithm.
SGVF & SGVFN *	*Velocity Feedforward Gain*	Sets the velocity feedforward gain in the PIV&**F_V** servo algorithm.
SMPER	*Maximum Allowable Position Error*	Sets the maximum allowable error between the commanded position and the actual position as indicated by the feedback device. If the error exceeds this limit, the 6270 activates the Shutdown output and sets the DAC output to zero (plus any SOFFS offset). If there is no offset, a valve will return to the null position and a rotary motor will freewheel to a stop. You can enable the ERROR command to continually check for this error condition, and when it occurs to branch to a programmed response defined in the ERRORP program.
SOFFS & SOFFSN	*Servo Control Signal Offset*	Sets an offset to the commanded analog output voltage, which is sent to the drive system. The SOFFSN command allows you to set an offset voltage when the position error is negative. The SOFFSN value tracks the SOFFS value until a separate SOFFSN value is entered. To return SOFFSN to the default mode in which it track SOFFS, issue the SOFFSN command with a minus sign (-) in the command field for the affected axis (e.g., SOFFSN,- restores axis 2 to the default mode).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SSFR	*Servo Frequency Ratio*	Sets the ratio between the update rate of the move trajectory and the update rate of the servo action. The intermediate position setpoints calculated by the trajectory generator is updated at a slower rate then the servo position correction. This command allows you to optimize this for your application. The default setting (SSF4) is sufficient for most applications.
SSWD & SSWG	*Setpoint Window Gains*	You may now activate a specified set of gains to be used when with a defined region either side of the position setpoint ("setpoint window"). The SSWD command is used to specify the distance on both sides of the position setpoint ("setpoint window") in which the gain set specified with the SSWG command is used. Specifically, the gain set is automatically invoked by the controller *after* the commanded move profile is complete and the actual position is within the setpoint window. The setpoint window includes a hysteresis loop equal to 25% of the value used in the SSWD command. NOTE: The gain set to be used must first be defined with the SGSET command. For more information, refer to the *Setpoint Window Gains* section later in this chapter.
STRGTD STRGTE STRGTT STRGTV	*Target Zone Distance* *Target Zone Mode Enable* *Target Zone Timeout Period* *Target Zone Velocity*	When using the Target Zone Mode, enabled with the STRGTE command, the actual position and actual velocity must be within the *target zone* (that is, within the distance zone defined by STRGTD and within the velocity zone defined by STRGTV). If the motor/load does not settle into the target zone before the timeout period set by STRGTT, the 6270 detects an error. To prevent subsequent commands/moves from being executed when this error condition occurs, you must enable the ERROR command to continually check for this error condition, and when it occurs to branch to a programmed response defined in the ERRORP program. Otherwise, subsequent commands/moves can be executed regardless of the actual position and velocity. This feature is explained in greater detail later in the *Target Zone* section.
TDAC & DAC	*Value of DAC Output*	Transfers [or assigns/compares] the output from the 6270's digital-to-analog converter. This is the analog control signal output at the 6270's CMD terminal.
TFB & FB	*Position of Servo Feedback Devices*	Transfers [or assigns/compares] the actual position of the transducers selected for feedback (see SFB).
TGAIN	*Transfer Servo Gains*	Transfers the currently active set of PIV&F gains. The servo gain set reported represents the last gain values specified with the individual servo gain commands (SGI, SGP, SGV, SGAF, and SGVF), or the last gain set enabled with the SGSET command.
TPC & PC	*Position Commanded*	Transfers [or assigns/compares] the commanded position (intermediate position setpoint) to the drive or valve.
TPER & PER	*Position Error*	Transfers [or assigns/compares] the error between the commanded position (TPC) and the actual position (TFB, or TPE, TLDT, TANI) as measured by the feedback device.
TSGSET	*Transfer Servo Gain Set*	Transfers a previously-saved set of servo gain parameters. A gain set is saved with the SGSET command.
TSTLT	*Transfer Servo Settling Time*	Transfers the time it took the last move to settle within the *target zone* (that is, within the distance zone defined by STRGTD and within the velocity zone defined by STRGTV). The Target Zone Mode does not need to be enabled to use this command.

* The negative gain commands (SGAFN, SGIN, SGPN, SGVN, and SGVFN) allow you to establish gains to be used when the position error is negative. The negative gain value will track the positive gain value until a separate negative gain value is entered. To re-establish the default mode where the negative gain tracks the positive gain, issue the negative gain command with a minus sign (-) in the command field for the affected axis (e.g., SGPN,- restores axis 2 to the default mode).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

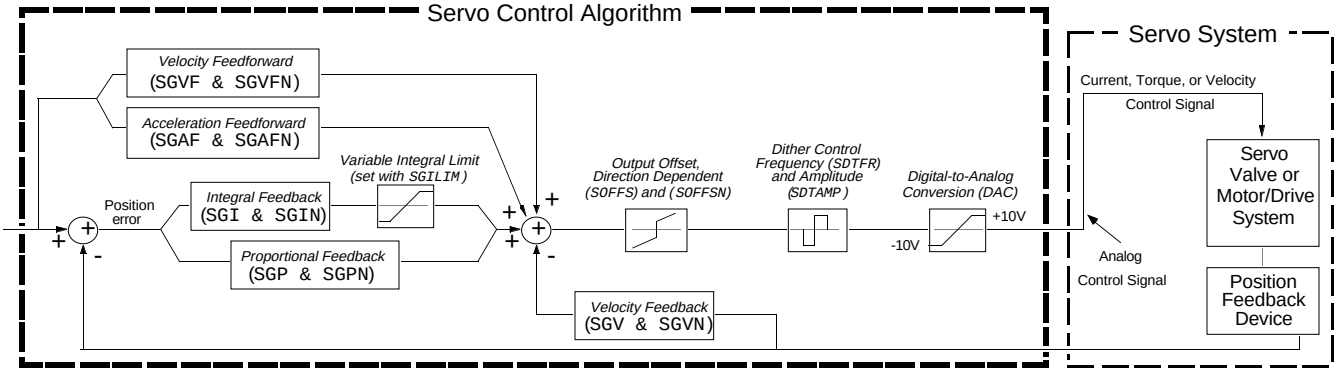
Servo Control Techniques

To ensure that you are tuning your servo system properly, you should understand the tuning techniques described in this section.

The 6270 employs a *PIV&F* servo control algorithm. The control techniques available in this system are as follows:

P .....Proportional Feedback (controlled with the SGP and SGPN commands)
I .....Integral Feedback (controlled with the SGI and SGIN commands)
V .....Velocity Feedback (controlled with the SGV and SGVN commands)
F .....Velocity and Acceleration Feedforward (controlled by the SGVF and SGAF and SGVFN and SGAFN commands, respectively)

The block diagram below shows these control techniques in relation to the servo control algorithm configuration. The following table presents a condensed summary of each control's effect on the servo system.



Gain	Stability	Damping	Disturbance Rejection	Steady State Error	Tracking Error
Proportional (SGP and SGPN)	Improve	Improve	Improve	Improve	Improve
Integral (SGI and SGIN)	Degrade	Degrade	Improve	Improve	Improve
Velocity Feedback (SGV and SGVN)	Improve	Improve	----------	----------	Degrade
Velocity Feedforward (SGVF and SGVFN)	----------	----------	----------	----------	Improve
Acceleration Feedforward (SGAF and SGAFN)	----------	----------	----------	----------	Improve

Proportional Feedback Control (SGP)

NOTE: *The proportional feedback gain (SGP) should never be set to zero, except when open-loop operation is desired.*

***Proportional feedback* is the most important feedback for stabilizing a servo system.** When the 6270 uses *proportional feedback*, the control signal is linearly proportional to the position error (the difference between the commanded position and the actual position—see TPER command). The proportional gain is set by the Servo Gain Proportional (SGP) command. Proportional feedback can be used to make the servo system more responsive, as well as reduce the steady state position error.

Since the control is proportional to the position error, whenever there is any disturbance (such as torque ripple or a spring load) forcing the load away from its commanded position, the proportional control can immediately output a signal to move it back toward the commanded position. This function is called *disturbance rejection.*

If you tune your system using only the proportional feedback, increasing the proportional feedback gain (SGP value) too much will cause the system response to be oscillatory, underdamped, or in some cases unstable.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Integral Feedback Control (SGI)

Using *integral feedback control*, the value of the control signal is integrated at a rate proportional to the feedback device position error. The rate of integration is set by the Servo Gain Integral (SGI) command.

In most hydraulic servo applications, it is best to set the SGI gain to zero (SGIØ,Ø) during the move. Then, after motion has stopped, use the proper SGI gain to hold the position.

The primary function of the integral control is to overcome friction and/or gravity and to reject disturbances so that steady state position error can be minimized or eliminated. This control action is important for achieving high system accuracy. *However, if you can achieve acceptable position accuracy by using only the proportional feedback (SGP), then there is no need to use the integral feedback control.*

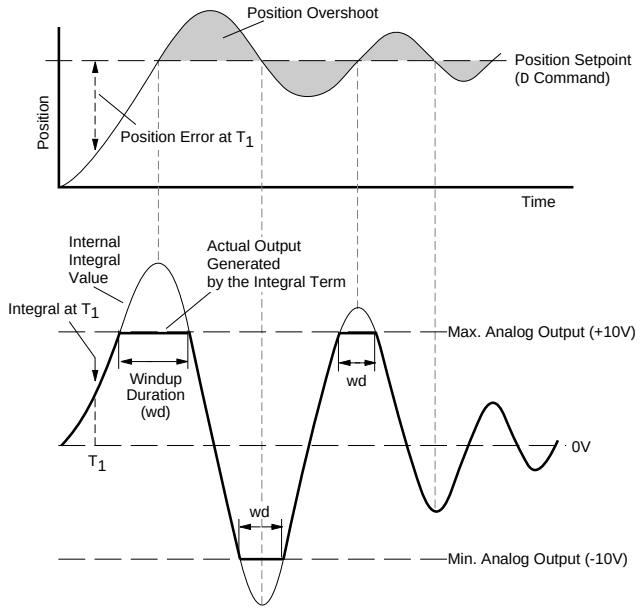
In the task of reducing position error, the integral gain (SGI) works differently than the proportional gain (SGP); this is because the magnitude of its control signal is not dependent on the magnitude of the position error as in the case of proportional feedback. If any position error persists, then the output of the integral term will ramp up over time until it is high enough to drive the error back to zero. Therefore, even a very small position error can be eliminated by the integral feedback control. By the same principle, integral feedback control can also reduce the tracking error when the system is commanded to cruise at constant velocity.

Controlling Integral Windup

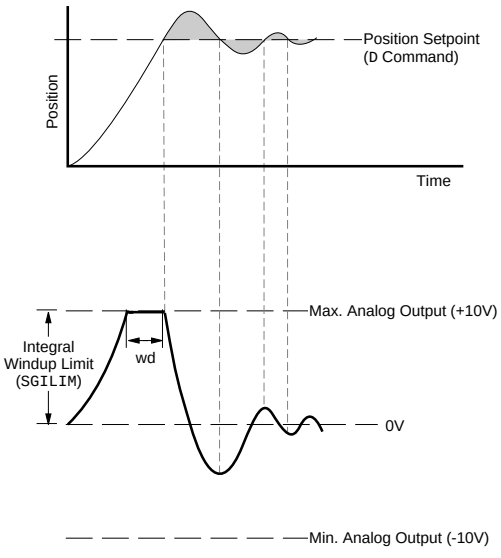
If integral control (SGI) is used and an appreciable position error has persisted long enough during the transient period (time taken to reach the setpoint), the control signal generated by the integral action can end up too high and saturate to the maximum level of the controller's analog control signal output. This phenomenon is called *integrator windup*.

After windup occurs, it will take a while before the integrator output returns to a level within the limit of the controller's output. Such a delay causes excessive position overshoot and oscillation. Therefore, the integral windup limit (SGILIM) command is provided for you to set the absolute limit of the integral and, in essence, turn off the integral action as soon as it reaches the limit; thus, position overshoot and oscillation can be reduced (see illustration below). The application of this feature is demonstrated in Step 4 of the *Tuning Procedure* below.

Without SGILIM



With SGILIM



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Velocity Feedback Control (SGV)

When *velocity feedback control* is used, the control signal is proportional to the feedback device's velocity (rate of change of the actual position). The Servo Gain Velocity (SGV) command sets the gain, which is in turn multiplied by the feedback device's velocity to produce the control signal. Since the velocity feedback acts upon the feedback device's velocity, its control action essentially anticipates the position error and corrects it before it becomes too large. Such control tends to increase damping and improve the stability of the system.

A high velocity feedback gain (SGV) can also increase the position tracking error when traveling at constant velocity. In addition, setting the velocity feedback gain too high tends to slow down (*overdamp*) the response to a commanded position change. If a high velocity feedback gain is needed for adequate damping, you can balance the tracking error by applying velocity feedforward control (increasing the SGVF value—discussed below).

Since the feedback device's velocity is derived by differentiating the feedback device's position with a finite resolution, the finite word truncation effect and any fluctuation of the feedback device's position would be highly magnified in the velocity value, and even more so when multiplied by a high velocity feedback gain. When the value of the velocity feedback gain has reached such a limit, the motor (or hydraulic cylinder, etc.) will *chatter* (high-frequency, low-amplitude oscillation) at steady state.

Velocity Feedforward Control (SGVF)

The purpose of velocity feedforward control is to improve *tracking performance*; that is, reduce the position error when the system is commanded to move at constant velocity. The tracking error is mainly attributed to three sources—friction, torque load, and velocity feedback control (SGV).

Velocity feedforward control is directed by the Servo Gain Velocity Feedforward (SGVF) setting, which is in turn multiplied by the rate of change (velocity) of the commanded position to produce the control signal. Consequently, because the control signal is now proportional to the velocity of the commanded position, the 6270 essentially anticipates the commanded position and initiates a control signal ahead of time to more closely follow (*track*) the commanded position.

Applications requiring linear interpolation can benefit from improved tracking performance; however, *if your application only requires short, point-to-point moves, velocity feedforward control is not necessary*.

Because velocity feedforward control is not in the servo feedback loop (see *Servo Control Algorithm* drawing above), it does not affect the servo system's stability. Therefore, there is no limit on how high the velocity feedforward gain (SGVF) can be set, except when it *saturates the control output* (tries to exceed the 6270's analog control signal range).

Acceleration Feedforward Control (SGAF)

The purpose of acceleration feedforward control is to improve position tracking performance when the system is commanded to accelerate or decelerate.

Acceleration feedforward control is directed by the Servo Gain Acceleration Feedforward (SGAF) setting, which is in turn multiplied by the acceleration of the commanded position to produce the control signal. Consequently, because the control signal is now proportional to the acceleration of the commanded position, the 6270 essentially anticipates the velocity of the commanded position and initiates a control signal ahead of time to more closely follow (*track*) the commanded position.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

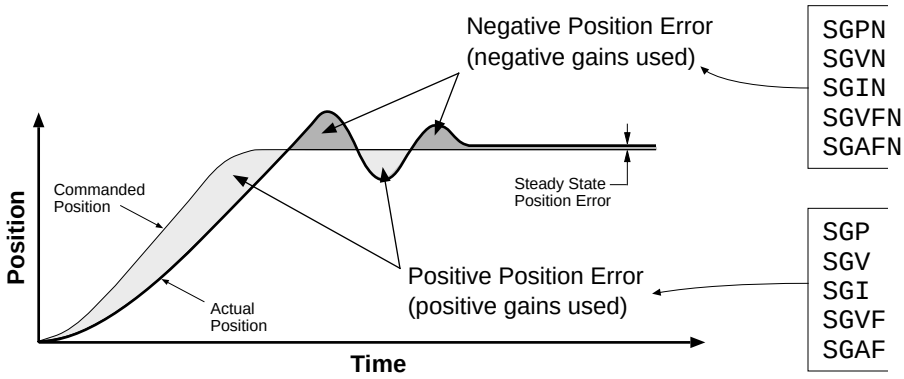
Same as velocity feedforward control, this control action can improve the performance of linear interpolation applications. In addition, it also reduces the time required to reach the commanded velocity. *However, if your application only requires short, point-to-point moves, acceleration feedforward control is not necessary.*

Acceleration feedforward control does not affect the servo system's stability, nor does it have any effect at constant velocity or at steady state.

Negative Gains

The negative gain commands (see list below) allow you to establish gains to be used when the position error is negative.

SGPN .....Proportional Gain Negative
SGVN .....Velocity Gain Negative
SGIN .....Integral Gain Negative
SGVFN.....Velocity Feedforward Gain Negative
SGAFN.....Acceleration Feedforward Gain Negative



The 6270 automatically switches between the positive and negative gains to correlate with the positive or negative position error. This is particularly useful when controlling hydraulic cylinders in which the different surface areas on each side of the piston react differently with the same gain settings.

Each negative gain changes with (tracks) the corresponding positive gain value until a negative gain command is executed. For example, the SGPN value automatically tracks the SGP value until an SGPN command is executed with a value other than the current SGP command value. After the negative gain command is executed, separate positive and negative gain commands must be used.

To re-establish the default mode where the negative gain tracks the positive gain, issue the negative gain command with a minus sign (-) in the command field for the affected axis (e.g., SGPN,- restores axis 2 to the default mode).

Gain Sets

An added dimension to the control techniques discussed in the previous section is to group the gains into "gains sets" that can be invoked to affect motion under certain conditions. Gain sets may be useful for applications in which you would like to invoke different gains a different portions of a move profile, or at rest, or based on an external process, etc.

The SGSET command allows you to save the currently active gains, control signal offsets (SOFFS and SOFFSN), and maximum position error (SMPER) setting, to a specified gain set (see list below).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SGPN .....Proportional Gain Negative
SGVN .....Velocity Gain Negative
SGIN .....Integral Gain Negative
SGVFN.....Velocity Feedforward Gain Negative
SGAFN .....Acceleration Feedforward Gain Negative
SGILIM .....Integral Windup Limit
SOFFS.....Servo Control Signal Offset
SOFFSN .....Servo Control Signal Offset Negative
SMPER.....Maximum Allowable Position Error

The gain set saved with the SGSET command can be enabled/recalled later with the SGENB command or the SSWG command. Using the SGENB command, the gains can be enabled during motion at any specified point in the profile, or when not in motion (see programming example below). If using the SSWG command, the gain set is referred to as the "setpoint window gain set" and is invoked after the commanded profile is complete (see *Setpoint Window Gains* below for details).

NOTE

The tuning gains saved to a given gain are specific to the current feedback source (selected with the last SFB command) at the time the gains were saved with the SGSET command. Later, when you enable the saved gain set, **make sure that the gain set you enable is appropriate to the feedback source you are using at the time**.

To display the gain values currently in effect, use the TGAIN command. To display the contents of a particular gain set, use the TSGSET command.

Setpoint Window Gains

You may activate a specified set of gains to be used when with a defined region either side of the position setpoint ("setpoint window").

The setpoint window is defined by the SSWD command, which establishes the distance on both sides of the position setpoint ("setpoint window") in which the gain set specified with the SSWG command is used. Specifically, the gain set is automatically invoked by the controller *after* the commanded move profile is complete and the position error is within the setpoint window. The SSWG gain set accommodates different proportional, integral, and velocity gains and offsets for each direction.

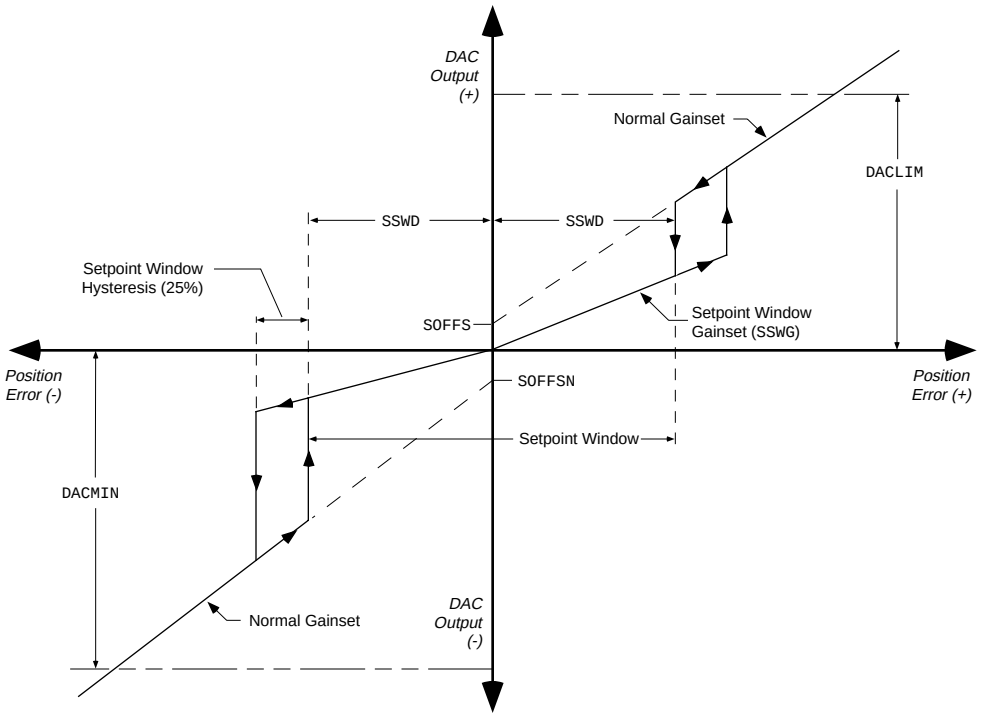
The setpoint window includes a hysteresis loop equal to 25% of the value used in the SSWD command.

If you want to disable the servo control loop when the position error is within the setpoint window, set all of the gain values to zero—you can do this by executing all the individual gain commands or by executing the SFBØ command. If you want to disable the offsets (SOFFS & SOFFSN) when the position error is within the setpoint window, set them to zero volts.

To disable the setpoint window gain feature, use the SSWGØ command.

To assign a gain set as the setpoint window gain set with the SSWG command, you must first define/save the gain set with the SGSET command (see programming example below).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



The diagram above makes two assumptions. First, for simplicity, only a proportional gain is being used. Second, related to the SSWG gain set, the proportional gains (SGP and SGPN) are lower than those used in the normal gain set and the offsets (SOFFS and SOFFSN) are set to zero.

The arrows on the above diagram illustrate the hysteresis loop. The SSWG gains are used until the position error increases to a value of 25% greater than the number specified in the SSWD command. At this point, the normal gain set is automatically substituted until the position error falls below the value in the SSWD command when the SSWG gains are returned.

Finally, the maximum positive DAC voltage is determined by the value in the DACLIM command and the maximum negative value is determined by the DACMIN command.

Target Zone — An alternative to SSWD

You can use the target zone settling mode to override the setpoint window distance (SSWD) and introduce distance and velocity end-of-move criteria to define when the controller switches to the setpoint window gains. When using the target zone settling mode (enabled with the STRGTE command), after completion of the commanded move profile, the actual position and actual velocity must be within the "target zone" (i.e., within the position band defined by STRGTD and within the velocity band defined by STRGTV) before motion can be determined complete. After that point, the controller will switch to the SSWG gains. For more information on the target zone mode, refer to the *Target Zone* section later in this chapter.

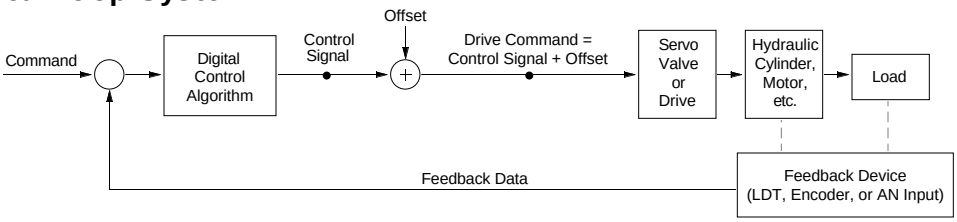
Programming Example

Example	Description
> SGP35	Set proportional gain to 35
> SGI3	Set integral gain to 3
> SGSET3	Save current gains as gain set #3 (to use when within the setpoint window)
> SSWG3	Assign gain set #3 as the setpoint window gain set
> SGP5Ø	While moving, use higher proportional gain
> SGIØ	While moving, use no integral gain,
> SGV4	While moving, use introduce velocity gain
> SSWD1ØØ	After the commanded move profile is complete, the controller will use gain set #3 if within 100 counts (125 counts including hysteresis) of the setpoint position

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

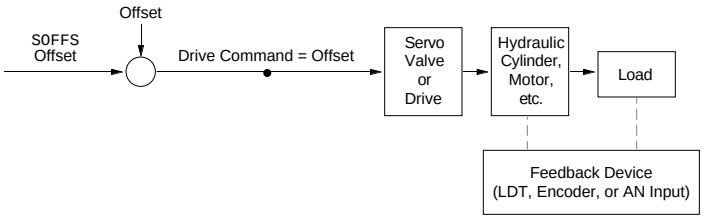
When the control algorithm is used, the whole servo system is a closed-loop system (see diagram below). It is called closed loop because the control algorithm accounts for both the command (position, velocity, tension, etc.) and the feedback data (from the LDT, encoder, or ANI input); therefore, it forms a *closed loop* of information flow.

Closed Loop System



When all gains are set to zero, the digital control algorithm is essentially disabled and the system becomes an open loop system (see diagram below). Open loop operation could be used during system setup or troubleshooting so that you can independently test the drive/motor or valve operation (the *Tuning Setup Procedure* below demonstrates open-loop operation).

Open Loop System



There are methods of entering the open loop mode:

- Disable each individual positive and negative gain value. This allows the controller to monitor the position error and signal a fault if the maximum limit is exceeded.
- Issue the SFBØ command. This automatically invokes the following conditions:
  - **WARNING** — The hardware and software end-of-travel limits are disabled. Make sure that it is safe to operate without end-of-travel limits before using the open-loop function.
  - Gain values (SGILIM, SGAF, SGAFN, SGI, SGIN, SGP, etc.) set to zero (open-loop operation).
  - SMPER value set to zero (position error is allowed to increase without causing a fault).
  - Subsequent attempts to change gain values or SMPER will cause an error message ("NOT ALLOWED IF SFBØ").
  - SOFFS and SOFFSN set to zero, but allows subsequent servo offset changes to affect motion.
  - SSWG set to zero (disables the Setpoint Window Gains feature).
  - Disables output-on-position (OUTPA - OUTPD) functions.
  - Any subsequent changes to PSET, PSETCLR, SCLD, SCLA, SCLV, SOFFS, and SOFFN are lost when another feedback source is selected.

  Recommendation — Use the Disable Drive On Kill mode, enabled with the KDRIVE command, so that the 6270 will shut down the valve/drive if a kill command (e.g., !K) is executed or if a kill input is activated. **CAUTION**: Shutting down a valve/cylinder system returns the valve to the null position; shutting down a rotary drive system allows the load to freewheel if there is no brake installed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tuning Setup Procedure

Use the following procedure to set up your servo system before completing the tuning procedures. You can perform this procedure for both axes simultaneously.

Before you set up for tuning:

Do not begin this procedure unless you are sure you have successfully completed these system connection, test, and configuration procedures provided in Chapter 3:

- Connect the valve or the drive (especially the drive's shutdown output).
- Connect and test the feedback devices.
- Connect and test the end-of-travel limits.
- Test the 6270's analog output.
- Attach the load and the feedback devices as required for your application.
- Configure the number of axes in use, drive fault level (if using a rotary drive), and feedback device resolution.
- Select the appropriate feedback source per axis with the SFB command (tuning parameters for each axis are specific to the currently selected feedback source).



WARNING



The tuning process requires operation of your system's electrical and mechanical components. Therefore, you should test your system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

EMERGENCY SHUTDOWN: You should be prepared to shut down the valve or drive during the tuning process (for instance, if the system becomes unstable or experiences a runaway). You can use the **ENBL** input (disconnect it from ground) to disable the 6270's analog output signal. An alternative is to issue the `@DRIVEØ` command to the 6270 over the communication interface, but this requires connecting a shutdown output to the drive. If the drive does not have a shutdown input, use a manual emergency stop switch to disable the valve's/drive's power supply.

Step 1 **Drive Users Only**: If you are using a rotary drive, make sure the power to the drive is off.

Step 2 Apply power to the 6270 only and issue the `DRIVE11` command. Measure the 6270's analog output between the **CMD+** and **CMD-** terminals on the **DRIVE** connector with both an oscilloscope to check for noise and a digital volt-meter (DVM) to monitor the analog output. Both readings should be very close to zero. If an offset exists, ignore it for now; it will be taken care of later in step 8.

Step 3 If your system has mechanical stops, manually move the load to a position mid-way between them.

Step 4 Enter these commands to zero all the gains and run the system in open loop:

Open-Loop Operation ☞

Command	Description
> SGPØ,Ø	Set the proportional feedback gain to zero
> SGVØ,Ø	Set the velocity feedback gain to zero
> SGIØ,Ø	Set the integral feedback gain to zero
> SGVFØ,Ø	Set the velocity feedforward gain to zero
> SGAFØ,Ø	Set the acceleration feedforward gain to zero

Step 5 **Drive Users Only**: Apply power to the drive. The motor shaft should be stationary or perhaps turning very slowly (velocity drives only). *A small voltage to a torque drive, with little or no load attached, will cause it to accelerate to its maximum velocity. Since the torque demand at such a low voltage is very small, you can prevent the shaft from moving by holding it.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 6 Observe the 6270's analog output noise level on the oscilloscope. Typically, the ideal noise level should be below 3.0mV, but inevitably you must determine the acceptable noise level for your application.

☞ *The control method (voltage or current output) is selected with internal jumpers. The default is voltage output. If you required current output, refer to the instructions in Chapter 6 to change the appropriate jumpers.*

If the noise level is acceptable, proceed to Step 7. If the noise level is too high:

a. Turn all the power off and tie the earth grounds of all the electrical components of your system to a single common earth ground point.

b. Shield the drive or valves properly and shield all the wiring that interconnect the components.

c. After you have completed a and b above, turn on the controller only and start over from Step 2. If the noise level is still unacceptable, consult the noise suppression techniques described in Appendix A.

Step 7 The purpose of this step is to ensure that a positive voltage on the 6270's analog control signal output (from the **CMD+** and **CMD-** terminals) results in the feedback device counting in the positive direction.

a. Using the SMPER command, set the maximum allowable position error to 1 inch by entering the SMPER1,1 command. This assumes the default scaling factor (1 distance value = 432 counts, or 1 inch) is still in effect.

b. Enter the TFB command to check the current position of the feedback devices. Record this number for later use.

c.

CAUTION

The offset introduced in this step may cause an acceleration to a high speed, if there is little or no load.

Enter the SOFFSØ.2 command to introduce an offset DAC output value of 0.2V to make the servo mechanism move slowly in the positive (clockwise or extension) direction. (Motion will stop when the maximum allowable position error is exceeded.) *If the load has a large stiction component, you may need to use a larger offset (SOFFS command) to overcome stiction and affect motion.*

d. Use the TFB command again to observe the feedback device's position. The value should have increased from the value observed in Step 7.b.

If the position reading decreases when using a positive SOFFS setting, turn off the 6270 (and the drive, if using one) and swap the **CMD+** and **CMD-** connections either at the 6270 or at the valve/drive, whichever is more accessible (this will not work for servo valves/drives that do not accept differential input). Then turn on the 6270 again, enter the DRIVE11 command, and repeat Steps 4 through 7.d. before proceeding to Step 8.

As an alternative to swapping **CMD+** *and* **CMD-***, you could use the appropriate feedback polarity reversal command (*LDTPOL, ENCPOL, *or* ANIPOL*).*

e. Enter the SOFFSØ command to *stop* the motor, and enter the DRIVE11 command to re-enable the drives.

Step 8 Having set the servo output offset to zero with the SOFFSØ command (see Step 7.e.), read the 6270's analog output with the DVM to determine if there is any offset caused by the electrical interconnections between the 6270 and the valve or drive.

☞ *If you are using current control, convert the offset from milliamps to volts (output range in mA ÷ 10V = mA/V) and enter the result in the* SOFFS *command.*

If the DVM reads anything other than zero, enter the DVM's reading (but with the opposite polarity) as the offset adjustment with the SOFFS command. For example, if the DVM reading is 0.015V, then enter SOFFS-Ø.Ø15. If, after doing this, the reading is still not zero, then fine-tune it by trying SOFFS entries of slightly different values until the DVM reading is between ±3.0mV.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 9 **Drive Users Only**: If you are using a velocity drive, motion may still be occurring due to the drive's balance/offset setting. If so, adjust the drive's balance/offset until motion stops. Consult the drive's user documentation for instructions.

Step 10 Proceed to the *Drive Tuning Procedure* section to tune the velocity drive (if you are using a torque drive or a valve, skip to the *Controller Tuning Procedure*).

Drive Tuning Procedure (Velocity Motor Drives Only)

The goals of the *Drive Tuning Procedure* are to:

1. Tune the drive to output the desired velocity at a given voltage from the 6270.
2. Tune the drive (iteratively) to achieve the desired response.

> **NOTE**
> Be sure to complete the *Tuning Setup Procedure* before proceeding with the following drive tuning procedure. Unlike the *Tuning Setup Procedure*, you must **tune one axis at a time**.

Step 1 **Tune the drive to output the desired velocity at a given voltage from the 6270**:

a. If your system has mechanical stops, manually move the load to a position mid-way between them.

b. Enter the SOFFS command to set the 6270's output voltage to its maximum level, 10.0 volts (SOFFS1Ø for axis 1, or SOFFS,1Ø for axis 2).

c. Adjust the drive gain factor (sometimes called the *tach gain*) such that when the 6270's command output is 10V, the velocity just reaches its maximum value (check the velocity with the TVELA command). Refer to your drive's user documentation if necessary.

> **EXAMPLE**
> Suppose your drive can run at a max. velocity of 7000 rpm (or 116.67 rps). If the drive gain factor is 20 rps/V, then the drive will reach the maximum velocity (116.67 rps) when the 6270's command output is only 5.833V. This means the full range of ±10V is not fully usable. To use the full range of ±10V, the gain factor has to be adjusted to 11.667 rps/V.
>
> Drive manufactures usually provide a potentiometer for adjusting this gain factor. Some manufacturers provide a few preset values selectable with jumpers or DIP switches.

Step 2 **Tune the drive (iteratively) to achieve the desired response**:

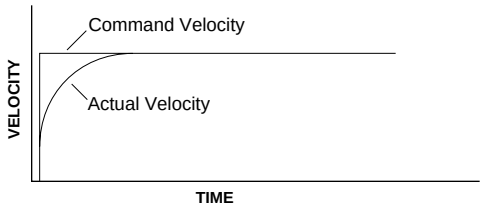
a. Enter the following commands to create and execute a step velocity command:

Command	Description
> DEF STEP	Begin definition of the program called STEP
- @SGPØ	Set the SGP gain to zero
- @SGIØ	Set the SGI gain to zero
- @SGVØ	Set the SGV gain to zero
- @SGAFØ	Set the SGAF gain to zero
- @SGVFØ	Set the SGVF gain to zero
- @SMPERØ	Disable checking the maximum allowable position error
- @SOFFSØ.5	Set the command output to 0.5 volts
- T1	Wait for 1 second
- @SOFFSØ	Set the command output to zero volts (stopping the motor)
- @SMPER1	Re-enable checking the maximum allowable position error
- END	End definition of the program called STEP
> STEP	Execute the program called STEP (the motor will move for 1 second and then stop)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

b. Observe the plot of the commanded velocity versus the actual velocity on the oscilloscope.

Using the tuning methods specified in the drive's user documentation, tune the drive to achieve a first-order response (no overshoot) as illustrated below—repeat Steps 2.a. and 3.b. as necessary.



Step 3 **Proceed to the *Controller Tuning Procedure* section to tune the 6270.**

Controller Tuning Procedure

The *Controller Tuning Procedure* leads you through the following steps:

1. Setup up for tuning.
2. Select the 6270's servo Sampling Frequency Ratios (SSFR).
3. Set the Maximum Position Error (SMPER).
4. Optimize the Proportional (SGP) and Velocity (SGV) gains.
5. Use the Integral Feedback Gain (SGI) to reduce steady state error.
6. Use the Velocity Feedforward Gain (SGVF) to reduce position error at constant velocity.
7. Use the Acceleration Feedforward Gain (SGAF) to reduce position error during acceleration and deceleration.

Before you tune the 6270:

Be sure to complete the *Tuning Setup Procedure* (and the *Drive Tuning Procedure*, if you are using a velocity drive) before proceeding with the following tuning procedure. Unlike the *Tuning Setup Procedure*, you must **tune one axis at a time**.

If your application requires switching between feedback sources on the same axis, then for each feedback source on each axis you must select the feedback source with the SFB command and repeat steps 3-7.

Step 1 **Set up for tuning**:

Use a computer (with a terminal emulator) or a dumb terminal to enter the commands noted in the steps below. To monitor system performance, you may use visual inspection, or use an analog type position transducer (potentiometer, LVDT, RVDT, etc.) to pick up the load's or motor's position displacement and monitor the transducer output on a digital storage oscilloscope.

Step 2 **Select the sampling frequency ratios (SSFR) and max. position error (SMPER)**

The 6270's control signal is computed by the digital signal processor (DSP). The velocity of the commanded position, the velocity of the feedback device's position, and the integral of the position error are used for various control actions. These measurements are derived by the DSP from the position values sampled periodically at a fixed rate; this sampling rate is called the *servo sampling frequency* (samples/second).

NOTE

The SSFR setting affects the dither frequency ratio (SDTFR setting) and the LDT position update rate (LDTUPD setting). If the sampling rate is too fast for the LDT, position errors or bad LDT reads will occur. The occurrence of read errors can be monitored with the TAS and [AS] command bit #27. Refer to the SSFR, SDTFR, and LDTUPD command descriptions in the ***6000 Series Software Reference Guide*** for more details.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Higher sampling frequencies improve the accuracy of the velocity and integral values derived. A higher sampling frequency can also improve the tracking of a rapidly changing or oscillating position. Therefore, the servo sampling frequency is a key parameter that influences the servo system's stability and closed loop bandwidth.

In addition to computing the 6270's control signal, the DSP also computes the commanded position trajectory. When the servo sampling frequency is increased, the motion trajectory update rate has to be decreased, and vice versa. The ratio between the servo sampling frequency and the trajectory update rate, called the *sampling frequency ratio*, depends on the requirements of your application and/or the dynamic characteristics of the system. The Servo Sampling Frequency Ratio (SSFR) command offers four selectable ratio settings. These four ratios and the actual sampling frequencies and sampling periods (reciprocal of sampling frequency) are shown below.

NOTE
Changing the active axes with the INDAX command will change the SSFR ratio.

	# of Axes Active (INDAX)	SSFR Command Setting	Servo Sampling Update Frequency (samples/sec.)	Servo Sampling Update Period (μsec)	Motion Trajectory Update Frequency (samples/sec.)	Motion Trajectory Update Period (μsec)	System Update Frequency (samples/sec.)	System Update Period (μsec)
	INDAX1	SSFR1	3030	330	3030	330	757	1320
	INDAX1	SSFR2	4000	250	2000	500	500	2000
	INDAX1	SSFR4	4651	215	1163	860	581	1720
	INDAX1	SSFR8	4878	205	610	1640	610	1640
	INDAX2	SSFR1	1667	600	1667	600	417	2400
	INDAX2	SSFR2	2272	440	1136	880	568	1760
Default →	INDAX2	SSFR4	2500	400	625	1600	625	1600
	INDAX2	SSFR8	2667	375	333	3000	333	3000

The general rule to determining the proper SSFR value is to first select the slowest servo sampling frequency that is able to give a satisfactory response. This can be done by experiment or based on the closed-loop bandwidth requirement for your application. (Keep in mind that increasing the SSFR value allows for higher bandwidths, but produces a rougher motion profile; conversely, decreasing the SSFR value provides a smoother profile, but makes the servo system less stable and slower to respond.)

As an example, if your application requires a closed-loop bandwidth of 300 Hz, you can determine the minimum servo sampling frequency by using the rule of thumb—setting the servo sampling frequency at least 8 times higher than the bandwidth frequency—the required minimum servo sampling frequency would be 2400 Hz. If two axes are running (INDAX2), then you should try using the SSFR4 setting.

The table below provides guidelines for various application requirements.

Application Requirement	SSFR1	SSFR2	SSFR4	SSFR8
XY Linear Interpolation	✔	✔		
Fast point-to-point motion			✔	✔
Regulation (speed, torque, etc.)			✔	✔
High natural frequency system				✔

Setting the Sampling Frequency Ratio

Select a sampling ratio (with the SSFR command) appropriate to your system now, before you proceed to tune each gain.

If you change the sampling frequency ratios (SSFR) after the tuning is complete and the new servo sampling frequency is lower than the previous one, the response may change (if your system bandwidth is quite high) and you may have to re-tune the system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 3 **Set the Maximum Position Error (SMPER)**:

The SMPER command allows you to set the maximum position error allowed before an error condition occurs. The position error, monitored once per system update period, is the difference between the commanded position and the actual position as read by the feedback device selected with the last SFB command. *Larger values allow greater oscillations/motion when unstable; therefore, smaller SMPER values are safer.*

When the position error exceeds the value entered by the SMPER command, an error condition is latched (see TAS or AS bit #23) and the 6000 controller issues a shutdown to the faulted axis and sets its analog output command to zero volts. To enable the system again, the appropriate DRIVE1 command must be issued, which also sets the commanded position equal to the actual feedback device position (incremental devices will be zeroed).

If the SMPER value is set to zero, the position error condition is not monitored, allowing the position error to accumulate without causing a fault.

Step 4 **Optimize the Proportional (SGP) and Velocity (SGV) gains (*see illustration for tuning process*)**:

a. Enter the following commands to create a step input profile *(use a comma in the first data field when tuning axis 2—e.g.,* D,100*)*:

Command	Description
> A999	Set acceleration to 999 units/sec^2
> AD999	Set deceleration to 999 units/sec^2
> V30	Set velocity to 30 units/sec
> D100	Set distance to 100 units

b. Start with an SGP command value of 0.5 (SGP0.5 or SGP,0.5).

c. Enter the GO1 or GO,1 command depending on which axis is being tuned at the time.

d. Observe the plot of the commanded position versus the actual position on the oscilloscope. If the response is already very oscillatory, lower the gain (SGP); if it is *sluggish* (overdamped), increase the SGP gain.

Repeat Steps 4.c. and 4.d. until the response is slightly under-damped.

e. Start with an SGV command value of 0.1 (SGV0.1 or SGV,0.1).

f. As you did in Step 4.c., enter GO1 or GO,1.

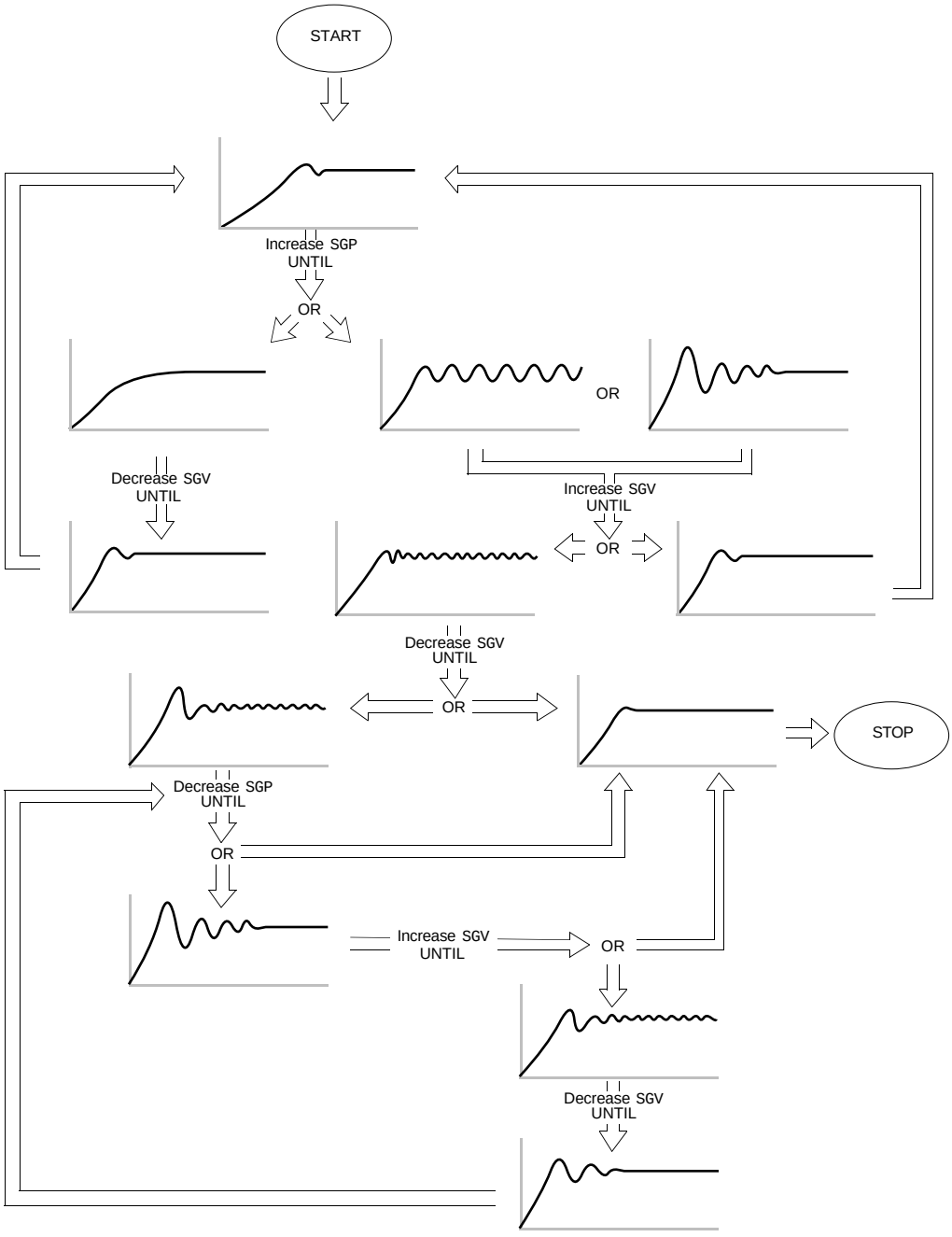
g. Observe the plot on the oscilloscope. If the response is *sluggish* (overdamped), reduce the SGV gain. *Repeat Steps 4.f. and 4.g. until the response is slightly under-damped.*

☞ *Refer to the* Tuning Scenario *section later in this chapter for a case example.*

h. The flow diagram below shows you how to get the values of the proportional and velocity feedback gains for the fastest, well-damped response in a step-by-step fashion. The tuning principle here is based on these four characteristics:

- Increasing the proportional gain (SGP) can speed up the response time and increase the damping.
- Increasing the velocity feedback gain (SGV) can increase the damping more so than the proportional gain can, but also may slow down the response time.
- When the SGP gain is too high, it can cause instability.
- When the SGV gain is too high, it can cause the motor (or valve, hydraulic cylinder, etc.) to chatter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Step 5 **Use the Integral Feedback Gain (SGI) to reduce steady state error**:

a. Determine the steady state position error (the difference between the commanded position and the actual position). You can determine this error value by the TPER command when the load is not moving.

Steady state position error is described earlier in the Performance Measurements *section.*

NOTE

If the steady state position error is zero or so small that it is acceptable for your application, **you do not need to use the integral gain**. For hydraulic applications, it is usually best to use a small SGI value, or use SGIØ while moving and use SGIn when stopped. The use of the Target Zone Settling Mode (STRGTE) is recommended.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

b. If you have to enter the integral feedback gain to reduce the steady error, start out with a small value (e.g., SGIØ.1). After the gain is entered, observe two things from the response:

- Whether or not the magnitude of steady state error reduces
- Whether or not the steady state error reduces to zero at a faster rate

c. Keep increasing the gain to further improve these two measurements until the overshoot starts to increase and the response becomes oscillatory.

d. There are three things you can do at this point (If these three things do not work, that means the integral gain is too high and you have to lower it.):

1st Lower the integral gain (SGI) value to reduce the overshoot.

2nd Check whether the 6270's analog output saturates the ±10V limit; you can do this by observing the signal from a digital oscilloscope. If it saturates, then lower the integral output limit by using the SGILIM command. This should help reduce the overshoot and shorten the settling time. Sometimes, even if the analog output is not saturated, you can still reduce the overshoot by lowering SGILIM to a value less than the maximum output value. *However, lowering it too much can impair the effectiveness of the integral feedback.*

☞ *If you are using current control, convert the offset from milliamps to volts and enter the result in the* SGILIM *command.*

3rd You can still increase the velocity feedback gain (SGV value) further, provided that it is not already at the highest possible setting (causing the motor or valve to chatter).

Step 6 **Use the Velocity Feedforward Gain (SGVF) to reduce position error at constant speed**:

a. Execute a continuous (MC1 command) move, setting the acceleration, deceleration and velocity values appropriate to your application. Set the SGVF value to be the product of SGP * SGV (if SGV = zero, set SGVF equal to SGP).

b. Check the position error at constant velocity by issuing the TPER command.

c. Increase SGVF to reduce the position error (repeat steps a and b as necessary).

Step 7 **Use the Acceleration Feedforward Gain (SGAF) to reduce position error during acceleration**:

a. Execute a continuous (MC1 command) move, setting the acceleration, deceleration and velocity values appropriate to your application. Set SGAF to 0.01 (SGAFØ.Ø1).

b. Check the position error during acceleration by issuing the TPER command.

c. Increase SGAF to reduce the position error (repeat steps a and b as necessary).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tuning Scenario

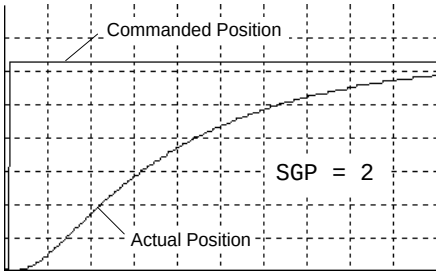
This example shows how to obtain the highest possible proportional feedback (SGP) and velocity feedback (SGV) gains experimentally by using the flow diagram illustrated earlier in Step 4 of the *Tuning Procedure*.

> **NOTE**
> The steps shown below (steps 1 - 11) represent the major steps of the process; the actual progression between these steps usually requires several iterations.

The motion command used for this example is a step command with a step size of 100. The plots shown are as they might appear on a scope (X axis = time, Y axis = position).

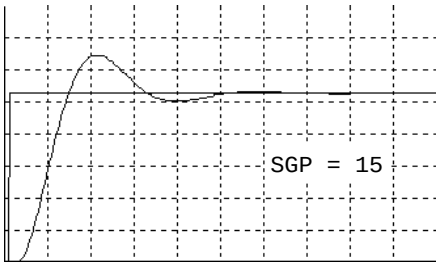
Step 1 For a starting trial, we set the proportional feedback gain (SGP) to 2. As you can see by the plot, the response is slow.

In the next step, we should increase SGP until the response is slightly underdamped.



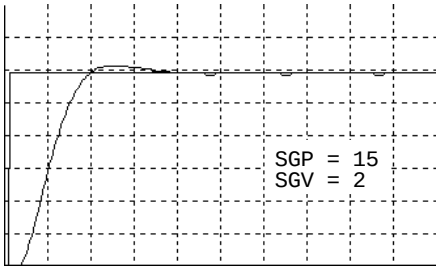
Step 2 With SGP equal to 15, the response becomes slightly underdamped (see plot).

Therefore, we should introduce the velocity feedback gain (SGV) to *damp out* the oscillation.



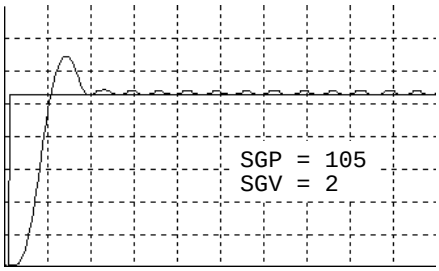
Step 3 With SGV equal to 2, the response turns out fairly well damped (see plot).

At this point, the SGP should be raised again until oscillation or excessive overshoot appears.



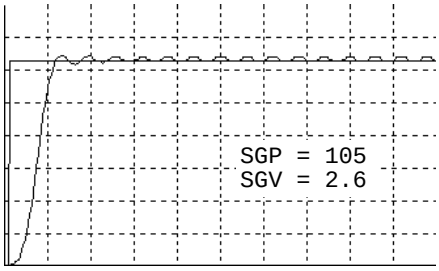
Step 4 As we iteratively increase SGP to 105, overshoot and chattering becomes significant (see plot). This means either the SGV gain is too low and/or the SGP is too high.

Next, we should try raising the SGV gain to see if it could damp out the overshoot and chattering.



Step 5 After the SGV gain is raised to 2.6, the overshoot was reduced but chattering is still quite pronounced. This means either one or both of the gains is too high.

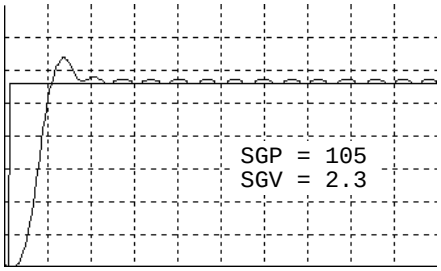
The next step should be to lower the SGV gain first.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

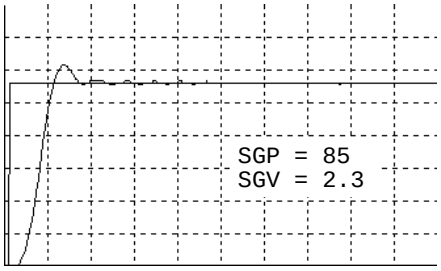
Step 6 Lowering the SGV gain to 2.3 does not help reduce the chattering by much.

Therefore, we should lower the SGP gain until chattering stops.



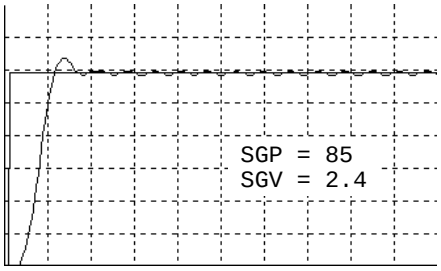
Step 7 Chattering stops after reducing the SGP gain to 85. However, the overshoot is still a little too high.

The next step should be to try raising the SGV to damp out the overshoot.



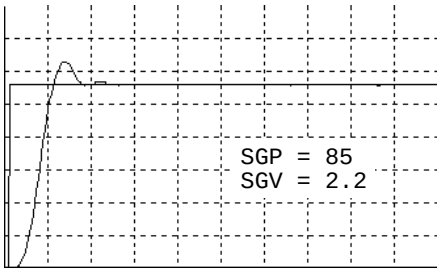
Step 8 After raising the SGV gain to 2.4, overshoot is reduced a little, but chattering reappears. This means the gains are still too high.

Next, we should lower the SGV gain until chattering stops.



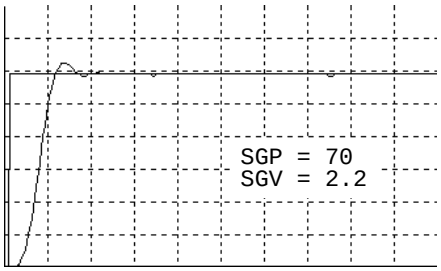
Step 9 After lowering the SGV gain to 2.2 (even less than in Step 7—2.3), chattering stops.

Next we should lower the SGP gain.

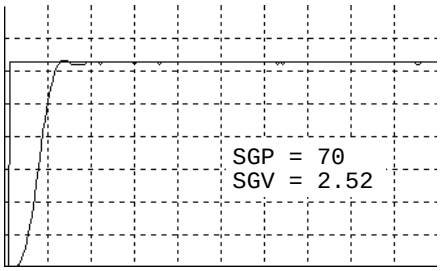


Step 10 Overshoot is reduced very little after lowering the SGP gain to 70. (The SGV gain might have been lowered too much in Step 9.)

Next, we should try raising the SGV gain again until the overshoot is gone.



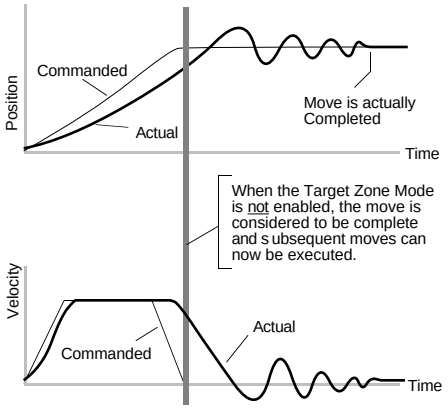
Step 11 When we raised the SGV gain to 2.52, the step response became fast and very stable.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Target Zone (Move Completion Criteria)

Under default operation (Target Zone Mode not enabled), the 6270's move completion criteria is simply derived from the move trajectory. The 6270 considers the current preset move to be complete when the commanded trajectory has reached the desired target position; after that, subsequent commands/moves can be executed for that same axis. Consequently, the next move or external operation can begin before the actual position has settled to the commanded position (see diagram).



To prevent premature command execution before the actual position settles into the commanded position, use the *Target Zone Mode*. In this mode, enabled with the STRGTE command, the move cannot be considered complete until the actual position and actual velocity are within the *target zone* (that is, within the distance zone defined by STRGTD and less than or equal to the velocity defined by STRGTV). If the load does not settle into the target zone before the timeout period set with the STRGTT command, the 6270 detects a *timeout error* (see illustration below).

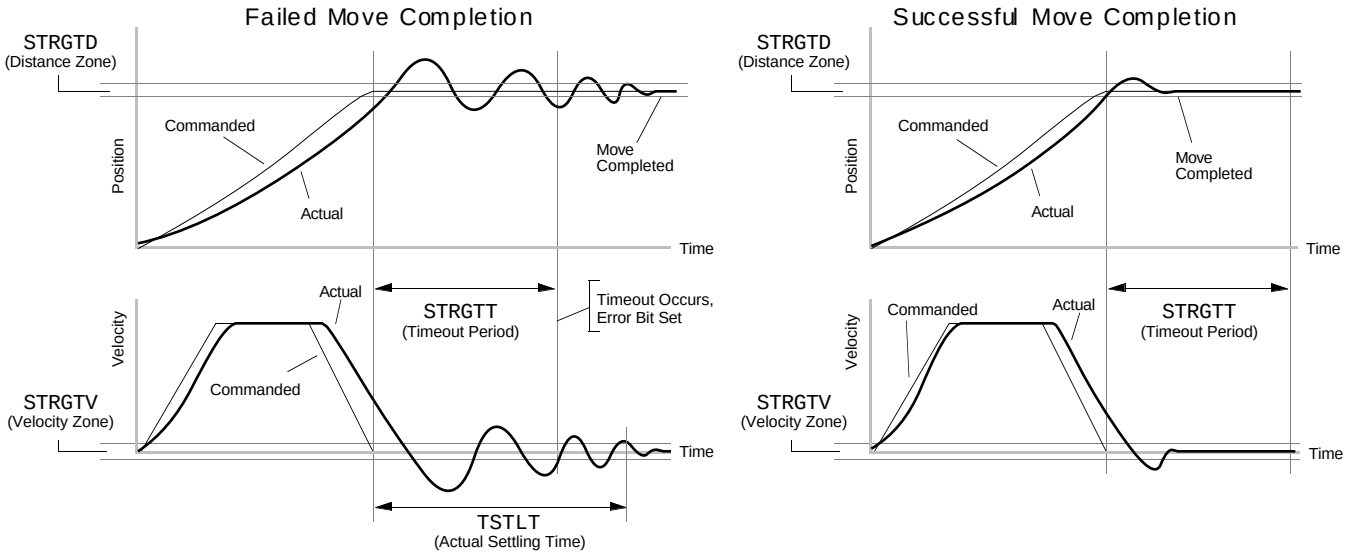
Refer to the Error Handling *section in the* **6000 Series Software Reference Guide** *for error program examples*

If the timeout error occurs, you can prevent subsequent command/move execution only if you enable the ERROR command to continually check for this error condition, and when it occurs to branch to a programmed response you can define in the ERRORP program.

As an example (assuming scaling is enabled and default scaling values are used), setting the distance zone to ±0.01 inches (STRGTD.Ø1), the velocity zone to ≤0.5 ips (STRGTVØ.5), and the timeout period to 1/2 second (STRGTT5ØØ), a move with a distance of 8 inches (D8) must end up between position 7.99 and 8.01 and settle down to ≤0.5 ips within 500 ms (1/2 second) after the commanded profile is complete.

Damping is critical.

To ensure that a move settles within the distance zone, it must be damped to the point that it will not move out of the zone in an oscillatory manner. This helps ensure the actual velocity falls within the target velocity zone set with the STRGTV command (see illustration below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Checking the Actual Settling Time

Using the TSTLT command, you can display the actual time it took the last move to settle into the target zone (that is, within the distance zone defined by STRGTD and less than or equal to the velocity defined by STRGTV). The reported value represents milliseconds. **This command is usable whether or not the Target Zone Settling Mode is enabled with the STRGTE command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑤

Feature Implementation

The information in this chapter will enable you to understand and implement the 6270's features into your application:

- Support Software (Motion Architect, DOS Disk)
- Safety Features
- Scaling
- End-of-Travel Limits
- Homing
- Positioning Modes
- Dithering Hydraulic Servo Valves
- User Interface Options:
  - Programmable I/O
  - Thumbwheel Interface
  - I/O Device Interface (including PLCs)
  - Joystick Interface
  - 14-Bit Analog Input Interface **(6270-ANI Option only)**
  - RP240 Remote Operator Panel Interface
  - Host Compumotor Control
- Variables
- Teach Mode
- S-Curve Profiling
- X-Y Linear Interpolation
- RS-232C Daisy-chaining

See Also, Chapter 4

Tuning-related features are described in chapter 4:

- *Selecting a servo sampling rate (SSFR)*
- *Selecting a max. position error (SMPER)*
- *Tuning Procedures*
- *Gains*
- *Gain Sets*
- *Setpoint Window Gain Sets*
- *Target Zone Settling Mode*

Before You Proceed With This Chapter

⚠ WARNING ⚠

Most of the features described in this chapter are used for operating your system's electrical and mechanical components. Therefore, before proceeding with this chapter, you should test your system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel. Be sure to complete all the installation and test procedures provided in Chapter 2 and Chapter 3, and complete the tuning procedures in Chapter 4 or the ***Servo Tuner User Guide***.

EMERGENCY SHUTDOWN: At all times, you should be prepared to shut down the valve or drive (e.g., if the system becomes unstable or experiences a runaway). You can use the **ENBL** input (disconnect it from ground) to disable the 6270's analog output signal. An alternative is to issue the `@DRIVEØ` command to the 6270 over the communication interface, but this requires connecting a shutdown output to the drive. If the drive does not have a shutdown input, use a manual emergency stop switch to disable the valve's/drive's power supply.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6000 Series Software Reference Guide

Since this chapter often refers to the 6000 Series Command Language employed by the 6270's operating system, keep the ***6000 Series Software Reference Guide*** nearby as a reference for programming guidelines and detailed command descriptions.

Support Software

The 6270 is shipped with two support software tools, Motion Architect™ and the *6000 DOS Support Disk*.

6000 DOS Support Disk

The *6000 DOS Support Disk* (p/n 95-012266-01) contains a program that provides terminal emulation and program editing capabilities specifically designed for use with any 6000 Series stand-alone product. Also included on the disk are sample 6000 command language programs. For more detailed user information, refer to the ***6000 DOS Support Disk Quick Reference***.

Motion Architect®

Motion Architect® is an intuitive Microsoft® Windows™ based programming tool. A brief description of Motion Architect's basic features is provided below. For more detailed user information, refer to the ***Motion Architect User Guide***.

- **System Configurator and Code Generator**: Automatically generate controller code of basic system set-up parameters (I/O definitions, encoder operations, etc.).
- **Tuning and Data Gathering Tool (*Servo Tuner* option for Motion Architect)**: Tune the 6270 and the attached servo drives (if any) and receive instant data feedback on customizable displays. The Servo Tuner™ option is an add-on module and does not automatically come with the basic Motion Architect software package. To order your copy of Motion Architect Servo Tuner, contact your local Automation Technology Center.
- **Program Editor**: Create blocks or lines of 6270 controller code, or copy portions of code from previous files. You can save program editor files for later use in BASIC, C, etc., or in the terminal emulator or test panel.
- **Terminal Emulator**: Communicating directly with the 6270, the terminal emulator allows you to type in and execute controller code and transfer code files to and from the 6270.
- **Test Panel and Program Tester**: You can create your own test panel to run your programs and check the activity of I/O, motion, system status, etc. This can be invaluable during start-ups and when fine tuning machine performance.
- **On-line Context-sensitive Help and Command Reference**: These on-line resources provide help information about Motion Architect, as well as interactive access to the contents of the ***6000 Series Software Reference Guide***.
- Options (available from your local Automation Technology Center or distributor):
  - **Servo Tuner™** (Tuning and Data Gathering Tool): Tune the 6270 and receive instant data feedback on customizable displays. The Servo Tuner option is an add-on module and does not automatically come with the basic Motion Architect software package.
  - **CompuCAM™**: CAD-to-Motion software allows you to translate DXF, HP-GL, and G-Code files into 6000 Series Language motion programs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6270 Safety Features

To help ensure a safe operating environment, you should take advantage of the 6270's safety features listed in the table below.

Feature	Description	See Also
Enable Input	The enable input (**ENBL**), found on pin #14 on the **AUX** connector, is provided as an emergency stop input to the 6270. When you open the **ENBL** input, with respect to **GND**, the analog output voltage between **CMD+** and **CMD-** is clamped to almost zero, and the shutdown outputs are activated on both axes. *Clamping occurs independent of the microprocessor and the DSP.* (The clamping circuit is also connected to the watchdog timer; if the 6270's microprocessor fails, the analog output voltage will be clamped.)	Chapter 3: *System Connections*
Shutdown Outputs (for drives only)	The 6270 uses the shutdown outputs to disable the servo drive if it detects a problem. Two types of relay outputs are found on both **DRIVE** connectors—**SHTNC** for drives that require a closed contact to disable the drive, and **SHTNO** for drives that require an open contact to disable the drive. The shutdown relay outputs are essential for smooth power-up and power-down of the system. The shutdown relay is active (disabling the drive) when no power is applied to the 6270. When the 6270 is powered up, the shutdown relay remains active until you issue the DRIVE11 command.	Chapter 3: *Motor Driver Connections*
Drive Fault Inputs (for drives only)	The drive fault (**DFT**) inputs, found on pin #5 of both **DRIVE** connectors, allows the drives to tell the 6270 if they encounter a fault condition. When a drive fault occurs, the 6270 stops motion (at the rate set with the LHAD command) and terminates program execution. No drive shutdown will result unless it is initiated with an ERROR*P* error program.	Chapter 3: *Motor Driver Connections*
End-of-travel Limit Inputs	End-of-travel limits prevent the load from crashing through mechanical stops, an incident that can damage equipment and injure personnel. You can use hardware or software limits, as your application requires. Hardware limits use the **CW** and **CCW** terminals on the **LIMITS** connector. Software limits are set with the LSCCW and LSCW commands.	Chapter 5: *End-of-Travel Limits*
User Fault Input	Using the INFNC*i*-F command, you can assign any of the programmable inputs the *user fault* function. You can then wire the input to activate when an external event, considered a *fault* by the user, occurs.	Chapter 5: *Input Functions*
Maximum Allowable Position Error	A *position error* (TPER) is defined as the difference between the commanded position (TPC) and the actual position as measured by the feedback device (TFB). The maximum allowable position error is set with the SMPER command. When the maximum allowable position error is exceeded (usually due to instability or loss of position feedback), the 6270 shuts down the drive and sets error status bit #12 (reported by the TER command). If SMPER is set to zero (SMPERØ), the position error will not be monitored.	Chapter 4: *Servo Tuning*

☞ *Programmed Error-Handling Responses*

When any of the safety features listed above are exercised (e.g., **ENBL** input is opened, **DFT** input is activated, etc.), the 6270 considers it an error condition. With the exception of the shutdown output activation, you can enable the ERROR command to check for the error condition, and when it occurs to branch to an assigned ERROR*P* program. Refer to the *Error Handling* section in the ***6000 Series Software Reference Guide*** for further information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Scaling

The scaling commands allow you to scale acceleration, deceleration, velocity, and position to values that are appropriate for the application. The SCALE, SCLA, SCLV, SCLD, PSCLA, and PSCLV commands are used to implement the scaling features. The default condition of the 6270 is with scaling enabled (SCALE1).

☞ *Scaling parameters are specific to the current feedback source. (See programming example on page 24.)*

The scaling factors for each axis are specific to the current feedback source (selected with the last SFB command). If your application requires switching between feedback sources on the same axis, then for each feedback source, you must issue the appropriate SFB command and enter the scaling parameters specific to operating with that feedback source.

NOTE

To maximize the efficiency of the 6270's microprocessor, the scaling multiplications are performed when the program is defined or downloaded. Therefore, you must enable scaling (SCALE) and define the scaling factors (SCLD, SCLA, SCLV, PSCLA, PSCLV) prior to defining (DEF), uploading (TPROG), or running (RUN) the program. This can be accomplished by defining all scaling factors via a terminal emulator just before defining or downloading the program; you should also put the scaling factors into the startup (STARTP) program.

When Scaling is Disabled

- All programmed accel and decel values are entered in encoder revs/sec^2, LDT inches/sec^2, or ANI volts/sec^2; encoder and LDT values are internally multiplied by the encoder resolution (ERES) or the LDT resolution (LDTRES) value to obtain acceleration and deceleration values in steps/sec^2 for the motion trajectory calculations.
- All programmed velocity values are entered in encoder revs/sec, LDT inches/sec, or ANI volts/sec; encoder and LDT values are internally multiplied by the encoder resolution (ERES) or LDT resolution (LDTRES) value to obtain velocity values in steps/sec for the motion trajectory calculations.
- The distance values (D and PSET) are entered in encoder, LDT, or ANI counts. These values are internally represented as steps.

Acceleration & Deceleration Scaling (SCLA/PSCLA)

If scaling is enabled (SCALE1), all accel/decel values entered are internally multiplied by the acceleration scaling factor (SCLA) to convert user units/sec^2 to encoder, LDT, or ANI steps/sec^2. Acceleration scaling affects the following commands: A, AA, AD, ADA, HOMAD, HOMADA, JOGA, JOGAA, JOGAD, JOGADA, JOYA, JOYAA, JOYAD, JOYADA, LHAD, LHADA, LSAD, and LSADA.

Path Scaling: If you are using the 6270's linear interpolation feature, the PA, PAA, PAD and PADA commands are affected by the path acceleration scaling factor (PSCLA).

As the acceleration scaling factor (SCLA/PSCLA) changes, the accel/decel command's range and its decimal places also change (see table below). An acceleration value with greater resolution than allowed will be truncated. For example, if scaling is set to SCLA1Ø, the A9.9999 command would be truncated to A9.9.

SCLA/PSCLA **Value**	**Decimal Places**	**Max. Accel/Decel**	**Min. Accel/Decel (resolution)**
1 - 9	0	LDT Feedback: $\frac{999.9999 \times LDTRES}{SCLA}$	LDT Feedback: $\frac{0.001 \times LDTRES}{SCLA}$
10 - 99	1		
100 - 999	2		
1000 - 9999	3	Encoder Feedback: $\frac{999.9999 \times ERES}{SCLA}$	Encoder Feedback: $\frac{0.001 \times ERES}{SCLA}$
10000 - 99999	4		
100000 - 999999	5		
		ANI Feedback: $\frac{818999.9181}{SCLA}$	ANI Feedback: $\frac{0.819}{SCLA}$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Velocity Scaling (SCLV/PSCLV)

If scaling is enabled (SCALE1), all velocity values entered are internally multiplied by the velocity scaling factor (SCLV) to convert user units/sec to encoder, LDT, or ANI steps/sec. Velocity scaling affects the following commands: V, HOMV, HOMVF, JOGVH, JOGVL, JOYVH, and JOYVL.

Path Scaling: If you are using the 6270's linear interpolation feature, the PV command is affected by the path velocity scaling factor (PSCLV). As the velocity scaling factor (SCLV/PSCLV) changes, the velocity command's range and its decimal places also change (see table below). A velocity value with greater resolution than allowed will be truncated. For example, if scaling is set to SCLV1Ø, the V9.9999 command would be truncated to V9.9.

SCLV/PSCLV Value **(steps/unit)**	**Velocity Resolution (units/sec)**	**Decimal Places**	**Max. Velocity Calculation**
1 - 9	1	0	LDT Feedback: $\frac{1000 \times LDTRES}{SCLV}$
10 - 99	0.1	1	
100 - 999	0.01	2	
1000 - 9999	0.001	3	Encoder Feedback: $\frac{1000 \times ERES}{SCLV}$
10000 - 99999	0.0001	4	
100000 - 999999	0.00001	5	ANI Feedback: $\frac{1000 \times 819}{SCLV}$

Distance Scaling (SCLD)

If scaling is enabled (SCALE1), D and PSET command values are internally multiplied by the distance scaling factor (SCLD). Since the SCLD units are in terms of steps/unit, all distances will thus be internally represented in encoder, LDT, or ANI steps. For instance, if the distance scaling factor is 10000 (SCLD1ØØØØ) and you enter a distance of 75 (D75), the actual distance moved will be 750,000 (10000 x 75) encoder or LDT steps.

LDT Users—Programming In Inches Or Millimeters

The default SCLD value is 432 (SCLD432,432), which allows LDT users to program in inches. To program in millimeters, use a SCLD value of 17 (SCLD17,17). These factors must be multiplied by the number of recirculations if the LDT uses more than one recirculation.

As the SCLD scaling factor changes, the distance command's range and its decimal places also change (see table below). A distance value with greater resolution than allowed will be truncated. For example, if scaling is set to SCLD4ØØ, the D1Ø5.2776 command would be truncated to D1Ø5.277.

SCLD **(steps/unit)**	**Distance Resolution (units)**	**Distance Range (units)**	**Decimal Places**
1 - 9	1.0	0 - ±99999999.9	1
10 - 99	0.10	0.0 - ±9999999.99	2
100 - 999	0.010	0.00 - ±999999.999	3
1000 - 9999	0.0010	0.000 - ±99999.9999	4
10000 - 99999	0.00010	0.0000 - ±9999.99999	5
100000 - 999999	0.00001	0.00000 - ±999.999999	5

NOTE **FRACTIONAL STEP TRUNCATION** **NOTE**

If you are operating in the incremental mode (MAØ), when the distance scaling factor (SCLD) and the distance value are multiplied, a fraction of one step may possibly be left over. This fraction is truncated when the distance value is used in the move algorithm. This truncation error can accumulate over a period of time, when performing incremental moves continuously in the same direction. To eliminate this truncation problem, set SCLD to 1, or a multiple of 10.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Scaling Example

Axis #1 controls a 4,000 step/rev servo motor/drive system (using a 1000-line encoder). Attached is a 5-pitch leadscrew that he wants to position in inches (4,000 steps/rev x 5 revs/inch = 20,000 steps/inch).

Axis #2 controls a servo valve and hydraulic cylinder using position feedback from an LDT with a gradient of 9.0227 µs/inch, and 2 recirculations. The user would like to program in inches.

Command	Description
> SFB1,3	Feedback devices: encoder for axis 1, LDT for axis 2
> ERES4ØØØ	Set encoder resolution to 4,000 steps/rev (post quadrature)
> LDTGRD,9.0227	Set LDT gradient to 9.0227µs/inch
> LDTRES,864	Set LDT resolution to 864 to accommodate 2 recirculations
> SCALE1	Enable scaling
> SCLD2ØØØØ,864	Distance scale factors (20,000 = 5 pitch * 4000 steps/rev)
> SCLV2ØØØØ,864	Velocity scale factors
> SCLA2ØØØØ,864	Acceleration and deceleration scale factors
> D5,4	Set move distance to 5 inches for axis 1 and 4 inches for axis 2
> A5,8	Set acceleration to 5 in/sec^2 on axis 1 and 8 in/sec^2 on axis 2
> V11,4	Set velocity to 11 in/sec on axis 1 and 4 in/sec on axis 2
> GO11	Initiate move on both axes

End-of-Travel Limits

The 6270 can respond to both hardware and software end-of-travel limits.

NOTE

If you are not using hardware end-of-travel limits in your application, you must disable them. There are two methods available:

- Use the LH command (e.g., to disable hardware and software limits on axes 1 and 2, issue the LHØ,Ø command).
- Find the **LIMITS** connector on the front panel and jumper the end-of-travel limit terminals (**CW** and **CW**) to the **GND** terminal.

Refer to Chapter 3, Installation, *for instructions to wire hardware end-of-travel limit switches.*

End-of-travel limits prevent the motor's load from traveling past defined limits. Once a hardware or software limit is reached, the 6270 will decelerate that axis at a rate specified with the LHAD or LSAD command. Typically, software and hardware limits are positioned in such a way that when the software limit is reached the motor will start to decelerate towards the hardware limit. This will allow for a much smoother stop at the hardware limit. Software limits can be used regardless of incremental or absolute positioning. Refer to the LH, LS, LHAD, LHADA, LSAD, and LSADA commands in the ***6000 Series Software Reference Guide*** for more information.

The example below uses the SCLD scaling command to define software limits in millimeters. Software limits are defined by the LSCW and LSCCW commands, and enabled with the LS command. The software limits are referenced from a position of absolute zero. Both software limits may be defined with positive values (axis #2 in example below) or negative values. *Care must be taken when performing incremental moves because the software limits are always defined in absolute terms.* They must be large enough to accommodate the moves, or a new zero point must be defined (using the PSET command) before each move.

NOTES

Soft Limits: To ensure proper motion when using soft end-of-travel limits, be sure to set the LSCW value to a greater absolute value than the LSCCW value. (*CW* refers to extension, *CCW* to retraction)

Linear Positioning Systems: In this user guide, it is assumed that you have connected the valve so that clockwise (CW) refers to extension and counter-clockwise (CCW) refers to retraction. This convention is accurate if you connect the valve as instructed in Chapter 3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example Set Up	**Command**	**Description**
	> SFB3,3	Use LDTs as position feedback devices for both axes
	> LDTRES432,432	Set LDT resolution to 432 (no recirculations)
	> SCALE1	Enable scaling
	> SCLD17,17	Distance scale factor (allows programming in millimeters)
	> SCLV17,17	Velocity scale factor
	> SCLA17,17	Acceleration and deceleration scale factor
	> LH3,3	Enable limits on axes 1 and 2
	> LHAD10,10	Hard limit deceleration
	> LSAD5,10	Soft limit deceleration
	> LSCW10,20	Establish CW (extension direction) soft limit
	> LSCCW0,2	Establish CCW (retract direction) soft limit
	> LS3,3	Enable soft limits on axes 1 and 2

Homing *(Using the Home Inputs)*

Refer to Chapter 3, Installation, *for instructions to wire hardware home limit switches.*

The HOM command initiates a sequence of moves that position an axis using the Home and/or the Z channel inputs. The result of any *homing* operation is a repeatable initial starting location. The home inputs to be used, the edge of those inputs, and the final approach direction may all be defined by the user. If the encoder's Z channel input is to be used, the HOMZ command must be enabled. The input polarity (normally-open or normally-closed) of the home input or switch is defined with the HOMLVL command.

The velocity for a move to the home position is specified with the HOMV command. The acceleration and deceleration rates are specified with the HOMA and HOMAD commands, respectively. (HOMAA and HOMADA are also used if you are using S-curve Profiling—see *S-Curve Profiling* section below for details.) If *backup to home* (HOMBAC) is enabled, the velocity of the final approach toward the home position is specified with the HOMVF command.

Enabling backup to home (HOMBAC) allows you to use two other homing features, HOMEDG and HOMDF. The HOMEDG command allows you to specify the side of the home switch on which to stop. The HOMDF command allows you to specify the final approach direction. If HOMBAC is not enabled, HOMEDG and HOMDF will have no effect on the homing algorithm (see Figures A and B).

Figures A and B show the homing operation when HOMBAC is not enabled. If a limit is encountered during the homing operation, the motion will be reversed and the home switch will be sought in the opposite direction. If a second limit is encountered, the homing operation will be terminated, stopping motion at the second limit.

After a homing operation is successfully completed, the absolute position register is reset to zero (applies also to the voltage register for ANI feedback).

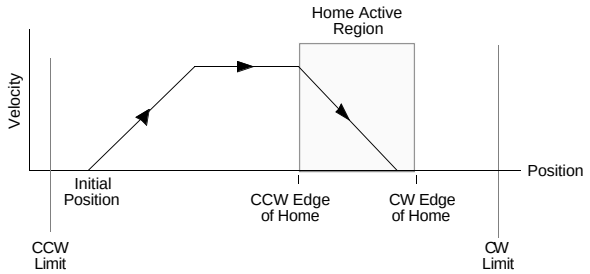


Figure A. Homing in a CW Direction (HOMØ) with backup to home disabled (HOMBACØ)

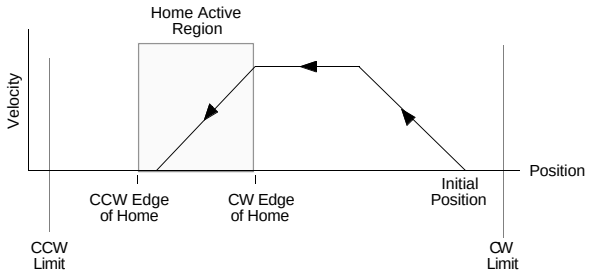


Figure B. Homing in a CCW Direction (HOM1) with backup to home disabled (HOMBACØ)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CW Homing with Backup to Home Enabled

The seven steps below describe a sample homing operation when HOMBAC is enabled (see Figure C). The final approach direction (HOMDF) is CW and the home edge (HOMEDG) is the CW edge.

> **NOTE**
>
> To better illustrate the direction changes in the backup-to-home operation, the illustrations in the remainder of this section show the backup-to-home movements with varied velocities. In reality, the backup-to-home movements are performed at the same velocity (HOMVF value).

Step 1 A CW home move is started with the HOMØ command at the HOMA and HOMAA accelerations. Default accel is 10 inches/sec^2.

Step 2 The HOMV velocity is reached (move continues at that velocity until home input goes active).

Step 3 The CCW edge of the home input is detected, this means the home input is active. At this time the move is decelerated at the HOMADA and/or HOMAD command values. It does not matter if the home input becomes inactive during this deceleration.

Step 4 After stopping, the direction is reversed and a second move with a peak velocity specified by the HOMVF value is started.

Step 5 This move continues until the CCW edge of the home input is reached.

Step 6 Upon reaching the CCW edge, the move is decelerated at the HOMAD and HOMAA command values, the direction is reversed, and another move is started in the CW direction at the HOMVF velocity.

Step 7 As soon as the home input CW edge is reached, this last move is immediately terminated. The load is at home and the absolute position register is reset to zero.

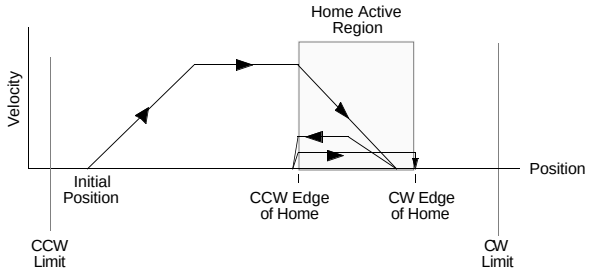


Figure C. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDGØ, HOMDFØ

Figures D through F show the homing operation for different values of HOMDF and HOMEDG, when HOMBAC is enabled.

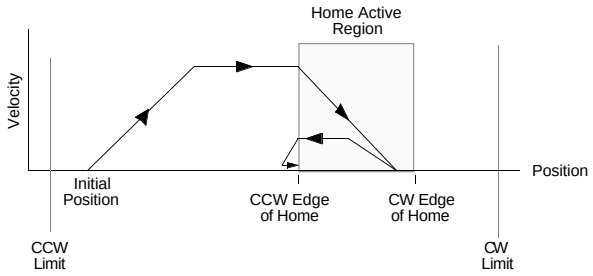


Figure D. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDG1, HOMDFØ

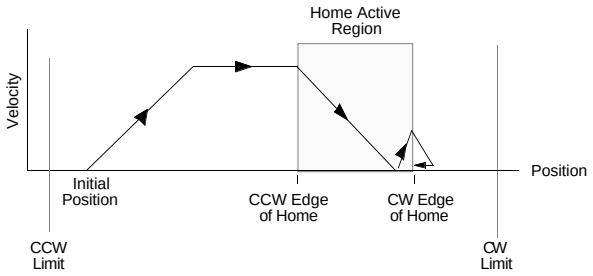


Figure E. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDGØ, HOMDF1

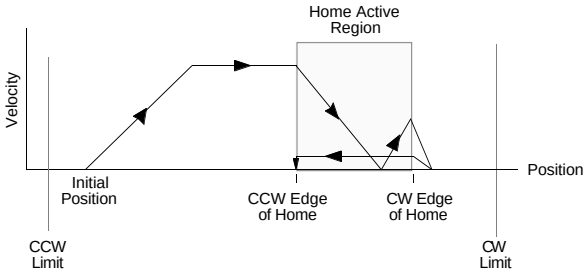


Figure F. Homing in a CW Direction (HOMØ) with HOMBAC1, HOMEDG1, HOMDF1

CCW Homing with Backup to Home Enabled

Figures G through J show the homing operation for different values of HOMDF and HOMEDG, when HOMBAC is enabled.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

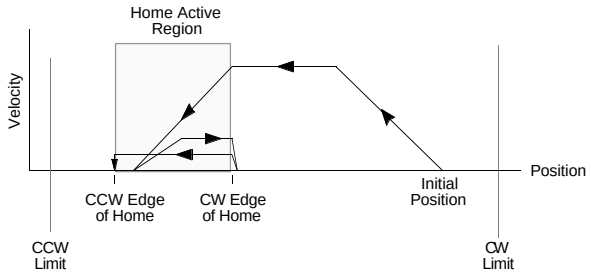


Figure G. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDG1, HOMDF1

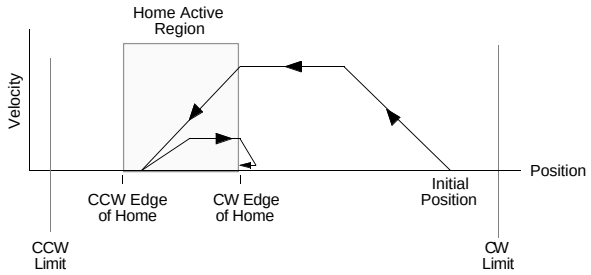


Figure H. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDF1

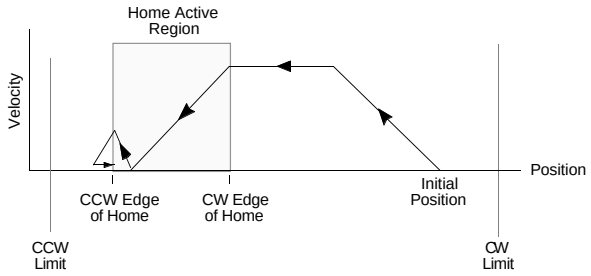


Figure I. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDG1, HOMDFØ

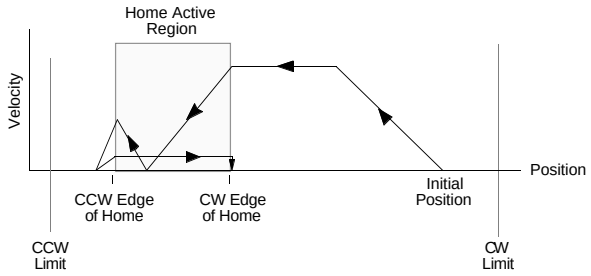


Figure J. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDFØ

Move HOME Using The Z-Channel

Figures K through O show the homing operation when homing to an encoder index pulse, or Z channel, is enabled (HOMZ1). The Z-channel will only be recognized after the home input is activated. It is desirable to position the Z channel within the home active region; this reduces the time required to search for the Z channel.

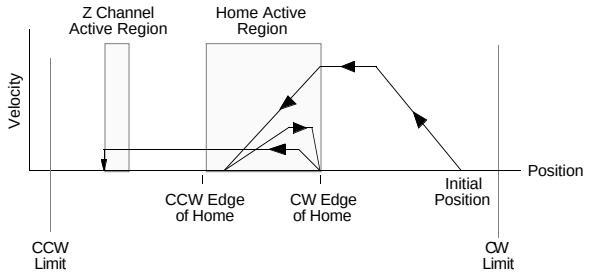


Figure K. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDG1, HOMDF1

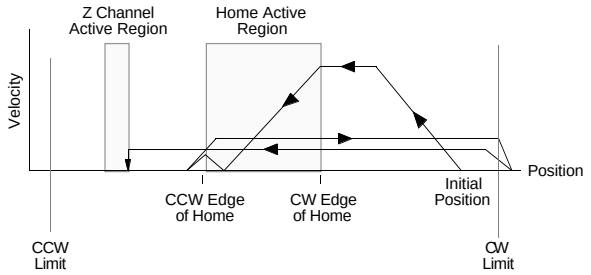


Figure L. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDFØ

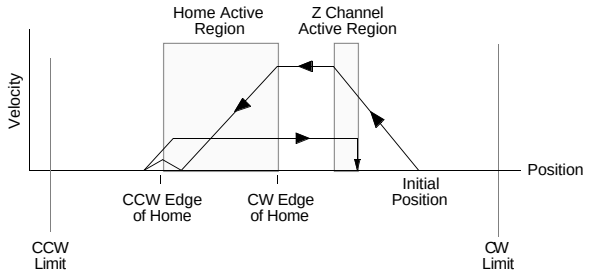


Figure M. Homing in a CCW Direction (HOM1) with HOMBAC1, HOMEDGØ, HOMDFØ

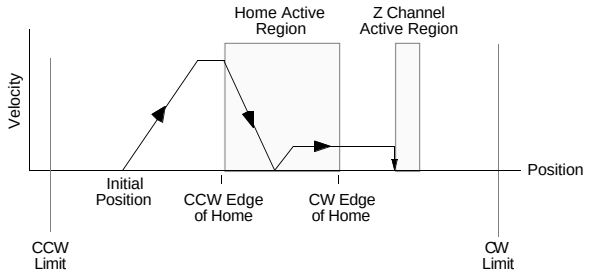


Figure N. Homing in a CW Direction (HOMØ) with HOMBACØ, HOMEDGØ, HOMDFØ

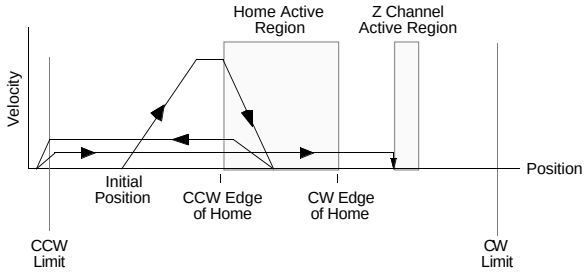


Figure O. Homing in a CW Direction (HOMØ) with HOMBACØ, HOMEDGØ, HOMDF1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Positioning Modes

The 6270 can be programmed to position in either the preset (incremental or absolute) mode or the continuous mode. You should select the mode that will be most convenient for your application. For example, a repetitive cut-to-length application requires incremental positioning. X-Y positioning, on the other hand, is better served in the absolute mode. Continuous mode is useful for applications that require constant movement of the load based on internal conditions or inputs, not distance.

Refer also to the Scaling *section above.*

Positioning modes require acceleration, deceleration, velocity, and distance commands (*continuous mode does not require distance*). The following table identifies these commands and their units of measure, and which scaling command affects them.

Parameter	Commands	Units (Unscaled)	Unit Scaling Command *
Acceleration	A, AA	Encoder Feedback: revs/sec^2 (rps^2) LDT Feedback: inches/sec^2 (ips^2) ANI Feedback: volts/sec^2 (vps^2)	SCLA or PSCLA
Deceleration	AD, ADA	rps^2 or ips^2 or vps^2	SCLA or PSCLA
Velocity	V	rps or ips or vps	SCLV or PSCLV
Distance	D	steps (counts from encoder or LDT or ANI input)	SCLD

* Scaling must first be enabled with the SCALE1 command. PSCLA and PSCLV are for linear interpolated moves.

Preset Mode

A *preset* move is a point-to-point move of a specified distance. You can select preset moves by putting the 6270 into preset mode (canceling continuous mode) using the MCØ command. Preset moves allow you to position the motor/load in relation to a defined zero reference position (*absolute mode*—enabled with the MA1 command) or in relation to the previous stopped position (*incremental mode*—enabled with the MAØ command).

Absolute Mode Moves

The absolute mode is the 6270's default power-up mode. A preset move in the Absolute Mode (MA1) moves the motor/load the distance that you specify from the *absolute zero position*. You can set the absolute position to any value with the PSET command. When the Go Home (HOM) command is issued, the absolute position register is automatically set to zero after reaching the home position.

The direction of an absolute preset move depends upon the motor's/load's position at the beginning of the move and the position you command it to move to. For example, if the motor/load is at absolute position +12,500, and you instruct it to move to position +5,000, it will move in the negative (CCW, retract) direction a distance of 7,500 steps to reach the absolute position of +5,000.

The 6270 retains the absolute position, even while the unit is in the incremental mode. You can use the Feedback Device Position Report (TFB) command to read the absolute position.

Example

Command	Description
> SFB3	Select LDT position feedback for axis 1
> LDTRES432	Set axis 1 LDT resolution to 432 steps (counts) per inch
> SCALEØ	Disable scaling
> MA1	Set the 6270 to the absolute positioning mode
> PSETØ	Set axis 1 current absolute position to zero
> A5	Set axis 1 acceleration to 5 ips^2
> V3	Set axis 1 velocity to 3 ips
> D4	Set axis 1 move to absolute position 4
> GO1	Initiate axis 1 move to absolute position 4
> D8	Set axis 1 move to absolute position 8
> GO1	Initiate axis 1 move (Since the motor/load was already at position 4, it moves 4 additional steps in the positive, or extension, direction.)
> DØ	Set axis 1 move to absolute position zero
> GO1	Initiate axis 1 move (Since the motor is at absolute position 8,000 the motor moves 8,000 steps in the negative, or retraction, direction.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Incremental Mode Moves

When using the incremental mode (MAØ), a preset move moves the motor/load the specified distance from its starting position. For example, if you are using a hydraulic positioning system to move the load 1.5 inches, you would specify a preset move with a distance of +648 steps (1.5 revs @ 432 steps/inch). Every time the 6270 executes this move, the hydraulic cylinder moves 1.5 inches from its resting position. You can specify the direction of the move by using the optional sign (D+648 or D-648). Whenever you do not specify the direction (e.g., D648), the unit defaults to the positive (CW or extension) direction.

Example

Command	Description
> SCALEØ	Disable scaling
> SFB3	Select LDT position feedback for axis 1
> LDTRES432	Set axis 1 LDT resolution to 432 steps (counts) per inch
> MAØ	Set axis 1 to Incremental Position Mode
> A2	Set axis 1 acceleration to 2 ips^2
> V5	Set axis 1 velocity to 5 ips
> D432	Set axis 1 distance to 432 steps (1 inch) in the extension direction
> GO1	Initiate motion on axis 1 (load moves 1 inch in extension direction)
> GO1	Repeat the move (load ends up at 2 inches from original position)
> D-864	Set axis 1 distance to 864 steps (2 inches) in the retract direction
> GO1	Initiate motion on axis 1 (load moves 2 inches in the retract direction and ends at its original starting position)

Continuous Mode

The Continuous Mode (MC) is useful in the following situations:

- Applications that require constant movement of the load
- Synchronize the motor to external events such as trigger input signals
- Changing the motion profile after a specified distance or after a specified time period (T command) has elapsed

Buffered vs. *Immediate* Commands

You can manipulate the motor movement with either buffered or immediate commands. After you issue the GO command, buffered commands are not executed unless the continuous command execution mode (COMEXC command) is enabled. Once COMEXC is enabled, buffered commands are executed in the order in which they were programmed. More information on the COMEXC mode is provided in the *Programming Guide* section of the ***6000 Series Software Reference Guide***.

The command can be specified as *immediate* by placing an exclamation mark (!) in front of the command. When a command is specified as immediate, it is placed at the front of the command queue and is executed immediately.

Example

Command	Description
> COMEXC1	Enable continuous command processing mode
> MC1	Sets axis 1 mode to continuous
> A1Ø	Sets axis 1 acceleration to 10
> V1	Sets axis 1 velocity to 1
> GO1	Initiates axis 1 move (Go)
> WAIT(1VEL=1)	Wait to reach continuous velocity
> T5	Time delay of 5 seconds
> S1	Initiate stop of axis 1 move
> WAIT(MOV=bØ)	Wait for motion to completely stop on axis 1
> COMEXCØ	Disable continuous command processing mode

The motor accelerates to 1 rps/ips and continues at 1 rps/ips for 5 seconds, at which point it decelerates to a stop. While in continuous mode, motion can also be stopped if:

- You issue an immediate Stop (!S) or Kill (!K) command
- The load trips an end-of-travel limit switch
- The load trips an input configured as a kill or stop input with the INFNCi-C or INFNCi-D commands, respectively.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

On-The-Fly Changes

You can change velocity and acceleration *on the fly* (while motion is in progress and in continuous mode) by issuing an immediate velocity (!V) and/or acceleration (!A) command followed by an immediate go (!GO). If the continuous command processing mode (COMEXC) is enabled, you can also make *on-the-fly* velocity and acceleration changes by using buffered commands (V and A), followed by a GO command.

NOTE

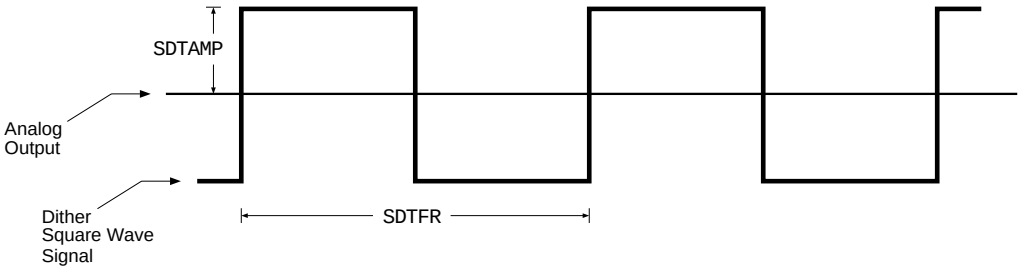
While the axis is moving, you cannot change the parameters of some commands (such as D and HOM). This rule applies during the COMEXC mode and even if you prefix the command with an immediate command identifier (!). For more information, refer to *Changing Command Parameters During Motion* in the *Programming Guide* section of the ***6000 Series Software Reference Guide***.

Example

Command	Description
> DEF prog1	Begin definition of program prog1
- COMEXC1	Enable continuous command processing mode
- MC1	Set axis 1 mode to continuous
- A1Ø	Set axis 1 acceleration to 10
- V1	Set axis 1 velocity to 1
- GO1	Initiate axis 1 move (Go)
- WAIT(1VEL=1)	Wait for motor to reach continuous velocity
- T3	Time delay of 3 seconds
- A5Ø	Set axis 1 acceleration to 50
- V1Ø	Set axis 1 velocity to 10
- GO1	Initiate acceleration and velocity changes on axis 1
- T5	Time delay of 5 seconds
- S1	Initiate stop of axis 1 move
- WAIT(MOV=bØ)	Wait for motion to completely stop on axis 1
- COMEXCØ	Disable continuous command processing mode
- END	End definition of program prog1

Dithering Hydraulic Valves

Dither is a square-wave signal added to the analog output and is used to keep the hydraulic valve moving slightly for the purpose of reducing stiction (see illustration below). Two commands are used to select the amplitude and frequency of the dither signal—SDTAMP and SDTFR.



SSFR (servo sampling frequency) / SDTFR = Dither Frequency (cycles/sec)

The SDTAMP command selects the amplitude of the dither signal in peak volts (see illustration). The SDTFR command selects the frequency ratio of the dither.

☞ *Refer to Step 2 in the* Controller Tuning Procedure *in Chapter 4 for a discussion on the servo sampling rate.*

The actual dither frequency is determined by the ratio of the servo sampling frequency (SSFR & INDAX settings) and the SDTFR value. For example, if the SSFR value is 4 and the INDAX value is 2 (default settings), the servo sampling rate is 2500 samples per second. Then, at SSFR4, an SDTFR value of 46 (default setting) would yield a 54.3 Hz dither frequency (2500/46 = 54.3). With an SDTFR command setting of 46, a positive voltage (SDTAMP) is added during 23 servo updates and a negative voltage is added during the next 23 servo updates.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

User Interface Options

The following are the three basic user interface options for controlling the 6270:

Stand-alone operation:

After defining and storing 6270 programs with a RS-232C terminal, you can operate the 6270 as a stand-alone controller. A program stored in the 6270 may interactively prompt the user for input as part of the program (via programmable I/O, thumbwheels, an RP240, or an RS-232C terminal).

PLC interface:

The 6270's programmable I/O may be connected to most PLCs. After defining and storing 6270 programs with a RS-232C terminal, the PLC typically executes programs, loads data, and manipulates inputs to the 6270. The PLC instructs the 6270 to perform the motion segment of a total machine process.

Host computer operation:

A computer may be used to control a motion or machine process. A PC can monitor processes and orchestrate motion by sending motion commands to the 6270 or by executing motion programs already stored in the 6270. This control might come from a BASIC or C program.

Of course, you can use any one, or combination, of these options in your application. Some application examples are provided below. The sections below discuss programmable I/O (including thumbwheel and PLC interfacing), the RP240 interface, host computer interfacing, stored programs, and other aspects of the 6270 that allow the user to apply the 6270 in the variety of interface options as noted.

User Interface Option	Application Example
Stand-alone:	
Programmable I/O and Thumbwheel/TM8 interface	**Cut-to-length**: Load the stock into the machine, enter the length of the cut on the thumbwheels, and activate a programmable input switch to initiate the predefined cutting process (axis #1). When the stock is cut, a sensor activates a programmable input to stop the cutting process and the 6270 then initiates a predefined program that indexes the stock forward (axis #2) into position for the next cut.
RP240 Remote Operator Panel interface	**Gainset selection**: Program the RP240 to select different servo gainsets. At start-up a set of gains for cold fluid can be selected. As the fluid heats up a new set of gains may be selected.
Joystick interface	**Coarse positioning**: Before machining a part it must be coarsely positioned. **Simulator**: Use analog inputs to provide motion in flight simulator.
14-bit analog interface (6270-ANI Option only)	**Injection Molding**: Use for feedback from a pressure sensor and use 6270 to maintain constant, programmable force.
PLC Interface	**X-Y point-to-point**: A PLC controls several tools to stack and bore several steel plates at once. The 6270 is programmed to move an X-Y table in a pre-programmed sequence. The 6270 moves the load when the inputs are properly configured, signals the PLC when the load is in position, and waits for the signal to continue to the next position.
Host Computer (PC) Interface	A BASIC program example is provided later in the section labeled *Host Computer Operation*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Programmable Inputs and Outputs

Refer to Chapter 3 for I/O connection instructions.

There are 26 programmable inputs (includes 2 trigger inputs on the **AUX** connector) and 26 programmable outputs (includes 2 auxiliary outputs on the **AUX** connector).

I/O circuit drawings and specifications are provided in Chapter 5.

***All** the 6270's inputs and outputs are optically isolated. 24 inputs and 24 outputs are on OPTO-22 compatible connectors for those applications that require interfacing to 120VAC I/O or to I/O with higher current requirements than the 6270 can support.*

Programmable inputs and outputs are provided to allow the 6270 to detect and respond to the state of switches, thumbwheels, electronic sensors, and outputs of other equipment such as drives and PLCs. Based on the state of the inputs and outputs, read with the [IN] and [OUT] commands, the 6270 can make program flow decisions and assign values to binary variables for subsequent mathematical operations. These operations and the associated program flow, branching, and variable commands are listed below.

Operation based on I/O State	Associated Commands	See Also*
I/O state assigned to a binary variable	[IN], [OUT], VARB	Chapter 5, *Variables* section
I/O state used as a basis for comparison in conditional branching & looping statements	[IN], [OUT], IF, ELSE, NIF, REPEAT, UNTIL, WAIT, WHILE, NWHILE	*Program Flow Control* section in the ***6000 Series Software Reference Guide***
I/O state used as a basis for a program interrupt (GOSUB) conditional statement	ONIN	*Program Interrupts* section in the ***6000 Series Software Reference Guide***

* Refer also to the command descriptions in the ***6000 Series Software Reference Guide***

As discussed below, you can program and check the status of each input and output with the INFNC and OUTFNC commands, respectively. To receive a binary report of the state (on or off) of the I/O, use the TIN command (inputs) or the TOUT command (outputs).

Using the INLVL and OUTLVL commands, you can define the logic levels of the 24 general-purpose inputs and outputs (including **OUT-A** and **OUT-B**) as positive or negative.

Output Functions

You can turn the 6270's 26 programmable outputs on and off with the Output (OUT or OUTALL) commands, or you can use the Output Function (OUTFNC) command to configure them to activate based on seven different situations.

The output functions are assigned with the OUTFNCi-<a>c command. The "i" represents the number of the output (the 24 general purpose outputs are outputs 1 through 24, and **OUT-A** and **OUT-B** are outputs 25 and 26). The "<a>" represents the number of the axis and is optional for the B, D, and G functions (see list below); when no axis specifier is given, the output will be activated when the condition occurs on either axis. The "c" represents the letter designator of the function (A, B, C, D, F, G or H). For example, the OUTFNC5-2D command configures output #5 to activate when axis #2 encounters a hard or soft limit.

NOTE

To activate the function of an output with the OUTFNC command, you must enable the output functions with the OUTFEN1 command.

A: Programmable Output (**default function**)
B: Moving/Not Moving (or In Position)
C: Program in Progress
D: Soft or Hard Limit Encountered
E: *<Not Used>*
F: Fault Output (indicates drive or user fault)
G: Max. Allowable Position Error Exceeded
H: Output on Position

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Output Status

As shown below, you can use the OUTFNC command to determine the current function and state (on or off) of one or all the outputs. The TOUT command also reports the outputs' state, but in a binary format in which the left-most bit represents output #1 and the right-most bit represents output #26.

Command	Description
> OUTFNC	Query status of all outputs; response indicating default conditions is: *OUTFNC1-A NO FUNCTION OUTPUT - STATUS OFF *OUTFNC2-A NO FUNCTION OUTPUT - STATUS OFF (response continues until all 26 outputs are reported)
> OUTFNC1	Query status of output #1; response indicating default conditions is: *OUTFNC1-A NO FUNCTION OUTPUT - STATUS OFF
> OUTFNC1-C	Change output #1 to function as a *Program in Progress* output
> OUTFNC1	Query status of output #1; response should be now be: *OUTFNC1-C PROGRAM IN PROGRESS - STATUS OFF
> TOUT	Query binary status report of all outputs; response indicating default conditions is: *TOUTØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØ

Programmable Output
(OUTFNCi-A)

The default function for the outputs is *Programmable*. As such, the output is used as a standard output, turning it on or off with the OUT or OUTALL commands to affect processes external to the 6270. To view the state of the outputs, use the TOUT command. To use the state of the outputs as a basis for conditional branching or looping statements (IF, REPEAT, WHILE, etc.), use the [OUT] command (refer to the *Conditional Looping and Branching* section in the ***6000 Series Software Reference Guide*** for details).

Moving/ Not Moving (In Position)
(OUTFNCi-<a>B)

When assigned the *Moving/Not Moving* function, the output will activate when the axis is commanded to move. As soon as the move is completed, the output will change to the opposite state.

If the target zone mode is enabled (STRGTE1), the output will not change state until the move completion criteria set with the STRGTD and STRGTV commands has been met. (Refer to the *Target Zone* section in Chapter 4 for more details on the target zone mode.) In this manner, the Moving/Not Moving output functions as an *In Position* output.

Example

The following example defines output 1 and output 2 as *Programmable* outputs and output 3 as a *Moving/Not Moving* output. Before the motor moves 4 units, output 1 turns on and output 2 turns off. These outputs remain in this state until the move is completed, then output 1 turns off and output 2 turns on. While the motor/load is moving, output 3 remains on.

Command	Description
> PS	Pauses command execution until the 6270 receives a Continue (!C) command
SCALE1	Enable scaling
MCØ	Sets axis 1 to Normal mode
MAØ	Select incremental positioning mode
A1Ø	Sets axis 1 acceleration to 10
V5	Sets axis 1 velocity to 5
D4	Sets axis 1 distance to 4 units
OUTFEN1	Enable output functions
OUTFNC1-A	Sets output 1 as a programmable output
OUTFNC2-A	Sets output 2 as a programmable output
OUTFNC3-1B	Sets output 3 as a axis 1 Moving/Not Moving output
OUT1Ø	Turns output 1 on and output 2 off
GO1	Initiates axis 1 move
OUTØ1	Turns output 1 off and output 2 on
!C	Initiates command execution to resume

Program in Progress
(OUTFNCi-C)

When assigned the *Program in Progress* function, the output will activate when a program is being executed. After the program is finished, the output's state is reversed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Limit Encountered
(OUTFNCi-<a>D)

When assigned the *Limit Encountered* function, the output will activate when a hard or soft limit has been encountered.

If a hard or soft limit is encountered, you will not be able to move the motor/load in that same direction until you clear the limit by changing direction (D) and issuing a GO command. (An alternative is to disable the limits with the LHØ command, but this is recommended only if the motor, cylinder, etc. is not coupled to the load.)

Fault Output
(OUTFNCi-F)

When assigned the *Fault Output* function, the output will activate when either the user fault input or the drive fault input (drive users only) becomes active. The user fault input is a general-purpose input defined as a user fault input with the INFNCi-F command. The drive fault input is found on the **DRIVE** connector, pin #5; make sure the drive fault active level (DRFLVL) is appropriate for the drive you are using.

Maximum Position Error Exceeded
(OUTFNCi-<a>G)

When assigned the *Max. Position Error Exceeded* function, the output will activate when the maximum allowable position error, as defined with the SMPER command, is exceeded.

The position error (TPER) is defined as the difference between the commanded position (TPC) and the actual position as measured by the feedback device (TFB). When the maximum position error is exceeded (usually due to lagging load, instability, or loss of position feedback), the 6270 shuts down the drive/valve and sets error status bit #12 (reported by the TER command if bit #12 of the ERROR command is enabled).

> **NOTE**
> If the SMPER command is set to zero (SMPERØ), the position error will not be monitored; thus, the *Maximum Position Error Exceeded* function will not be usable.

Output on Position
(OUTFNCi-H)

The *Output on Position* function for axis 1 (OUTFNC25-H) can be assigned only to output #25 (**OUT-A**), and the *Output on Position* function for axis 2 (OUTFNC26-H) can be assigned only to output #26 (**OUT-B**). The position being monitored is the currently selected feedback device. The *Output on Position* parameters are configured with the OUTPA and OUTPB commands:

1st data field (b): 1 enables the *output on position* function; Ø disables the function. If a SFB command is executed, the function is disabled.

2nd data field (b): 1 sets the position comparison in the 3rd data field (r) to an incremental position;
Ø sets the position comparison in the 3rd data field (r) to an absolute position.

3rd data field (r): Represents the scalable distance with which the actual feedback device position is to be compared (distance is either incremental or absolute, depending on the setting of the 2nd data field). The feedback device used per axis depends on which one is assigned with the SFB command. *Remember that encoder feedback may not be used for axis 2.*

4th data field (i): Represents the time (in milliseconds) the output is to stay active. If this data field is set to Ø, the output will stay active for as long as the actual distance equals or exceeds the distance specified in the 3rd data field. (This is valid only for the absolute mode—2nd data field set to Ø)

If an incremental distance is used for comparison (2nd data field set to 1), the output activates when the feedback device position is greater than or equal to the specified distance, and stays active for the specified time.

If an absolute distance is used for comparison (2nd data field set to Ø), the output activates when the feedback device position is greater than or equal to the specified absolute distance, and stays active for the specified time.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE

The output activates only during motion; thus, issuing a PSET command to set the absolute position counter to activate the output on position will not turn on the output until the next motion occurs.

Example

Command	Description
> OUTFEN1	Enable programmable output functions
> OUTFNC25-H	Set OUT-A (output #25) as *output on position* output for axis 1
> OUTFNC26-H	Set OUT-B (output #26) as *output on position* output for axis 2
> OUTPA1,Ø,+5Ø,5Ø	Turn on OUT-A for 50 ms when the actual position of axis 1 is greater than or equal to absolute position +50
> OUTPB1,Ø,+2.4,2ØØ	Turn on OUT-B for 200 ms when the actual position of axis 2 is greater than or equal to absolute position +2.4

Input Functions

The input functions are assigned with the INFNCi-<a>c command. The "i" represents the number of the input (the 24 general purpose inputs are inputs 1 through 24, and **TRG-A** and **TRG-B** are inputs 25 and 26). The "<a>" represents the number of the axis, if required. The "c" represents the letter designator of the function (A through Q). For example, the INFNC5-2D command configures output #5 to function as a *stop* input, stopping motion on axis #2 when activated.

NOTE

To activate the function of an input with the INFNC command, you must first enable the input functions with the INFEN1 command. Because the INFEN1 command enables the drive fault input, you should verify the fault active level (DRFLVL) is set properly.

A: No Function (**default**)
B: BCD Program Select
C: Kill
D: Stop
E: Pause/Continue
F: User Fault
H: Position Latch (**TRG-A & TRG-B Only**)
J: Jog+ (CW)
K: Jog- (CCW)
L: Jog Speed Select
P: Program Select
Q: Program Security

Input Status

As shown below, you can use the INFNC command to determine the current function and state (on or off) of one or all the inputs. The TIN command also reports the inputs' state, but in a binary format in which the left-most bit represents input #1 and the right-most bit represents input #26.

Example

Command	Description
> INFNC	Query status of all inputs; response indicating default conditions is: *INFNC1-A NO FUNCTION INPUT - STATUS OFF *INFNC2-A NO FUNCTION INPUT - STATUS OFF (response continues until all 26 inputs are reported)
> INFNC1	Query status of input #1; response indicating default conditions is: *INFNC1-A NO FUNCTION INPUT - STATUS OFF
> INFNC1-D	Change input #1 to function as a Stop input
> INFNC1	Query status of input #1; response should be now be: *INFNC1-D STOP INPUT - STATUS OFF
> TIN	Query binary status report of all inputs; response indicating default conditions is: *TINØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØ

Input Debounce Time

Using the Input Debounce Time (INDEB) command, you can change the input debounce time for all 24 general-purpose inputs (one debounce time for all 24), or you can assign a unique debounce time to each of the 2 trigger inputs.

General-Purpose Input Debounce: The input debounce time for the 24 general-purpose inputs is the period of time that the input must be held in a certain state before the 6270 recognizes it. This directly affects the rate at which the inputs can change state and be recognized.

Trigger Input Debounce: For trigger inputs, the debounce time is the time required between a trigger's initial active transition and its secondary active transition. This allows rapid recognition of a trigger, but prevents subsequent bouncing of the input from causing a false position capture.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The INDEB command syntax is INDEB<i>,<i>. The first <i> is the input number and the second <i> is the debounce time **in even increments** of milliseconds (ms). The debounce time range is 1 - 250 ms. The default debounce time is 4 ms for the 24 general-purpose inputs, and 24 ms for the 2 trigger inputs (**TRG-A** & **TRG-B**). If the first <i> is in the range 1 - 24, the specified debounce time is assigned to all 24 general-purpose inputs. If the first <i> is 25 or 26, the specified debounce is assigned only to the specified trigger input.

For example, the INDEB5,6 command assigns a debounce time of 6 ms to all 24 general-purpose inputs. The INDEB26,12 command assigns a debounce time of 12 ms only to input #26, which is trigger B (**TRG-B**).

No Function (INFNCi-A)

When an input is defined as a *No Function* input (default function), the input is used as a standard input. You can then use this input to synchronize or trigger program events. To view the current state of the inputs, use the TIN command. To use the state of the inputs as a basis for conditional branching or looping statements (IF, REPEAT, WHILE, etc.), use the [IN] command (refer to the *Conditional Looping and Branching* section in the ***6000 Series Software Reference Guide*** for details).

Example

Command	Description
> DEF prog1	Begin definition of program prog1
- INFEN1	Enable input functions
- INFNC1-A	No function for input 1
- INFNC2-A	No function for input 2
- INFNC3-D	Input 3 is a stop input
- A1Ø	Set acceleration
- V1Ø	Set velocity
- D5	Set distance
- WAIT(IN=b1XX)	Wait for input 1
- GO1	Initiate motion
- IF(IN=bX1)	If input 2
- TLDT	Transfer LDT position
- NIF	End IF statement
- END	End prog1
> RUN prog1	Initiate program prog1

BCD Program Select (INFNCi-B)

Inputs can be defined as *BCD program select* inputs. This allows you to execute defined programs (DEF command) by activating the program select inputs. Program select inputs are assigned BCD weights. The table to the right shows the BCD weights of the 6270's inputs when inputs 1- 8 are configured as program select inputs. The inputs are weighted with the least weight on the smallest numbered input.

Input #	BCD Weight
Input 1	1
Input 2	2
Input 3	4
Input 4	8
Input 5	10
Input 6	20
Input 7	40
Input 8	80

If inputs 6, 9, 10 and 13 are selected instead of inputs 5, 6, 7 and 8, then the weights would be as follows:

Input #6 = 10
Input #9 = 20
Input #10 = 40
Input #13 = 80

Since 100 programs can be defined, a maximum of 8 inputs are required to select all possible programs. Up to 400 programs may be defined if you have the -M (extended memory) option.

The program number is determined by the order in which the program was downloaded to the 6270. The program number can be obtained through the TDIR command.

If the inputs are configured as in the above table, activating inputs 2 and 3 will execute program #6. Activating inputs 1, 4, and 6 will execute program #29.

To execute programs using the program select lines, enable the INSELP command. Once enabled, the 6270 will continuously scan the input lines and execute the program selected by the active program select lines. To disable scanning for program select inputs, enter !INSELPØ or place INSELPØ in a program that can be selected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Once enabled (INSELP1), the 6270 will run the program number that the active program select inputs and their respective BCD weights represent. After executing and completing the selected program, the 6270 will scan the inputs again. If a program is selected that has not been defined, no program will be executed.

The INSELP command also determines how long the program select input must be maintained before the 6270 executes the program. This delay is referred to as *debounce time* (but is not affected by the INDEB setting). The following examples demonstrate how to select programs via inputs.

Example

Command	Definition
> RESET	Return 6270 to power-up conditions
> ERASE	Erase all programs
> DEF prog1	Begin definition of program prog1
- TLDT	Transfer position of LDTs
- END	End program
> DEF prog2	Begin definition of program prog2
- TREV	Transfer software revision
- END	End program
> DEF prog3	Begin definition of program prog3
- TSTAT	Transfer statistics
- END	End program
> INFNC1-B	Input 1 is a BCD program select input
> INFNC2-B	Input 2 is a BCD program select input
> INFEN1	Enable input functions
> INSELP1,5Ø	Enable scanning of inputs, levels must be maintained for 50ms

You can now execute programs by making a contact closure from an input to ground to activate the input.

- Activate input 1 to execute program #1
- Activate input 2 to execute program #2
- Activate input 1 & 2 to execute program #3

Kill (INFNCi-C)

An input defined as a *Kill* input will stop motion at the rate set with the hard limit (LHAD & LHADA) commands. The program currently in progress will also be terminated, the commands currently in the command buffer will be eliminated, and the drive will be left in the enabled state (DRIVE1).

Disabling the Drive or Valve on a Kill

If your application requires you to *disable* (*shut down* or *de-energize*) the valve or drive in a Kill situation, set the 6270 to the *Disable Drive on Kill* mode with the KDRIVE1 command. In this mode, a kill command or kill input will shut down the valve or drive immediately. If you are using a drive, the shutdown outputs are activated and the motor will then be allowed to *free wheel* (without control from the drive) to a stop. If you are using a hydraulic valve, the kill will set the command output to zero, causing the valve to return immediately to the neutral (*null*) position and hold the load at that position. To re-enable the valve or drive, issue the DRIVE11 command.

Stop (INFNCi-D)

An input defined as a *Stop* input will stop motion on any one or all axes. Deceleration is controlled by the programmed (AD/ADA) deceleration ramp. After the Stop input is received, further program execution is dependent upon the COMEXS command setting:

COMEXSØ: Upon receiving a stop input, program execution will be terminated and every command in the buffer will be discarded.

COMEXS1: Upon receiving a stop input, program execution will pause, and all commands following the command currently being executed will remain in the command buffer (*but the move in progress will not be saved*).

You can resume program execution (but not the move in progress) by issuing an immediate Continue (!C) command or by activating a pause/resume input (i.e., a general-purpose input configured as a pause/continue input with the INFNCi-E command—see below). *You cannot resume program execution while the move in progress is decelerating.*

COMEXS2: Upon receiving a stop input, program execution will be terminated, but the INSELP value is retained. This allows external program selection, via inputs defined with the INFNCi-B or INFNCi-aP commands, to continue.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pause/Continue (INFNCi-E)

An input defined as a *Pause/Continue* input will affect motion and program execution depending on the COMEXR command setting, as described below. In both cases, when the input is activated, the current command being processed will be allowed to finish executing before the program is paused.

COMEXRØ: Upon receiving a pause input, only program execution will be paused; any motion in progress will continue to its predetermined destination. Releasing the pause input or issuing a !C command will resume program execution.

COMEXR1: Upon receiving a pause input, both motion and program execution will be paused; the motion stop function is used to halt motion. Releasing the pause input or issuing a !C command will resume motion and program execution. *You cannot resume program execution while the move in progress is decelerating.*

User Fault (INFNCi-F)

An input defined as a *User Fault* input will set error status bit 7 (reported by TER and [ER]), and act as a Kill (K) command. Once this bit has been set, the error program (ERRORP) will be initiated if the specific error condition was enabled (ERRORxxxx xx1). Within the error program, a response to the fault condition can be initiated.

Position Latch (INFNCi-H)

Certain applications (such as coordinate measurement machines) require latching the current position upon receiving an input. *This function can only be assigned to the two trigger inputs (inputs 25 & 26).*

NOTE

When configured as position latch inputs, the triggers cannot be affected by the input enable (INEN) command.

Trigger input specs and circuit drawings provided in Chapter 6.

When a trigger input defined as a *Position Latch* input is activated, the commanded position and the position of all feedback devices on both axes are captured at one time. The position information is stored in registers and is available at the next system update through the use of transfer and assignment/comparison commands (see table below).

Captured Information	Transfer Command	Assignment/Comparison Command
Commanded Position	TPCC	PCC
LDT Position	TPCL	PCL
Encoder Position	TPCE	PCE
ANI Position	TPCA	PCA
ANI Voltage	TCA	CA

Position Latch Accuracy is Greater for Selected Feedback Source (SFB)

If you are capturing the position/value of a feedback source (LDT, encoder, or ANI) that is currently selected with the SFB command, the captured position is interpolated from the last sampled position and velocity of the feedback device and the time elapsed since the last sample. The position sample rate is determined by the SSFR and INDAX command settings (system update rate). The position capture accuracy is ±50 µs * current velocity.

If you are capturing the position of a device that is not selected with the SFB command, the last sampled position is simply stored as the captured position.

***Exceptions*:**

- Regardless of the SFB selection, if **TRG-A** is used to capture the encoder position on axis #1, the position is latched in hardware within ±1 encoder count (at max. encoder frequency) after the input is activated. Encoder position capture with **TRG-B** is affected by the SFB selection as noted above.
- The captured commanded position is always interpolated from the last sampled position (of the feedback device selected with the SFB command) and position error, and the time elapsed since the last sample.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

☞ *Input debounce time programmable with INDEB command.*

Each *position latch* input has a 24-ms debounce time. Therefore, the maximum rate that the input can capture positions is 40 times per second. However, if your application requires a shorter debounce time, you can change it with the INDEB command (refer to the *Input Debounce Time* section provided earlier in this chapter).

System status bits #25 and #26, reported with the TSS and SS commands, are set to 1 when the position has been captured on the respective trigger input (**TRG-A** and **TRG-B**, respectively). As soon as the captured information is transferred (TPCC, TPCL, TPCE, TPCA, or TCA) or assigned/compared PCC, PCL, PCE, PCA, or CA), the respective system status bit is cleared, but the information is still available from the register until it is overwritten by a new latch from the trigger input.

The captured *position* values (commanded, LDT, encoder, and ANI) are affected by the current state of distance scaling (SCLD), position offset (PSET), feedback polarity (LDTPOL, ENCPOL, or ANIPOL), and commanded direction polarity (CMDDIR). However, the captured ANI voltage value (TCA and CA) is not affected and is reported only in volts.

Jogging the Motor

(INFNCi-aJ)
(INFNCi-aK)
(INFNCi-aL)

In some applications, you may want to manually move (*jog*) the load. You can configure the jog function with the INFNC command.

You must define the jogging velocity with the Jog Velocity High (JOGVH) and Jog Velocity Low (JOGVL) commands. The acceleration and deceleration of the JOG move can be configured using JOGA, and JOGAD respectively. (If you are using S-curve profiles, you must also specify JOGAA and JOGADA.)

You can define three different inputs per axis for jogging: extend CW (positive direction) Jog input (INFNCi-aJ), retract CCW (negative direction) Jog Input (INFNCi-aK), and Jog Speed Select High/Low (INFNCi-aL). You must also enable the jogging feature with the JOG command.

Once you set up these parameters, you can attach a switch to the jog inputs that you defined and perform jogging. The following example shows how you can define a program to set up jogging.

Step 1

Command	Description
> DEF prog1	Begin definition of program prog1
- JOGA25	Set jog acceleration
- JOGAD25	Set jog deceleration
- JOGVL.5	Sets low-speed jog velocity
- JOGVH5	Sets high-speed jog velocity
- INFEN1	Enable input functions
- INFNC1-1J	Sets input 1 as a CW jog input
- INFNC2-1K	Sets input 2 as a CCW jog input
- INFNC3-1L	Sets input 3 as a speed-select input
- JOG1	Enables Jog function for axis 1
- END	End program definition

Step 2 Activate input 1 to move the load in the positive direction at a velocity of 0.5 (until input 1 is released).

Step 3 Activate input 2 to move the load in the negative direction at a velocity of 0.5 (until input 2 is released).

Step 4 Activate input 3 to switch to high-speed jogging.

Step 5 Repeat steps 2 and 3 to perform high-speed jogging.

One-to-One Program Select

(INFNCi-aP)

Inputs can be defined as *One-to-One Program Select* inputs (INFNCi-aP). This allows programs defined by the DEF command to be executed by activating an input. Different from BCD Program Select inputs, One-to-One Program Select inputs correspond directly to a specific program number. The program number is determined by the order in which the program was downloaded to the 6270. The program number can be obtained through the TDIR command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To execute programs using the program select lines, enable the INSELP command for one-to-one program selection. Once enabled, the 6270 will continuously scan the input lines and execute the program selected by the active program select line. To disable scanning of the program select lines, enter !INSELPØ, or place INSELPØ in a program that can be selected.

Command	Description
> RESET	Return 6270 to power-up conditions
> ERASE	Erase all programs
> DEF proga	Begin definition of program proga
- TLDT	Transfer position of LDT
- END	End program
> DEF progb	Begin definition of program progb
- TREV	Transfer software revision
- END	End program
> DEF progc	Begin definition of program progc
- TSTAT	Transfer statistics
- END	End program
> INFNC1-1P	Input 1 will select proga
> INFNC2-2P	Input 2 will select progb
> INFNC3-3P	Input 3 will select progc
> INFEN1	Enable input functions
> INSELP2,5Ø	Enable scanning of inputs

You can now execute programs by making a contact closure from an input to ground to activate the input:

- Activate input 1 to execute program #1 (proga)
- Activate input 2 to execute program #2 (progb)
- Activate input 3 to execute program #3 (progc)

Program Security (INFNCi-Q)

Issuing the INFNCi-Q command enables the *Program Security* feature and assigns the *Program Access* function to the specified programmable input.

The program security feature denies you access to the DEF, DEL, ERASE, MEMORY, and INFNC commands until you activate the program access input. Being denied access to these commands effectively restricts altering the user memory allocation. If you try to use these commands when program security is active (program access input is not activated), you will receive the error message *ACCESS DENIED. *The INFNCi-Q command is not saved in battery-backed RAM, so you may want to put it in the start-up program (STARTP).*

For example, once you issue the INFNC22-Q command, input #22 is assigned the program access function and access to the DEF, DEL, ERASE, MEMORY, and INFNC commands will be denied until you activate input #22.

Thumbwheel Interface

You can connect the 6270's programmable I/O to a bank of thumbwheel switches to allow operator selection of motion or machine control parameters.

The 6270 allows two methods for thumbwheel use. One method uses Compumotor's TM8 thumbwheel module or IM32 input module. The other allows you to wire your own thumbwheels.

The TM8 requires a multiplexed BCD input scheme to read thumbwheel data. Therefore, a decode circuit must be used for thumbwheels. Compumotor recommends that you purchase Compumotor's TM8 module if you desire to use a thumbwheel interface. The TM8 contains the decode logic; therefore, only wiring is needed.

The 6270 commands that allow for thumbwheel data entry are:

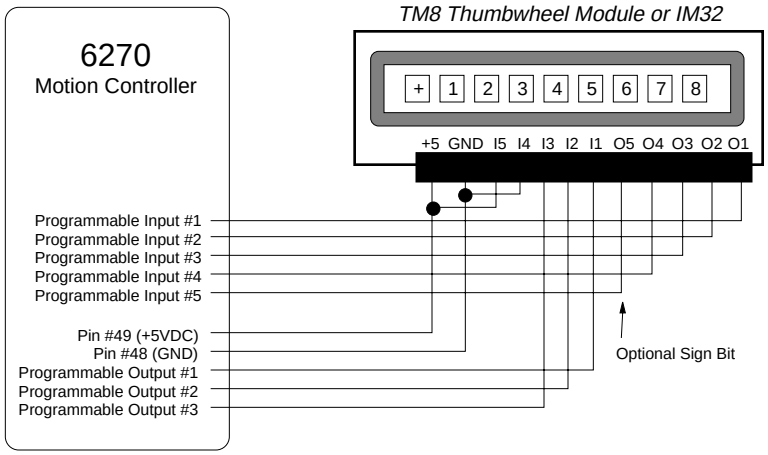
Command	Description
INSTW	Establish thumbwheel data inputs (TM8)
OUTTW	Establish thumbwheel data outputs (TM8)
TW	Read thumbwheels or PLC inputs
INPLC	Establish PLC data inputs (Other thumbwheel module)
OUTPLC	Establish PLC data outputs (Other thumbwheel module)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the TM8 & IM32 Modules

To use Compumotor's TM8 or IM32 Modules, follow the procedures below.

Step 1 Wire your TM8 or IM32 module to the 6270 as shown below.



Step 2 Configure your 6270 as follows:

Command	Description
> OUTTW1,1-3,Ø,1Ø	Configure thumbwheel output set 1 as follows: outputs 1-3 are strobe outputs, 10 ms strobe time per digit read. **The minimum strobe time recommended for the TM8 module is 10 ms**.
> INSTW1,1-4,5	Configure thumbwheel input set 1 as follows: inputs 1-4 are data inputs, input 5 is a sign input.
> INLVLØØØØØ	Inputs 1-5 configured active low

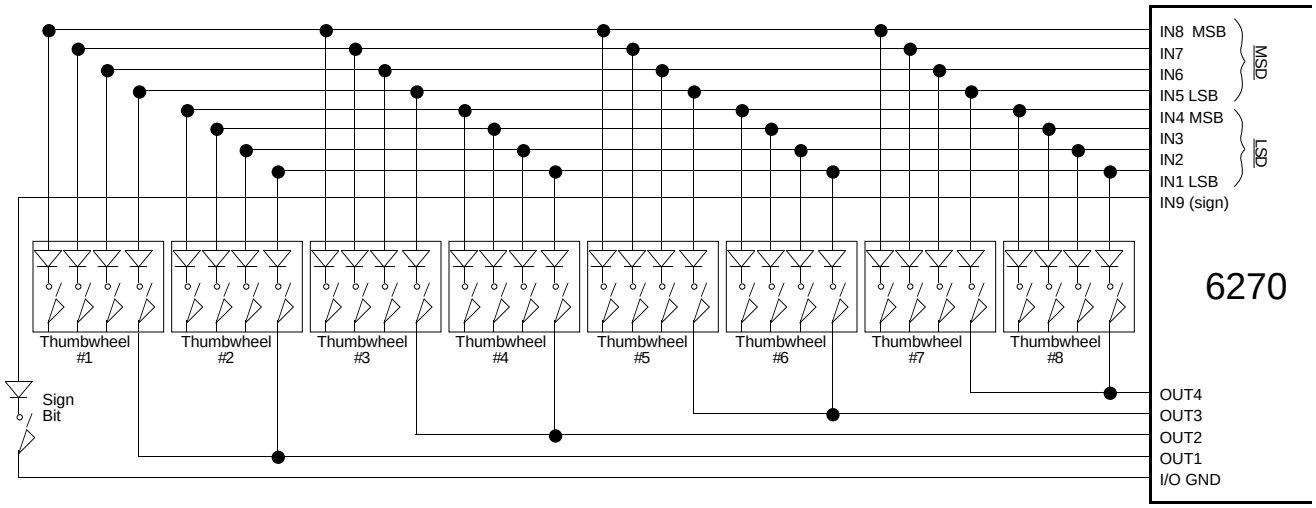
Step 3 Set the thumbwheel digits on your TM8 module to **+12345678**. To verify that you have wired your TM8 module(s) correctly and configured your 6270 I/O properly, enter the following commands:

Command	Description
> VAR1=TW1	Assign data from all 8 thumbwheel digits to VAR1
> VAR1	Displays the variable—*VAR1=+Ø.12345678**.** If you do not receive the response shown, return to step 1 and retry.

Using your own Thumbwheel Module

As an alternative to Compumotor's TM8 Module, you can use your own thumbwheels. The 6270's programming language allows direct input of BCD thumbwheel data via the programmable inputs. Use the following steps to set up and read the thumbwheel interface. Refer to the ***6000 Series Software Reference Guide*** for descriptions of the commands used below.

Step 1 Wire your thumbwheels according to the following schematic.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 2 Set up the inputs and outputs for operation with thumbwheels. The data valid input will be an input which the operator holds active to let the 6270 read the thumbwheels. This input is not necessary; however, it is often used when interfacing with PLCs.

Command	Description
> OUTPLC1,1-4,Ø,12	Configure PLC output set 1 as follows: outputs 1-4 are strobe outputs, no output enable bit, 12 ms strobe time per digit read.
> INPLC1,1-8,9	Configure PLC input set 1 as follows: inputs 1-8 are data inputs, input 9 is a sign input, no data valid input.
> INLVLØØØØØØØØØ	Inputs 1-9 configured active low

Step 3 The thumbwheels are read sequentially by outputs, which strobe two digits at a time. The sign bit is optional. Set the thumbwheels to **+12345678** and type in the following commands:

Command	Description
> VAR1=TW5	Assign data from all 8 thumbwheel digits to VAR1
> VAR1	Displays the variable—*VAR1=+Ø.12345678**.** If you do not receive the response shown, return to step 1 and retry.

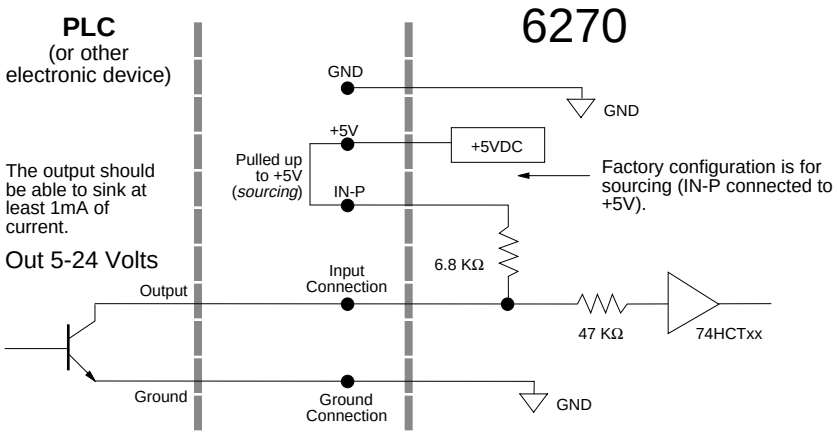
I/O Device Interface

The 6270's optically-isolated programmable I/O may be connected to electronic devices with discrete inputs and outputs, including PLCs.

When the 6270 is shipped from the factory, all 26 programmable outputs (including **OUT-A** and **OUT-B**) and the 24 general-purpose programmable inputs are pulled up to +5VDC using the pull-up (**OUT-P** and **IN-P**) terminals on the **AUX** connector. If +5VDC logic is not to be used, disconnect the pull-up terminal(s) from the **+5V** supply and use an external power source of up to 24VDC. The trigger inputs (**TRG-A** through **TRG-D**) are internally tied to 5V, but can have up to 24V connected to them.

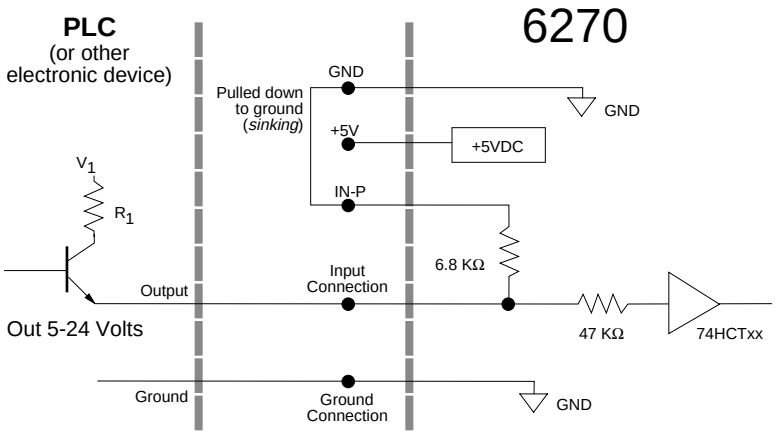
For higher current or voltages above 24VDC, use external signal conditioning such as OPTO-22 compatible I/O signal conditioning racks. Contact your local Compumotor distributor for information on these products.

Connecting to a Sinking Output Device ☞

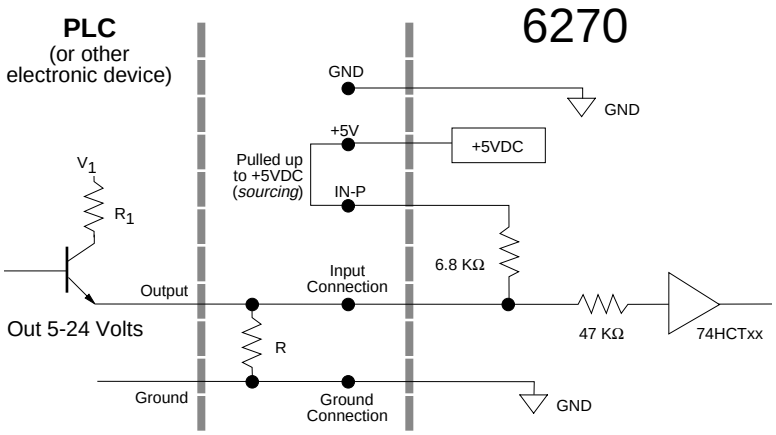


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connecting to a Sourcing Output Device ☞



If you will be connecting to a combination of sourcing and sinking outputs, connect **IN-P** to **+5V** to accommodate sinking output devices. Then for each individual input connected to a sourcing output, wire an external resistor between the 6270's input terminal and ground (see illustration below). The resistor provides a path for current to flow from the device when the output is active.



Typical value for R = 450Ω (assuming R_1 = 0)
Note: The value of R may vary depending on the value of R_1 and V_1

Joystick Interface

The 6270 has three 8-bit analog input channels (CH1 - CH3). The analog inputs are configured as three discrete single-ended inputs, with an input range of 0.0V to 2.5V. These inputs can be used to control an axis with a joystick. The voltage value on the analog inputs can be read using the ANV or TANV commands. *Refer to Chapter 3 for connection procedures.*

The Daedal JS6000 joystick is compatible with the 6270. To order the JS6000, contact Daedal at (800) 245-6903 or contact your local distributor.

Joystick control can be achieved by simply connecting a joystick potentiometer to one of the analog inputs. Joystick operation is enabled with the JOY1 command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Travel limitations in potentiometers and voltage drops along the cables may make it impossible to achieve the full 0.0V to 2.5V range at the joystick input. Therefore, you must configure the 6270 to optimize the joystick's usable voltage range. This configuration will affect the *velocity resolution*. The velocity resolution is determined by the following equation:

$$\frac{\text{maximum velocity set with the JOYVH or the JOYVL command}}{\text{voltage range between the joystick's no-velocity region (}\textit{center deadband}\text{) and its maximum-velocity region (}\textit{end deadband}\text{)}}$$

To establish the velocity resolution, you must define the full-scale velocity and the usable voltage.

Define Full-Scale Velocity

You must define the full-scale velocity for your application with the JOYVH and JOYVL commands. Both commands establish the maximum velocity that can be obtained per axis by deflecting the potentiometer fully CW or fully CCW. The JOYVH command establishes the *high* velocity range (selected if the joystick select input is high—sinking current). The JOYVL command establishes the *low* velocity range (selected if the joystick select input is low—not sinking current).

The JOYAXL and JOYAXH commands define which analog channels are to be used with which joystick axes when the joystick select input is low or high, respectively.

Define Usable Voltage

Use the commands described in the following table to establish the joystick's usable velocity range.

The analog-to-digital converter is an 8-bit converter with a voltage range of 0.0V to 2.5V. With 8 bits to represent this range, there are 256 distinct voltage levels from 0.0V to 2.5V. 1 bit represents 2.5/256 or 0.00976 volts/bit.

Command	Name	Purpose
JOYEDB	End Deadband	This command defines voltage levels (shy of the 0.0V and 2.5V endpoints) at which maximum velocity occurs. Specifying an end deadband effectively decreases the voltage range of the analog input to compensate for joysticks that cannot reach the 0.0V and 2.5V endpoints.
JOYCTR (or JOYZ)	Center Voltage*	This command defines the voltage level for the center of the analog input range (the point at which zero velocity will result). As an alternative, you can use the JOYZ command, which reads the current voltage on the joystick input and considers it the center voltage. You can check the center voltage by typing in JOYCTR[cr].**
JOYCDB	Center Deadband	This command defines the voltage range on each side of the center voltage in which no motion will occur (allows for minor drift or variation in the joystick center position without causing motion).

* Because the center voltage can be set to a value other than the exact center of the potentiometer's voltage range, and because there could be two different velocity resolutions, the CW velocity resolution may be different than CCW velocity resolution.

** Because of finite voltage increments, the 6270 may not report back exactly what you specified with the JOYCTR command.

Joystick Control Inputs

The table below lists the 6270's four joystick control inputs and their active levels and what the active levels affect.

Joystick Input Bit	Active Level	Effect of Active Level
Axis select	Ø	Selects JOYAXL
	1	Selects JOYAXH
Velocity select	Ø	Selects JOYVL
	1	Selects JOYVH
Joystick release	Ø	Exit Joystick Mode (equiv. to JOYØØ command). To use the joystick again, issue the JOY11 command.
	1	Stay in Joystick Mode
Joystick trigger (general purpose)	Ø or 1	Interpreted by user program (status is reported with the TINO and INO commands)
Joystick auxiliary (general purpose)	Ø or 1	Interpreted by user program (status is reported with the TINO and INO commands)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Typical Applications

A typical joystick application is two-axis, in which a high velocity range is required to move to a region, then a low velocity range is required for a fine search. After the search is completed it is necessary to record the load positions, then move to the next region. The joystick trigger input can be used to indicate that the position should be read. The joystick release is used to exit the joystick mode and continue with the motion program.

Joystick Set Up Example

The following table describes the requirements of the application (rotary) described above, and how the 6270 is configured to satisfy those requirements. The resulting joystick voltage configuration is illustrated below. Given: one analog input channel is used for each axis.

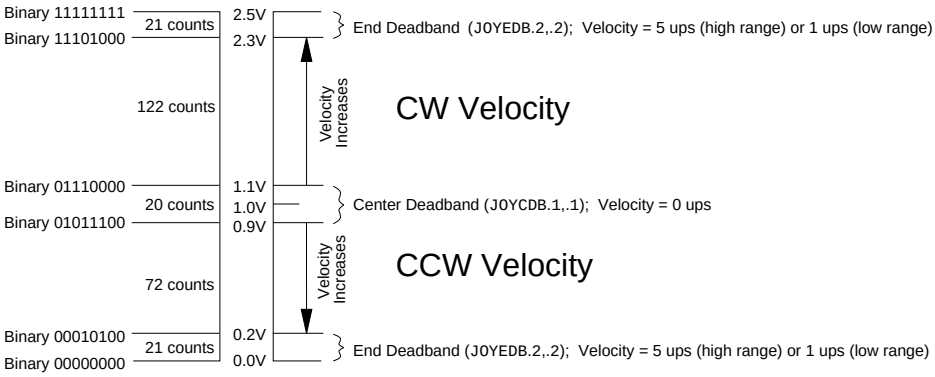
Requirement	Configuration
Set max. high-range velocity to 5 units/sec (on both axes)	Type in the `JOYVH5,5` command
Set max. low-range velocity to 1 units/sec (on both axes)	Type in the `JOYVL1,1` command
No velocity when voltage is at 1.0V	Set center voltage with `JOYCTR1,1,` command, or set voltage level at both analog inputs to 1.0V and type in `JOYZ11`
Joystick cannot reliably rest at 1.0V, but can rest within ±0.1V of 1.0V	Set center deadband of 0.1V with `JOYCDB.1,.1` command (0.1V is the system default)
Joystick can only produce maximum of 2.3V and minimum of 0.2V	Set end deadband to get max. velocity at 2.3V or 0.2V with the `JOYEDB.2,.2` command. Voltage range: CW = 1.1V to 2.3V (1.2V total) CCW = 0.9V to 0.2V (0.7V total) Voltage resolution: see below

The high-range velocity resolutions (at 5 units/sec max.) are calculated as follows:

CW: $\frac{\text{5 units/sec (ups)}}{\text{voltage range of 1.2V (122 counts)}} = 0.041$ ups/count; CCW: $\frac{\text{5 ups}}{\text{voltage range of 0.7V (72 counts)}} = 0.069$ ups/count

The low-range velocity resolutions (at 1 units/sec max.) are calculated as follows:

CW: $\frac{\text{1 ups}}{\text{voltage range of 1.2V (122 counts)}} = 0.008$ ups/count; CCW: $\frac{\text{1 ups}}{\text{voltage range of 0.7V (72 counts)}} = 0.014$ ups/count



Analog Voltage Override

Before you actually wire the analog inputs, you can simulate their activation in software by using the ANVO command. For instance, `ANVO1.2,1.6,1.8` overrides the hardware analog input channels—1.2V on channel 1, 1.6V on channel 2, and 1.8V on channel 3. The ANVO values are used in any command or function that references the analog input channels, but only those channels for which ANVOEN is set to 1 (e.g., Given `ANVOENØ11`, the ANVO values 1.6V and 1.8V are referenced for analog channels 2 and 3 only); otherwise, the voltage is read from the analog input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6270-ANI Option Offers14-Bit Analog Input

The 6270-ANI option offers two ±10V, 14-bit analog inputs (one **ANI** terminal found on each of the **DRIVE** connectors). Each input has an anti-aliasing filter and is sampled at the servo sample rate (set with the SSFR command). The voltage value of the ANI inputs can be transferred to the terminal with the TANI command, or used in an assignment or comparison operation with the [ANI] command (e.g., IF(1ANI<2.4)). The position value of the ANI inputs can be transferred to the terminal with the TPANI command, or used in an assignment or comparison operation with the [PANI] command (e.g., WAIT(1PANI>421)).

Three common applications of the ANI inputs are:

- Position command to the control loop
- Position feedback to the control loop
- A force or torque feedback signal

Programming Example

The following programming example will read the analog inputs into the 6270 and set the commanded analog output of each axis to that value. If you have a torque drive, this provides open-loop torque control.

Command	Description
> SGPØ,Ø	Turn off servo proportional feedback gain
> SGIØ,Ø	Turn off servo integral feedback gain
> SGVØ,Ø	Turn off servo velocity feedback gain
> SGAFØ,Ø	Turn off servo acceleration feedforward gain
> SGVFØ,Ø	Turn off servo velocity feedforward gain
> SOFFSØ,Ø	Set the offset to zero. The analog output will be 0 volts.
> L	Enter an infinite loop
VAR1=1ANI	Read value of ANI analog input #1 into variable #1
VAR2=2ANI	Read value of ANI analog input #2 into variable #2
SOFFS(VAR1),(VAR2)	Assign the voltages from ANI analog inputs #1 & #2 to the analog output for axes #1 & # 2, respectively
T.Ø1	Set time delay to 10 milliseconds
LN	End loop

4-20 mA Feedback

The analog inputs can be used to monitor a process using 4-20 mA feedback. This is accomplished by simply attaching a resistor from the **ANI** input to **AGND** (e.g., a 500Ω, 1% resistor would facilitate a 2-10V input). In this way, the current is converted to a voltage that can be monitored with the ANI input. The PSET command can be used to set the input readings to user coordinates.

ANI as a Feedback Device

The ANI analog inputs, when selected as a feedback source with the SFB command, is assumed to provide position information. With this feedback it is possible to solve applications that require positioning to a voltage, rather than positioning to a known position. Some example applications are as follows:

- Using a potentiometer as feedback (mechanical motion is mimicked by the 6270)
- Maintaining a force while position changes due to fluid evacuating a chamber
- Opening or closing a valve as another process changes

RP240 Remote Operator Panel Interface

The 6270 is directly compatible with the Compumotor RP240 Remote Operator Panel. This section describes how to use the 6270 with the RP240. RP240 connections are demonstrated in Chapter 3, *Installation*.

NOTE

Refer to the ***Model RP240 User Guide*** (p/n 88-012156-01) for user information on *Hardware Specifications*, *Mounting Guidelines*, *Environmental Considerations*, and *Troubleshooting*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Operator Interface Features

The RP240 is used as the 6270's *operator interface*, not a program entry terminal. As an operator interface, the RP240 offers the following features:

- Displays text and variables
- 8 LEDs can be used as programmable status lights
- Operator data entry of variables: read data from RP240 into variables and command value substitutions (see table in Appendix C of software guide)

Typically the user creates a program in the 6270 to control the RP240 display and RP240 LEDs. The program can read data and make variable assignments via the RP240's keypad and function keys.

The 6000 Series software commands for the RP240 are listed below. Detailed descriptions are provided in the ***6000 Series Software Reference Guide*** and the ***Model RP240 User Guide***. The example below demonstrates the majority of the 6000 Series commands for the RP240.

DCLEAR.....Clear The RP240 Display
DJOG.....Enter RP240 Jog Mode
DLED.....Turn RP240 LEDs On/Off
DPASS .....Change RP240 Password
DPCUR .....Position The Cursor On The RP240 Display
[DREAD] .....Read RP240 Data
[DREADF] .....Read RP240 Function Key
DREADI.....RP240 Data Read Immediate Mode
DVAR.....Display Variable On RP240
DWRITE.....Display Text On The RP240 Display

NOTE: If you are not using an RP240, the **RP240** connector can be used as a one-way (transmit only) RS-232C port. The DWRITE command can then be used to send text to a remote RS-232C device, such as an SX.

Example

```
Command                          Description
>  DEF panel1                    Define program panel1
-  REPEAT                        Start of repeat loop
-   DCLEARØ                      Clear display
-   DWRITE"SELECT A FUNCTION KEY"  Display text "SELECT A FUNCTION KEY"
-   DPCUR2,2                     Move cursor to line 2 column 2
-   DWRITE"DIST"                 Display text "DIST"
-   DPCUR2,9                     Move cursor to line 2 column 9
-   DWRITE"GO"                   Display text "GO"
-   DPCUR2,35                    Move cursor to line 2 column 35
-   DWRITE"EXIT"                 Display text "EXIT"
-   VAR1 = DREADF                Input a function key
-   IF (VAR1=1)                  If function key #1 hit
-     GOSUB panel2               GOSUB program panel2
-   ELSE                         Else
-     IF (VAR1=2)                If function key #2 hit
-       DLED1                    Turn on LED #1
-       GO1                      Start motion on axis #1
-       DLEDØ                    Turn off LED #1
-     NIF                        End of IF (VAR1=2)
-   NIF                          End of IF (VAR1=1)
-  UNTIL (VAR1=6)                Repeat until VAR1=6 (function key 6)
-  DCLEARØ                       Clear display
-  DWRITE"LAST FUNCTION KEY = F"   Display text "LAST FUNCTION KEY = F"
-  DVAR1,1,Ø,Ø                   Display variable 1
-  END                           End of panel1
>
>  DEF panel2                    Define prog panel2
-   DCLEARØ                      Clear display
-   DWRITE"ENTER DISTANCE"       Display text "ENTER DISTANCE"
-   D(DREAD)                     Enter distance number from RP240
-  END                           End of panel2
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the Default Mode

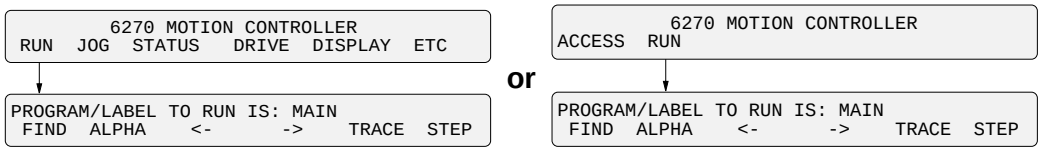
On power-up, the 6270 will automatically default to a mode in which it controls the RP240 with the menu-driven functions listed below.

- Run a stored program (RUN, STOP, PAUSE and CONTINUE functions)
- Jog (open loop and closed loop)
- Display the status of:
  - System (TSS)
  - Axis (TAS)
  - I/O (TIN and TOUT)
  - Limits (TLIM) and Enable input (TINO bit #6)
  - Joystick analog channel voltage values (TANV) and input states (TINO)
  - Position: Encoder (TPE), LDT (TLDT), ANI volts (TANI), Commanded (TPC), Current feedback device (TFB), and Position error (TPER)
  - Firmware revision levels for 6270, DSP and RP240 (TREV)
- Access RP240 menu functions with a security password
- RESET the 6270
- View and edit tuning gains and gain sets
- View numeric (VAR) and string (VARS) variables, and edit numeric variables

The flow chart on the next page illustrates the RP240's menu structure in the default operating mode (when no 6270 user program is controlling the RP240). Press the ***Menu Recall*** key to back up to the previous screen. The menu functions are described in detail below.

To disable these menus, the start-up program (the program assigned with the STARTP command) must contain the DCLEARØ command.

Running a Stored Program



After accessing the RUN menu, press ***F1*** to "find" the names of the programs stored in the 6270's memory; pressing ***F1*** repeatedly displays subsequent programs in the order in which they were stored in BBRAM. To execute the program, press the ***ENTER*** key.

To type in a program name at the location of the cursor, first select alpha or numeric characters with the ***F2*** function key (characters will be alpha if an asterisk appears to the right of ALPHA, or numeric if no asterisk appears). If alpha, press the up (2) or down (8) keys to move through the alphabet, if numeric, press the desired number key. Press ***F3*** to move the cursor to the left, or ***F4*** to move the cursor to the right.

Only user programs defined with DEF and END may be executed from this menu. Contouring paths cannot be executed from this men; they must be executed from the terminal emulator with the PRUN command, or you can place the PRUN (name of path) command in a user program and then execute that program from this menu.

When a program is RUN and TRACE is selected (TRACE*), the RP240 display will trace all program commands as they are executed. This is different from the TRACE command in that the trace output goes to the RP240 display, not to a terminal via the RS-232C port.

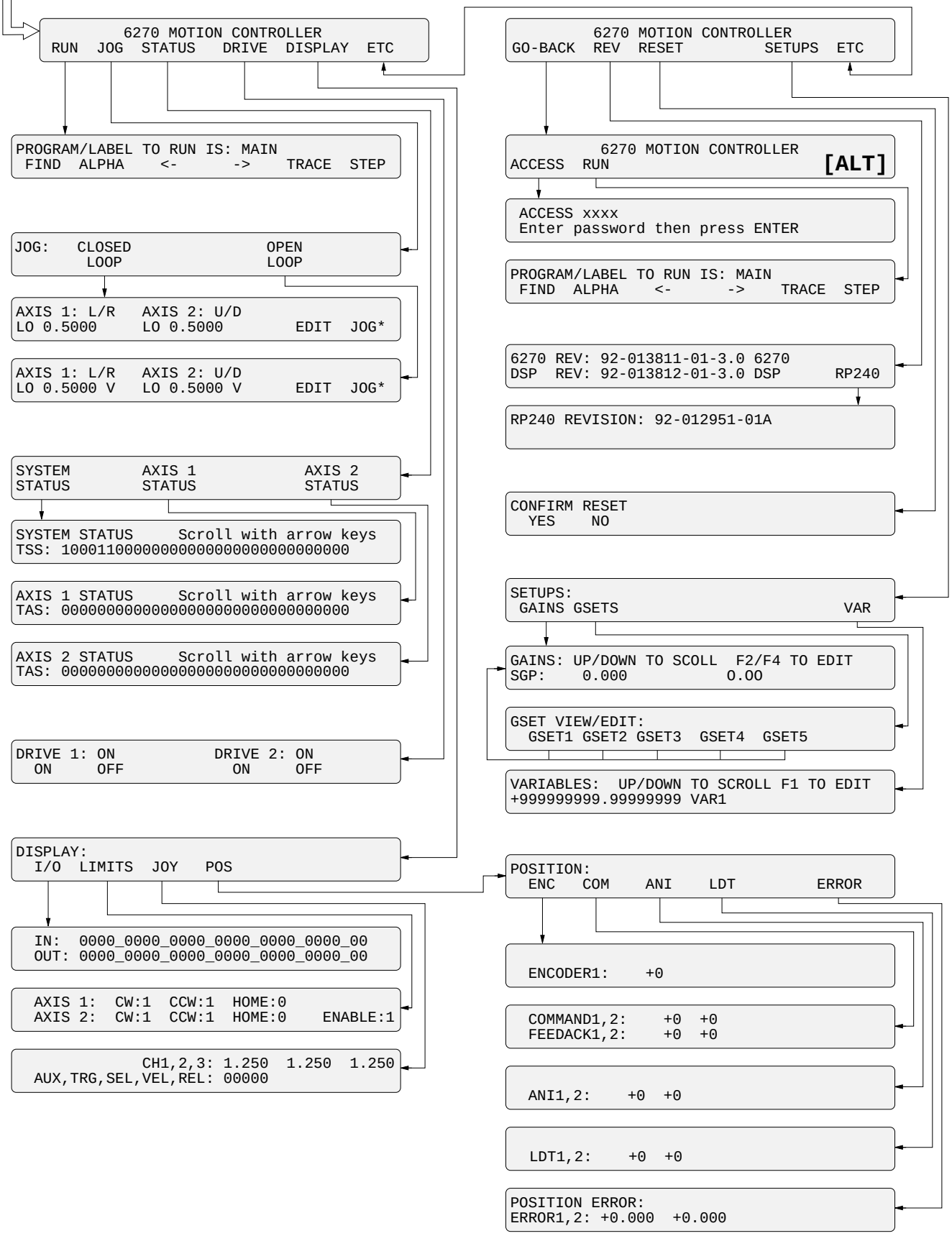
HINT: If you wish to display each command as it is executed, select STEP and TRACE and press the ENTER key to step through the program.

When a program is RUN and STEP is selected, step mode has been entered. This is similar to the STEP command, but when selected from the RUN menu the step mode allows single stepping by pressing the ***ENTER*** key. Both RP240 trace mode and step mode are exited when program execution is terminated.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

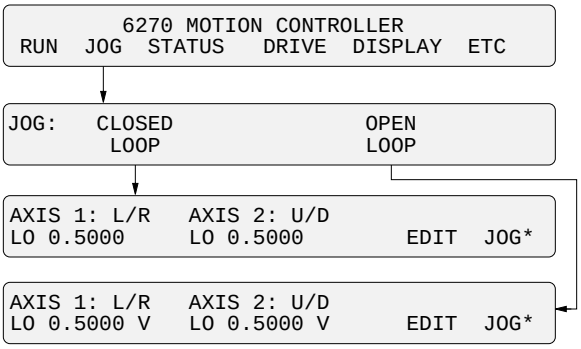
Default Power-up Menu

- If you change the password (default is “6270”) with the DPASS command, the menu marked with **[ALT]** becomes the power-up menu.
- Arrows indicate the menu path when you press the corresponding function key below the menu item.
- To back up to the previous menu item, press the *MENU RECALL* button.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Jogging



You can jog individual axes by pressing the RP240 arrow keys. Pressing an arrow key on the numeric keypad will start motion and releasing the arrow key will stop motion. The left and right arrow keys correspond to axis #1 negative and positive direction, respectively. The up and down arrows keys are for axis #2 negative and positive direction, respectively.

The HI and LO values in the closed-loop jog menu represent the velocity in units of inches/sec (if scaling is enabled, the value is multiplied by the programmed SCLV value). The HI and LO values in the open-loop jog menu represent volts.

To edit the jog velocity/voltage values:

1. Press the ***F5*** function key under EDIT (edit mode indicated with an asterisk).
2. Press the ***F1*** or the ***F3*** function keys to select the HI and LO values for axis 1 and 2, respectively (cursor appears under the first digit of the value selected).
3. Using the numeric keyboard, enter the value desired.
4. Repeat steps 2 and 3 for all values to be changed.
5. Press ***ENTER*** when finished editing.
6. To jog with the new velocity/voltage values, first press the ***F6*** function key (under JOG) to enable the arrow keys again.

Closed-loop jog acceleration and deceleration values are specified by JOGA and JOGAD commands, respectively.

Open Loop Jogging

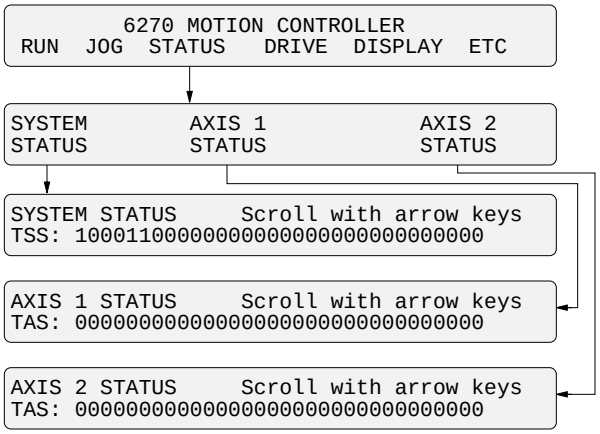
When using the open-loop jog mode, you should be aware of these conditions:

- **WARNING** — The hardware and software end-of-travel limits are disabled. Make sure that it is safe to operate without end-of-travel limits before using the open-loop jog function.
- Gain values (SGILIM, SGAF, SGAFN, SGI, SGIN, SGP, etc.) set to zero (open-loop operation).
- SMPER value set to zero (position error is allowed to increase without causing a fault).
- Subsequent attempts to change gain values or SMPER will cause an error message ("NOT ALLOWED IF SFBØ").
- SOFFS and SOFFSN set to zero, but allows subsequent servo offset changes to affect motion.
- SSWG set to zero (disables the Setpoint Window Gains feature).
- Disables output-on-position (OUTPA - OUTPD) functions.
- Any subsequent changes to PSET, PSETCLR, SCLD, SCLA, SCLV, SOFFS, and SOFFN are lost when another feedback source is selected.

Recommendation — Use the Disable Drive On Kill mode, enabled with the KDRIVE command, so that the 6270 will shut down the valve/drive if a kill command (e.g., !K) is executed or if a kill input is activated. **CAUTION**: Shutting down a valve/cylinder system returns the valve to the null position; shutting down a rotary drive system allows the load to freewheel if there is no brake installed.

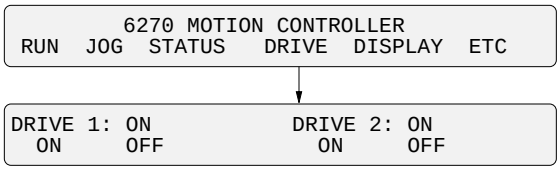
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Status Reports: *System & Axis*



After accessing the desired status menu, you can ascertain the function and status of each system (TSS) or axis (TAS) status bit by pressing the arrow keys on the numeric keypad.

Enabling and Disabling the Valves or Drives

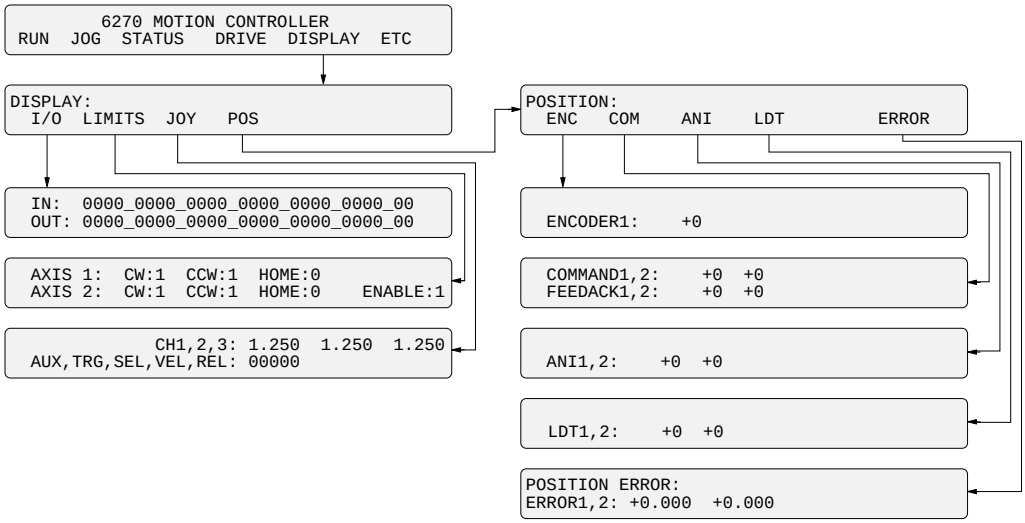


In the Drive menu, the current status of the valves/drives is displayed. To enable or disable valve/drive #1, press ***F1*** or ***F2***; press ***F4*** or ***F5*** to enable or disable valve/drive #2. This menu offers the same functionality as the DRIVE command.

CAUTION

Shutting down a valve/cylinder system returns the valve to the null position. Shutting down a rotary drive system allows the load to freewheel if there is no brake installed.

Status Reports: *I/O, Limits, Joystick, & Position*



I/O Menu:

- Input bit pattern (left to right): Bits 1-24 are the general-purpose programmable inputs, bits 25 &26 are triggers A and B (**TRG-A** and **TRG-B** on the **AUX** connector).
- Output bit pattern (left to right): Bits 1-24 are the general-purpose programmable outputs, bits 25 &26 are auxiliary outputs A and B (**OUT-A** and **OUT-B** on the **AUX** connector).

LIMITS Menu:

- "CW" refers to the hardware end-of-travel limit imposed when counting in the positive direction; this usually implies extension for cylinders. "CCW" refers to limit imposed when counting in the negative direction; this usually implies retraction for cylinders.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- A “1” indicates that the input is grounded, “0” indicates not grounded.

  The end-of-travel limits (CW and CCW) must be grounded to allow motion (this is reversed if the active level is reversed with the LHLVL command).

  The Enable input (**ENBL** terminal on the **AUX** connector) must also be grounded before motion is allowed. When not grounded, the analog output is held to zero volts and the shutdown outputs are activated.

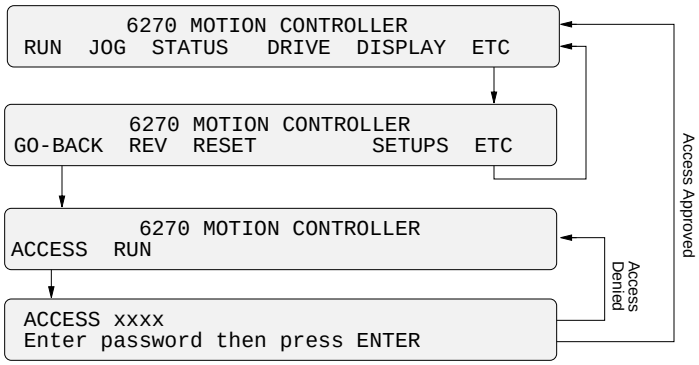
JOY Menu:

- The voltage levels present on the 3 analog channel on the 25-pin D **JOYSTICK** connector are listed.
- Also listed is the status of the joystick inputs (“1” indicates that the input is grounded/low, “0” indicates not grounded). AUX, TRG, SEL, VEL, and REL correspond respectively to pins 19, 18, 15, 16, and 17 on the **JOYSTICK** connector.

POS Menu (and sub-menus):

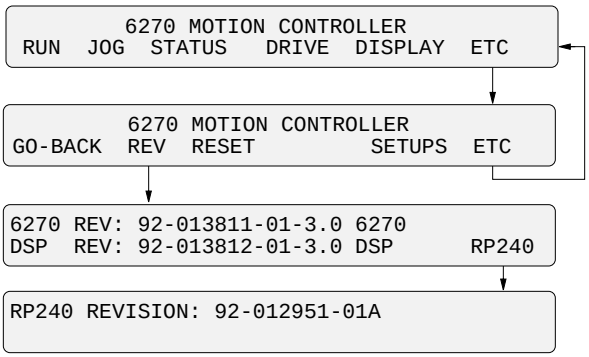
- The position and position error values shown are continually updated.
- Position values (except for ANI) are subject to the SCLD scaling factor (if scaling is enabled—SCALE1), PSET offset value, feedback polarity (ENCPOL and LDTPOL), and commanded direction polarity (CMDDIR). The ANI reading is always volts and is not scaled.
- “FEEDBACK” in the COM sub-menu refers to the feedback device currently selected with the SFB command (default is LDT).
- Position error = command position - actual (“feedback”) position.

Access Security



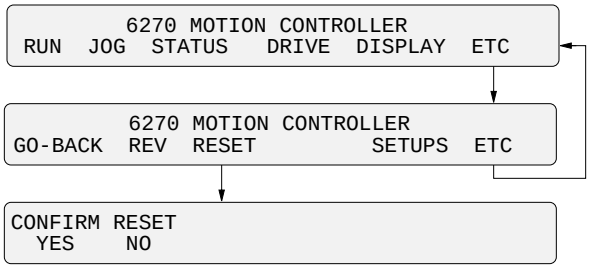
If the RP240 password is modified with the DPASS command to be other than DPASS6270 (the default password is “6270”) the access menu then becomes the new default menu after power-up or executing a RESET. After that, the new password must then be entered to access the original default menu (see Access Approved path in illustration above). If the operator does not know the new password, all he or she can do is execute programs stored in the 6270 (RUN).

Revision Levels



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Resetting the 6270

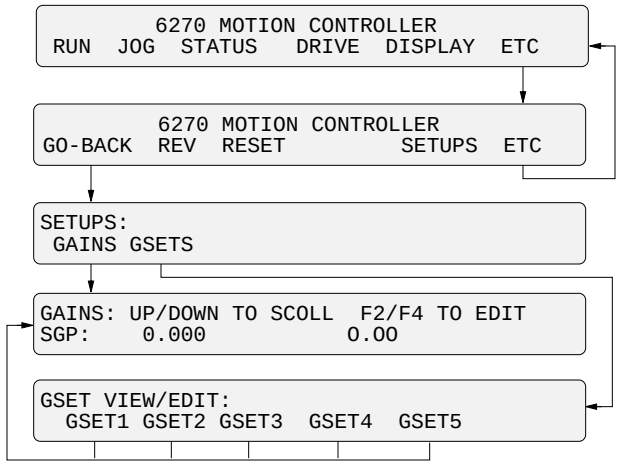


After accessing the RESET menu, press the ***F1*** key to execute a reset (or press ***F2*** to cancel and exit the menu). The reset is identical to issuing a RESET command or cycling power to the 6270. If a start-up program has been assigned with the STARTP command, that program will be executed.

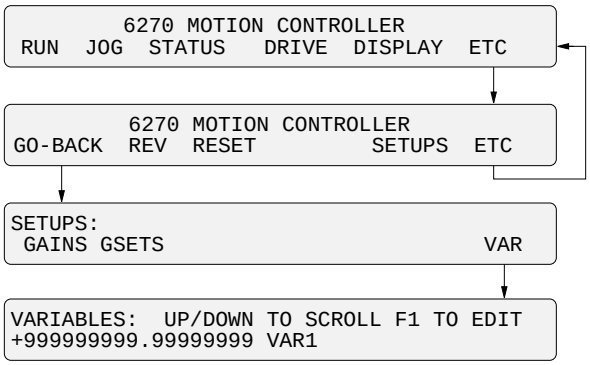
CAUTION

Executing a reset will restore all command values (those not saved in programs or variables) to their factory default setting.

Viewing and Editing Gains and Gain Sets



Viewing and Editing Variables



From the VAR menu, you can view all 150 numeric variables. The value and text of the first 25 numeric and string variables are presented side by side (e.g., the numeric value of VAR1 is displayed next to the text string in VARS1).

If no text is assigned to the corresponding string variable, the name of the variable is displayed (see example above).

To edit the displayed numeric variable, press ***F2*** (cursor appears) and type the new value on the numeric keypad.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Another choice for a user interface is to use a host computer and execute a motion program using the RS-232C serial interface. A host computer may be used to run a motion program interactively from a BASIC or C program (high-level program controls the 6270 and acts as a user interface). A BASIC program example is provided below.

```
10  '      6000 Series Serial Communication BASIC Routine
12  '                        6000.BAS
14  '
16  ' ***********************************************************
18  '
20  ' This program will set the communications parameters for the
22  ' serial port on a PC to communicate with a 6000 series
24  ' stand-alone product.
26  '
28  ' ***********************************************************
30  '
100 '*** open com port 1 at 9600 baud, no parity, 8 data bits, 1 stop bit
110 '*** activate Request to Send (RS), suppress Clear to Send (CS), suppress
120 '*** DATA set ready (DS), and suppress Carrier Detect (CD) ***
130 OPEN "COM1:9600,N,8,1,RS,CS,DS,CD" FOR RANDOM AS #1
140 '
150 '*** initialize variables ***
160 MOVE$ = ""        ' *** commands to be sent to the product ***
170 RESPONSE$ = ""    ' *** response from the product ***
180 POSITION$ = ""    ' *** feedback position reported ***
190 SETUP$ = ""       ' *** setup commands ***
200 '
210 '*** format the screen and wait for the user to start the program ***
220 CLS : LOCATE 12, 20
230 PRINT "Press any key to start the program"
240 '
250 '*** wait for the user to press a key ***
260 PRESS$ = INKEY$
270 IF PRESS$ = "" THEN 260
280 CLS
290 '
300 '*** set a pre-defined move to make ***
310 SETUP$ = "ECHO1:ERRLVL0:LH0,0:"
320 MOVE$ = "A100,100:V2,2:D1000,1000:GO11:TFB:"
330 '
340 '
400 '*** send the commands to the product ***
410 PRINT #1, SETUP$
420 PRINT #1, MOVE$
430 '
500 '*** read the response from the TPE command ***
510 '   *** the controller will send a leading "+" or "-" in response to the TFB command to
520 '   *** indicate which direction travel has occurred. ***
530 WHILE (RESPONSE$ <> "+" AND RESPONSE$ <> "-") ' *** this loop waits for the "+"
540   RESPONSE$ = INPUT$(1, #1)                   ' *** or "-" characters to be returned
550 WEND                                          ' *** before reading the position ***
560 '
570 WHILE (RESPONSE$ <> CHR$(13))        ' *** this loop reads one character at a time
580   POSITION$ = POSITION$ + RESPONSE$  ' *** from the serial buffer until a carriage
590   RESPONSE$ = INPUT$(1, #1)          ' *** return is encountered ***
600 WEND
610 '
620 '*** print the response to the screen ***
630 LOCATE 12, 20: PRINT "Position is " + POSITION$
640 '
650 'END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Variables

HINT: *Numeric and string variables may be edited via the RP240 (see* RP240 Remote Operator Panel Interface *section earlier in this chapter).*

The 6270 has 3 types of variables (numeric, binary, and string). There are 150 numeric variables, numbered 1 - 150. There are 25 binary and string variables, numbered 1 - 25. Each type of variable is designated with a different command. The VAR command designates a numeric variable, the VARB command designates a binary variable, and the VARS command designates a string variable.

Variables do not share the same memory (i.e., VAR1, VARB1, and VARS1 can all exist at the same time and operate separately).

Numeric variables are used to store numeric values with a range of -999,999,999.ØØØØØØØØ to 999,999,999.99999999. Mathematical, trigonometric, and boolean operations are performed using numeric variables.

Binary variables can be used to store 32-bit binary or hexadecimal values. Binary variables can also store I/O, system, axis, or error status (e.g., the VARB2=IN.12 command assigns input bit 12 to binary variable 2). Bitwise operations are performed using binary variables.

String variables are used to store message strings of 20 characters or less. These message strings can be predefined error messages, user messages, etc.

NOTE

The programming examples in this section make use of the colon (:) command delimiter to allow entering more than one command per line.

Converting Between Binary and Numeric Variables

Using the Variable Type Conversion (VCVT) operator, you can convert numeric values to binary values, and vice versa. The operation is a signed operation as the binary value is interpreted as a two's complement number. Any *don't cares* (X) in a binary value is interpreted as a zero (Ø).

If the mathematical statement's result is a numeric value, then VCVT converts binary values to numeric values. If the statement's result is a binary value, then VCVT converts numeric values to binary values.

Numeric to Binary

Example	Description/Response
> VAR1=-5	Set numeric variable value = -5
> VARB1=VCVT(VAR1)	Convert the numeric value to a binary value
> VARB1	*VARB1=11Ø1_1111_1111_1111_1111_1111_1111_1111

Binary to Numeric

Example	Description/Response
> VARB1=bØØ1Ø_Ø11Ø_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ	Set binary variable = +100.0
> VAR1=VCVT(VARB1)	Convert the binary value to a numeric value
> VAR1	*VAR1=+1ØØ.Ø

Performing Operations with Numeric Variables

Some Math Operations Reduce Precision

The following math operations reduce the precision of the return value: Division and Trigonometric functions yield 5 decimal places; Square Root yields 3 decimal places; and Inverse Trigonometric functions yield 2 decimal places.

Mathematical Operations

The following examples demonstrate how the 6270 can perform math operations with its numeric variables. Operator precedence occurs from left to right (e.g., VAR1=1+1+1*3 sets VAR1 to 9, not 5).

Addition (+)

Example	Response
> VAR1=5+5+5+5+5+5+5 : VAR1	+VAR1=35.Ø
> VAR23=1ØØØ.565	
> VAR11=VAR1+VAR23 : VAR11	*VAR11=+1Ø35.565
> VAR1=VAR1+5 : VAR1	*VAR1=+4Ø.Ø

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Subtraction (-)

Example	Response
> VAR3=2Ø-1Ø	
> VAR2Ø=15.5	
> VAR3=VAR3-VAR2Ø : VAR3	*VAR3=-5.5

Multiplication (*)

Example	Response
> VAR3=1Ø	
> VAR3=VAR3*2Ø : VAR3	*VAR3=+2ØØ.Ø

Division (/)

Example	Response
> VAR3=1Ø	
> VAR2Ø=15.5 : VAR2Ø	*+15.5
> VAR3=VAR3/VAR2Ø : VAR3	*+Ø.64516
> VAR3Ø=75 : VAR3Ø	*+75.Ø
> VAR19=VAR3Ø/VAR3 : VAR19	*+116.25Ø23

Square Root

Example	Response
> VAR3=75	
> VAR2Ø=25	
> VAR3=SQRT(VAR3) : VAR3	*+8.66Ø
> VAR2Ø=SQRT(VAR2Ø)+SQRT(9)	
> VAR2Ø	*+8.Ø

Trigonometric Operations

The following examples demonstrate how the 6270 can perform trigonometric operations with its numeric variables.

Sine

Example	Response
> RADIANØ	
> VAR1=SIN(Ø) : VAR1	*VAR1=+Ø.Ø
> VAR1=SIN(3Ø) : VAR1	*VAR1=+Ø.5
> VAR1=SIN(45) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=SIN(6Ø) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=SIN(9Ø) : VAR1	*VAR1=+1.Ø
> RADIAN1	
> VAR1=SIN(Ø) : VAR1	*VAR1=+Ø.Ø
> VAR1=SIN(PI/6) : VAR1	*VAR1=+Ø.5
> VAR1=SIN(PI/4) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=SIN(PI/3) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=SIN(PI/2) : VAR1	*VAR1=+1.Ø

Cosine

Example	Response
> RADIANØ	
> VAR1=COS(Ø) : VAR1	*VAR1=+1.Ø
> VAR1=COS(3Ø) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=COS(45) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=COS(6Ø) : VAR1	*VAR1=+Ø.5
> VAR1=COS(9Ø) : VAR1	*VAR1=+Ø.Ø
> RADIAN1	
> VAR1=COS(Ø) : VAR1	*VAR1=+1.Ø
> VAR1=COS(PI/6) : VAR1	*VAR1=+Ø.866Ø3
> VAR1=COS(PI/4) : VAR1	*VAR1=+Ø.7Ø711
> VAR1=COS(PI/3) : VAR1	*VAR1=+Ø.5
> VAR1=COS(PI/2) : VAR1	*VAR1=+Ø.Ø

Tangent

Example	Response
> RADIANØ	
> VAR1=TAN(Ø) : VAR1	*VAR1=+Ø.Ø
> VAR1=TAN(3Ø) : VAR1	*VAR1=+Ø.57735
> VAR1=TAN(45) : VAR1	*VAR1=+1.Ø
> VAR1=TAN(6Ø) : VAR1	*VAR1=+1.732Ø5
> RADIAN1	
> VAR1=TAN(Ø) : VAR1	*VAR1=+Ø.Ø
> VAR1=TAN(PI/6) : VAR1	*VAR1=+Ø.57735
> VAR1=TAN(PI/4) : VAR1	*VAR1=+1.Ø
> VAR1=TAN(PI/3) : VAR1	*VAR1=+1.732Ø5

Inverse Tangent (Arc Tangent)

Example	Response
> RADIANØ	
> VAR1=SQRT(2)	
> VAR1=ATAN(VAR1/2) : VAR1	*VAR1=+35.26
> VAR1=ATAN(.57735) : VAR1	*VAR1=+3Ø.Ø

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Boolean Operations

The 6270 has the ability to perform Boolean operations with its numeric variables. The following examples illustrate this capability. Refer to the ***6000 Series Software Reference Guide*** for more information.

Boolean And (&)

```
Example                          Response
> VAR1=5 : VAR2=-1
> VAR3=VAR1 & VAR2 : VAR3        *VAR3=+Ø.Ø
```

Boolean Or (|)

```
Example                          Response
> VAR1=5 : VAR2=-1
> VAR3=VAR1 | VAR2 : VAR3        *VAR3=+1.Ø
```

Boolean Exclusive Or (^)

```
Example                          Response
> VAR1=5 : VAR2=-1
> VAR3=VAR1 ^ VAR2 : VAR3        *VAR3=+1.Ø
```

Boolean Not (~)

```
Example                          Response
> VAR1=5
> VAR3=~(VAR1) : VAR3            *VAR3=+Ø.Ø
> VAR1=-1
> VAR3=~(VAR1) : VAR3            *VAR3=+1.Ø
```

Performing Operations with Binary Variables

The 6270 has the ability to perform bitwise functions with its binary variables. The following examples illustrate the bit manipulation capabilities of the 6270.

Bitwise And (&)

```
Example    > VARB1=b11Ø1 : VARB1
  Response   *VARB1=11Ø1_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX

Example    > VARB1=VARB1 & bXXX1 11Ø1 : VARB1
  Response   *VARB1=XXØ1_XXØX_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX

Example    > VARB1=hØØ32 FDA1 & h1234 43E9 : VARB1
  Response   *VARB1=ØØØØ_ØØØØ_11ØØ_ØØØØ_ØØ1Ø_1ØØØ_Ø1Ø1_1ØØØ
```

Bitwise Or (|)

```
Example    > VARB1=h32FD : VARB1
  Response   *VARB1=11ØØ_Ø1ØØ_1111_1Ø11_ØØØØ_ØØØØ_ØØØØ_ØØØØ

Example    > VARB1=VARB1 | bXXX1 11Ø1 : VARB1
  Response   *VARB1=11X1_11Ø1_1111_1X11_XXXX_XXXX_XXXX_XXXX

Example    > VARB1=hØØ32 FDA1 | h1234 43E9 : VARB1
  Response   *VARB1=1ØØØ_Ø1ØØ_11ØØ_Ø11Ø_1111_1111_Ø111_1ØØ1
```

Bitwise Exclusive Or (^)

```
Example    > VARB1=h32FD ^ bXXX1 11Ø1 : VARB1
  Response   *VARB1=XXX1_1ØØ1_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX

Example    > VARB1=hØØ32 FDA1 ^ h1234 43E9 : VARB1
  Response   *VARB1=1ØØØ_Ø1ØØ_ØØØØ_Ø11Ø_11Ø1_Ø111_ØØ1Ø_ØØØ1
```

Bitwise Not (~)

```
Example    > VARB1=~(h32FD) : VARB1
  Response   *VARB1=ØØ11_1Ø11_ØØØØ_Ø1ØØ_1111_1111_1111_1111

Example    > VARB1=~(b1Ø1Ø XX11 Ø1Ø1) : VARB1
  Response   *VARB1=Ø1Ø1_XXØØ_1Ø1Ø_XXXX_XXXX_XXXX_XXXX_XXXX
```

Shift Left to Right (>>)

```
Example    > VARB1=h32FD >> h4 : VARB1
  Response   *VARB1=ØØØØ_11ØØ_Ø1ØØ_1111_1Ø11_ØØØØ_ØØØØ_ØØØØ

Example    > VARB1=b1Ø1Ø XX11 Ø1Ø1 >> b11 : VARB1
  Response   *VARB1=ØØØ1_Ø1ØX_X11Ø_1Ø1X_XXXX_XXXX_XXXX_XXXX
```

Shift Right to Left (<<)

```
Example    > VARB1=h32FD << h4 : VARB1
  Response   *VARB1=Ø1ØØ_1111_1Ø11_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ

Example    > VARB1=b1Ø1Ø XX11 Ø1Ø1 << b11 : VARB1
  Response   *VARB1=ØXX1_1Ø1Ø_1XXX_XXXX_XXXX_XXXX_XXXX_XØØØ
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Teach Mode

The Teach Mode is simply a method of storing (*teaching*) variable data and later using the stored data as a source for motion program parameters. The variable data can be any value that can be stored in a numeric (VAR) variable (e.g., position, acceleration, velocity, etc). The variable data is stored into a *data program*, which is an array of *data elements* that have a specific address from which to write and read the variable data. Data programs do not contain 6000 Series commands.

The information below describes the principles of using the data program in a teach mode application. Following that is a teach mode application example in which the joystick is used to teach position data to be used in a motion program.

Teach Mode Basics

The basic process of using a data program for teach mode applications is as follows:

1. Initialize a data program.
2. Teach (store/write) variable data into the data program.
3. Read the data elements from the data program into a motion program.

Initialize a Data Program

This is accomplished with the DATSIZ command. The DATSIZ command syntax is DATSIZi<,i>. The first integer (i) represents the number of the data program (1 - 50). You can create up to 50 separate data programs. The data program is automatically given a specific program name (DATPi). The second integer represents the total number of data elements (up to 6,500) you want in the data program. Upon issuing the DATSIZ command, the data program is created with all the data elements initialized with a value of zero.

The data program has a tabular structure, where the data elements are stored 4 to a line. Each line of data elements is called a *data statement*. Each element is numbered in sequential order from left to right (1 - 4) and top to bottom (1 - 4, 5 - 8, 9 - 12, etc.). You can use the TPROG DATPi command ("i" represents the number of the data program) to display all the data elements of the data program.

For example, if you issue the DATSIZ1,13 command, data program #1 (called DATP1) is created with 13 data elements initialized to zero. The response to the TPROG DATP1 command is depicted below. Each line (*data statement*) begins with DATA=, and each data element is separated with a comma.

```
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø
```

Each data statement, comprising four data elements, uses 39 bytes of memory. The memory for each data statement is subtracted from the memory allocated for user programs (see MEMORY command).

Teach the Data to the Data Program

The data that you wish to write to the data elements in the data program must first be placed into numeric variables (VAR). Once the data is stored into numeric variables, the data elements in the data program can be edited by using the Data Pointer (DATPTR) command to move the data pointer to that element, and then using the Data Teach (DATTCH) command to write the datum from the numeric variable into the element.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When the DATSIZ command is issued, the internal data pointer is automatically positioned to data element #1. Using the default settings for the DATPTR command, the numeric variable data is written to the data elements in sequential order, incrementing one by one. When the last data element in the data program is written, the data pointer is automatically set to data element #1 and a warning message (*WARNING: POINTER HAS WRAPPED AROUND TO DATA POINT 1) is displayed. The warning message does not interrupt program execution.

The DATPTR command syntax is DATPTRi,i,i. The first integer (i) represents the data program number (1 through 50). The second integer represents the number of the data element to point to (1 through 6500). The third integer represents the number of data elements by which the pointer will increment after writing each data element from the DATTCH command, or after recalling a data element with the DAT command.

The DATTCH command syntax is DATTCHi<,i,i,i>. Each integer (i) represents the number of a numeric variable. The value of the numeric variable will be stored into the data element(s) of the currently active data program (i.e., the program last specified with the last DATSIZ or DATPTR command). As indicated by the number of integers in the syntax, the maximum number of variable values that can be stored in the data program per DATTCH command is 4. Each successive value from the DATTCH command is stored to the data program according to the pattern established by the third integer of the DATPTR command.

As an example, suppose data program #1 is configured to hold 13 data elements (DATSIZ1,13), the data pointer is configured to start at data element #1 and increment 1 data element after every value stored from the DATTCH command (DATPTR1,1,1), and the values of numeric variables #1 through #3 are already assigned (VAR1=2, VAR2=4, VAR3=8). If you then enter the DATTCH1,2,3 command, the values of VAR1 through VAR3 will be assigned respectively to the first three data elements in the data program, leaving the pointer pointing to data element #4. The response to the TPROG DATP1 command would be as follows (the text is highlighted to illustrate the final location of the data pointer after the DATTCH1,2,3 command is executed):

```
*DATA=2.Ø,4.Ø,8.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø
```

If you had set the DATPTR command to increment 2 data elements after every value from the DATTCH command (DATPTR1,1,2), the data program would be filled differently and the data pointer would end up pointing to data element #7:

```
*DATA=2.Ø,Ø.Ø,4.Ø,Ø.Ø
*DATA=8.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø,Ø.Ø,Ø.Ø,Ø.Ø
*DATA=Ø.Ø
```

Recall the Data from the Data Program

After storing (*teaching*) your variables to the data program, you can use the DATPTR command to point to the data elements and the DATi ("i" = data program number) data assignment command to read the stored variables to your motion program. *You cannot recall more than one data element at a time; therefore, if you want to recall the data in a one-by-one sequence, the third integer of the DATPTR command must be a 1 (this is the default setting).*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Summary of Related 6000 Series Commands

NOTE: A detailed description of each command is provided in the **6000 Series Software Reference Guide**.

DATSIZ..... Establishes the number of data elements a specific data program is to contain. A new DATPi program name is automatically generated according to the number of the data program (i = 1 through 50). The memory required for the data program is subtracted from the memory allocated for user programs (see MEMORY command).

DATPTR..... Moves the data pointer to a specific data element in any data program. This command also establishes the number of data elements by which the pointer increments after writing each data element from the DATTCH command and after recalling each data element with the DAT command.

DATTCH..... Stores the variable data into the data program specified with the last DATSIZ or DATPTR command. After the data is stored, the data pointer is incremented the number of times entered in the third integer of the DATPTR command. *The data must first be assigned to a numeric variable before it can be taught to the data program.*

TDPTR..... Responds with a 3-integer status report (i,i,i): First integer is the number of the active data program (the program # specified with the last DATSIZ or DATPTR command); Second integer is the location number of the data element to which the data pointer is currently pointing; Third integer is the increment set with the last DATPTR command.

[DPTR].... From the currently active data program, uses the number of the data pointer's location in a numeric variable assignment operation or a conditional statement operation.

[DATPi].. The name of the data program created after issuing the DATSIZ command. The integer (i) represents the number of the data program. Data programs can be deleted just like any other user program (e.g., DEL DATP1).

[DATi].... From the data program specified with i, assigns the numeric value of the data element (currently pointed to by the data pointer) to a specified variable parameter in a 6000 series command (e.g., D(DAT3),(DAT3)).

Teach Mode Application Example

In this example, 2 axes of the 6270 are used to move a 2-axis stage. This example illustrates a common method of teaching a path by using the joystick to move the load into position, teach the position (triggered by the Joystick Release input), then move to the next position. Five positions will be taught from each axis (2 axes at one trigger), for a total of 10 data elements in the data program. After all 10 positions are taught to the data program, the 6270 will automatically move both axes to a home position, move to each position that was taught, and then return to the home position.

For the sake of brevity, this example is limited to teaching 10 position data points; however, in a typical application, many more points would be taught. Also, it is assumed that end-of-travel and home limits are wired and a homing move has been programmed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

What follows is a suggested method of programming the 6270 for this application. To accomplish the teach mode application, a program called MAIN is created, comprising three subroutines: SETUP (to set up for teach mode), TEACH (to teach the positions), and DOPATH (to implement a motion program based on the positions taught).

The joystick operation in this example is based on setting the Joystick Axes Select input (pin #15 on the Joystick connector) to high to select analog input channels #1 and #2 (pins #1 and #2) for joystick use, and using the Joystick Release input (pin #17) to trigger the position teach operation.

Step 1 **Initialize a Data Program.**

> DEL DATP1	Delete data program #1 (DATP1) in preparation for creating a new data program #1
> DATSIZ1,1Ø	Create data program #1 (named DATP1) with an allocation of 10 data elements. Each element is initialized to zero.

Step 2 **Define the SETUP Subroutine.** Note that the SETUP subroutine need only run once.

> DEF SETUP	Begin definition of the subroutine called SETUP
- JOYVH3,3	Set the high velocity speed to 3 units/sec
- JOYVL.2,.2	Set the low velocity to 0.2 units/sec
- JOYAXH1,2	When axes select input is set high, apply analog input 1 to axis 1 and apply analog input 2 to axis 2
- VAR1=Ø	Set variable #1 equal to zero
- VAR2=Ø	Set variable #2 equal to zero
- DRIVE11	Enable the drives for both axes
- MA11	Enable the absolute positioning mode for both axes
- END	End definition of the subroutine called SETUP

Step 3 **Define the TEACH Subroutine.**

> DEF TEACH	Begin definition of the subroutine called TEACH
- HOM11	Home both axes (absolute position counter is set to zero after homing move)
- DATPTR1,1,1	Select data program #1 (DATP1) as the current active data program, and move the data pointer to the first data element. After each DATTCH value is stored to DATP1, increment the data pointer by 1 data element.
- REPEAT	Set up a repeat/until loop
- JOY11	Enable joystick mode on both axes. At this point, you can start moving the axes into position with the joystick. While using the joystick, command processing is stopped here until you activate the joystick release input. Activating the joystick release input disables the joystick mode and allows the subsequent commands to be executed (assign the current positions to the variables and then store the positions in the data program).
- VAR1=1PM	Set variable #1 equal to the position of motor 1
- VAR2=2PM	Set variable #2 equal to the position of motor 2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

-	DATTCH1,2	Store variable #1 and variable #2 into consecutive data elements. (The first time through the repeat/until loop, variable #1 is stored into data element #1 and variable #2 is stored into data element #2. The data pointer is automatically incremented once after each data element and ends up pointing to the third data element in anticipation of the next DATTCH command.)
-	WAIT(INO.5=b1)	Wait for the joystick release input to be de-activated
-	UNTIL(DPTR=1)	Repeat the loop until the data pointer wraps around to data element #1 (data program full)
-	END	End definition of the subroutine called TEACH

Step 4 **Define the DOPATH Subroutine.**

>	DEF DOPATH	Begin definition of the subroutine called DOPATH
-	HOM11	Move both axes to the home position (absolute counters set to zero)
-	A5Ø,5Ø	Set up the acceleration
-	V3,3	Set up the velocity
-	DATPTR1,1,1	Select data program #1 (DATP1) as the current active data program, and set the data pointer to the first data element. Increment the data pointer one element after every data assignment with the DAT command. *If you wanted to move only axis 1 down the taught path, you would set the increment (third integer) to a 2, thus accessing only the axis 1 stored positions.*
-	REPEAT	Set up a repeat/until loop
-	D(DAT1),(DAT1)	The position of axis 1 and axis 2 are recalled into the distance command
-	GO11	Move to the position
-	T.5	Wait for 0.5 seconds
-	UNTIL(DPTR=1)	Repeat the loop until the data pointer wraps around to data element #1 (all data elements have been read)
-	HOM11	Move both axes back to the home position
-	END	End definition of the subroutine called DOPATH

Step 5 **Define the MAIN Program (Include SETUP, TEACH, and DOPATH).**

>	DEF MAIN	Begin definition of the program called MAIN
-	SETUP	Execute the subroutine called SETUP
-	TEACH	Execute the subroutine called TEACH
-	DOPATH	Execute the subroutine called DOPATH
-	END	End definition of the program called MAIN

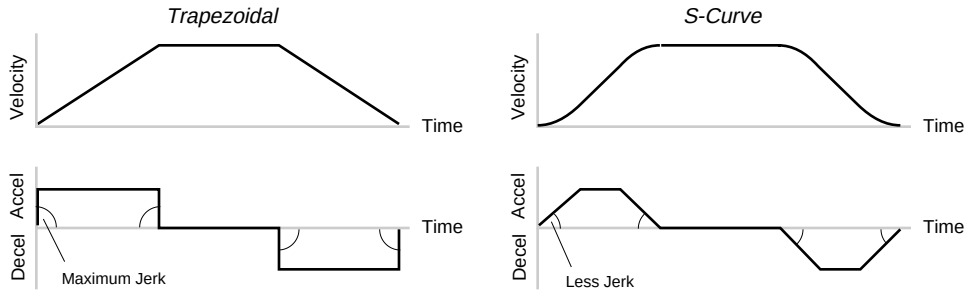
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 6 **Run the `MAIN` Program and Teach the Positions with the Joystick.**

1. Enter the `MAIN` command to execute the teach mode program and set the joystick's *axis select* input to high.
2. Use the joystick to move to the position to be taught.
3. Once in position, activate the *joystick release* input to teach the positions. Two positions (one for each axis) are taught each time you activate the joystick release input.
4. Repeat steps 2 and 3 for the remaining four teach locations. After triggering the joystick release input the fifth time, the 6270 will home the axes, repeat the path that was taught, and then return both axes to the home position.

S-Curve Profiling

The 6270 allows you to perform *S-curve* move profiles, in addition to the usual trapezoidal profiles. S-curve profiling provides smoother motion control by reducing the *jerk* (rate of change) in acceleration and deceleration portions of the move profile (see drawing below).



S-curves improve position tracking.

Because S-curve profiling reduces jerk, it improves position tracking performance in servo systems, especially in linear interpolation applications.

S-curve programming requirements.

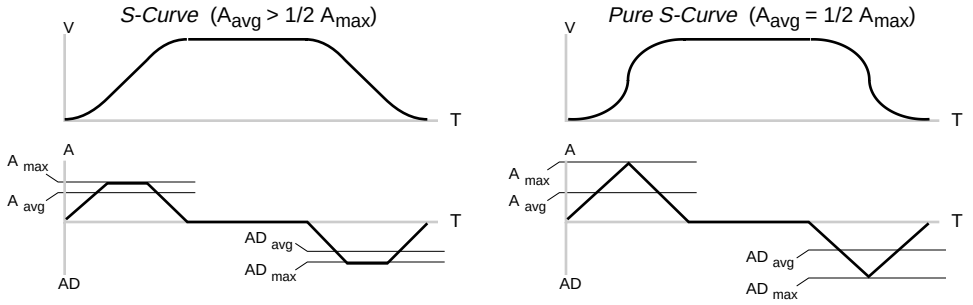
To program an S-curve profile, you must use the *average accel/decel* commands provided in the 6000 Series programming language. For every maximum accel/decel command (e.g., `A`, `AD`, `HOMA`, `HOMAD`, `JOGA`, `JOGAD`, etc.) there is an *average* command for S-curve profiling (see table below).

Maximum Accel/Decel Commands:		Average (S-Curve) Accel/Decel Commands:	
Command	**Function**	**Command**	**Function**
`A`	Acceleration	`AA`	Average Acceleration
`AD`	Deceleration	`ADA`	Average Deceleration
`HOMA`	Home Acceleration	`HOMAA`	Average Home Acceleration
`HOMAD`	Home Deceleration	`HOMADA`	Average Home Deceleration
`JOGA`	Jog Acceleration	`JOGAA`	Average Jog Acceleration
`JOGAD`	Jog Deceleration	`JOGADA`	Average Jog Deceleration
`JOYA`	Joystick Acceleration	`JOYAA`	Average Joystick Acceleration
`JOYAD`	Joystick Deceleration	`JOYADA`	Average Joystick Deceleration
`LHAD`	Hard Limit Deceleration	`LHADA`	Average Hard Limit Deceleration
`LSAD`	Soft Limit Deceleration	`LSADA`	Average Soft Limit Deceleration
`PA`	Path Acceleration	`PAA`	Average Path Acceleration
`PAD`	Path Deceleration	`PADA`	Average Path Deceleration

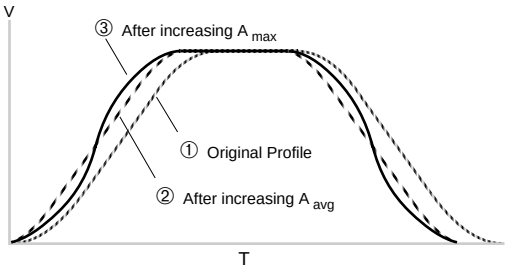
The command values for average accel/decel (`AA`, `ADA`, etc.) and maximum accel/decel (`A`, `AD`, etc.) determine the characteristics of the S-curve. To smooth the accel/decel ramps, you must enter average accel/decel command values that satisfy the equation $1/2\ A_{max} \le A_{avg} < A_{max}$, where A_{max} represents maximum accel/decel and A_{avg} represents average accel/decel. Given this requirement, the following conditions are possible:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- If $A_{avg} > 1/2\ A_{max}$, but $A_{avg} < A_{max}$, you have achieved an S-curve profile with a variable period of constant accel/decel (see drawing below).
- If $A_{avg} = 1/2\ A_{max}$, you have achieved what is called a *Pure S-curve* profile in which there is no period of constant accel/decel and jerk is at an absolute minimum (see drawing below).



- Once you enter an A_{avg} value that is ≠ zero and satisfies $1/2\ A_{max} \leq A_{avg} < A_{max}$, S-curve profiling is enabled, but only in the operation that uses that particular A_{avg} command. For example, entering a HOMAA command enables S-curve acceleration profiling only for homing moves, not for other functions such as jogging (which would require the JOGAA command). To return to the default trapezoidal profiling mode, enter an A_{avg} value of zero, or set $A_{avg} = A_{max}$.
- If $A_{avg} = A_{max}$, a trapezoidal profile results, but can be changed to an S-curve by specifying a new A_{avg} value less than A_{max}, or set A_{max} greater than A_{avg}.
- If $A_{avg} < 1/2\ A_{max}$, or $A_{avg} > A_{max}$, when you try to initiate motion, the move will not be executed and an error message, *INVALID CONDITIONS FOR S_CURVE ACCELERATION–FIELD *n*, will be displayed.
- If A_{avg} = zero or if you never enter an A_{avg} command, the 6270 defaults to trapezoidal profiling and the A_{avg} command value will always match the A_{max} command value. However, if you enter an A_{avg} **deceleration** of zero, you will receive the error message *INVALID DATA–FIELD n, where n is the number of the data field.
- If you never enter the maximum (A_{max}) or average (A_{avg}) decel command values (AD or ADA, HOMAD or HOMADA, etc.), the average decel value will always match, or *track*, the average accel value (AA, HOMAA, etc.). However, once you change the maximum decel, the average decel will no longer track the average accel.
- If you increase the A_{avg} value above the pure S-curve level ($A_{avg} > 1/2\ A_{max}$), the time required to reach the target velocity and the target distance decreases; however, increasing A_{avg} also increases jerk. After increasing A_{avg}, you can reduce the jerk by increasing A_{max} (see illustration); **however, increasing A_{max} requires a greater torque to achieve the commanded velocity at the mid-point of the acceleration profile.**



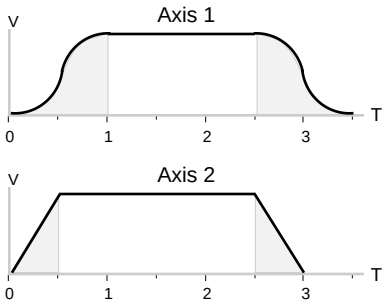
- You can calculate the profile's accel/decel time with the following equations (calculation method is identical for S-curve **and** trapezoidal):

$$\text{Time of accel or decel} = \frac{\text{Velocity}}{A_{avg}} \qquad \text{or} \qquad \text{Time of accel or decel} = \sqrt{\frac{2 * \text{Distance}}{A_{avg}}}$$

- Accel/decel scaling (SCLA or PSCLA) affects A_{avg} the same as it does for A_{max}, regardless of if the profile is trapezoidal or S-curve (see *Scaling* section in Chapter 5).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example	**Example**	**Description**
	> LDTRES432	Set resolution to 432 counts/rev
	> SCALEØ	Disable scaling
	> @MAØ	Select incremental positioning mode
	> @D5ØØØ	Set distances to 5000 counts
	> A1Ø,1Ø	Set max. accel to 10 (both axes)
	> AA5,1Ø	Set avg. accel to 5 on axis 1, and 10 on axis 2
	> AD1Ø,1Ø	Set max. decel to 10 (both axes)
	> ADA5,1Ø	Set avg. decel to 5 on axis 1, and 10 on axis 2
	> V5,5	Set velocity to 5 on both axes
	> GO	Execute motion on both axes



Axis 1 executes a pure S-curve profile that takes 1 second to reach a velocity of 5 and 1 second to return to zero velocity. Axis 2 executes a trapezoidal profile that takes 0.5 seconds to reach a velocity of 5 and 0.5 seconds to return to zero velocity.

X-Y Linear Interpolation

The 6270 allows you to perform *X-Y linear interpolation*, the process of moving two orthogonal (right angle) linear axes to achieve linear (straight line) motion. The task is to derive appropriate move parameters to move from a current location to a new location, where each position is specified by a set of *Cartesian coordinates*. Both axes must start, accelerate, decelerate, and stop in a synchronized manner.

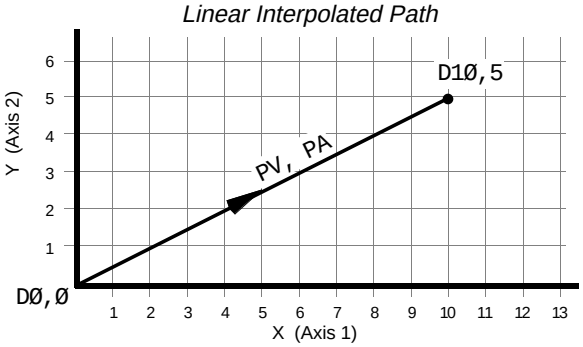
The Initiate Linear Interpolated Motion (GOL) command initiates linear interpolation moves based on the parameters set with the D, PA, PAD, and PV commands. You simply enter the desired path acceleration (PA), the path deceleration (PAD), and the path velocity (PV) to arrive at the point in space (*end point*) specified with the distance (D) command; the 6270 internally calculates each axis' actual move profiles to achieve a straight-line path with these parameters.

You can scale the acceleration, velocity, and distance with the PSCLA, PSCLV, and SCLD commands, respectively (see example below).

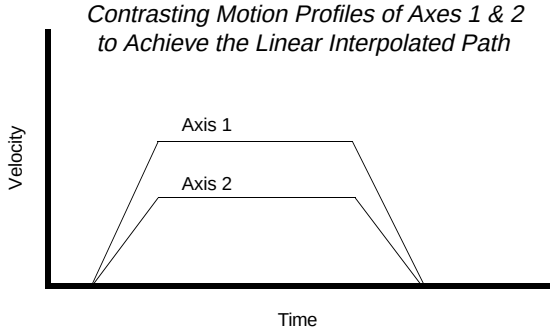
The GOL command starts motion on either or both axes. If the GOL command is issued without any arguments, motion will be started on both axes.

Example	**Command**	**Description**
	> SCALE1	Enable scaling
	> PSCLA432	Set path acceleration scale factor to 432 counts/unit
	> PSCLV432	Set path velocity scale factor to 432 counts/unit
	> @SCLD432	Set distance scale factor to 432 counts/unit on all axes
	> PA25	Set the path acceleration to 25 units/sec^2
	> PAD2Ø	Set the path deceleration to 20 units/sec^2
	> PV2	Set the path velocity to 2 units/sec
	> D1Ø,5	Set the distance to 10 & 5 units on axes 1 & 2, respectively
	> GOL11	Initiate linear interpolated motion on both axes (**see figure below**) (a GOL command could have been issued instead of a GOL11 command)

Linear Interpolated Path



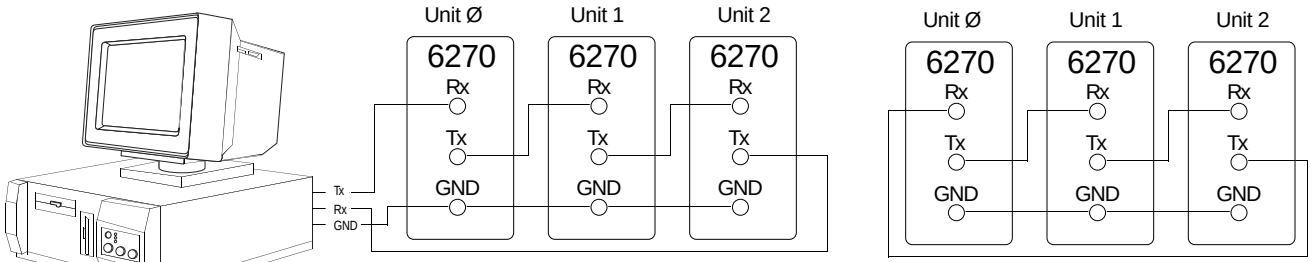
Contrasting Motion Profiles of Axes 1 & 2 to Achieve the Linear Interpolated Path



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RS-232C Daisy-Chaining

Up to ninety-nine 6270s may be daisy-chained. There are two methods of daisy-chaining: one uses a computer or terminal as the controller in the chain; the other uses a 6270 as the master controller. The figure below illustrates examples of both daisy-chain types for three 6270s. *Be sure to use the* **Rx**, **Tx** *and* **GND** *on the* **AUX** *connector, not the* **RP240** *connector.*

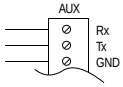


Daisy Chain to a Computer or Terminal

Stand-alone Daisy Chain

Standard 9-Pin COM Port Pin Outs:	Standard 25-Pin COM Port Pin Outs:
Pin 3 = Transmit (Tx) Pin 2 = Receive (Rx) Pin 5 = Ground (GND)	Pin 2 = Transmit (Tx) Pin 3 = Receive (Rx) Pin 7 = Ground (GND)

NOTE: Use Rx, Tx and GND on the 6270's AUX connector, not the RP240 connector.



Follow these steps to implement daisy-chaining:

Step 1 To enable and disable communications on a particular 6270 unit in the chain, you must establish a unique device address using the unit's address DIP switches or the Daisy-chain Address (ADDR) command.

DIP switches: Instructions for accessing and changing these DIP switch settings are provided in the *Optional DIP Switch Settings* section in Chapter 6. Device addresses set with the DIP switches range from 0 to 7.

ADDR command: The ADDR command automatically configures unit addresses for daisy chaining by disregarding the DIP switch setting. This command allows up to 99 units on a daisy chain to be uniquely addressed.

Sending ADDRi to the first unit in the daisy chain sets its address to be (i). The first unit in turn transmits ADDR(i + 1) to the next unit to set its address to (i + 1). This continues down the daisy chain until the last unit of (n) daisy-chained units has its address set to (i + n).

Setting ADDR to Ø re-enables the unit's daisy-chain address configured on its internal DIP switch.

Note that a 6270 with the default device address of zero (0) will send an initial power-up start message similar to the following:

```
*PARKER 627Ø  -  2 AXIS MOTION CONTROLLER
*NO REMOTE PANEL
*ANALOG INPUT OPTION NOT DETECTED
*ADVANCED FOLLOWING FEATURES NOT DETECTED
```

Step 2 Connect the daisy-chain with a terminal as the master (see diagram above).

It is necessary to have the error level set to 1 for all units on the daisy-chain (ERRLVL1). When the error level is not set to 1, the 6270 sends ERROK or ERRBAD prompts after each command, which makes daisy-chaining impossible. Send the ERRLVL1 command from the master terminal (without an address prefix, all units will respond to ERRLVL1).

Command	Description
> ERRLVL1	Set error level to 1

After this has been accomplished a carriage return sent from the terminal will not cause any 6270 to send a prompt. Verify this. Instructions below

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

show how to set the error level to 1 automatically on power-up by using the 6270's power-up start program (highly recommended).

After the error level for all units has been set to ERRLVL1, send a 6000 series command to all units on the daisy-chain by entering that command from the master terminal.

Command	Description
> OUT1111	Turn on outputs #1 - #4 on all units
> A5Ø,5Ø	Set acceleration to 50 for all axes (all units, both axes)

To send a 6000 series command to one particular unit on the daisy-chain, prefix the command with the appropriate unit's device address and an underline:

Command	Description
> 2_OUTØ	Turn off output #1 on unit #2
> 4_OUTØ	Turn off output #1 on unit #4

To receive data from a 6270, you **must** prefix the command with the appropriate unit's device address and an underline:

Command	Description
> 1_A	Request acceleration information from unit #1
> *A5Ø,5Ø	Response from unit #1

Use the (E) command to enable/disable RS-232C communications for an individual unit. If all 6270 units on the daisy chain are enabled, commands without a device address identifier will be executed by all units. Because of the daisy-chain's serial nature, the commands will be executed approximately 1 ms per character later on each successive unit in the chain (assuming 9600 baud).

Units with the RS-232C disabled (EØ) will not respond to any commands, except E1; however, characters are still echoed to the next device in the daisy chain.

Command	Description
> 3_EØ	Disable RS-232C on unit #3
> VAR1=1	Set variable #1 to 1 on all other units
> 3_E1	Enable RS-232C on unit #3
> 3_VAR1=5	Set variable #1 to 5 on unit #3

Verify communication to all units by using the techniques described above.

Step 3 Now that communication is established programming of the units can begin (alternately, units can be programmed individually by connecting the master terminal to one unit at a time). To allow daisy-chaining between multiple 6270s, the ERRLVL1 command must be used to prevent units from sending error messages and command prompts. In every daisy-chained unit the ERRLVL1 command should be placed in the program that is defined as the STARTP program:

Command	Description
> DEF chain	Begin definition of program chain
– ERRLVL1	Set error level to 1
– GOTO main	Go to program main
– END	End definition of program chain
> STARTP chain	Designates program chain as the power-up program

To define program main for unit #0:

Command	Description
> Ø_DEF main	Begin definition of program main on unit #0
– Ø_GO	Start motion
– Ø_END	End definition of program main on unit #0

Step 4 After all programming is completed program execution may be controlled by either a master terminal (diagram above), or by a master 6270 (diagram above).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Daisy-Chaining from a Computer or Terminal

Controlling the daisy-chain from a master computer or terminal follows the examples above:

Command	Description
> Ø_RUN main	Run program main on unit #0
> 1_RUN main	Run program main on unit #1
> 2_GO1	Start motion on unit #2 axis #1
> 3_2A	Get A command response from unit #3 axis #2

Daisy-Chaining from a Master 6270

Controlling the daisy-chain from a master 6270 (the first unit on the daisy-chain) requires stored programs in the master 6270 which can control program and command execution on the slave 6270s. The example below demonstrates the use of the WRITE command to send commands to other units on the daisy-chain.

NOTE
The last unit on the daisy-chain must have RS-232C echo disabled (ECHOØ command).

Master 6270's main program:

Command	Description
> DEF main	Program main
– L	Indefinite loop
– WHILE (IN.1 = bØ)	Wait for input #1 to go active
– NWHILE	
– GOL	Initiate linear interpolated move
– WHILE (IN.1 = b1)	Wait for input #1 to go inactive
– NWHILE	
– WRITE"2_D2ØØØ,4ØØØ"	Send message "2_D2ØØØ,4ØØØ" down the daisy chain
– WRITE"2_ACK"	Send message "2_ACK" down the daisy chain
– LN	End of loop
– END	End of program main

6270 unit #2 ack program:

Command	Description
> DEF ack	Program ack
– GO11	Start motion on both axes
– END	End of program ack

Daisy-Chaining and RP240s

RP240s cannot be placed in the 6270 daisy chain; RP240s can only be connected to the designated RP240 port on a 6270. It is possible to use only one RP240 with a 6270 daisy-chain to input data for multiple units on the chain. The example below (for the 6270 master with an RP240 connected) reads data from the RP240 into variables #1 (*data1*) & #2 (*data2*), then sends the messages 3_Ddata1,data2<CR> and 3_GO<CR>.

Command	Description
> L	Indefinite loop
VAR1=DREAD	Read RP240 data into variable #1
VAR2=DREAD	Read RP240 data into variable #2
EOTØ,Ø,Ø,Ø	Turn off <CR>
WRITE"3_D"	Send message "3_D" down the daisy chain
WRVAR1	Send variable #1 data down the daisy chain
WRITE","	Send message "," down the daisy chain
EOT13,Ø,Ø,Ø	Turn on <CR>
WRVAR2	Send variable #2 data down the daisy chain
WRITE"3_GO"	Send message "3_GO" down the daisy chain
LN	End of loop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑥

Hardware Reference

Use this chapter as a quick-reference tool for 6270 system specifications (general specifications, I/O circuit drawings & pin outs, and DIP switch & jumper settings).

General Specifications

Parameter	Specification
Power	
AC or DC Input	85-240VAC (single-phase), 50/60Hz, 1.8A @ 120VAC; or 110 - 340VDC
Status LED	Refer to the *Common Problems & Solutions* table in Chapter 7
Environmental	
Operating Temperature	32°F to 122°F (0°C to 50°C)
Storage Temperature	-22°F to 185°F (-30°C to 85°C)
Humidity	0% to 95% non-condensing
Performance	
Position Range	±2,147,483,648 counts
Velocity Range	1 to 1,600,000 counts/sec
Acceleration Range	1 to 2,147,483,648 counts/sec^2
Velocity Accuracy	±0.02% of maximum rate
Velocity Repeatability	±0.02% of set rate
Motion Trajectory Update Period	Default is 1.6 ms (depends on SSFR and INDAX command values)
Servo Sampling Update Period	Default is 400 µs (depends on SSFR and INDAX command values)
System Update Period	Default is 1.6 ms (depends on SSFR and INDAX command values)
LDT Position Sampling Period	Default is 1.6 ms (depends on SSFR, INDAX, and LDTUPD values)
RS-232C Interface	
Connections	3-wire (Rx, Tx and GND) connection to the AUX connector (opto iso)
Maximum number of daisy-chained 6270s	Up to 99 units
Address settings	DIP switch Selectable (see *Optional DIP Switch Settings*), or use ADDR
Communication Parameters	9600 baud (auto-baud option—see *Optional DIP Switch Settings*), 8 data bits, 1 stop bit, no parity bit, full duplex
Inputs (see also *I/O Pin Outs & Circuit Drawings*)	**All inputs are optically isolated**
Home, CW/CCW Limits, Joystick Trigger, Joystick Release, Axes Select, Joystick Velocity, Drive Fault (DFT), Enable (ENBL)	TTL-compatible*; internal 6.8 KΩ pull-ups to 5V; voltage range is 0 - 24V.
Linear Displacement Transducer (LDT)	Differential comparator inputs for displacement or Start/Stop pulse. Maximum frequency = 48 MHz.
Incremental Encoder	Differential comparator accepts two-phase quadrature encoders with differential (recommended) or single-ended outputs (+5VDC TTL-compatible*). Maximum frequency = 1.2 MHz. Minimum time between transitions = 833 ns.
24 Programmable	TTL-compatible* with internal 6.8 KΩ pull-up (connect IN-P to +5V to source current or connect IN-P to GND to sink current). Voltage range = 0V - 24V. 50-pin plug is compatible with OPTO-22™ signal conditioning equipment. Controllable with the 6000 Series programming language.
Triggers (triggers [TRG-A and TRG-B] on AUX connector)	TTL-compatible* with internal 6.8 KΩ pull-up to +5VDC. Controllable with the 6000 Series programming language.
Analog (Joystick)	Voltage range = 0 - 2.5VDC, 8-bit A/D converter. **Input voltage must not exceed 5V.**
Analog (±10V optional inputs)	Voltage range = ±10VDC, 14-bit A/D converter (accuracy is 12 bits). **(6270-ANI only)**

* TTL-compatible voltage levels: Low ≤ 0.4V, High ≥ 2.4V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifications Table *(continued)*

Outputs (see also *I/O Pin Outs & Circuit Drawings*)	**All outputs are optically isolated**
26 Programmable (includes OUT-A and OUT-B on AUX connector)	Can be pulled up by connecting OUT-P to +5V on the AUX connector, or to a user-supplied voltage of up to 24V. 50-pin plug is compatible with OPTO-22™ signal conditioning equipment. Controllable with the 6000 Series programming language.
LDT Interrogate (INTR)	TTL-compatible*, open collector output. Max. voltage in OFF state (not sinking current) = 24V, max. current in ON state (sinking) = 30mA.
Command Out (CMD)	12-bit DAC. Load should be > 2KΩ impedance. Output is jumper selectable (see *Optional DIP Switch and Jumper Settings* section below): ±10V (default), ±20mA, ±50mA, ±60mA, ±80mA, ±100mA, or ±150mA.
Shutdown (SHTNO, SHTNC, and COM)	Shutdown relay output. Max. rating: 175VDC, 0.25A, 3W.

* TTL-compatible voltage levels: Low ≤ 0.4V, High ≥ 2.4V

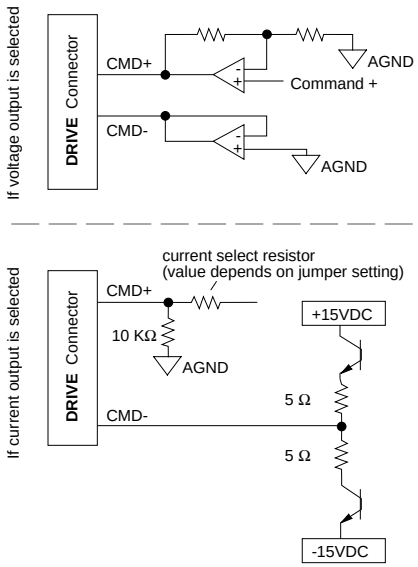
I/O Pin Outs & Circuit Drawings

This section, organized by connector, provides pin outs and circuit drawings for all 6270 inputs and outputs. **All inputs and outputs are optically isolated.**

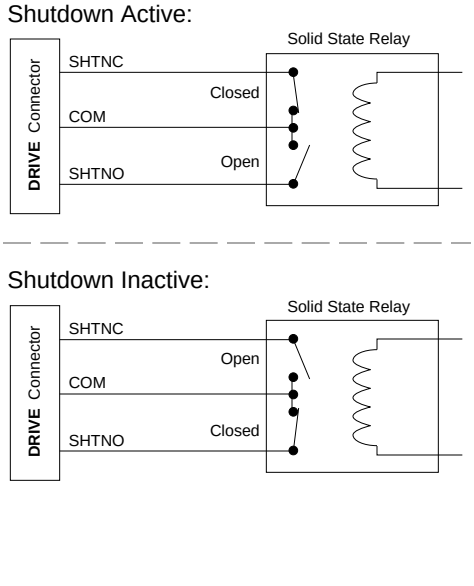
Drive Connectors

Pin #	In/Out	Name	Description
1	-----	SHLD	Shield—internally connect to chassis (earth) ground.
2	-----	COM	Signal common to which the shutdown relay outputs are referenced. Electrically isolated from any ground.
3	OUT	SHTNC	Shutdown relay output to drives that require a closed contact to disable the drive(see circuit drawing below). The shutdown relay is active (disabling the drive) when no power is applied to the 6270. When the 6270 is powered up, the shutdown relay remains active until you issue the `DRIVE11` command. Shutdown active (`DRIVEØØ`): This output is internally connected to COM. Shutdown inactive (`DRIVE11`): This output is disconnected from COM.
4	OUT	SHTNO	Shutdown relay output to drives that require an open contact to disable the drive(see circuit drawing below). The shutdown relay is active (disabling the drive) when no power is applied to the 6270. When the 6270 is powered up, the shutdown relay remains active until you issue the `DRIVE11` command. Shutdown active (`DRIVEØØ`): This output is disconnected from COM. Shutdown inactive (`DRIVE11`): This output is internally connected to COM.
5	IN	DFT	Drive fault input. Set active level with the `DRFLVL` command. The drive fault input will not be recognized until you enable the input functions with the `INFEN1` command. — see circuit diagram below
6	-----	AGND	Analog ground. This is the ground reference for the ANI input.
7	IN	ANI	±10V, 14-bit analog input (12-bit accuracy). ANI Voltage reported with `[ANI]` & `TANI` commands. ANI position (counts) reported with `[PANI]` & `TPANI` commands or `[FB]` & `TFB` commands. **(6270-ANI option only).**
8	OUT	CMD-	Command signal return. — see circuit diagram below
9	OUT	CMD+	Command output signal (jumper selectable: ±10V is default; current settings are available). — see circuit diagrams

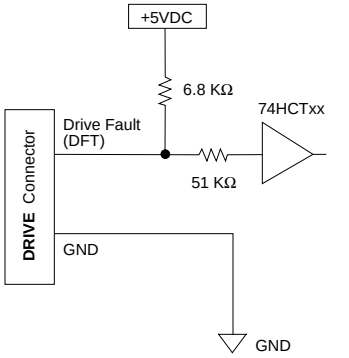
Internal Command Signal Output Circuit



Internal Shutdown Output Circuit



Internal Drive Fault Input Circuit

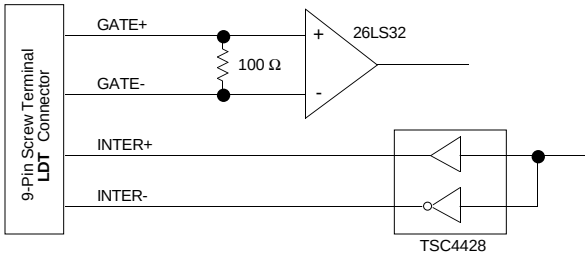


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Linear Displacement Transducer (**LDT**) Connectors

Pin	In/Out	Name	Description
9	-----	+15V	Power connection for LDT
8	OUT	INTR+	Interrogate output to LDT (+) (sampling rate default is 1.5 ms, adjust with LDTUPD command)
7	OUT	INTR-	Interrogate output to LDT (-)
6	IN	GATE+	Return pulse from LDT (+)
5	IN	GATE-	Return pulse from LDT (+)
4	-----	-15V	Power connection for LDT
3	-----	RSVD	Reserved for future enhancements
2	-----	GND	Isolated logic ground
1	-----	SHLD	Internally connected to chassis ground (earth)

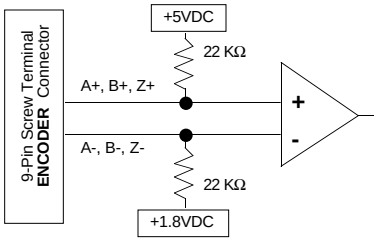
Internal Encoder Input Circuit



Encoder Connector (For Use With Incremental Encoder Only)

Pin	In/Out	Name	Compumotor E Series Encoder Cable Colors	Description
9	OUT	+5V	Red	+5VDC output to power the encoder
8	IN	A +	Brown	A+ channel quadrature signal
7	IN	A -	Brown/White	A- channel quadrature signal
6	IN	B +	Green	B+ channel quadrature signal
5	IN	B -	Green/White	B- channel quadrature signal
4	IN	Z +	Orange	Z+ channel quadrature signal
3	IN	Z -	Orange/White	Z- channel quadrature signal
2	-----	GND	Black	Isolated logic ground
1	-----	SHLD	Shield	Internally connected to chassis ground (earth)

Internal Encoder Input Circuit



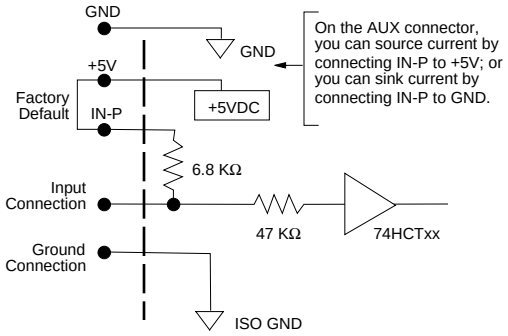
Programmable I/O Connectors

Pin	OUTPUTS Connector	INPUTS Connector
49	+5VDC	+5VDC
47	Output #1 (LSB)	Input #1 (LSB)
45	Output #2	Input #2
43	Output #3	Input #3
41	Output #4	Input #4
39	Output #5	Input #5
37	Output #6	Input #6
35	Output #7	Input #7
33	Output #8	Input #8
31	Output #9	Input #9
29	Output #10	Input #10
27	Output #11	Input #11
25	Output #12	Input #12

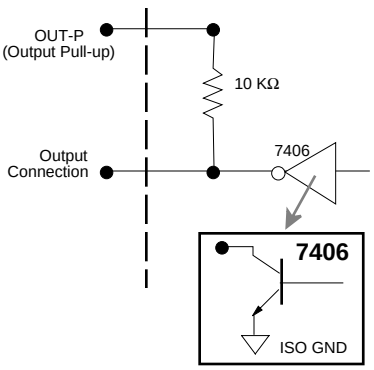
Pin	OUTPUTS Connector	INPUTS Connector
23	Output #13	Input #13
21	Output #14	Input #14
19	Output #15	Input #15
17	Output #16	Input #16
15	Output #17	Input #17
13	Output #18	Input #18
11	Output #19	Input #19
09	Output #20	Input #20
07	Output #21	Input #21
05	Output #22	Input #22
03	Output #23	Input #23
01	Output #24 (MSB)	Input #24 (MSB)

All even-numbered pins are connected to logic ground.

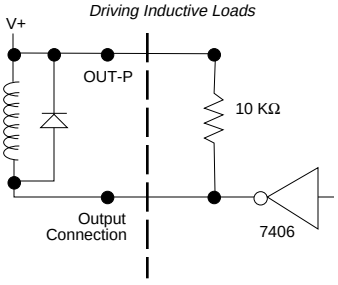
Internal Programmable Input Circuit



Internal Programmable Output Circuit



Driving Inductive Loads



Use an external diode when driving inductive loads. Connect the diode in parallel to the inductive load, attaching the anode to the 6270 output and the cathode to the supply voltage of the inductive load.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

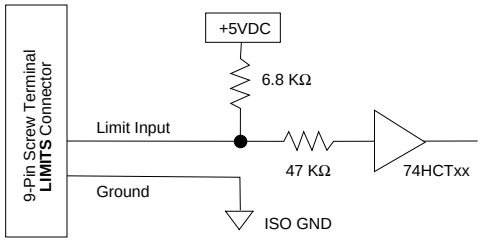
Auxiliary (**AUX**) Connector

Pin	In/Out	Name	Description
1	IN	Rx	Receive input for RS-232C interface
2	OUT	Tx	Transmit output for RS-232C interface
3	------	GND	Isolated ground for RS-232C interface
4	------	SHLD	*Shield*—Internally connected to chassis ground (earth)
5	OUT	+5V	Connect to **OUT-P** to power the 26 programmable outputs. 1.8A limit (applies to total load on all of the I/O connectors) — e.g., if 2 encoders are drawing at total of 500mA, then 1.3A is left for other purposes.
6	OUT	OUT-P	Internally connected to pull-up resistors for the 24 general-purpose programmable outputs. Connecting this input to the +5V pin (this is already done at the factory) makes the outputs TTL compatible. Connection to other voltages (max. = 24V) allows for compatibility with other signal levels.
7	IN	IN-P	Internally connected to pull-up resistors for the 24 general-purpose programmable inputs. Connect this input to the +5V pin (already done at the factory) to source current; or connect to GND to sink current. Connection to other voltages (max. = 24V) allows for compatibility with other signal levels. Does not apply to triggers.
8	IN	TRG-A	Trigger input A: Like programmable inputs, but can function as a position latch input that latches positions within 1 encoder count of the trigger being activated (see `INFNC` command). Internal circuit is identical to the limit input (see limit input circuit drawing below).
9	IN	TRG-B	Same function as trigger input A above
10	------	GND	Isolated ground
11	OUT	OUT-A	Auxiliary programmable output A: Function and circuit is identical to the other 24 programmable outputs (see programmable I/O circuit drawing above)
12	OUT	OUT-B	Same function as auxiliary programmable output A above
13	------	GND	Isolated ground
14	IN	ENBL	Enable input. Normally grounded. When contact to ground is broken, the analog output signal to the drive is set to ØV and the shutdown outputs are activated; this occurs independent of the DSP and the microprocessor. Internal input circuit is identical to the limit circuit below.

Limits Connector

Pin	In/Out	Name	Description
9	IN	1CW	Clockwise limit input for axis 1
8	IN	1CCW	Counter-clockwise limit input for axis 1
7	IN	1HOM	Home limit input for axis 1
6	—	GND	Isolated ground
5	IN	2CW	Clockwise limit input for axis 2
4	IN	2CCW	Counter-clockwise limit input for axis 2
3	IN	2HOM	Home limit input for axis 2
2	—	GND	Isolated ground
1	—	SHLD	*Shield*—Internally connected to chassis ground (earth)

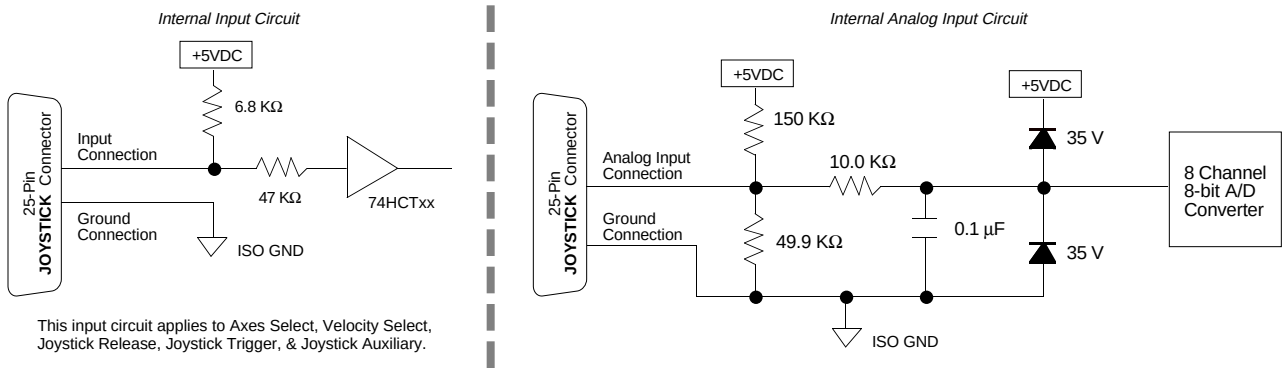
Internal Limit Input Circuit



Joystick Connector

Pin	In/Out	Name	Description
1	IN	Analog Channel 1	8-bit, analog input for joystick control of axis. Input voltage must not exceed 5V.
2	IN	Analog Channel 2	8-bit, analog input for joystick control of axis. Input voltage must not exceed 5V.
3	IN	Analog Channel 3	8-bit, analog input for joystick control of axis. Input voltage must not exceed 5V.
8	—	Shield	Shield
14	—	Ground	Isolated Ground
15	IN	Axes Select	If using only one analog channel, you can use this input to alternately control axes 1 or 2
16	IN	Velocity Select	Input to select high or low velocity range (as defined with `JOYVH` or `JOYVL` command)
17	IN	Joystick Release	When low (grounded), joystick mode can be enabled. When high (not grounded), program execution will continue with the first command after the joystick enable (`JOY`) statement.
18	IN	Joystick Trigger	Status of this active-low input can be read by a program (using the `INO` or `TINO` commands) to control program flow, or to enter the 6270 into joystick mode.
19	IN	Joystick Auxiliary	Status of this active-low input can be read by a program (using the `INO` or `TINO` commands) to control program flow.
23	OUT	+5VDC (out)	+5VDC power output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



RP240 Connector

Pin	In/Out	Name	Description
5	—	Shield	Shield
4	OUT	Tx	Transmit output to RP240's Rx input
3	IN	Rx	Receive input from RP240's Tx output
2	—	Ground	Ground
1	OUT	+5VDC (out)	+5VDC power output

Optional DIP Switch and Jumper Settings

The 6270 is equipped with a four-position DIP switch package you can use to select the 6270's device address (used for daisy-chaining multiple 6270s with one RS-232C circuit), and to use the 6270's auto baud rate feature.

NOTE

As an alternative to setting the address with the DIP switch package, you can automatically establish an *n + 1* unique address configuration with the ADDR command. Refer to the *RS-232C Daisy-Chaining* section in Chapter 5 for details.

The 6270 is also equipped with jumpers you can use to select a variety of command output voltage or current settings.

Accessing the DIP Switch Package and Jumpers

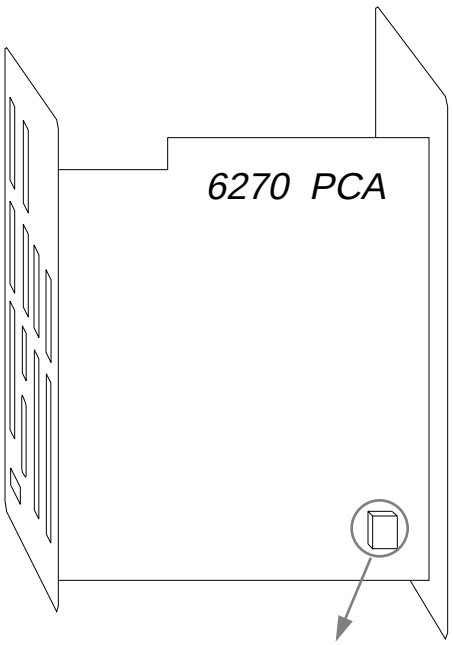
CAUTION

While handling the 6270 printed circuit assembly (PCA), be sure to observe proper grounding techniques to prevent electro-static discharge (ESD).

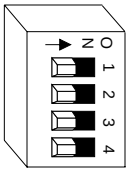
1. Remove power before removing the 6270's enclosure.
2. Lay the 6270 on its back (the side with the mounting brackets).
3. Using a Phillips screwdriver, remove the four screws on the front panel and the two screws on the bottom panel.
4. Gently slide the enclosure away from the chassis.

The illustration below shows the DIP switch package and jumpers and lists the optional settings.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



DIP Switch
Factory Default Setting Shown



Switch #3	Switch #2	Switch #1	Device Address	
OFF	OFF	OFF	Ø	(default)
OFF	OFF	ON	1	
OFF	ON	OFF	2	
OFF	ON	ON	3	
ON	OFF	OFF	4	
ON	OFF	ON	5	
ON	ON	OFF	6	
ON	ON	ON	7	

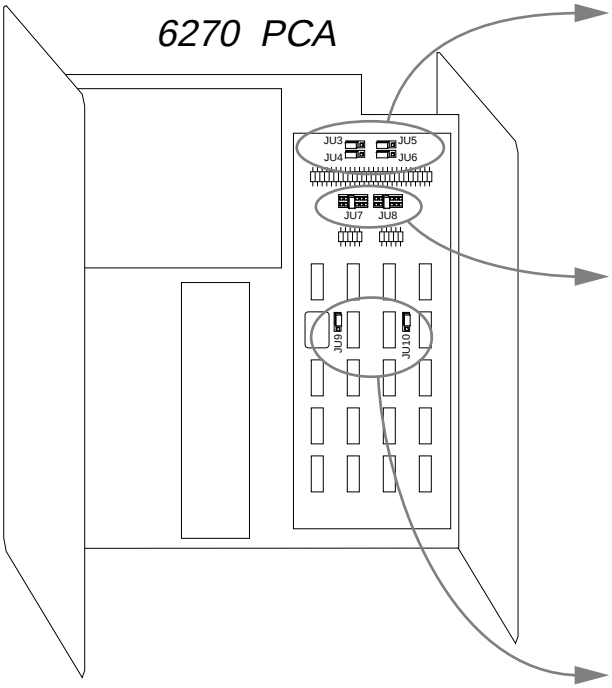
* Device address is checked upon power up or reset.

Switch #4 ON = Auto Baud Enabled
Switch #4 OFF = Auto Baud Disabled (default)

Following these steps to implement the Auto Baud feature:

1. Change Switch #4 to the ON position.
2. Connect the terminal to the 6270's RS-232C serial port on the AUX connector.
3. Power up the terminal.
4. Cycle power to the 6270 and immediately press the space bar several times.
5. The 6270 should send a message to the terminal confirming the baud rate (i.e., `*9600`). You should now be communicating to the 6270 via the terminal. If no baud rate messages is received, verify steps 1 through 3 and repeat step 4.
6. Change Switch #4 to the OFF position.
7. Cycle power to the 6270. You should be communicating to the 6270 via the terminal at the previously determined rate.

NOTE: If Auto Baud is enabled, the 6270 performs its auto baud routine every time it is powered up or reset. The 6270 is capable of matching 1200, 2400, 4800, and 9600 baud. Once the baud rate has been determined, the 6270 stores that baud rate in non-volatile memory; therefore, Switch #4 should be set to the OFF position after the baud rate has been determined.



CURRENT OR VOLTAGE CONTROL

Note: Two jumper settings are required for each axis

JUMPER	FUNCTION	OPTIONS
JU5 and JU6	Axis #1	Position 1/2 = ±10 volt output Position 2/3 = current output (select range with JU8)
JU3 and JU4	Axis #2	Position 1/2 = ±10 volt output Position 2/3 = current output (select range with JU7)

CURRENT RANGE OPTIONS

JUMPER	FUNCTION	OPTIONS
JU8	Axis #1	Position 1/2 = ± 20 mA Position 3/4 = ± 50 mA Position 5/6 = ± 60 mA Position 7/8 = ± 80 mA Position 9/10 = ± 100 mA Position 11/12 = ± 150 mA
JU7	Axis #2	Position 1/2 = ± 20 mA Position 3/4 = ± 50 mA Position 5/6 = ± 60 mA Position 7/8 = ± 80 mA Position 9/10 = ± 100 mA Position 11/12 = ± 150 mA

LDT FEEDBACK OPTIONS

JUMPER	FUNCTION	OPTIONS
JU9	Axis #1	Position 1/2 = pulse duration feedback Position 2/3 = start/stop feedback
JU10	Axis #2	Position 1/2 = pulse duration feedback Position 2/3 = start/stop feedback

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑦

Troubleshooting

The information in this chapter will enable you to isolate and resolve system hardware and software problems.

Troubleshooting Basics

If your system does not function properly (or as you expect it to operate), the first thing that you must do is identify and isolate the problem. When you have accomplished this, you can effectively begin to resolve the problem.

The first step is to isolate each system component and ensure that each component functions properly when it is run independently. You may have to dismantle your system and put it back together piece by piece to detect the problem. If you have additional units available, you may want to exchange them with existing components in your system to help identify the source of the problem.

Determine if the problem is mechanical, electrical, or software-related. Can you repeat or re-create the problem? Do not attempt to make quick rationalizations about problems. Random events may appear to be related, but they are not necessarily contributing factors to your problem. You must carefully investigate and decipher the events that occur before the subsequent system problem.

You may be experiencing more than one problem. You must isolate and solve one problem at a time. Log (document) all testing and problem isolation procedures. You may need to review and consult these notes later. This will also prevent you from duplicating your testing efforts.

If you are having difficulty isolating a problem be sure to document all occurrences of the problem along with as much specific information, such as time of occurrence, 6270 status, and anything else that was happening when the problem occurred.

Once you have isolated a problem, take the necessary steps to resolve it. Refer to the problem solutions contained in this chapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Reducing Electrical Noise

For detailed information on reducing electrical noise, refer to Appendix A.

Error Messages and Debug Tools

A list of all possible error messages, and their causes, is provided in the ***6000 Series Software Reference Guide***. For instructions on using the 6270's program debug tools (Trace mode, Single-Step mode, I/O activation, bad command detection, etc.) refer to *Program Debug Tools* in the *Programming Guide* section of the ***6000 Series Software Reference Guide***.

Common Problems & Solutions

The following table presents some guidelines to help you isolate problems with your motion control system. Some common symptoms are listed along with a list of possible causes and remedies.

- Look for the symptom that most closely resembles what you are experiencing.
- Look through the list of possible causes so that you better understand what may be preventing proper operation.
- Start from the top of the list of remedies and use the suggested procedures to isolate the problem.
- Refer to other sections of the manual for more information on 6270 set up, system connections, and feature implementation. You may also need to refer to the ***6000 Series Software Reference Guide***.

Problem	Cause	Solution
Erratic operation	1. Electrical Noise 2. Improper shielding 3. Improper wiring	1. Reduce electrical noise or move the 6270 away from noise source (refer also to Appendix A) 2. Ground Joystick Release input (refer also to Appendix A) 3. Check wiring for opens, shorts, and mis-wired connections
LEDs: DISABLED1/2 LEDs are red	1. Shutdown input active 2. No AC power to drive or valve 3. Drive or valve not connected 4. Position error (LDT not connected) 5. LDT read error (LDT not connected, LDT failure, or LDTUPD setting too fast)	1. Issue DRIVE11 command 2. Check AC power to drive or valve 3. Connect drive or valve 4. Connect LDT and issue DRIVE11 command 5.a. Connect LDT and issue DRIVE11 command 5.b. Replace faulty LDT and issue DRIVE11 command 5.c. Increase the LDTUPD value and issue DRIVE11 command NOTE: To enable an axis (DRIVE11) without an LDT connected, connect **GATE+** to **GND** on the **LDT** connector.
LEDs: STATUS LED is off	1. No AC power	1. Check AC power
LEDs: STATUS LED is red	1. Internal Board Monitor Alarm (BMA) has detected a non-recoverable fault	1.a. Recycle power to the 6270 1.b. Ensure +5V is not shorted to GND on the I/O connections
Missing counts from feedback device	1. Improper wiring 2. Feedback device slipping 3. Feedback device too hot 4. Electrical noise 5. Feedback device frequency too high 6. LDT read error (LDT not connected, LDT failure, or LDTUPD setting too fast)	1. Check wiring 2. Check and tighten feedback device coupling 3. Reduce encoder temperature with heatsink, thermal insulator, etc. 4.a. Shield wiring (refer also to Appendix A) 4.b. Use encoder with differential outputs 5. Peak encoder frequency must be below 1.2 MHz post-quadrature; peak frequency must account for velocity ripple 6.a. Connect LDT and issue DRIVE11 command 6.b. Replace faulty LDT and issue DRIVE11 command 6.c. Increase the LDTUPD value and issue DRIVE11 command NOTE: To enable an axis (DRIVE11) without an LDT connected, connect **GATE+** to **GND** on the **LDT** connector.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Problems, Causes & Solutions *(continued)*

Problem	Cause	Solution
Load does not move in joystick mode	1. Joystick Release input not grounded 2. Improper wiring	1. Ground Joystick Release input 2. Check wiring for opens, shorts, and mis-wired connections
Runaway (if encoder or LDT counts positive when turned clockwise or extended)	1. Direction connections reversed	1. Switch CMD– with the CMD+ connection to drive or valve NOTE: The CMD+/– Connection is not differential. Do not connect CMD+ to ground on your valve/drive
No Motion	1. Status LED off or red 2. Limits engaged 3. Drive fault level incorrect. 4. Improper wiring 5. Load is jammed 6. No torque from motor 7. Maximum position error exceeded 8. Drive has activated the Drive fault input 9. ENBL input is not grounded to GND	1. See status LED problems above. 2a. Move load off of limits or disable limits with `LHØ,Ø` 2b. If using soft limits, make sure `LSCW>LSCCW` 3. Set drive fault level using `DRFLVLxx` (for S, Z, and K drives, use `DRFLVL11`) 4. Check command, shutdown, drive fault, and limit connections. 5. Remove power and clear jam 6. See problem: *No Torque* 7. Check to see if `TAS` bit #23 is set, and issue the `DRIVE1` command to the axis that exceeded the position error limit. 8a. Check to see if `TAS` bit #14 is set, and check the `DRFLVL` command to ensure the drive fault level is correct. 8b. Inspect the drive to determine the cause 9. Ground the ENBL input to GND and reset
No RS-232C Communication	1. Improper RS-232C Interface or communication parameters 2. RS-232C disabled 3. In daisy chain, unit may not be set to proper address	1. See *RS-232C Troubleshooting* section 2. Enable RS-232C with the `E` command (all units if daisy-chained) 3. Verify DIP switch settings (see *Optional DIP Switch Settings* in Chapter 6), verify proper application of the `ADDR` command
No Torque/Force	1. Improper wiring 2. No power to drive 3. Drive or valve failed 4. Drive faulted 5. Shutdown issued	1. Check wiring to the drive, or valve, as well as other system wiring 2. Check power to drive 3. Check drive or valve status 4. Check drive status 5. Enable drive or valve with `DRIVE11`
Power-up Program does not execute	1. ENBL input is not grounded to GND 2. `STARTP` program is not defined	1. Ground the ENBL input to GND and reset 2. Check the response to the `STARTP` command. If no program is reported, define the `STARTP` program and reset
Program access denied: receive the message `*ACCESS DENIED` when trying to use the `DEF`, `DEL`, `ERASE`, `INFNC`, or `MEMORY` commands	1. Program security function has been enabled (`INFNCi-Q`) and the program access input has not been activated	1.a. Activate the assigned program access input, perform your programming changes, then deactivate the program access input. 1.b. Refer to the instructions in the `INFNC` command description in the ***6000 Series Software Reference Guide***.
Program execution: stops at the `INFEN1` command	1. `INFEN1` enables drive fault monitoring, but the drive fault level (`DRFLVL`) command is set incorrectly for the drive being used.	1. Issue the correct `DRFLVL` command for your drive (refer to the `DRFLVL` command)
Program execution: the first time a program is run, the move distances are incorrect. Upon downloading the program the second time, move distances are correct.	1. Scaling parameters were not issued when the program was downloaded; or scaling parameters have been changed since the program was defined	1. Issue the scaling parameters (`SCALE1`, `SCLA`, `SCLD`, `SCLV`, `PSCLA`, `PSCLV`) before saving any programs
Programmable inputs not working	1. IN-P (input pullup) not connected 2. If external power supply is used, the grounds must be connected together 3. Improper wiring	1.a. When inputs will be pulled down to 0V by an external device, connect IN-P to +5V or to another positive supply 1.b. When inputs will be pulled up to 5V or higher by an external device, connect IN-P to 0V 2. Connect external power supply's ground to ground (GND) 3. Check wiring for opens, shorts, and mis-wired connections
Programmable outputs not working	1. Output connected such that it must source current (pull to positive voltage) 2. OUT-P not connected to +5V or other positive voltage source 3. If external power supply is used, the grounds must be connected together 4. Improper wiring	1. Outputs are open-collector and can only sink current -- change wiring. 2. Connect OUT-P to +5V supplied or other voltage in system 3. Connect the external power supply's ground to ground (GND) 4. Check wiring for opens, shorts, and mis-wired connections

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Problems, Causes & Solutions *(continued)*

Problem	Cause	Solution
Trigger inputs not working	1. Improper wiring	1. Check wiring for opens, shorts, and mis-wired connections
Wrong Direction—Stable	1. Command output (CMD) connections and feedback device connections or mounting are reversed	1. Software remedy: Issue the `CMDDIR1` command to the affected axis. This reverses the polarity of the commanded direction and the feedback direction so that servo stability is maintained. Hardware remedy: Switch CMD- with the CMD+ connection to drive or valve (if your valve or drive does not accept differential outputs this will not work). You will also have to change the feedback device wiring or mounting so that it counts in same direction as the commanded direction.
Wrong Direction—Unstable	1. Not tuned properly 2. Phase of encoder reversed or mounting of ANI input or LDT is such that it counts in the opposite direction as the commanded direction.	1. Refer to Chapter 4 for tuning instructions 2. Software remedy: For the affected axis, issue the appropriate feedback polarity reversal command (`LDTPOL1` if LDT, `ANIPOL1` if ANI analog input, or `ENCPOL1` if encoder). Hardware remedy: If encoder feedback, swap the PHA+ and PHA- connections to the 6270. If LDT or ANI feedback, change the mounting so that the counting direction is reversed.
Wrong Speed or Distance	1. Wrong resolution setting 2. Wrong scaling value	1.a. Encoder feedback: Check and set resolution on 6270 with `ERESx,x` 1.b. LDT feedback: Check and set resolution on 6270 with `LDTRESx,x` 2. Check the scaling parameters (`SCALE1`, `SCLA`, `SCLD`, `SCLV`, `PSCLA`, `PSCLV`)

RS-232C Troubleshooting

If you are having problems communicating with the 6270, try the following procedure to troubleshoot the communications interface.

1. Power-up your computer or terminal *and then* power-up the 6270.
2. The serial port of your computer/terminal may require hardware handshaking. If so, you must disable handshaking with your terminal emulator software package. You can also disable hardware handshaking by connecting the computer's/terminal's RTS & CTS lines together (usually pins 4 and 5) and DSR & DTR lines together (usually pins 6 to 20).
3. Verify that the computer/terminal and 6270 are configured to the same baud rate, number of data bits, number of stop bits, and parity. If your terminal is not capable of 9600 baud, you can use the 6270's *auto-baud* function to automatically set the 6270's baud rate equal to the terminal's baud rate. Refer to the *Optional DIP Switch and Jumper Settings* section in Chapter 6 for instructions.
4. Check to make sure you are using DC common or signal ground as your reference, *not* earth ground.
5. Cable lengths for RS-232C should not exceed 50 feet. As with any control signal, be sure to shield the cable to earth ground at one end only.
6. Press the return key several times. The cursor should move down one or two lines each time you press the return key. If your terminal displays garbled characters, check the terminal's protocol set-up; the baud rate setting probably does not match the 6270's setting (see step ③ above). The problem could also be caused by a poor ground connection.
7. If the cursor does not move after pressing the space bar:
   a. Disconnect the RS-232C cable from the 6270.
   b. Connect the RS-232C cable's Rx and Tx lines together at the end that connects to the 6270.
   c. Press the space bar. If the cursor does not move, either the computer (or terminal) or the cable is defective.
8. Once you are able to make the cursor move, enter some characters. These characters should appear on the computer or terminal display. If each character appears twice, your host is set to half-duplex; set it to full-duplex.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A: Reducing Electrical Noise

Noise-related difficulties can range in severity from minor positioning errors to damaged equipment from runaway loads crashing blindly through limit switches. In microprocessor-controlled equipment such as the 6270, the processor constantly retrieves instructions from memory in a controlled sequence. If an electrical disturbance occurs, it may cause the processor to misinterpret an instruction or access the wrong data. This can be catastrophic to the program and force you to reset the processor.

Sources of Noise

Being invisible, electrical noise can be very mysterious, but it invariably comes from the following sources:

- Power line noise
- Externally conducted noise
- Transmitted noise
- Ground loops

The following electrical devices are notorious for generating unwanted electrical noise conditions:

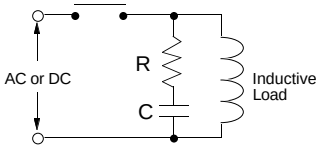
- Coil-driven devices: conducted and power line noise
- SCR-fired heaters: transmitted and power line noise
- Valves, motors & motor drives: transmitted and power line noise
- Welders (electric): transmitted and power line noise

Power Line Noise

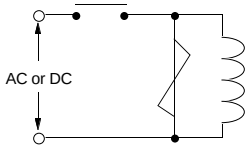
Power line noise is usually easy to resolve due to the wide availability of line filtering equipment for the industry. Only the most severe situations call for an isolation transformer. Line filtering equipment is required when other devices connected to the local power line are switching large amounts of current, especially if the switching occurs at a high frequency.

Any device having coils is likely to disrupt the power line when it is switched off. Surge suppressers, such as metal oxide varistors (MOVs) are capable of limiting this type of electrical noise. A series resistor/capacitor (RC) network across the coil is also effective (resistance: 500 to 1,000 Ω; capacitance: 0.1 to 0.2 μF). Coil-driven devices (inductive loads) include relays, solenoids, contractors, clutches, brakes, and motor starters.

Typical RC Network



MOV

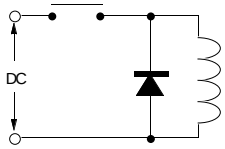


Externally Conducted Noise

Externally-conducted noise is similar to power line noise, but the disturbances are created on signal and ground wires that are connected to the 6270. This kind of noise can get into logic circuit ground or into the processor power supply and scramble the program. The problem here is that control equipment often shares a common DC ground wire that may be connected to several devices, such as a DC power supply, programmable controller, remote switches, etc. When a noisy device such as a relay or solenoid is attached to the DC ground, it may cause disturbances within the 6270.

To solve a noise problem caused by DC mechanical relays and solenoids, you can connect a diode backwards across the coil to clamp the induced voltage *kick* that the coil will produce. The diode should be rated at 4 times the coil voltage and 10 times the coil current. Using solid state relays is another way to eliminate this problem.

Diode



Multiple devices on the same circuit should be grounded together at a single point.

Furthermore, power supplies and programmable controllers often have DC common tied to Earth (AC power ground). As a rule, it is preferable to have the 6270 signal ground or DC common floating with respect to Earth. This prevents noisy equipment which is grounded to Earth from sending noise into the 6270. When floating the signal ground is not possible, you should make the Earth ground connection at only one point.

In many cases, optical isolation may be required to completely eliminate electrical contact between the 6270 and a noisy environment. Solid state relays provide this type of isolation.

Transmitted Noise

Transmitted noise is picked up by external connections to the 6270, and in severe cases can attack the 6270 when there are no external connections. The 6270's sheet metal enclosure will typically shield the electronics from this, but openings in the enclosure for connections and front panel controls may *leak*.

When high current contacts open, they draw an arc, producing a burst of broad spectrum radio frequency noise that can be picked up on a limit switch or other wiring. High-current and high-voltage wires have an electrical field around them and may induce noise on signal wiring, especially when they are tied in the same wiring bundle or conduit.

When this kind of problem occurs, you should consider shielding signal cables or isolating the signals. A proper shield surrounds the signal wires to intercept electrical fields, but this shield must be tied to Earth to drain the induced voltages. At the very least, wires should be run in twisted pairs to limit straight line antenna effects.

Installing the 6270 in a NEMA enclosure ensures protection from this kind of noise, unless noise-producing equipment is also mounted inside the enclosure. Connections external to the enclosure must be shielded.

Even the worst noise problems in environments near 600 amp welders and 25kW transmitters have been solved using enclosures, conduit, optical isolation, and single-point ground techniques.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Ground Loops

Ground Loops are the most mysterious noise problems. They seem to occur most often in systems where a control computer is using RS-232C communication. Symptoms like garbled transmissions and intermittent operation are typical.

The problem occurs in systems where multiple Earth ground connections exist, particularly when these connections are far apart.

Ground Loops—Noise Scenario

Suppose a 6270 is controlling an axis, and the limit switches use an external power supply. The 6270 is controlled by a computer in another room. If the power supply Common is connected to Earth, the potential exists for ground loop problems. This is because most computers have their RS-232C signal common tied to Earth. The loop starts at the 6270 system limit switch ground, goes to Earth through the drive, and on to Earth at the computer. From there, the loop returns to the 6270 system through RS-232C signal ground. If a voltage potential exists between drive Earth and remote computer Earth, ground current will flow through the RS-232C ground, creating unpredictable results.

The way to test for and ultimately eliminate a ground loop is to lift or *cheat* Earth ground connections in the system until the symptoms disappear.

Defeating Noise

The best time to handle electrical noise problems is before they occur. When a motion system is in the design process, the designer should consider the following set of guidelines for system wiring (in order of importance):

1. Put surge suppression components on all electrical coils: Resistor/capacitor filters, MOVs, Zener and clamping diodes.
2. Shield all remote connections, use twisted pairs. Shields should be tied to Earth at one end.
3. Put all microelectronic components in an enclosure. Keep noisy devices outside. Watch internal temperature.
4. Ground signal common wiring at one point. Float this ground from Earth if possible.
5. Tie all mechanical grounds to Earth at one point. Run chassis and motor grounds to the frame, and the frame to Earth.
6. Isolate remote signals. Solid state relays or opto isolators are recommended.
7. Filter the power line. Use common RF filters, and use an isolation transformer for worst case.

A noise problem must be identified before it can be solved. The obvious way to approach a problem situation is to eliminate potential noise sources until the symptoms disappear, as in the case of ground loops. When this is not practical, use the above guidelines to *shotgun* the installation.

References

Information about the equipment referred to may be obtained by calling the numbers listed below.

- Corcom line filters, (214) 386-5515
- Opto-22 optically isolated relays, (408) 496-6611
- Crydom optically isolated relays, (415) 463-2250
- Potter Brumfield optically isolated relays, (812) 386-1000
- Teal power line isolation filters, (800) 888-8325

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix B: Alphabetical Command List

Command Name	Command Description
[<cr>]	Carriage Return
[<lf>]	Line Feed
[:]	Colon
!	Immediate Command Identifier
@	Global Command Identifier
;	Begin Comment
$	Label Deceleration
#	Step Through a Program
'	Enter Data (Single quote)
[.]	Bit Select
["]	Begin and End String
[\ ]	ASCII Character Designator
[=]	Assignment or Equivalence
[>]	Greater Than
[>=]	Greater Than or Equal
[<]	Less Than
[<=]	Less Than or Equal
[<>]	Not Equal
[( )]	Operation Priority Level
[+]	Addition
[-]	Subtraction
[*]	Multiplication
[/]	Division
[&]	Boolean And
[|]	Boolean Or
[^]	Boolean Exclusive Or
[~( )]	Boolean Not
[>>]	Shift from Right to Left
[<<]	Shift from Left to Right
A	Acceleration
[A]	Acceleration Assignment
AA	Average Acceleration for S-curve
AD	Deceleration
[AD]	Deceleration Assignment
ADA	Average Deceleration for S-curve
ADDR	Daisy-Chain Address
[AND]	And
[ANI]	Analog Input Value (for -ANI Option)
ANIPOL	ANI Input Polarity
[ANV]	Analog Input Value
ANVO	Analog Input Voltage Override
ANVOEN	Analog Input Voltage Override Enable
[AS]	Axis Status Value
[ATAN( )]	Inverse Tangent
[b]	Binary Identifier
BP	Set a Program Break Point
BREAK	Terminate Program Execution
C	Continue
[CA]	Value of Captured ANI Input
CMDDIR	Commanded Direction Polarity
COMEXC	Enable Continuous Command Mode
COMEXL	Continue Command Execution on Limit
COMEXR	Continue Motion on Pause/Resume Input
COMEXS	Continue Command Execution on Stop
[COS( )]	Cosine

Command Name	Command Description
D	Distance
[D]	Distance Assignment
[DAC]	Current DAC Commanded Voltage
DACLIM	DAC Limit
DACMIN	Minimum DAC Output Voltage
[DAT]	Data Assignment
DATA	Data Statement
[DATP]	Data Program
DATPTR	Set Data Pointer
DATRST	Reset Data Pointer
DATSIZ	Data Program Size
DATTCH	Data Teach
DCLEAR	Clear RP240 Display
DEF	Define a Program/Subroutine
DEL	Delete a Program/Subroutine
DJOG	Enable RP240 Jog Mode
DLED	Turn RP240 LEDs On/Off
DPASS	Set RP240 Password
DPCUR	Position Cursor on RP240 Display
[DPTR]	Data Pointer Assignments
[DREAD]	Read Numeric Keypad on RP240
[DREADF]	Read Function Key on RP240
DREADI	RP240 Data Read Immediate Mode
DRFLVL	Drive Fault Level
DRIVE	Drive Shutdown
DVAR	Display Variable on RP240 Display
DWRITE" "	Write Text to the RP240 Display
E	RS-232C Enable
ECHO	Echo Enable
ELSE	Else Condition of IF Statement
ENCPOL	Encoder Polarity
END	Program/Subroutine End
EOL	End of Line Terminating Characters
EOT	End of Transmission Characters
[ER]	Error Value
ERASE	Erase all Programs/Subroutines
ERES	Encoder Resolution
ERRBAD	Bad Prompt
ERRDEF	Program Definition Prompt
ERRLVL	Error Detection Level
ERROK	Good Prompt
ERROR	Error Program Enable
ERRORP	Error Program
[FB]	Position of Feedback Device
GO	Initiate Motion
GOL	Initiate Linear Interpolated Motion
GOSUB	Execute a Subroutine with Return
GOTO	Execute a Subroutine without Return
[h]	Hexadecimal Identifier
HALT	Terminate Program Execution
HELP	Applications Help
HOM	Go Home
HOMA	Home Acceleration
HOMAA	Average Homing Acceleration
HOMAD	Home Deceleration
HOMADA	Average Homing Deceleration
HOMBAC	Home Backup Enable
HOMDF	Home Direction Final
HOMEDG	Home Reference Edge

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Name	Command Description
HOMLVL	Home Active Level
HOMV	Home Velocity
HOMVF	Home Velocity Final
HOMZ	Home to Z-channel Enable
IF( )	If Statement
[IN]	Input Status
INDAX	Participating Axes
INDEB	Input Debounce Time
INDUSE	Enable/Disable User Status
INDUST	User Status
INEN	Input Enable
INFEN	Input Function Enable/Disable
INFNC	Input Function
INLVL	Input Active Level
[INO]	Other Input Status
INPLC	Establish PLC Data Inputs
INSELP	Select Program Enable
INSTW	Establish Thumbwheel Data Inputs
JOG	Jog Mode Enable
JOGA	Jog Acceleration
JOGAA	Average Jogging Acceleration
JOGAD	Jog Deceleration
JOGADA	Average Jogging Deceleration
JOGVH	Jog Velocity High
JOGVL	Jog Velocity Low
JOY	Joystick Mode Enable
JOYA	Joystick Acceleration
JOYAA	Average Joystick Acceleration
JOYAD	Joystick Deceleration
JOYADA	Average Joystick Deceleration
JOYAXH	Joystick Analog Input High
JOYAXL	Joystick Analog Input Low
JOYCDB	Joystick Center Deadband
JOYCTR	Joystick Center
JOYEDB	Joystick End Deadband
JOYVH	Joystick Velocity High
JOYVL	Joystick Velocity Low
JOYZ	Joystick Zero
JUMP	Jump to a Subroutine without Return
K	Kill Motion
<ctrl>K	Immediate Kill
KDRIVE	Disable Drive on Kill
L	Loop
[LDT]	Position of LDT
LDTGRD	LDT Gradient
LDTPOL	LDT Polarity
LDTRES	LDT Resolution
LDTUPD	LDT Position Sampling Update Rate
LH	Hard Limit Enable
LHAD	Hard Limit Deceleration
LHADA	Average Hard Limit Deceleration
LHLVL	Hard Limit Active Level
[LIM]	Limit Status
LN	End Loop
LS	Soft Limit Enable
LSAD	Soft Limit Deceleration
LSADA	Average Soft Limit Deceleration
LSCCW	Soft Limit CCW Range
LSCW	Soft Limit CW Range
LX	Terminate Loop

Command Name	Command Description
MA	Absolute / Incremental Mode Enable
MC	Preset / Continuous Mode Enable
MEMORY	Configure Memory
[MOV]	Axis Moving Status
NIF	End IF Statement
[NOT]	Not
NWHILE	End WHILE Statement
ONCOND	On Condition Enable
ONIN	On an Input Condition Gosub
ONP	On Program
ONUS	On a User Status Condition Gosub
ONVARA	On Variable 1 Condition Gosub
ONVARB	On Variable 2 Condition Gosub
[OR]	Or
OUT	Output State
[OUT]	Output Status
OUTALL	Multiple Output State
OUTEN	Output Enable
OUTFEN	Output Function Enable
OUTFNC	Output Function
OUTLVL	Output Active Level
OUTPA	Output on Position — Axis 1
OUTPB	Output on Position — Axis 2
OUTPLC	Establish PLC Strobe Data Outputs
OUTTW	Establish Thumbwheel Strobe Data Outputs
PA	Path Acceleration
PAA	Average Path Acceleration for S-curve
PAD	Path Deceleration
PADA	Average Path Deceleration for S-curve
[PANI]	Position of ANI Inputs (6270-ANI Only)
[PC]	Commanded Position
[PCA]	Position of Captured ANI Input Voltage
[PCC]	Captured Commanded Position
[PCE]	Position of Captured Encoder
[PCL]	Position of Captured LDT
[PE]	Position of Encoder
[PER]	Position Error
[PI]	Pi (π)
PS	Pause Program Execution
PSCLA	Path Acceleration Scale Factor
PSCLV	Path Velocity Scale Factor
PSET	Establish Absolute Position
PV	Path Velocity
RADIAN	Radian Enable
[READ]	Read a Value from RS-232C port
REPEAT	Repeat Statement
RESET	Reset 6270
RUN	Execute a Program/Subroutine
S	Stop Motion
SCALE	Enable/Disable Scale Factors
SCLA	Accel / Decel Scale Factor
SCLD	Distance Scale Factor
SCLV	Velocity Scale Factor
SDTAMP	Servo Dither Amplitude
SDTFR	Servo Dither Frequency Ratio
SFB	Servo Feedback Source
SGAF	Servo Acceleration Feedforward Gain
SGAFN	Acceleration Feedforward — Negative
SGENB	Servo Gain Set Enable
SGI	Servo Integral Feedback Gain
SGIN	Integral Feedback Gain — Negative

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Name	Command Description
SGILIM	Servo Integral Windup Limit
SGP	Servo Proportional Feedback Gain
SGPN	Proportional Feedback Gain — Negative
SGSET	Servo Gain Set Save
SGV	Servo Velocity Feedback Gain
SGVN	Velocity Feedback Gain — Negative
SGVF	Servo Velocity Feedforward Gain
SGVFN	Velocity Feedforward Gain — Negative
[SIN( )]	Sine
SMPER	Servo Max. Allowable Position Error
SOFFS	Servo Control Signal Offset
SOFFSN	Control Signal Offset — Negative
[SQRT( )]	Square Root
[SS]	System Status
SSFR	Servo Sampling Frequency Ratio
SSWD	Setpoint Window Distance
SSWG	Setpoint Window Gain Set
STARTP	Set Power-up Program
STEP	Program Step Mode Enable
STRGTD	Servo Target Zone Distance
STRGTE	Servo Target Zone Mode Enable
STRGTT	Servo Target Zone Timeout Period
STRGTV	Servo Target Zone Velocity
T	Time Delay
[TAN( )]	Tangent
TANI	Transfer Analog Input Value (6270-ANI only)
TANV	Transfer Analog Input Value
TAS	Transfer Axis Status
TCA	Transfer Value of Captured ANI Input
TCMDER	Transfer Command Error
TDAC	Transfer Digital-to-Analog Converter Voltage
TDIR	Transfer Directory
TDPTR	Transfer Data Pointer Status
TER	Transfer Error Status
TEX	Transfer Program Execution Status
TFB	Transfer Position of Feedback Devices
TGAIN	Transfer All Servo Gains
[TIM]	Current Timer Value
TIMST	Start Timer
TIMSTP	Stop Timer
TIN	Transfer Input Status
TINO	Transfer Other Input Status
TLABEL	Transfer Labels
TLDT	Transfer Position of LDTs
TLIM	Transfer Limit Status
TMEM	Transfer Memory Usage
TOUT	Transfer Output State
TPANI	Transfer Position of ANI Inputs (6270-ANI only)
TPC	Transfer Position Commanded
TPCA	Transfer Voltage of Captured Analog Input
TPCC	Transfer Captured Commanded Position
TPCE	Transfer Position of Captured Encoder
TPCL	Transfer Position of Captured LDT
TPE	Transfer Position of Encoder
TPER	Transfer Position Error
TPROG	Transfer Program
TRACE	Program Trace Mode Enable
TRANS	Translation Mode Enable
TREV	Transfer Revision Level
TSGSET	Transfer Servo Gain Set
TSS	Transfer System Status
TSTAT	Transfer Servo Controller Status
TSTLT	Transfer Servo Settling Time
TTIM	Transfer Time
TUS	Transfer User Status
TVEL	Transfer Present Commanded Velocity
TVELA	Transfer Present Actual Velocity
[TW]	Thumbwheel Data Read
UNTIL( )	Until Part of REPEAT Statement
[US]	User Status
V	Velocity
[V]	Velocity Assignment
VAR	Numeric Variable
VARB	Binary Variable
VARS	String Variable
VCVT( )	Variable Type Conversion
[VEL]	Current Velocity
WAIT( )	Wait for a Specific Condition
WHILE( )	While a Condition is True
WRITE" "	Transmit String to RS-232C port
WRVAR	Transmit Variable to RS-232C port
WRVARB	Transmit Binary Variable to RS-232C port
WRVARS	Transmit String Variable to RS-232C port

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

I N D E X

6000 DOS Support Disk 27, 56
6000 Series Command Language 56
6000 Series Software Reference Guide iii, 56
6270
  description 1
  features 2
  ship kit 3

A

absolute position
  absolute positioning mode 64
  absolute zero position 64
  status 64
AC input voltage 5, 105
acceleration
  acceleration feedforward control (SGAF) 38
  change on the fly 66
  range 105
  S-curve profiling 99
  scaling 58
access to RP240 functions 88
active high/active low convention iii
actual position 31
address
  daisy chain 103
  DIP switch settings 109
airborne contaminants 7
alphabetical command list 117
ambient temperature 7
analog inputs 20
  override 81
analog output
  range 106
  selecting voltage or current 25, 110
  setting max & min limits 25
ANI input 21, 82
  4-20 mA feedback 82
  connections 21
    test 23
  feedback for servo control 30
  position 31, 88
  units of measure iii
application examples 67
application requirements 27
assumptions
  skills required to use the 6270 i
auto-baud 4, 109, 114
auxiliary (AUX) connector 108
auxiliary +5V output 16
auxiliary programmable outputs 2, 17, 70
axes
  number of 25
axes select input 20
axis status 87

B

baud rate 4, 109
  auto baud feature 109
BCD program select input 72
bench test 4
binary variables 91, 93
bitwise operations (and, or, not, etc.) 93
block diagram, system hardware 2
Boolean operations 93
buffered commands 65

C

cables
  custom
    encoder 21
    I/O 21
  RS-232C 114
  shielding 21
CAD-to-motion software 1, 56
capture positions
  encoder, LDT, ANI, commanded 74
CCW end-of-travel limits, *see limits, end-of-travel*
center joystick position 80
chattering servo response 32
circuit drawings 106
  analog output 106
  command 106
  drive enable 106
  drive fault 106
  encoder input 107
  joystick/analog input 108
  LDT input 107
  limit inputs 108
  programmable I/O 108
  shutdown 106
  trigger input 108
closed-loop operation 14, 30
color codes
  encoder cable 16
command
  buffer 73
    after pause 74
    after stop 73
  list, alphabetical 117
  queue 65
command, servo output 30
commanded position 31
communication
  daisy-chaining 102
  parameters 4
  test 5
  troubleshooting 114
  via RP240 terminal (to an external RS-232C device) 83
CompuCAM™ 56
computer-to-terminal conversion 4
conduit 7, 21
configuration
  address 103
  controller 25
  DAC output limits 25
  DIP switch settings 109
  inputs 71
  jogging 75
  jumper settings 25, 109, 110
  LDT feedback type 110
  outputs 68
  output– voltage or current 25, 110
  system ii
  thumbwheel 77
connections
  AC power cable 5
  analog input 19
  ANI inputs 21
  cable extensions 21
  daisy-chain 102
  drive 9
  electronic I/O devices 78
  encoder 15
  end-of-travel limits 14
  factory default 3
  home limit 14
  installation process overview ii
  joystick 19
  LDT 15
  PLC 78
  RP240 19
  RS-232C 4, 102
  thumbwheels 76
  TM8 Thumbwheel Module 77
  valve 9
  VM50 adaptor 18
contaminants 7
continue (!C) 73
continue execution on pause/resume (COMEXR) 74
continue execution on stop (COMEXS) 73
continue input 74
continuous command execution mode (COMEXC) 20, 65
continuous positioning mode 64, 65
control signal 30
controller output saturation 30
conventions
  active high/active low iii
  direction of motion ii
  units of measure iii
coordinate measurement machines 74
critically damped servo response 32
Crosstalk™ 4
current setings, output to valve 25, 110
current settings, output to valve 110
CW end-of-travel limits, *see limits, end-of-travel*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D

DAC output, limiting 25
daisy-chain 102, 109
  including RP240 104
damping 32
data bits 4
data storage – teach mode 94
DC common 115
DC ground wire 115
DC input voltage 105
deadband
  joystick 80
debounce time
  position capture 74
  program select input 73
  programmable inputs 71
debug tools 84, 112
deceleration
  change on the fly 66
  S-curve profiling 99
  scaling 58
default hardware connections 3
defeating noise 116
device address, *see address*
dimensions 8
DIP switch settings
  address 102, 109
  baud rate 109
direction of motion ii
distance
  fractional step truncation 59
  setpoint window 40
distance scaling (SCLD) 60
disturbance 32
  rejection of 36
dithering hydraulic valves 66
DOS Support Disk 56
drive 57
  connections 9, 106
    test 22
  fault input 71
  fault level (DRFLVL 25
  on/off status 5, 87
  shutdown
    on kill 27, 73
  tuning procedure 45

E

Earth ground 5, 115, 116
electric codes iii
electrical noise 7, 21, 44, 112, 115
electro-static discharge (ESD) 109
electronic sensors 18
electronics concepts i
emergency shutdown 43, 55
enable input 16, 57, 113
  test 23
encoder
  compatibility 15
  Compumotor encoder cable colors 16
  connections 15, 107
    test 23
  custom cabling 21
  differential outputs 16
  feedback for servo control 30
  position 31, 88
    after ENBL stop 16
    capture 74
  resolution 26, 58
  single-ended outputs 16
  test 22
  units of measure iii
  Z channel 14
end point
  linear interpolation 101
end-of-travel limits, *see limits, end of travel*
error handling 57
  error level 1 on power up 102
  error program 16
extend ii

F

factory settings
  connections 3
  DIP switches 109
fault output 70
feedback data 30
feedback device polarity reversal 114
feedback device selection 25
full duplex 4

G

gains
  definition 30
  editing via the RP240 89
  gain set 30
    saving, recalling 39
    setpoint window 40
  gain sets 30
  negative 39
  setpoint window 40
  specific to feedback source 25
  tuning
    controller 46
    drive 45
    setup 43
general specifications 105
grounding 7, 116

H

hard limits, *see limits, end-of-travel*
heat 7
high I/O state iii
home input, *see limits, home*
homing 61
  home limit, *see limits, home*
  home reference position 14
  zeroing the absolute position 61
host computer operation 67, 90
humidity 7, 105
hydraulic system components i
hysterisis, setpoint window 40

I

I/O cabling 21
I/O device interface 78
immediate commands 65
immediate stop 65
in position output 69
IN-P, *see inputs, programmable, pull-up*
incremental encoder, *see encoder*
inductive loads 107
inputs
  active high/active low iii
  analog 19, 79, 108
    ANI option 21, 67, 82
    overriding 81
  drive 9, 106
    fault 57
  enable (ENBL) 16, 57, 113
  encoder 15, 21, 107
    capture (position latch) 74
  end-of-travel limits 14, 57, 60, 65, 108
  home limits 14, 61, 108
  jogging 75
  joystick 19, 79, 108
  kill 73
  LDT 15, 107
  no function 72
  one-to-one program select 75
  pause/continue 74
  PLC 78
  program security 76
  program select 73, 75
  programmable 17, 68, 107
    change from sourcing to sinking 78
    debounce time 72
    function assignments 71
    kill 65
    pin outs 18, 107
    polarity iii, 68
    problems 113
    pull-up 3, 17, 78, 108
    screw terminal connections 18
    status 68, 71, 87
    stop 65
    test 23
  RP240 19
  specifications 105
  stop 73
  testing 22
  thumbwheel 77
  trigger 18
  triggers 108
    debounce time 71
    position latch 74
  user fault 57, 74
  valve 9
instability 32
installation
  ANI analog input connections 21
  drive connections 9
  enable input connection 16
  encoder connections 15
  I/O cables 21
  joystick connections 19
  LDT connections 15
  limit connections 14
  mounting 8
  PLC connections 78
  precautions 7
  procedure 8
  programmable I/O connections 17
  RP240 connections 19

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

installation *(continued)*
test/verification 22
thumbwheels 76
trigger connections 18
valve connections 10
VM50 connections 18
installation process overview ii
integral feedback control (SGI) 37
integral windup limit (SGILIM) 37
interface options 67
interpolation – linear 101
interrogation rate 107

J

jerk, acceleration
reducing 99
jogging 75
RP240 jog mode 86
speed select high/low (INFNCi-aL) 75
status 86
velocity 86
high (JOGVH) 75
low (JOGVL) 75
joystick
application example 67, 96
auxiliary input 21
center deadband 80
center voltage 80
connections 19, 108
test 23
interface 67, 79
release input 20
select input 80
status 87
test 22
trigger input 21
velocity resolution 80
jumper settings 109

K

kill
assigned input function 65, 73
effect on drive 27, 73

L

LDT
connections 15, 107
test 23
distance scaling 59
feedback for servo control 30
feedback type (start-stop, or pulse) 110
gradient 26
interrogation rate 107
position 31, 88
capture 74
read error 112
update rate 26, 105
recirculation 26
resolution 26, 58
units of measure iii
LEDs 5, 112
RP240 83
limit to DAC output 25
limits 14
connector pin outs 108
end-of-travel 14, 22, 57, 60, 65
connections 14
used as basis to activate output 70
home 14, 61, 108
polarity 15
test 22
status 87
test 22
linear and rotary motion ii
linear interpolation 101
acceleration scaling 58
end point 101
velocity scaling 59
low I/O state iii

M

master/slave daisy-chain 102, 104
mathematical operations 91
maximum position error exceeded 70
memory – locking 76
menus
RP240 85
microelectronic components 116
motion
test 22
Motion Architect 27, 56
servo tuner option 27, 29
motion control concepts i
motion trajectory update 47, 105
mounting 8
panel layout 8
move completion criteria 53
moving/not moving status 69

N

National Electric Code Handbook iii
negative gains 39
no function input 72
noise, electrical 7, 21, 44, 112
suppression 115
limits & triggers 21
on analog inputs 19
numeric variables 91
display & edit with RP240 89

O

on-the-fly velocity & accel changes 66
one-to-one program select input 75
open loop operation 30, 42, 43
operating system 56
operating temperature 105
operator interface – RP240 83
oscillatory servo response 32, 37
output saturation 31
outputs
activate on position 68, 70
active high/active low iii
analog 106
selecting voltage or current output 25, 110
configuration 68
fault output 70
limit encountered 70
maximum position error exceeded 70
moving/not moving (in position) 69
OUT-A 17, 70
OUT-B 17, 70
program in progress 69
programmable 17, 68, 78, 107
function assignments 68
inductive loads 107
pin outs 18, 107
polarity iii, 68
problems 113
pull-up 3, 17, 78, 107
screw terminal connections 18
status 68, 69, 87
switching thresholds 17
test 23
shutdown 9, 57
specifications 106
over-damped servo response 32
overshoot 33, 37

P

panel layout – mounting 8
parity 4
password
RP240 83
password, RP240 88
pause/continue input 74
PC-Talk™ 4
performance specifications 105
peripheral system components ii
pin outs 106
auxiliary connector 108
drive connector 106
encoder connector 16, 107
joystick connector 19, 108
LDT connector 15, 107
limits connector 14, 108
programmable I/O connector 18, 107
RP240 connector 109
triggers 108
valve/drive connector 9
PIV&F gains 36
PLC interface 67, 78
application example 67
point-to-point move 64
polarity
end-of-travel limits iii, 14
home input iii, 14
programmable I/O iii, 68
trigger inputs 19
position
absolute 64
actual (based on feedback device) 31
after ENBL stop 16
ANI 88
capture 74
commanded 31, 88
capture 74
encoder capture 74
error 31, 88, 112
max. allowable 25, 57
home 14
incremental 65

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

position *(continued)*
latch 17, 74
LDT 88
offset 25
overshoot 37
positioning modes 64
range 105
response (servo) 31
types 32
setpoint 31
status 87
tracking error 31
used to activate output 70
zeroed after homing 61
potentiometer 19
joystick 80
power cable connection 5
power line noise 115
power-up 5
start-up program (STARTP) 24, 84
problems 113
pre-wired connections 3
precautions
installation 7
mounting 8
preset positioning mode 64
Procomm™ 4
program in progress 69
programmable inputs, *see inputs, programmable*
programmable outputs, *see outputs, programmable*
programming
command data fields iii
debug tools 112
debugging via RP240 84
downloading programs from Motion Architect 56
error messages 112
error programs 57
power-up program 24
problems 113
program editing
in Motion Architect 56
program security 76
program selection
BCD 72
debounce time 73
one-to-one 75
skills required i
proportional feedback control (SGP) 36
pulse duration LDT feedback, selecting 110

R

reading inputs and outputs 68
thumbwheel data 76
recirculation 26
reference documents iii
RESET
check for address & auto baud 109
via the RP240 89
resolution
encoder 26
joystick 80
LDT 26
response – servo 32
retract ii
revision levels for 6270, DSP, RP240 88
rise time 33
rotation ii
RP240 82
access security, password 88
application example 67
connections 19, 109
test 23
editing gains & gain sets 89
in daisy chain 104
menu structure 85
send text via the connector 83
test 22
view & edit variables 89
RS-232C communication 4, 116
connections 4
daisy-chaining 102
disable handshaking 114
RP240 connector 83
specifications 105
test 5
troubleshooting 114
runaway motor 14, 113

S

s-curve profiling 99
safety features 57
saturation of the control output 30
scaling 58, 64
specific to feedback source 25
servo
control methods/types 36
open loop operation 43
sampling frequency 21, 30, 46, 47, 105
tuning, *see tuning*
Servo Tuner™ 56
setpoint 31
setpoint window
distance 40
gain set 40
settling time 33
actual 54
shielding 7, 21
shift left to right (>>) 93
shift right to left (<<) 93
shipment – inspection 3
shut down in case of emergency 43, 55
shutdown 9, 57, 106
on kill 27, 73
single-ended encoders 16
single-step mode 2, 84
sinking/sourcing current 17, 78
sourcing/sinking current 17
specifications 105
square-wave signal (dither) 66
stability 32
causing wrong direction 114
stand-alone operation 67
start-up program (STARTP) 24
problems 113
start/stop LDT feedback, selecting 110
status
assigned to binary variable 91
axis 87
inputs 71, 87
joystick inputs 20, 87
LEDs 5, 112
limits 87
numeric & string variables 89
outputs 69, 87
position
capture 74
RP240 display 87
system 87
steady-state 33
position error 31
stiction 66
overcoming 44
stop
assigned input function 65
effect on program execution 73
stop bits 4
storage temperature 105
string variables 91
display with RP240 89
support software 56
CompuCAM 56
DOS Support Disk 56
Motion Architect 56
Servo Tuner 56
surge suppression 115, 116
system connections 9
system status 87
system update rate 26, 47, 105

T

target zone 53
affects moving/not moving output 69
criterion for entering setpoint window 41
timeout error 53
teach mode 94
temperature
operating 105
terminal emulation 56
6000 DOS support disk 56
Motion Architect 56
test
ANI inputs 23
bench test procedure 5
enable input 22
encoder 22
joystick 22
limits 22
motion 22
programmable I/O 22
RP240 22
RS-232C communication 5
system installation 22
thumbwheels
application example 67
connections 17, 77
TM8 module 76
use of 76
timeout error 53
trace mode 2, 84
transient 33

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

transmitted noise 115
travel limit, *see limits, end-of-travel*
trigger inputs
  connections 18, 108
  position latch 74
trigonometric operations 92
troubleshooting
  common problems & solutions 112
  diagnostic LEDs 112
  error messages 112
  methods 111
  RS-232C 114
TTL-compatible voltage levels 105, 106
tuning 29, 30
  6270 tuning procedure 46
  gains, definition 36
  PIV&F algorithm 36
  process flow diagram 48
  related 6000 series commands 34
  scenario (case example) 51
  setup procedure 43
  target zone mode 53
  velocity drive tuning procedure 45

U

under-damped servo response 32
units of measure iii
unstable 32
user fault input 57, 70, 74
user interface options 67

V

valve
  connections 9
    test 22
  dithering 66
  stiction 66
  voltage or current operation 110
variables
  binary 91
  conversion between binary & numeric 91
  numeric 91
    display & edit with RP240 89
    teach mode 94
  string 91
    display with RP240 89
velocity
  accuracy 105
  change on the fly 66
  joystick 20
  range 105
  repeatability 105
  resolution 80
  scaling 59
  velocity feedback control (SGV) 38
  velocity feedforward control (SGVF) 38
verification – bench test 5
verification – installation test 22
VM50 adaptor 18
voltage control – valve 25, 110

W-Z

watchdog timer 57
window, setpoint
  distance 40
  gain set 40
windup of the integral action 37
X-Y linear interpolation 101
Z channel output 15
zero position after homing 61

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP