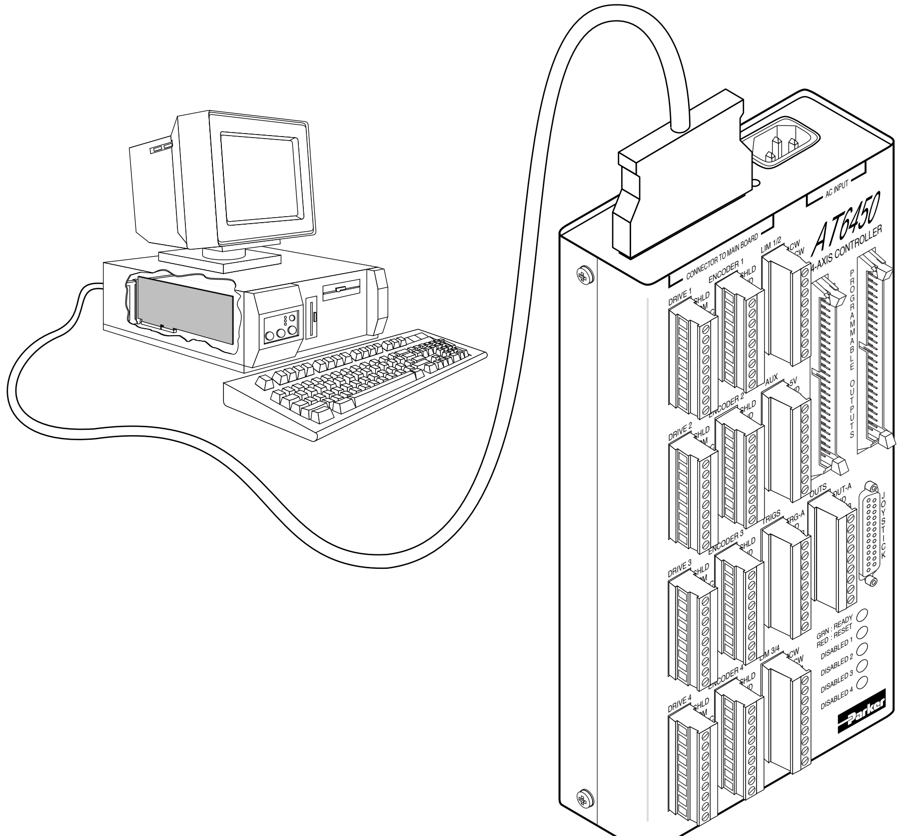
Compumotor

AT6250 & AT6450 Servo Controller

Installation Guide



Compumotor Division
Parker Hannifin Corporation
p/n 88-016412-01A August 1997

IMPORTANT

User Information

⚠ WARNING ⚠

6000 Series products are used to control electrical and mechanical components of motion control systems. You should test your motion system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

Technical Assistance ⇨ Contact your local automation technology center (ATC) or distributor, **or ...**

North America and Asia:
Compumotor Division of Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070 or (707) 584-7558
Fax: (707) 584-3793
FaxBack: (800) 936-6939 or (707) 586-8586
BBS: (707) 584-4059
e-mail: tech_help@cmotor.com
Internet: http://www.compumotor.com

Europe (non-German speaking)**:**
Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
Telephone: +44 (0)1202 69 9000
Fax: +44 (0)1202 69 5750

Germany, Austria, Switzerland:
HAUSER Elektronik GmbH
Postfach: 77607-1720
Robert-Bosch-Str. 22
D-77656 Offenburg
Telephone: +49 (0)781 509-0
Fax: +49 (0)781 509-176

Parker Automation

Product Feedback Welcome
E-mail: 6000user@cmotor.com

AT6250 & AT6450

Product & Documentation Enhancements

Topic	**Description**
New Documentation Set	The AT6250/AT6450 User Guide (p/n 88-014179-01), which combined hardware and software documentation together in one document, is replaced by this document, AT6250/AT6450 Installation Guide (p/n 88-016412-01). This document describes how to install and troubleshoot the AT6250 and AT6450 hardware. Programming related issues are covered in the 6000 Series Programmer's Guide (p/n 88-014540-01) and the 6000 Series Software Reference (p/n 88-012966-01).
Miscellaneous Corrections and Clarifications	**Corrections:** • The end-of-travel limit inputs (CW & CCW, home inputs (HOM), drive fault inputs (DFT), enable input (ENBL), and joystick inputs (pins 15-19 on the JOYSTICK connector) are HCMOS compatible (switching levels: low ≤ 1.00V, high ≥ 3.25V). Previously, they were documented as TTL compatible (low ≤ 0.4V, high ≥ 2.4V). • Recommended connections to the Dynaserv drive have been changed: connect the Dynaserv's pin #50 (AGND) to the AT6n50's AGND terminal (instead of to the CMD- terminal, as previously documented). **Clarifications:** • All inputs and outputs are optically isolated from the internal microprocessor (not from the other inputs and outputs). • Programmable I/O pullup voltage reference: You must select either the on-board +5V terminal or an external 5-24VDC power supply to power the IN-P or OUT-P pull-up resistors. Connecting IN-P or OUT-P to the +5V terminal and to an external supply will **damage the AT6n50**.

ABOUT THIS BOOK

Purpose of This Book

This book describes how to install and troubleshoot the AT6250 and AT6450 hardware. Programming related issues are covered in the *6000 Series Programmer's Guide* and the *6000 Series Software Reference*.

> **"AT6n50"**
> Hereafter, the AT6250 and AT6450 are referred to collectively as the "AT6n50."

What You Should Know

To install and troubleshoot the AT6n50, you should have a fundamental understanding of:

- IBM PC-compatible computer hardware and software operations
- Electronics concepts, such as voltage, switches, current, etc.
- Mechanical motion control concepts, such as torque, velocity, distance and force

Related Publications

- *6000 Series Software Reference*, Parker Hannifin Corporation, Compumotor Division; part number 88-012966-01
- *6000 Series Programmer's Guide*, Parker Hannifin Corporation, Compumotor Division; part number 88-014540-01
- Current *Parker Compumotor Motion Control Catalog*
- Operations user guide for your IBM-compatible computer
- Schram, Peter (editor). *The National Electric Code Handbook (Third Edition)*. Quincy, MA: National Fire Protection Association

Online Manuals This manual (in Acrobat PDF format) is available from our web site: `http://www.compumotor.com`

CHAPTER ONE

1 Installation

IN THIS CHAPTER
• Product ship kit list
• Things to consider before you install the AT6n50
• General specifications table
• Installing the AT6n50 PC card
• Mounting the AT6n50 AUX board
• Connecting all electrical components (includes specifications)
• Testing the installation
• Tuning the AT6n50 (refer to *Servo Tuner User Guide* or to Appendix A)
• Preparing for what to do next

What You Should Have (ship kit)

If an item is missing, call the factory (see phone numbers on inside front cover).

Part Name			Part Number
One of the following line items:			
AT6250 120VAC version (with ship kit)			**AT6250-120**
AT6250 240VAC version (with ship kit)			**AT6250-240**
AT6250 DC version (with ship kit)			**AT6250-DC**
AT6450 120VAC version (with ship kit)			**AT6450-120**
AT6450 240VAC version (with ship kit)			**AT6450-240**
AT6450 DC version (with ship kit)			**AT6450-DC**
Ship kit:	High density cable, 5-foot (standard):	AT6250	71-012832-05
		AT6450	71-011468-05
	High density cable, 15-foot (optional):	AT6250	71-012832-15
		AT6450	71-011468-15
	120VAC power cord (AT6n50-120 only)		44-000054-01
	240VAC EIA power cord connector (AT6n50-240 only)		43-011905-01
	Mounting clips (AC versions only)		53-012068-02
	Mounting screws (AC versions only)		51-006037-01
	(This installation guide)		88-016412-01
	6000 Series Software Reference		88-012966-01
	6000 Series Programmer's Guide		88-014540-01
	Motion Architect diskettes:	Disk 1	95-013070-01
		Disk 2	95-013070-02
		Driver & Samples Disk	95-016324-01
	Operating system/DOS Support diskette:	AT6250	95-014176-01
		AT6450	95-014175-01

Analog Input Option (p/n OPT-AT6N50)-A)

OPT-AT6N50-A is an optional analog input card that installs into your computer's ISA slot, next to the AT6n50 PC card. It provides up to four ±10V 14-bit analog inputs (two inputs for the AT6250 and four inputs for the AT6450).

Before You Begin

⚠ **WARNINGS** ⚠

The AT6n50 is used to control 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.

Always remove power to the computer before:

- Installing or removing the AT6n50 PC card
- Adjusting the DIP switches on the AT6n50 PC card after it is installed in the computer
- Connecting electrical devices to the AT6n50 PC card after it is installed in the computer

Recommended Installation Process

This chapter is organized sequentially to best approximate a typical installation process.

1. Review the general specifications.
2. Perform configuration/adjustments (if necessary).
3. Install the AT6n50 PC card (and the ANI option card, if you ordered one) in your computer.
4. Mount the AT6n50 AUX board.
5. Connect all electrical system components.
6. Test the installation.
7. Mount the motor and couple the load.
8. Tune the AT6n50 for optimum performance (if you are using Servo Tuner, refer to the instructions in the *Servo Tuner User Guide*; otherwise, refer to Appendix A).
9. Program your motion control functions. Programming instructions are provided in the *6000 Series Programmer's Guide* and the *6000 Series Software Reference*. We recommend using the programming tools provided in Motion Architect (found in your ship kit). You can also benefit from our iconic programming interface, Motion Builder (sold separately).

Electrical Noise Guidelines

- 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.
- Noise suppression guidelines for I/O cables are provided on page 21.
- General noise-reduction guidelines are also provided in the Parker Compumotor catalog.

General Specifications

Parameter	Specification
Power	
AT6n50 PC Card	5VDC @ 3.5A from the PC-AT bus
AT6n50-120V AUX Board (AC or DC input)	90-132VAC, 50/60Hz, 1.5A @120VAC, single-phase; or connect an external power source (5VDC ±10% @ 1.6A) to a +5V terminal on the AUX connector
AT6n50-240V AUX Board (AC or DC input)	90-264VAC, 50/60Hz, 0.75A @ 240VAC, single-phase; or connect an external power source (5VDC ±10% @ 1.6A) to a +5V terminal on the AUX connector
AT6n50-DC AUX Board	+5VDC ±5% @ 1.6A ; ±15VDC @ 50mA
Status LEDs/fault detection	Refer to Diagnostic LEDs on page 28
Environmental	
Operating Temperature	32-122°F (0-50°C)
Storage Temperature	-22-185°F (-30-85°C)
Humidity	0-95% non-condensing
Performance	
Position Range & Accuracy	Position range: ±2,147,483,648 counts; Accuracy: ±0 counts from preset total
Velocity Range, Accuracy, & Repeatability	Range: 1-2,000,000 counts/sec; Accuracy: ±0.02% of maximum rate; Repeatability: ±0.02% of set rate
Acceleration Range	1-24,999,975 counts/sec^2
Motion Trajectory Update Rate	See SSFR command description in the 6000 Series Software Reference
Servo Sampling Update Rate	See SSFR command description in the 6000 Series Software Reference
Inputs	All inputs are optically isolated from the microprocessor (not from the other inputs).
Home (HOM), EOT Limits (CW/CCW), Enable (ENBL), Drive Fault (DFT), Joystick inputs (pins 15-19: Axes Select, Velocity Select, Trigger, Release, & Auxiliary)	HCMOS compatible*; internal 6.8 KΩ pull-ups to +5V; voltage range is 0-24V.
Triggers (TRG-A through TRG-D)	TTL compatible*; internal 6.8 KΩ pull-ups to +5V; voltage range is 0-24V.
Encoder	Differential comparator accepts two-phase quadrature incremental encoders with differential (recommended) or single-ended outputs. Maximum voltage = 5VDC. Switching levels (TTL-compatible): Low ≤ 0.4V, High ≥ 2.4V. Maximum frequency = 1.6 MHz. Minimum time between transitions = 625 ns.
Analog (joystick connector)	Voltage range = 0-2.5VDC, 8-bit A/D converter. Input voltage must not exceed 5V.
24 General-Purpose Programmable	HCMOS compatible* with internal 6.8 KΩ pull-ups to IN-P terminal. To source current, connect IN-P to the +5V terminal or to an external 5-24VDC supply (**but not to both**); to sink current, connect IN-P to the GND terminal. IN-P can handle 0-24V with max. current of 100mA. Voltage range = 0-24V. 50-pin plug is compatible with OPTO-22™ equipment.
Outputs	All outputs are optically isolated from the microprocessor (not from the other outputs).
General-Purpose Programmable and OUT-A through OUT-D	Open collector output with 4.7 KΩ pull-ups. Can be pulled up by connecting OUT-P to the +5V terminal or to an external supply of up to 24V (**but not to both**). OUT-P can handle 0-24V with max. current of 50mA. Outputs will sink up to 30mA or source up to 5mA at 5-24VDC. 50-pin plug is compatible with OPTO-22™ signal conditioning equipment.
+5V Output	Internally supplied +5VDC. +5V terminals are available on multiple connectors. Load limit (total load for all I/O connections) is 1.4A for the AT6n50-120 and 0.4A for the AT6n50-240. Load limit for the DC versions of the AUX boards depends on your external +5V supply.
Command Out (CMD)	±10V analog output. 12-bit DAC. Load should be > 2KΩ impedance.
Auxiliary Analog Output (ANA)	±10V analog output. 8-bit DAC. Load should be > 2KΩ impedance. Accuracy is ±5%.
Shutdown (SHTNO, SHTNC, and COM)	Shutdown relay output. Max. rating: 175VDC, 0.25A, 3W.

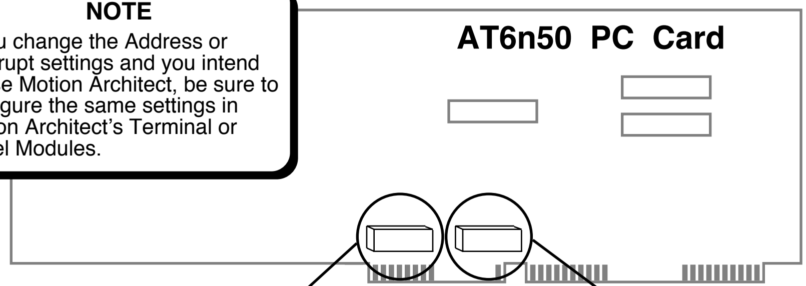
* HCMOS-compatible switching voltage levels: Low ≤ 1.00V, High ≥ 3.25V.
TTL-compatible switching voltage levels: Low ≤ 0.4V, High ≥ 2.4V.

Specifications for the optional ANI analog input PC card (part number OPT-AT6N50-A):

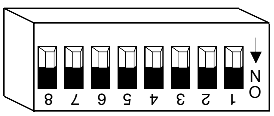
Power	5VDC @ 0.8A from the PC-AT bus
Analog inputs	AT6250 has 2 analog inputs, AT6450 has 4. Voltage range = -10VDC to +10VDC, 14-bit A/D converter.
Status LED	Green power LED above the screw-terminal connector indicates power to the card.

Default Settings
Address = 300 Hex (768 Decimal)
Transfer Mode = 16-Bit
Interrupts Selected = None

NOTE
If you change the Address or Interrupt settings and you intend to use Motion Architect, be sure to configure the same settings in Motion Architect's Terminal or Panel Modules.

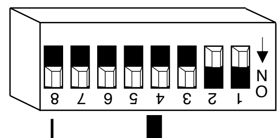


DIP Switch S2
Factory Default Setting Shown



= ON position

DIP Switch S1
Factory Default Setting Shown



INTERRUPTS (S2)

Switch #	Interrupt	Description	Default Setting
1	IRQ3	Serial Port (COM4)	OFF
2	IRQ4	Serial Port (COM3)	OFF
3	IRQ5	Parallel Printer	OFF
4	IRQ7	Parallel Printer	OFF
5	IRQ10	Unassigned	OFF
6	IRQ11	Unassigned	OFF
7	IRQ12	Unassigned	OFF
8	IRQ15	Unassigned	OFF

Switches are positive-true (ON selects the interrupt setting). Only one switch may be ON at one time.

ADDRESS (S1, Switches 1-7)

Switch #	Binary Value Decimal	Binary Value Hex	Default Setting
1	512	200	OFF
2	256	100	OFF
3	128	80	ON
4	64	40	ON
5	32	20	ON
6	16	10	ON
7	8	8	ON

Switches are negative-true (OFF selects the address value). The sum of the binary values of DIP switches 1-7 comprise the PC card's device address. The device address must be an even multiple of eight.

TRANSFER MODE (S1, Switch 8)

Switch #8 ON = 16-Bit Transfer Mode (default)
Switch #8 OFF = 8-Bit Transfer Mode

The 16-bit transfer mode offers higher performance than the 8-bit mode.

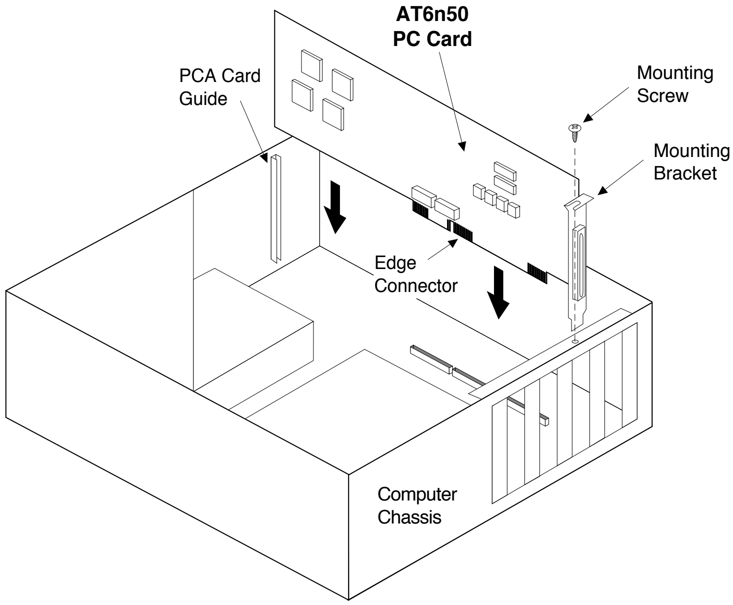
Computer I/O Address Space Map

Address Range Decimal	Address Range Hex	Description
000-255	000-0FF	Used by AT mother-board
496-511	1F0-1FF	Hard Disk (AT)
512-527	200-20F	Game Controller
528-543	210-21F	Expansion Unit
568-571	238-23B	Bus Mouse
572-575	23C-23F	Alternate Bus Mouse
624-639	270-27F	Parallel Printer Port
688-734	2B0-2DF	EGA Card
736-743	2E0-2E7	GPIB
744-751	2E8-2EF	Serial Port (4)
760-767	2F8-2FF	Serial Port (2)
768-799	**300-31F**	**Prototype Card**
800-815	320-32F	Hard Disk (XT)
888-895	378-37F	Parallel Printer
896-911	380-38F	SDLC
928-943	3A0-3AF	SDLC
944-955	3B0-3BB	Monochrome Card
956-959	3BC-3BF	Parallel Printer
960-975	3C0-3CF	EGA Card
976-991	3D0-3DF	CGA Card
1000-1007	3E8-3EF	Serial Port (3)
1008-1015	3F0-3F7	Floppy Disk
1016-1023	3F8-3FF	Serial Port (1)

PC Card Installation

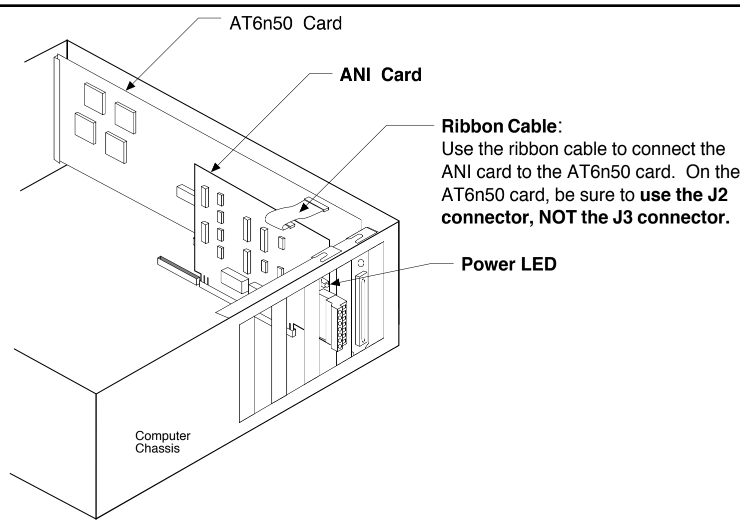
> **CAUTION**
>
> While handling the AT6n50 card, be sure to observe proper grounding techniques to prevent electro-static discharge (ESD).

1. Turn off the power to the computer.
2. Remove the computer's cover to access the internal slots where peripheral cards are added.
3. Select an expansion slot in your PC and remove the sheet metal bracket that covers the associated external access slot. *Save the screw*. On IBM-compatible computers, this is at the rear access panel where all external connections are made. This bracket is replaced by the bracket on the end of the AT6n50 card.
4. Select a 16-bit slot to install the AT6n50 card.
5. Insert the bottom corner of the AT6n50 PC card into the card guide slot near the front of the computer. Ease both ends of the card simultaneously down into the computer until the card's edge connector reaches the computer's mating connector (see drawing below). Adjust the card until the edge connectors align and press it down into the mating connector.
6. Using the screw that secured the original access slot cover bracket, fasten the AT6n50 mounting bracket to the computer chassis (see drawing below).



ANI Card Installation

If you ordered the ANI analog input card (p/n OPT-AT6N50-A) with your AT6n50 system, install it now.

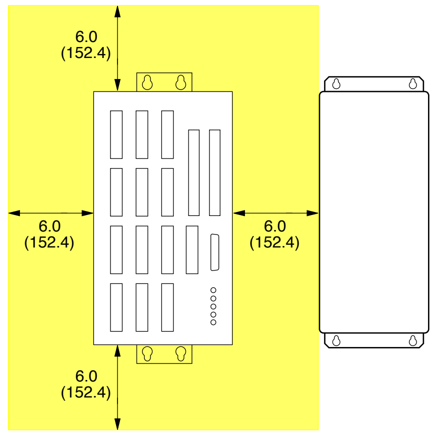


Mounting

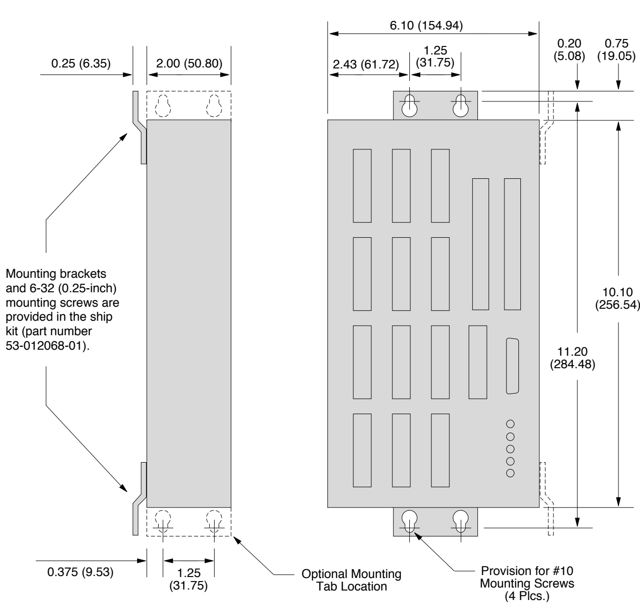
PRECAUTIONS

- To mount the AUX board, you must first attach the mounting brackets. The mounting brackets and 6-32 (0.25-inch) mounting screws are provided in the ship kit.
- Do not use screws longer than 0.25 inches (6.35 mm); they protrude too far into the sheet metal and may damage the AUX board.
- Mount the AUX board within the reach of the supplied 5-foot indexer-to-AUX cable. **You may not lengthen or shorten this cable.** A 15-foot cable is available (part number 71-012832-15).
- Environmental Considerations:
  - Humidity. Keep below 95%, non-condensing.
  - Airborne Contaminants, Liquids. Particulate contaminants, especially electrically conductive material, such as metal shavings and grinding dust, can damage the AT6n50. Do not allow liquids or fluids to come in contact with the AT6n50 or its cables.
  - Temperature. Operate the AT6n50 AUX board in ambient temperatures between 32°F (0°C) and 122°F (50°C). Provide a minimum of 6 inches (152.4 mm) of unrestricted air-flow space around the AUX board (see illustration). Fan cooling may be necessary if adequate air flow is not provided.

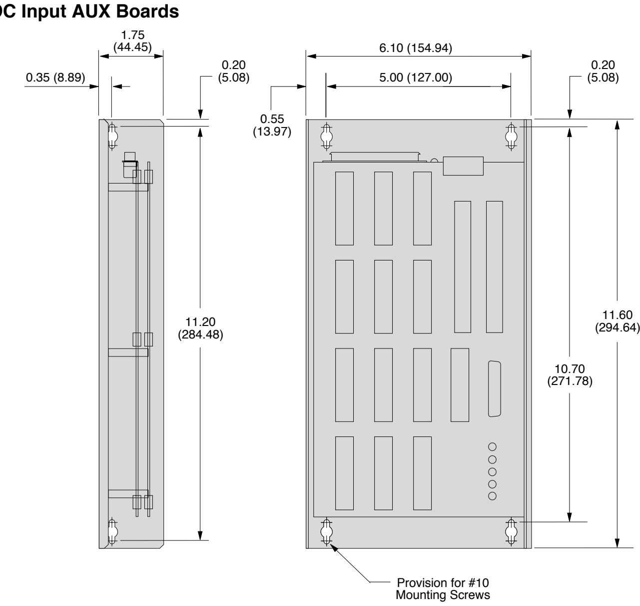
Minimum Airflow Space = 6 in (152.4 mm)



AC Input AUX Boards

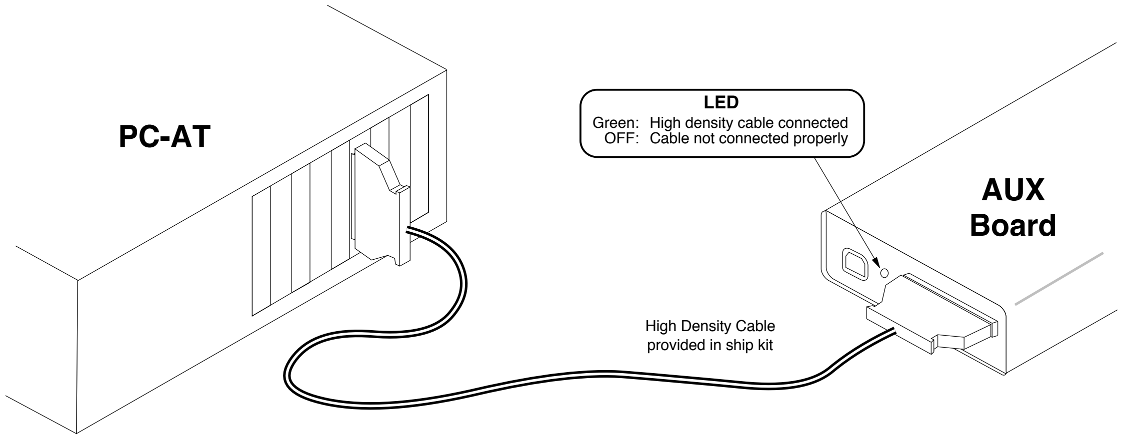


DC Input AUX Boards



Electrical Connections

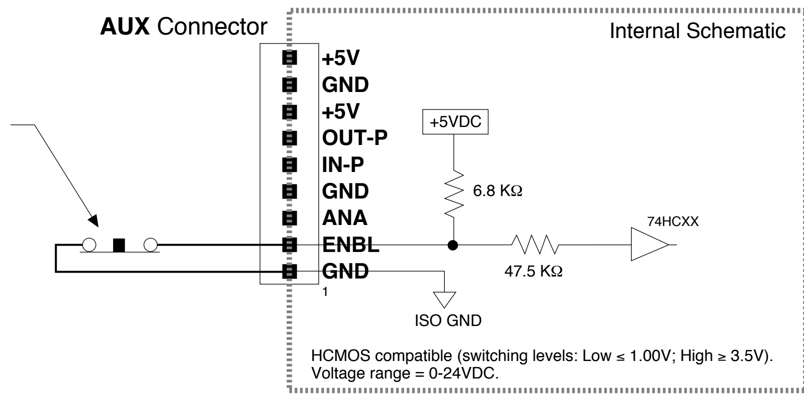
AUX Board Cable



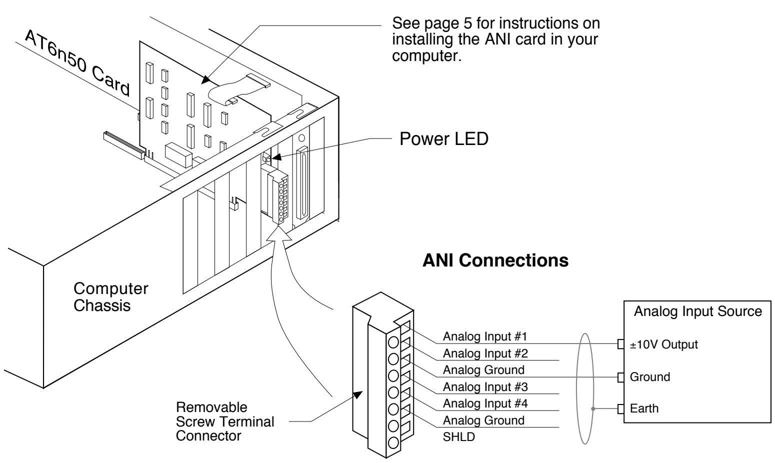
Enable Input (ENBL) — Emergency Stop Switch

ENBL connected to GND (normally-closed switch). If this connection is opened, motion is killed and the program in progress is terminated.

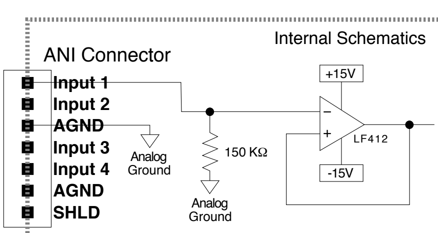
If the ENBL input is not grounded when motion is commanded, motion will not occur and the error message "`WARNING: ENABLE INPUT ACTIVE`" will be displayed in the terminal emulator.



ANI Analog Input (ANI card only – OPT-AT6N50-A)

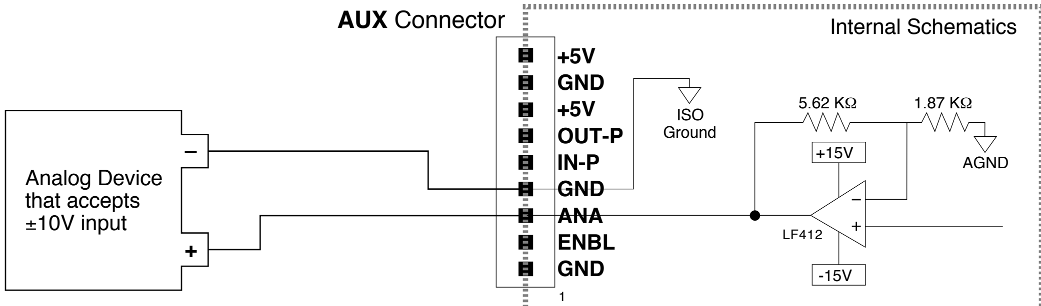


NOTES: • Inputs #3 and #4 are available only with the AT6450-ANI.
• Specs: voltage range is -10V to +10V; 14-bit A/D converter.



- **Each input** is a ±10V analog input with a 14-bit analog-to-digital converter.
- The ANI input is sampled at the servo sampling rate (see table for `SSFR` command).
- Voltage value reported with the `TANI` and `ANI` commands; Position value (819 counts/volt) reported with the `TPANI` and `PANI` commands.

Auxiliary Analog Output – ANA (for "half axis" control)



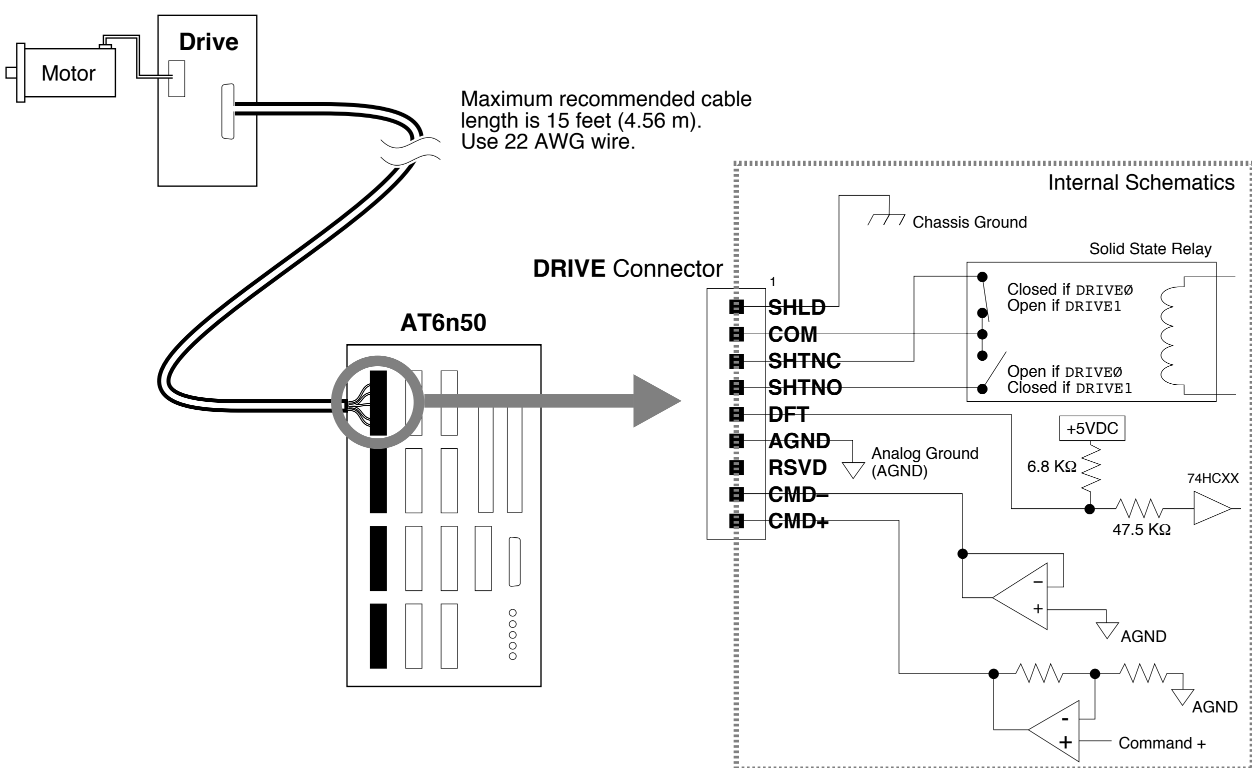
- **ANA** is a ±10V analog output with an 8-bit DAC. Load impedance should be > 2KΩ impedance.
- ANA voltage is set with the `OUTANA` command (see 6000 Software Reference for details).
- ANA is NOT affected by the ENBL switch.

Motor Drivers

WARNING

REMOVE AC POWER FIRST before connecting or disconnecting the drives.

CONNECTIONS & INTERNAL SCHEMATICS



PIN OUTS & SPECIFICATIONS (9-pin DRIVE Connector)

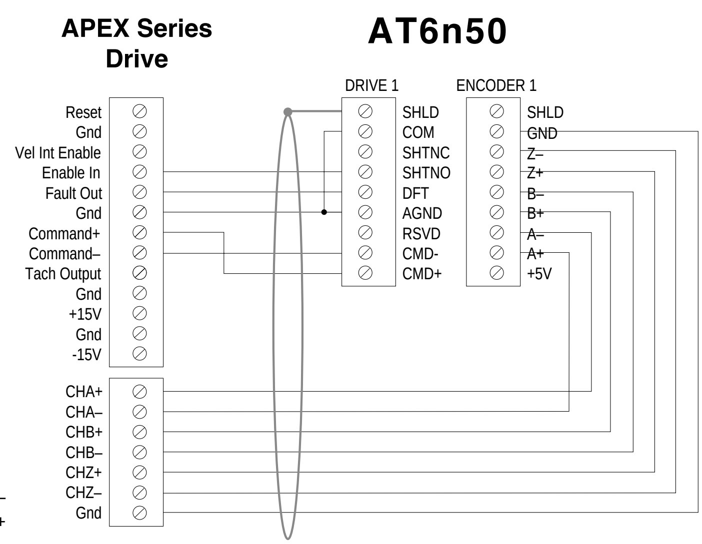
Name	In/Out	Description and Specifications
SHLD	—	Shield—Internally connected to chassis (earth) ground.
COM	—	Signal common for shutdown. Not connected to any ground or other COM.
SHTNC	OUT	Shutdown relay output to drives that require a closed contact to disable the drive. The shutdown relay is active (disabling the drive) when no power is applied to the AT6n50. When the AT6n50 is powered up, the shutdown relay remains active until you issue the `DRIVE1111` command. Max. rating: 175VDC, 0.25A, 3W. Shutdown active (`DRIVEØØØØ`): this output is internally connected to COM (see schematic above). Shutdown inactive (`DRIVE1111`): this output is disconnected from COM (see schematic above).
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 AT6n50. When the AT6n50 is powered up, the shutdown relay remains active until you issue the `DRIVE1111` command. Max. rating: 175VDC, 0.25A, 3W. Shutdown active (`DRIVEØØØØ`): this output is disconnected from COM (see schematic above). Shutdown inactive (`DRIVE1111`): this output is internally connected to COM (see schematic above).
DFT	IN	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. This input is HCMOS compatible (switching levels: Low ≤ 1.00V, High ≥ 3.25V). Internal 6.8 KΩ pull-ups to +5VDC. Voltage range for the DFT input is 0-24V.
AGND	—	Analog ground.
RSVD	—	reserved
CMD–	—	Command signal return.
CMD+	OUT	Command output signal. ±10V analog output. 12-bit DAC. Load should be > 2KΩ impedance.

CONNECTIONS TO SPECIFIC DRIVES

APEX Series Drive

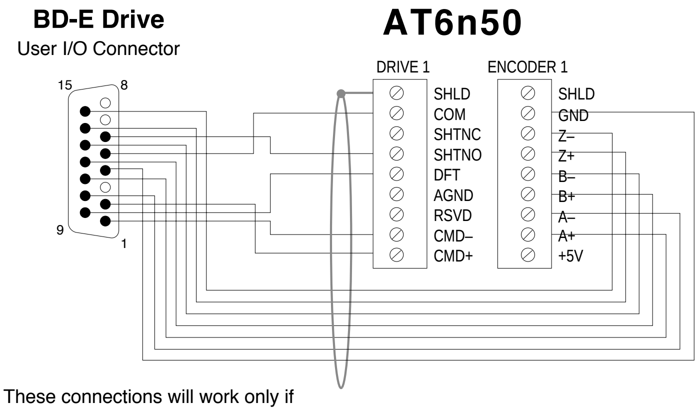
APEX Series Drive		AT6n50
Enable In	↔	SHTNO
Fault Out	↔	DFT
Gnd	↔	AGND
Command+	↔	CMD+
Command–	↔	CMD–
CHA+	↔	A–
CHA–	↔	A+
CHB+	↔	B+
CHB–	↔	B–
CHZ+	↔	Z+
CHZ–	↔	Z–
Gnd	↔	GND

NOTE:
Apex Series CHA+ connected to AT6n50's A–
Apex Series CHA– connected to AT6n50's A+



BD-E Drive

BD-E Drive		AT6n50
V2 (pin 1)	↔	CMD–
V1 (pin 2)	↔	CMD+
GND (pin 4)	↔	GND
RST (pin 5)	↔	COM
+15V (pin 6)	↔	SHTNO
FT (pin 9)	↔	DFT
AOP (pin 10)	↔	A–
AOP (pin 11)	↔	A+
BOP (pin 12)	↔	B+
BOP (pin 13)	↔	B–
ZOP (pin 14)	↔	Z+
ZOP (pin 15)	↔	Z–

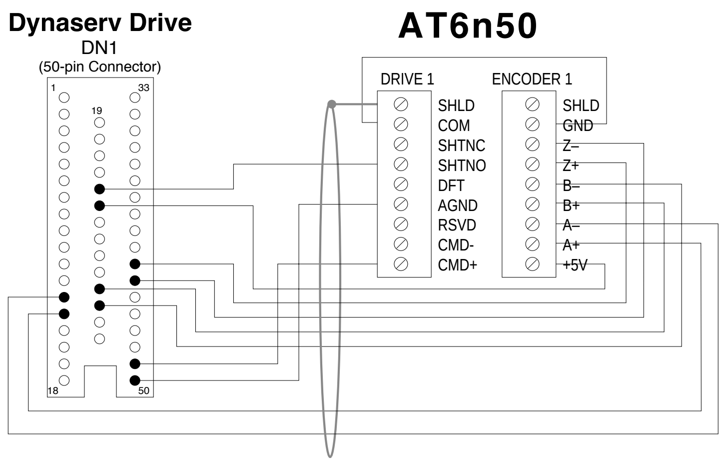


NOTE: These connections will work only if BD-E jumper LK2 is set to position B (not the factory default position).

Dynaserv Drive

Dynaserv Drive		AT6n50
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)	↔	AGND

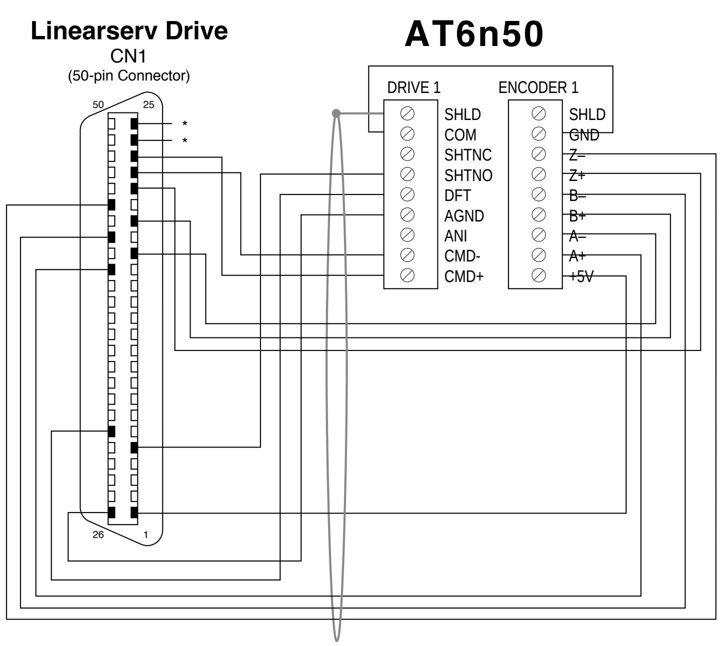
NOTE:
Dynaserv A+ connected to AT6n50's A–
Dynaserv A– connected to AT6n50's A+
AT6n50 GND connected to AT6n50 COM



Linearserv Drive

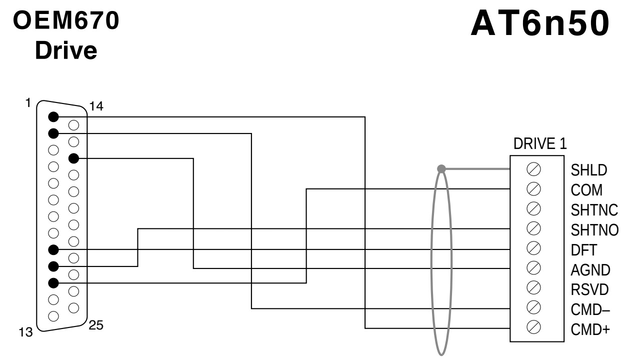
Linearserv Drive		AT6n50
Com+ (pin 01)	↔	+5V
Servo On– (pin 05)	↔	SHTNO
A+ (pin 17)	↔	A– **
B+ (pin 19)	↔	B+
Z+ (pin 21)	↔	Z+
Agnd-TQ (pin 22)	↔	CMD- *
Vin-TQ (pin 23)	↔	CMD+ *
Agnd-VEL (pin 24)	↔	CMD- *
Vin-VEL (pin 25)	↔	CMD+ *
Com– (pin 26)	↔	AGND
Ready+ (pin 31)	↔	DFT
A– (pin 41)	↔	A+ **
B– (pin 43)	↔	B–
Z– (pin 45)	↔	Z–

NOTE:
* When the Linearserv is in Torque Mode, connect Linearserv pins 23 & 22 to CMD+ & CMD-. When in the Velocity Mode, connect pins 25 & 24 are CMD+ & CMD-.
** Connect Linearserv A+ to AT6n50 A–.
** Connect Linearserv A– to AT6n50 A+.
Connect AT6n50 GND to AT6n50 COM.



OEM670 Drive

OEM670 Drive		AT6n50
CMD+ (pin 1)	↔	CMD+
CMD– (pin 2)	↔	CMD–
FAULT (pin 9)	↔	DFT
ENABLE (pin 10)	↔	SHTNO
GND (pin 11)	↔	COM
GND (pin 16)	↔	AGND



SV Drive

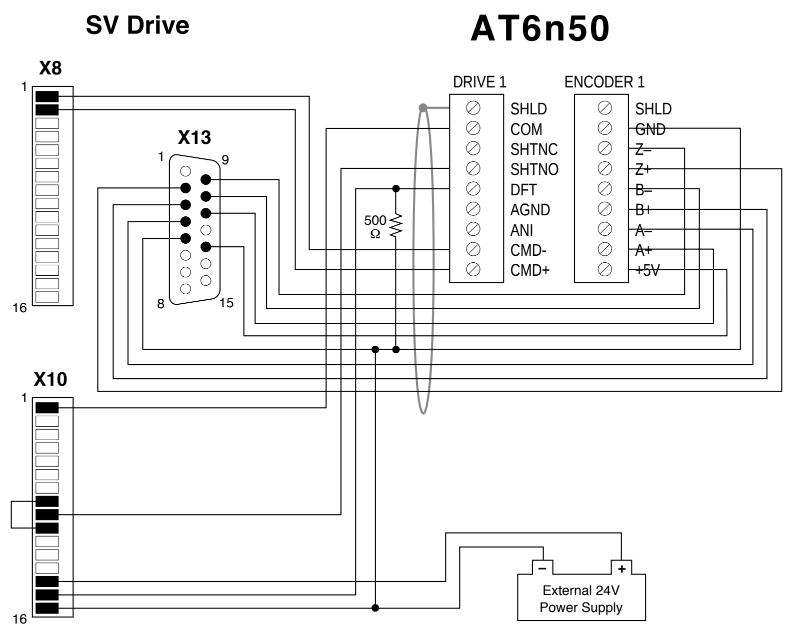
SV Drive		AT6n50
SOLL1+ (X8 pin 01)	↔	CMD+
SOLL1– (X8 pin 02)	↔	CMD–
N (X13 pin 02)	↔	Z+
B (X13 pin 03)	↔	B+
A (X13 pin 04)	↔	A– *
GND (X13 pin 05)	↔	GND
N/ (X13 pin 09)	↔	Z–
B/ (X13 pin 10)	↔	B–
A/ (X13 pin 11)	↔	A+ *
+5V (X13 pin 13)	↔	+5V
ENABLE GND (X10 pin 08)	←	Short these two terminals
+24V OUT GND (X10 pin 10)	←	
ENABLE (X10 pin 01)	↔	COM
+24V OUT (X10 pin 09)	↔	SHTNO
+24V IN (X10 pin 14)	↔	+24V ** (Ext. Supply)
Fault Output (X10 pin 15)	↔	DFT ***
GND for +24V (X10 pin 16)	↔	GND ** (& Ext. Supply)

NOTE:

* Connect SV A+ (called "A") to AT6n50 A–. Connect SV A– (called "A/") to AT6n50 A+.

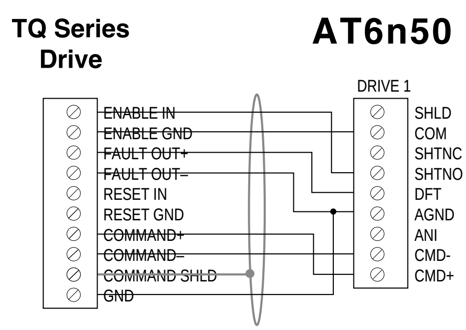
** Connect SV's X10 pins 14 & 16 to an external 24V power supply. Also connect SV X10 pin 16 to AT6n50 GND.

*** Connect a 500Ω resistor between the AT6n50's GND and DFT terminals.



TQ Series Drive

TQ Series Drive		AT6n50
ENABLE IN (pin 1)	↔	SHTNO
ENABLE GND (pin 2)	↔	COM
FAULT OUT+ (pin 3)	↔	DFT
FAULT OUT– (pin 4)	↔	AGND
COMMAND+ (pin 7)	↔	CMD+
COMMAND– (pin 8)	↔	CMD–
COMMAND SHLD (pin 9)	↔	(cable shield)
GND (pin 10)	↔	AGND



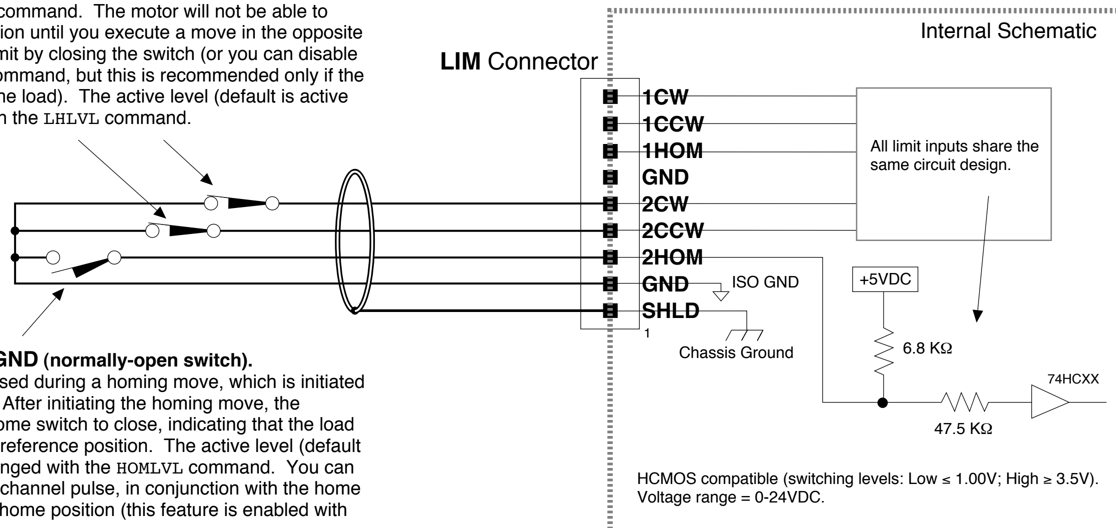
End-of-Travel and Home Limit Inputs

NOTES

- Motion will not occur on a particular axis until you do one of the following:
  - Install end-of-travel (**CW** & **CCW**) limit switches
  - Disable the limits with the `LHØ` command (recommended only if load is not coupled)
  - Change the active level of the limits with the `LHLVL` command
- Refer to the Basic Operations Setup chapter in the 6000 Series Programmer's Guide for in-depth discussions about using end-of-travel limits and homing.

CONNECTIONS & INTERNAL SCHEMATICS

CW & CCW connected to GND (normally-closed switches). Mount each switch such that the load forces it to open before it reaches the physical travel limit (leave enough room for the load to stop). When the load opens the switch, the axis stops at the decel value set with the `LHAD` command. The motor will not be able to move in that same direction until you execute a move in the opposite direction and clear the limit by closing the switch (or you can disable the limits with the `LHØ` command, but this is recommended only if the motor is not coupled to the load). The active level (default is active low) can be changed with the `LHLVL` command.



HOM connected to GND (normally-open switch). The home limit input is used during a homing move, which is initiated with the `HOM` command. After initiating the homing move, the controller waits for the home switch to close, indicating that the load has reached the "home" reference position. The active level (default is active low) can be changed with the `HOMLVL` command. You can also use an encoder's Z channel pulse, in conjunction with the home switch, to determine the home position (this feature is enabled with the `HOMZ1` command).

PIN OUTS & SPECIFICATIONS (9-pin LIM Connectors)

Pin #	In/Out	Name, Axes 1 & 2	Name, Axes 3 & 4	Description
9	IN	1CW	3CW	Positive-direction end-of-travel limit input.
8	IN	1CCW	3CCW	Negative-direction end-of-travel limit input.
7	IN	1HOM	3HOM	Home limit input.
6	—	GND	GND	Isolated ground.
5	IN	2CW	4CW	Positive-direction end-of-travel limit input.
4	IN	2CCW	4CCW	Negative-direction end-of-travel limit input.
3	IN	2HOM	4HOM	Home limit input.
2	—	GND	GND	Isolated ground.
1	—	SHLD	SHLD	Shield—Internally connected to chassis ground (earth).

Specification for all limit inputs

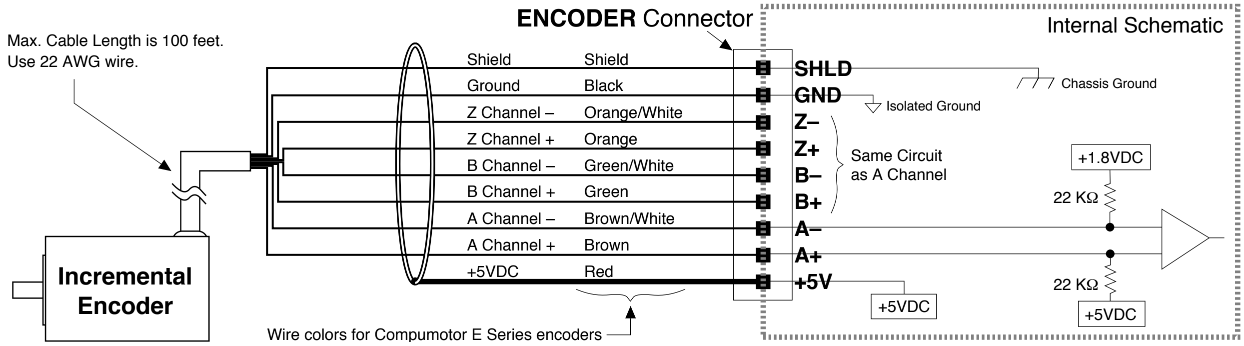
HCMOS compatible (voltage levels: Low ≤ 1.00V, High ≥ 3.25V); internal 6.8 KΩ pull-ups to +5V; voltage range is 0-24V.

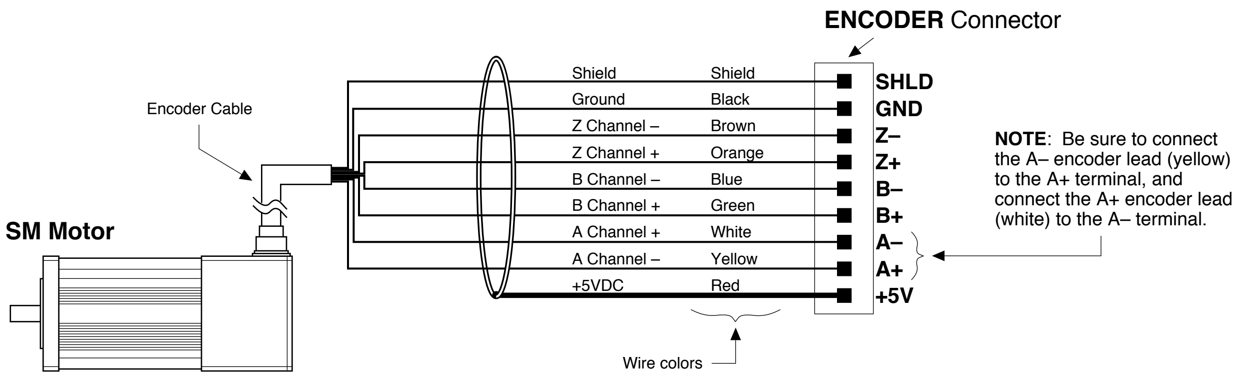
Active level for CW & CCW is set with the `LHLVL` command (default is active low, requiring normally-closed switch).

Active level for HOM is set with the `HOMLVL` command (default is active low, requiring normally-open switch).

Encoders

CONNECTIONS & INTERNAL SCHEMATICS





PIN OUTS & SPECIFICATIONS (9-pin ENCODER Connectors)

Pin #	In/Out	Name	Description
1	-----	SHLD	Shield—Internally connected to chassis ground (earth).
2	-----	GND	Isolated logic ground.
3	IN	Z–	Z– Channel quadrature signal input.
4	IN	Z+	Z+ Channel quadrature signal input.
5	IN	B–	B– Channel quadrature signal input.
6	IN	B+	B+ Channel quadrature signal input.
7	IN	A–	A– Channel quadrature signal input.
8	IN	A+	A+ Channel quadrature signal input.
9	OUT	+5V	+5VDC output to power the encoder.

Specification for all encoder inputs

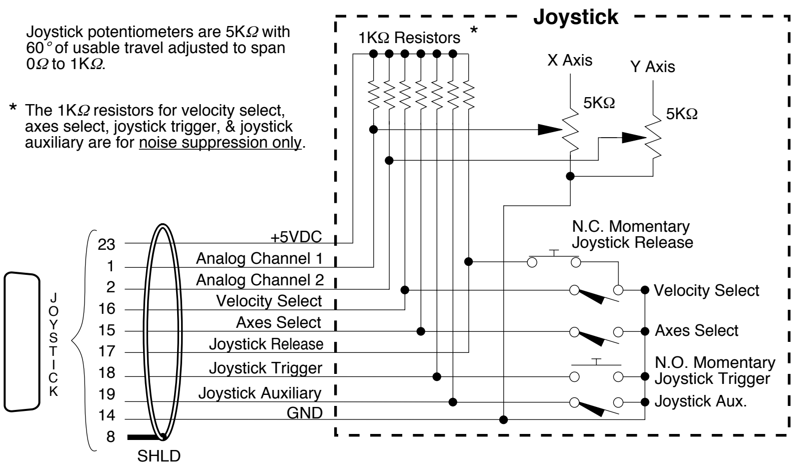
Differential comparator accepts two-phase quadrature incremental encoders with differential (recommended) or single-ended outputs. Max. frequency is1.6 MHz. Minimum time between transitions is 625 ns. TTL-compatible voltage levels: Low ≤ 0.4V, High ≥ 2.4V. Maximum input voltage is 5VDC.

Requirements for Non-Compumotor Encoders

- Use incremental encoders with two-phase quadrature output. An index or Z channel output is optional. **Differential outputs are recommended.**
- It must be a 5V (< 200mA) encoder to use the AT6n50's **+5V** output. Otherwise, it must be separately powered with TTL-compatible (low ≤ 0.4V, high ≥ 2.4V) or open-collector outputs.
- If you are using a single-ended encoder, leave the **A–**, **B–** and **Z–** terminals on the AT6n50 unconnected.

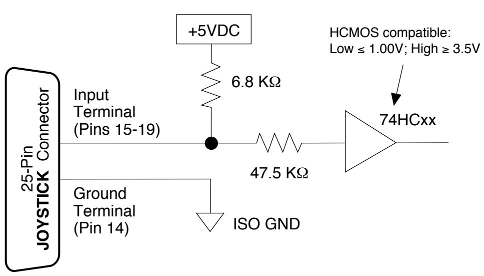
Joystick & Analog Inputs

CONNECTIONS



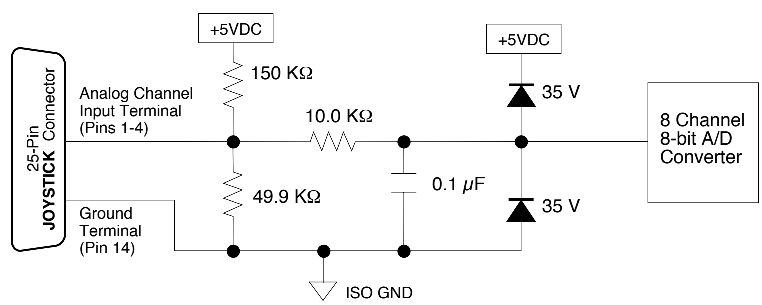
INTERNAL SCHEMATICS

Joystick Input Circuit



This input circuit applies to Axes Select, Velocity Select, Joystick Release, Joystick Trigger, & Joystick Auxiliary.

Analog Channel Input Circuit



PIN OUTS & SPECIFICATIONS

Pin	In/Out	Name	Description
1	IN	Analog Channel 1	Analog input for joystick control of axis. Voltage range is 0-2.5VDC, 8-bit A/D converter. CAUTION: Input voltage must not exceed 5VDC.
2	IN	Analog Channel 2	(same description as pin 1 above).
3	IN	Analog Channel 3	(same description as pin 1 above).
4	IN	Analog Channel 4	(same description as pin 1 above).
8	—	Shield	Shield (chassis ground).
14	—	Ground	Ground (isolated).
15	IN	Axes Select	If using one joystick, you can use this input to alternately control axes 1 & 2 or axes 3 & 4. *
16	IN	Velocity Select	Input to select high or low velocity range (as defined with the `JOYVH` or `JOYVL` commands). *
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 (`JOY1`) 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 AT6n50 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.

* Input voltage range for pins 15-19 is 0-24VDC. HCMOS compatible (switching voltage levels: Low ≤ 1.00V, High ≥ 3.25V).

Trigger Inputs

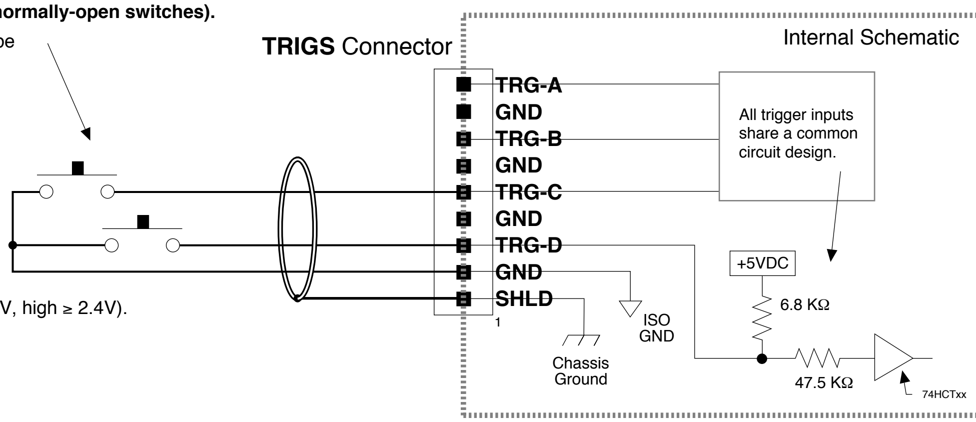
Trigger Inputs connected to GND (normally-open switches).

The active level (default is active low) can be changed with the `INLVL` command.

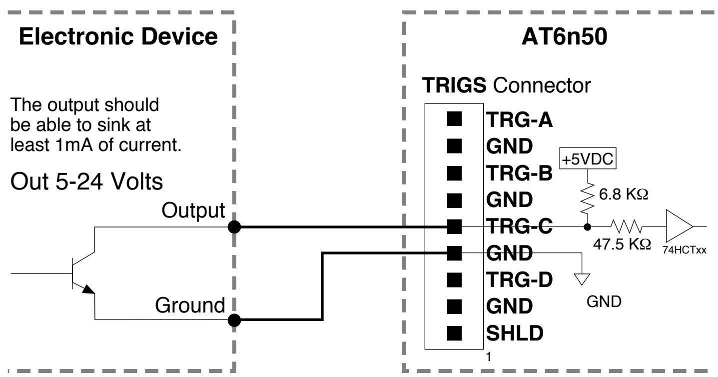
These inputs are like the general-purpose inputs on the 50-pin header. The differences are (1) the triggers are always internally pulled up to +5V and are TTL compatible; and (2) the triggers can be programmed with the `INFNCi-H` command to function as position capture inputs and registration inputs.

TTL compatible (switching levels: low ≤ 0.4V, high ≥ 2.4V). Voltage range = 0-24V.

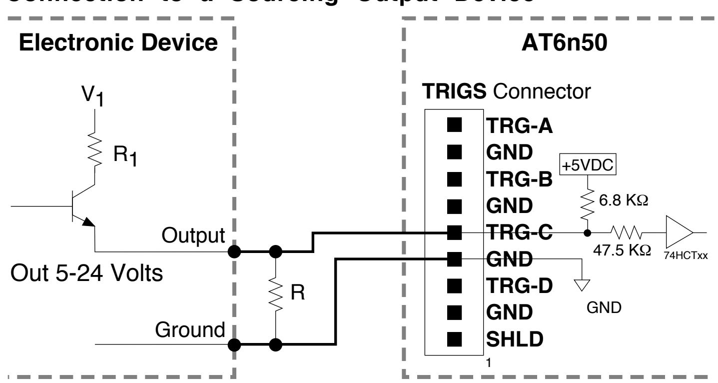
AT6250 Only: TRG-D is not available.



Connection to a Sinking Output Device



Connection to a Sourcing Output Device



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.

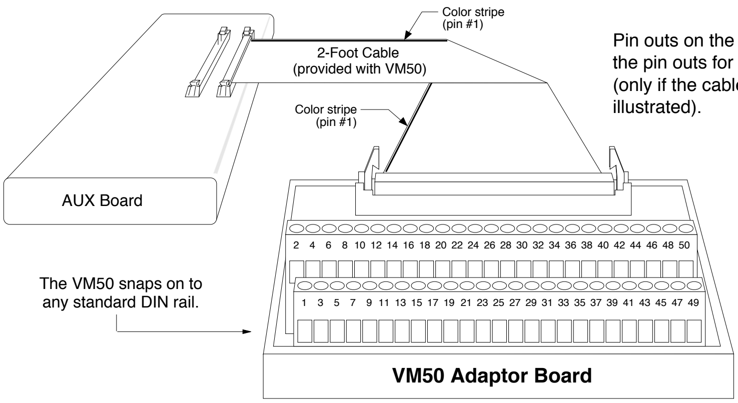
The resistor provides a path for current to flow from the device when the output is active.

PROGRAMMING TIP

Connecting to a sinking output? Set the trigger input's active level to low with the `INLVL` command (`Ø` = active low, default setting). **Connecting to a sourcing output?** Set the trigger input's active level to high with the `INLVL` command (`1` = active high). Thus, when the output is active, the `TIN` status command will report a "`1`" (indicates that the input is active), regardless of the type of output that is connected. For details on setting the active level and checking the input status refer to the `INLVL` and `TIN` command descriptions in the 6000 Series Software Reference.

General-Purpose Programmable Inputs & Outputs

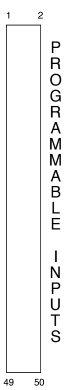
VM50 ADAPTOR — for screw-terminal connections



Pin outs on the VM50 are identical to the pin outs for the 50-pin connectors (only if the cable is connected as illustrated).

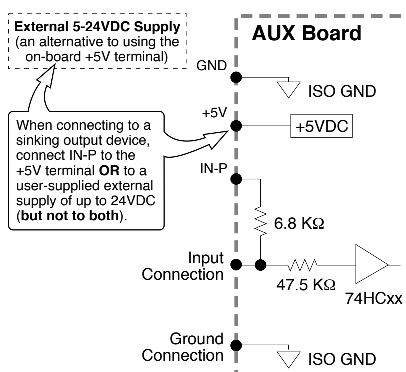
PIN OUTS & SPECIFICATIONS

Inputs



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

Internal Schematic



Specifications

HCMOS-compatible voltage levels (low ≤ 1.00V, high ≥ 3.25V).

Voltage range = 0-24V.

Sourcing Current: On the **AUX** connector, connect **IN-P** to **+5V** <u>or</u> connect **IN-P** to an external 5-24VDC supply (<u>but not to both</u>).

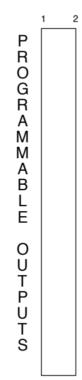
Sinking Current: On the **AUX** connector, connect **IN-P** to **GND**.

50-pin plug is compatible with OPTO-22™ signal conditioning equipment.

STATUS: Check with `TIN` or `INFNC`.

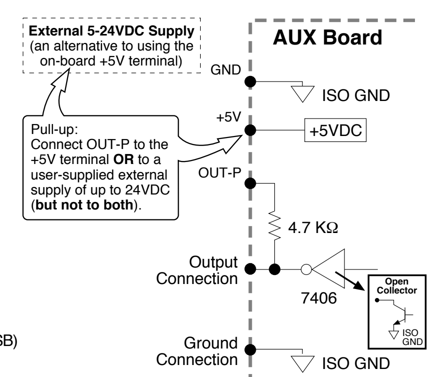
Active level: Default is active low, but can be changed to active high with the `INLVL` command.

Outputs



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

Internal Schematic



Specifications

Includes the 24 general-purpose outputs (**PROGRAMMABLE OUTPUTS** connector), and **OUT-A** thru **OUT-D** (**OUTS** connector). **AT6250**: **OUT-C** is used only as a general-purpose output, not for the Output On Position function; **OUT-D** is not available.

Open collector outputs; will sink up to 30mA or source up to 5mA at 5-24VDC.

Pull-up connection on **AUX** connector: Connect **OUT-P** to **+5V**, <u>or</u> to an external 5-24VDC power supply (<u>but not to both</u>).

50-pin plug is compatible with OPTO-22™ signal conditioning equipment.

STATUS: Check with `TOUT` or `OUTFNC`.

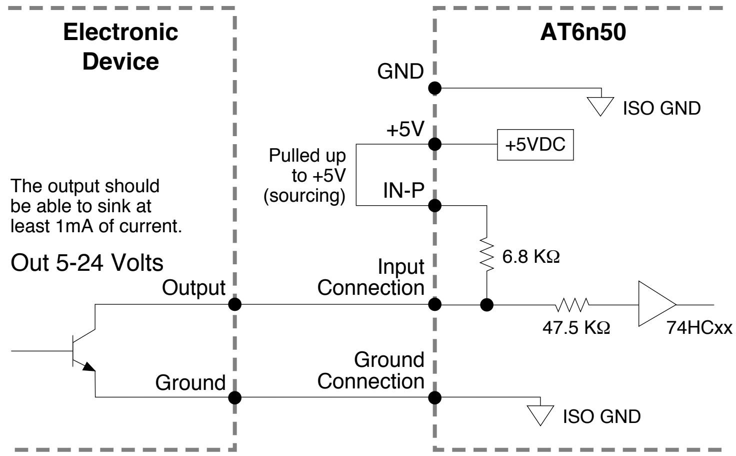
Active level: Default is active low, but can be changed to active high with the `OUTLVL` command.

NOTE: All even-numbered pins are connected to a common logic ground (DC ground)
LSB = least significant bit; MSB = most significant bit

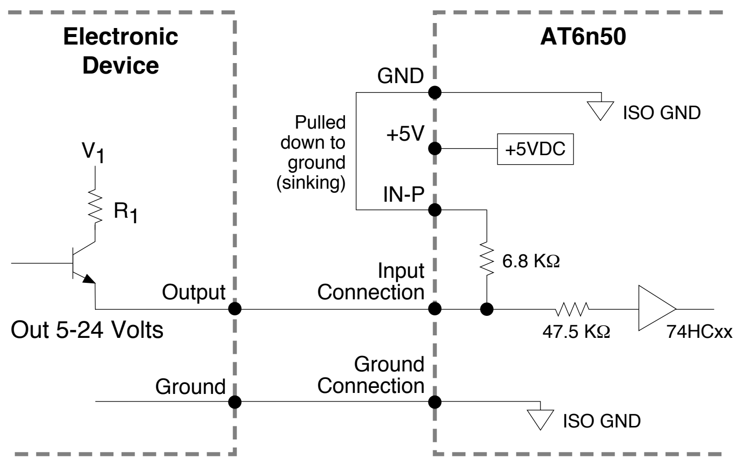
CAUTION: You must select <u>either</u> the on-board **+5V** terminal <u>or</u> an external power supply to power the **IN-P** and **OUT-P** pull-up resistors. Connecting **IN-P** or **OUT-P** to the **+5V** terminal <u>and</u> an external supply will **damage the AT6n50**.

INPUT CONNECTIONS — Connecting to electronic devices such as PLCs

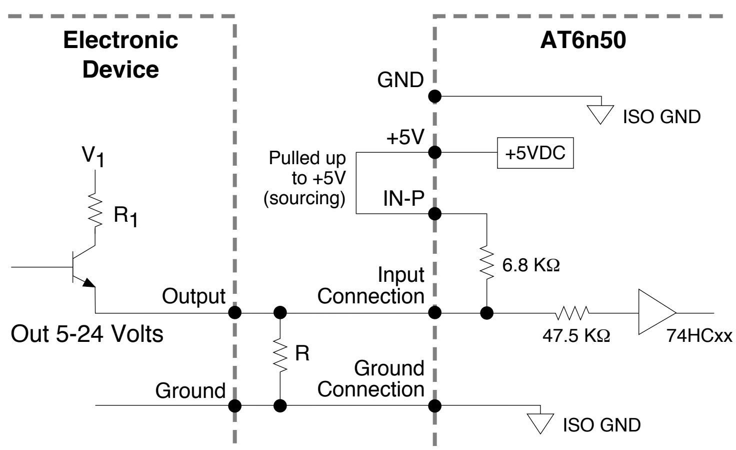
Connection to a Sinking Output Device



Connection to a Sourcing Output Device



Connection to a Combination of Sinking & Sourcing Outputs



Typical value for R = 450Ω (assuming R_1 = 0)

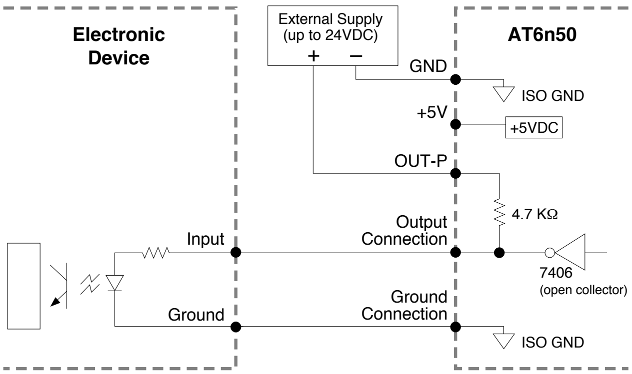
Note: The value of R may vary depending on the value of R_1 and V_1.

NOTE: If you will be connecting to a combination of sourcing and sinking outputs, connect **IN-P** to **+5V** (or to an external 5-24VDC supply) to accommodate sinking output devices. Then for each individual input connected to a sourcing output, wire an external resistor between the AT6n50's programmable input terminal and ground (see "R" in above drawing). The resistor provides a path for current to flow from the device when the output is active.

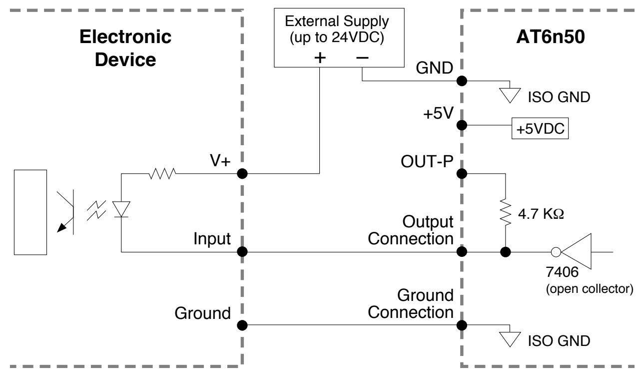
> **PROGRAMMING TIP**
>
> **Connecting to a sinking output?** Set the input's active level to low with the `INLVL` command (Ø = active low).
>
> **Connecting to a sourcing output?** Set the input's active level to high with the `INLVL` command (`1` = active high).
>
> Thus, when the output is active, the `TIN` status command will report a "`1`" (indicates that the input is active), regardless of the type of output that is connected.
>
> Details on setting the active level and checking the input status are provided in the 6000 Series Programmer's Guide. Refer also to the `INLVL` and `TIN` command descriptions in the 6000 Series Software Reference.

OUTPUT CONNECTIONS — for electronic devices such as PLCs

Connection to a Sinking Input (active high)

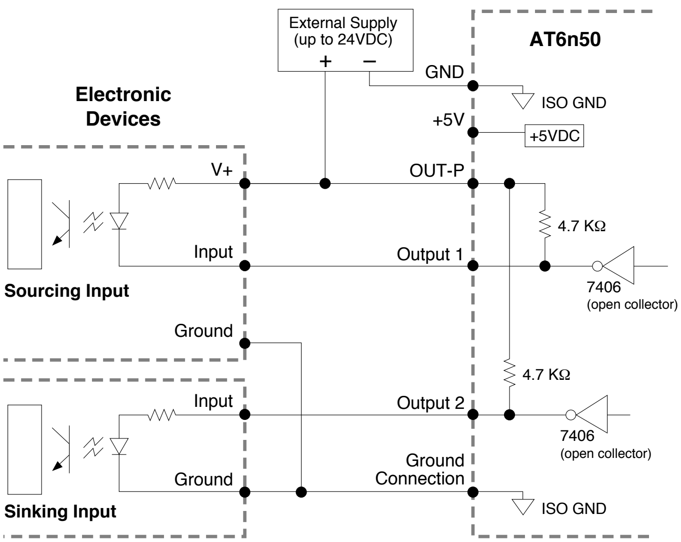


Connection to a Sourcing Input (active low)



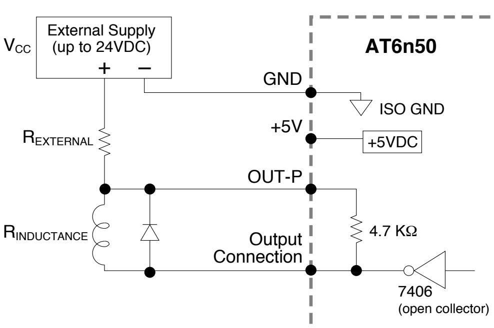
NOTE: It is not necessary to use the OUT-P pin for a sourcing input.

Connection to a Combination of Sinking & Sourcing Inputs



Combinations of sourcing and sinking inputs can be accommodated at the same voltage level. Be aware of the input impedance of the sourcing input module, and make sure that there is enough current flowing through the input module while in parallel with the OUT-P pull-up resistor.

Connection to an Inductive Load (active low)



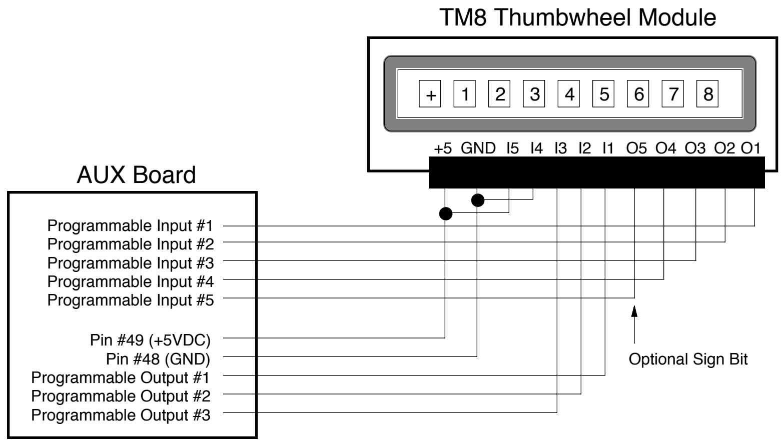
Use an external diode when driving inductive loads. Connect the diode in parallel to the inductive load, attaching the anode to the AT6n50 output and the cathode to the supply voltage of the inductive load, via an external resistor. To size the external resistor, use this formula:

$$\frac{V_{CC}}{R_{EXTERNAL} + R_{INDUCTANCE}} \leq 30mA$$

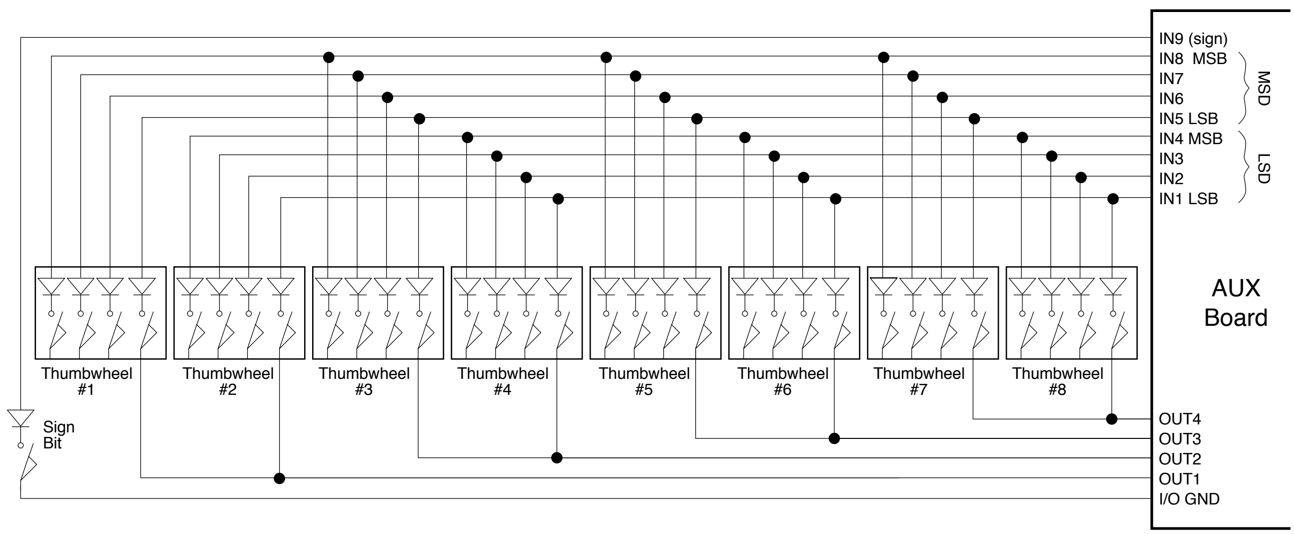
> **PROGRAMMING TIP**
>
> **Connecting to an active-high sinking input?** Set the output's active level to high with the `OUTLVL` command (`1` = active high).
>
> **Connecting to an active-low sourcing input?** Set the output's active level to low with the `OUTLVL` command (Ø = active low).
>
> Thus, when the AT6n50's output is activated, current will flow through the attached input and the `TOUT` status command will report a "`1`" (indicates that the output is active), regardless of the type of input that is connected.
>
> Details on setting the active level and checking the output status are provided in the 6000 Series Programmer's Guide. Refer also to the `OUTLVL` and `TOUT` command descriptions in the 6000 Series Software Reference.

THUMBWHEEL CONNECTIONS — for entering BCD data

Connection to the Compumotor TM8 Module



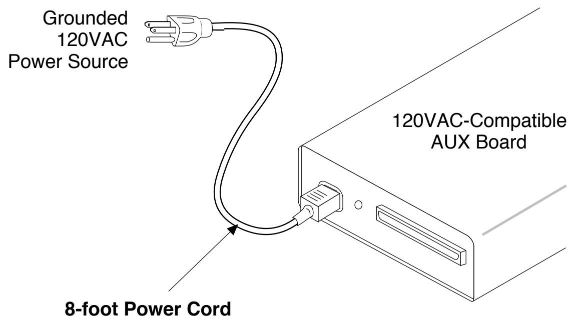
Connection to your own Thumbwheel Module



Input Power

AC Input AUX Boards

120VAC Input (AT6n50-120)



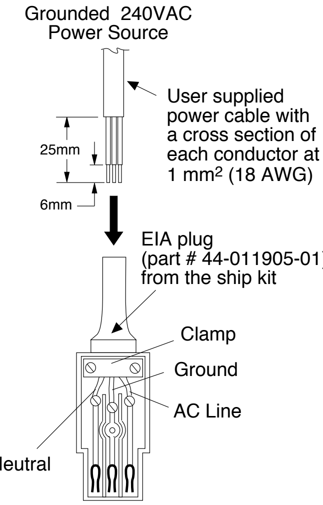
8-foot Power Cord
(part # 44-000054-01)
provided in the ship kit

Cable Lead Color	Signal
Black	AC Line
White	AC Neutral
Green	Earth Ground

Power Input: 90-132VAC, 50/60Hz, 1.5A @ 120VAC, single-phase

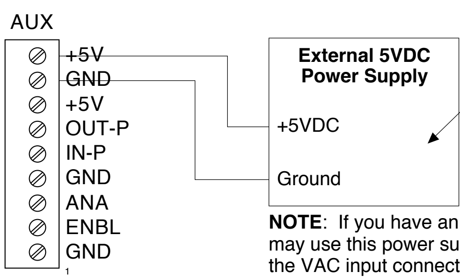
240VAC Input (AT6n50-240)

1. Remove the cover on the EIA plug.
2. Build your power cable as shown. **DO NOT USE A 120VAC CABLE**.
3. Slide the power cable into the plug.
4. Connect the conductor wires.
5. Lock the conductor wires under the clamp.
6. Reassemble the plug.



Power Input: 90-264VAC, 50/60Hz, 0.75A @ 240VAC, single-phase

DC Input (alternative to AC input)



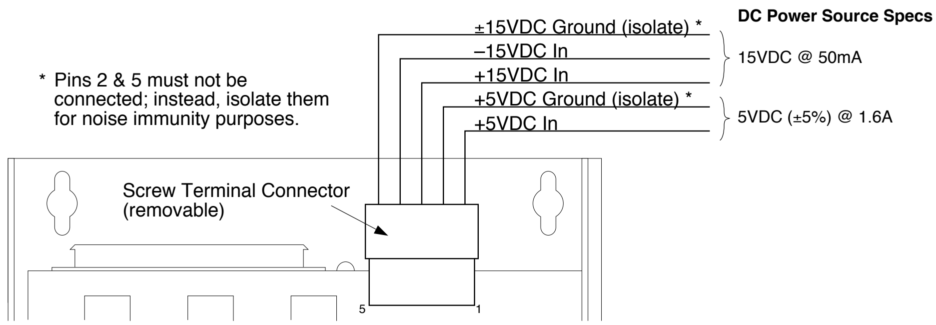
Power Input Specification

+5VDC (±5%) @ 1.6A. This provides the power for all the +5V terminals on all I/O connectors.

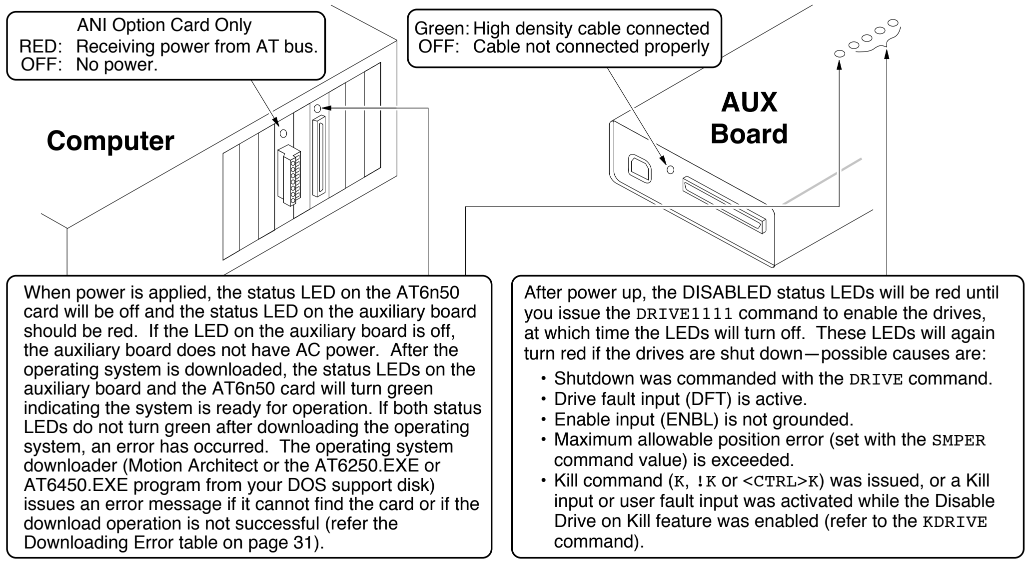
NOTE: If you have an AC version of the AUX board, you may use this power supply method as an alternative to the VAC input connection illustrated above.

DC Input AUX Boards

* Pins 2 & 5 must not be connected; instead, isolate them for noise immunity purposes.



STATUS LEDs

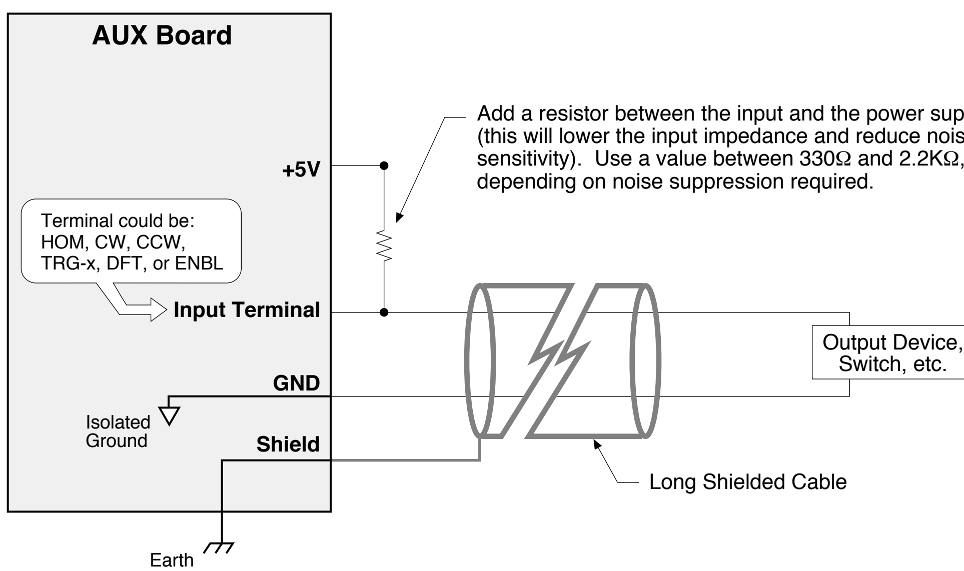


Lengthening I/O Cables

Bear in mind that lengthening cables increases noise sensitivity. (The maximum length of cables is ultimately determined by the environment in which the equipment will be used.) If you lengthen the cables, follow the precautions below to minimize noise problems.

- Use a minimum wire size of 22 AWG.
- Use twisted pair shielded cables and connect the shield to a **SHLD** terminal on an AUX board. Leave the other end of the shield disconnected.
- Do not route I/O signals in the same conduit or wiring trays as high-voltage AC wiring.

Reducing noise on limit (HOM, CW & CCW), trigger (TRG-x), drive fault (DFT), and enable (ENBL) inputs. If you are experiencing noise problems, try adding resistors to reduce noise sensitivity (see illustration below).



Test Setup

⚠ WARNING ⚠

- The test procedures below require you to exercise the I/O connected to your AT6n50 system; therefore, make sure that exercising the I/O will not damage equipment or injure personnel.
- The procedures below are designed to be executed with the drives not connected to the AT6n50; therefore, do not proceed until you have disconnected the drives from the AT6n50.

- **Terminal Emulation**. To communicate with the AT6n50, you will need a terminal emulation program. We recommend you use Motion Architect, a Windows-based program that is included in your ship kit. Motion Architect provides terminal emulation and program editor features as part of its ensemble of programming tools.

  Using Motion Architect:

  1. Power up the AT6n50 AUX board.
  2. To install Motion Architect, insert Disk 1 into your computer's disk drive and run the Setup program (setup.exe). Follow the instructions in the Setup program. **NOTE**: Be sure to install the driver and operating system files for your product. The Setup program will prompt you for the "6000 Driver and Samples" disk and your product's "Operating System and DOS Support Software" disk (you can find these disks in your AT6n50 ship kit).
  3. In the Setup program's last dialog (indicating that Motion Architect has been installed successfully), select "Yes, I want to run Motion Architect now" and click the "Finish" button to launch Motion Architect.
  4. From Motion Architect's main menu, click on the "Product" pull-down menu and click on "Selection" to invoke the "6000 Series Product Selection" dialog box. In the dialog box, select "AT6250" or "AT6450" and click the Okay button.
  5. From Motion Architect's main menu, click on "Terminal" to launch the terminal emulator.
  6. Download the operating system by clicking on the "Transfers" pull-down menu and selecting "Send Operating System". After the operating system is downloaded, the terminal window will display a startup message, followed by a command prompt (>); this indicates that you are communicating with the AT6n50:

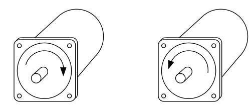
```
* PARKER COMPUMOTOR AT6N50 - N AXIS SERVO CONTROLLER
* ANALOG INPUT OPTION NOT INSTALLED

>
```

- **Verify Factory Settings**. The test procedures below are based on the factory-default active levels for the AT6n50's inputs and outputs. Verify these settings with the following *status* commands:

Command Entered	Response Should Be
INLVL	*INLVLØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ
HOMLVL	*HOMLVLØØØØ
LHLVL	*LHLVLØØØØ_ØØØØ
OUTLVL	*OUTLVLØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ

Connections	Test Procedure	Response Format (left to right)
End-of-travel and Home Limits	**NOTE**: If you are not using end-of-travel limits, issue the Disable Limits (`@LHØ`) command and ignore the first two bits in each response field. 1. Enable the hardware end-of-travel limits with the `@LH3` command. 2. Close the end-of-travel switches and open the home switches. 3. Enter the `TLIM` command. The response should be `*TLIM11Ø_11Ø_11Ø_11Ø`. 4. Open the end-of-travel switches and close the home switches. 5. Enter the `TLIM` command. The response should be `*TLIMØØ1_ØØ1_ØØ1_ØØ1`. 6. Close the end-of-travel switches and open the home switches (return to original config.). 7. Enter the `TLIM` command. The response should be `*TLIM11Ø_11Ø_11Ø_11Ø`.	`TLIM` response format: bit 1 = Axis 1 positive (CW) limit bit 2 = Axis 1 negative (CCW) limit bit 3 = Axis 1 home (HOM) limit bit 4 = Axis 2 positive (CW) limit bit 5 = Axis 2 negative (CCW) limit bit 6 = Axis 2 home (HOM) limit bit 7 = Axis 3 positive (CW) limit bit 8 = Axis 3 negative (CCW) limit bit 9 = Axis 3 home (HOM) limit bit 10 = Axis 4 positive (CW) limit bit 11 = Axis 4 negative (CCW) limit bit 12 = Axis 4 home (HOM) limit "CW" means positive travel. "CCW" means negative travel. "HOM" means home.

Analog Output Signal	1. If the servo drives are connected to the AT6n50's DRIVE connectors, disconnect them now.
	2. Set all gains on all axes to zero by entering these commands: `@SGPØ` <cr>, `@SGIØ` <cr>, `@SGVØ` <cr>, `@SGAFØ` <cr>, and `@SGVFØ` <cr>.
	3. Enter the `@DRIVE1` command to enable the AT6n50 to send out the analog command.
	4. Set the DAC output limit (all axes) to 10 volts by entering the `@DACLIM1Ø` command.
	5. Drive the analog output to the maximum positive range by entering the `@SOFFS1Ø` command.
	6. Enter the `TDAC` command to check the analog output value. The response should be `*TDAC+1Ø,+1Ø,+1Ø,+1Ø`.
	7. Using a Digital Volt Meter (DVM), measure the actual analog output voltage between the CMD+ (analog command) and CMD- (analog command return) terminals on each axis. 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 AT6n50's DAC is not functioning properly.
	8. Repeat steps 5 through 7, using these servo output offset values for step 5:
	`@SOFFS-1Ø`
	`@SOFFSØ`
	`@SOFFS5`
	`@SOFFS-5`
		`TDAC` response □(output voltage):
		±axis 1, ±axis 2, ±axis 3, ±axis 4

Encoder	1. Enter these commands: `L`<cr>, `TPE`<cr>, `T.3`<cr>, and then `LN`<cr>. This will begin a continuous display of the encoders 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): ±encoder1, ±enc1, ±enc3, ±enc4 Direction of rotation:  Clockwise (positive counts) Counter-clockwise (negative counts)
ANI Analog Input Feedback **(ANI option card 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 from the connector on the ANI option card. 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, ±ANI input #3, ±ANI input #4 NOTE ANI feedback is measured in volts
Programmable Inputs (incl. triggers TRG-A through TRG-D)	1. Open the input switches or turn off the device driving the inputs. 2. Enter the `TIN` command. The response should be `*TINØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ`. 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_1111`.	`TIN` response: bits 1-24 = prog. inputs 1-24 bits 25-28 = TRG-A, TRG-B, TRG-C and TRG-D AT6250: TRG-D is not available.
Programmable Outputs (incl. outputs OUT-A through OUT-D)	1. Enter the `OUTALL1,28,1` command to turn on (sink current on) all programmable outputs. Verify that the device(s) connected to the outputs activated properly. 2. Enter the `TOUT` command. The response should be `*TOUT1111_1111_1111_1111_1111_1111_1111`. 3. Enter the `OUTALL1,28,Ø` command to turn off all programmable outputs. Verify that the device(s) connected to the outputs de-activated properly. 4. Enter the `TOUT` command. The response should be `*TOUTØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ_ØØØØ`.	`TOUT` response: bits 1-24 = prog. outputs 1-24 bits 25-28 = OUT-A, OUT-B, OUT-C, and OUT-D AT6250: OUT-C is used as a general-purpose output only, not for the Output On Position feature. OUT-D is not available.
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 `*TINOØØØØ_ØØØØ`. 3. Close the ENBL switch, and close the joystick switches or turn on the device. 4. Enter the `TINO` command. The response should be `*TINO1111_11ØØ`.	`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 (ENBL) input bits 7 & 8 are not used

Tuning the AT6n50

Before tuning the AT6n50, mount and couple the motors as required for your application.

To assure optimum performance, you should tune your servo system. The goal of the tuning process is to define the gain settings, servo performance, and feedback setup (see command list below) that you can incorporate into your application program. (Typically, these commands are placed into a setup program – see examples in the Basic Operations Setup chapter of the *6000 Series Programmer's Guide*).

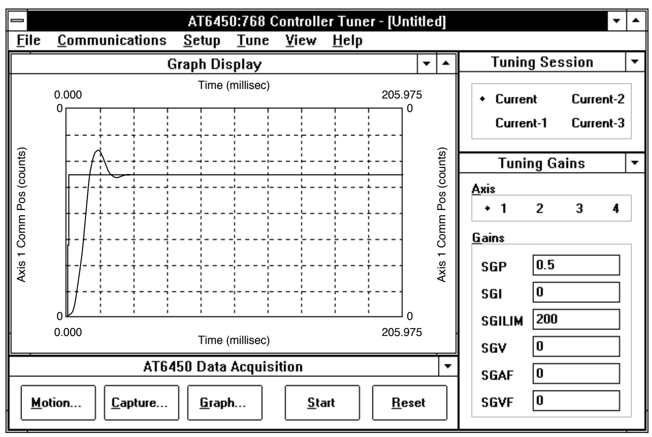
If you are using the Servo Tuner package (see note below), refer to the *Servo Tuner User Guide* for tuning instructions. If you are not using Servo Tuner, refer to the tuning instructions in Appendix A (page 33).

Servo Tuning Software Available

To effectively tune the AT6n50 controller (and any velocity drives you may be using), use the interactive tuning features in the Servo Tuner™. It greatly improves your efficiency and gives you powerful graphical tools to measure the performance of the system.

Servo Tuner is included as an integral element of Motion Builder™, an optional icon-based programming tool. Servo Tuner is also available as an optional add-on module to Motion Architect (it does not automatically come with the basic Motion Architect software package). Instructions for using Servo Tuner are provided in the Servo Tuner User Guide and in Motion Builder's online Help system and Motion Builder Startup Guide & Tutorial.

To order Motion Builder or the Servo Tuner add-on module to Motion Architect, contact your local Automation Technology Center (ATC) or distributor.



Tuning-Related Commands (see 6000 Series Software Reference or the Servo Tuner User Guide for details)

Tuning Gains:

- `SGP` .... Sets the proportional gain in the **P**IV&F servo algorithm.
- `SGI` .... Sets the integral gain in the P**I**V&F servo algorithm.
- `SGV` .... Sets the velocity gain in the PI**V**&F servo algorithm.
- `SGAF` .... Sets the acceleration feedforward gain in the PIV&$\mathbf{F_a}$ algorithm.
- `SGVF` .... Sets the velocity feedforward gain in the PIV&$\mathbf{F_v}$ algorithm.
- `SGILIM` .... 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.
- `SGENB` .... 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.
- `SGSET` .... Saves the presently-defined set of PIV&F gains as a 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.

Servo Performance:

- `INDAX` .... Selects the number of available axes to use.
- `SSFR` .... Sets the ratio between the update rate of the move trajectory and the update rate of the servo action. Affects the servo sampling update, the motion trajectory update, and the system update.

Feedback Setup:

- `SFB` .... Selects the servo feedback device. Options are encoder or ANI input. IMPORTANT: Parameters for scaling, tuning gains, max. position error (`SMPER`), and position offset (`PSET`) are specific to the feedback device selected (with the `SFB` command) at the time the parameters are entered.
- `ERES` .... Encoder resolution.
- `SMPER` .... Sets the maximum allowable error between the commanded position and the actual position as measured by the feedback device. If the error exceeds this limit, the controller activates the Shutdown output and sets the DAC output to zero (plus any `SOFFS` offset). If there is no offset, the motor will freewheel to a stop. You can enable the `ERROR` command to continually check for this error condition (`ERROR.12-1`), and when it occurs to branch to a programmed response defined in the `ERRORP` program.

What's Next?

By now, you should have completed the following tasks, as instructed earlier in this chapter:

1. Review the general specifications — see page 3.
2. Perform configuration/adjustments, if necessary — see page 4.
3. Install the AT6n50 PC card (and the ANI option card if you ordered one) — see page 5.
4. Mount the AT6n50 AUX board — see page 6.
5. Connect all electrical system components — see pages 7-21.
6. Test the installation — see pages 22-23.
7. Mount the motor and couple the load.
8. Tune the AT6n50 (see *Servo Tuner User Guide* or Appendix A for instructions).

Program Your Motion Control Functions

You should now be ready to program your AT6n50 for your application. Knowing your system's motion control requirements, refer now to the *6000 Series Programmer's Guide* for descriptions of the AT6n50's software features and instructions on how to implement them in your application. Be sure to keep the *6000 Series Software Reference* at hand as a reference for the 6000 Series command descriptions.

For assistance with your programming effort, we recommend that you use the programming tools provided in Motion Architect for Windows (found in your ship kit). Additional powerful programming and product interface tools are available (see below).

Motion Architect

Motion Architect® is a Microsoft® Windows™ based 6000 product programming tool (included in your ship kit). Motion Architect provides these features:

- **System configurator and code generator**: Automatically generate controller code for basic system set-up parameters (I/O definitions, feedback device operations, etc.).
- **Program editor**: Create blocks or lines of 6000 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 AT6n50, you can type in and execute controller code, transfer code files to and from the AT6n50.
- **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 technical references**: These on-line resources provide help information about Motion Architect, as well as access to hypertext versions of the *6000 Series Software Reference* and the *6000 Series Programmer's Guide*.

Other Software Tools Available

To Order these software packages, contact your local Automation Technology Center (ATC) or distributor.

Motion Builder™. A Windows-based iconic programming interface that removes the requirement to learn the 6000 programming language.

CompuCAM™. A CAD-to-Motion (CAM) program that allows you to easily translate DXF, HP-GL, and G-Code files into 6000 Series Language motion programs. Windows environment.

DDE6000™. Facilitates data exchange between the AT6n50 and Windows™ applications that support the dynamic data exchange (DDE) protocol. NetDDE™ compatible.

Motion OCX Toolkit™. Motion OCX Toolkit provides 32-bit Ole Custom Controls (OCXs) designed to run under Windows 95 or Windows NT.

Motion Toolbox™. A library of LabVIEW® virtual instruments (VIs) for programming and monitoring the AT6n50. Available for the Windows environment.

CHAPTER TWO

2 Troubleshooting

IN THIS CHAPTER
• Troubleshooting basics: - Reducing electrical noise - Diagnostic LEDs - Test options - Technical support • Solutions to common problems • Corrective actions in response to operating system download errors • Product return procedure

Troubleshooting Basics

When 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? Random events may appear to be related, but they are not necessarily contributing factors to your problem. You may be experiencing more than one problem. You must isolate and solve one problem at a time.

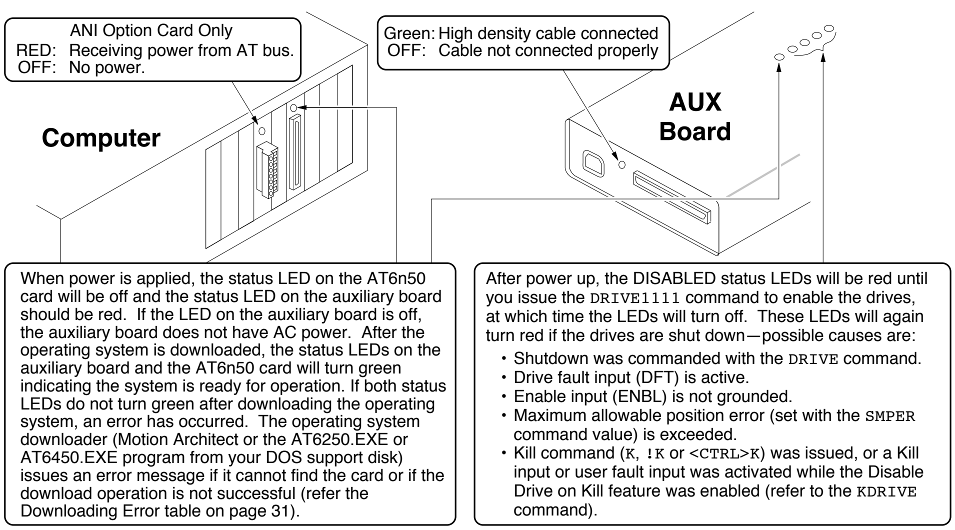
Log (document) all testing and problem isolation procedures. Also, if you are having difficulty isolating a problem, be sure to document all occurrences of the problem along with as much specific information as possible. You may need to review and consult these notes later. This will also prevent you from duplicating your testing efforts.

Once you isolate the problem, refer to the problem solutions contained in this chapter. If the problem persists, contact your local technical support resource (see *Technical Support* below).

Reducing Electrical Noise

Refer to the guidelines on page 21. General information on reducing electrical noise can be found in the Engineering Reference section of the Parker Compumotor catalog.

Diagnostic LEDs



Test Options

- **Test Panel**. Motion Architect's Panel Module allows you to set up displays for testing system I/O and operating parameters. Motion Architect is provided in your ship kit.
- **Hardware Test Procedure** (see pages 22-23).

Technical Support

If you cannot solve your system problems using this documentation, contact your local Automation Technology Center (ATC) or distributor for assistance. If you need to talk to our in-house application engineers, please contact us at the numbers listed on the inside cover of this manual. (These numbers are also provided when you issue the `HELP` command.)
NOTE: Compumotor maintains a BBS and website that contains the latest software upgrades and late-breaking product documentation, a FaxBack system, and a tech support email address.

Common Problems & Solutions

> **NOTE**
> Some software-related causes are provided because it is sometimes difficult to identify a problem as either hardware or software related.

Problem	Cause	Solution
Communication errors.	1. Communication program looking for card at wrong address. 2. Address conflict. 3. AT6n50 card not properly seated.	1. Select correct address for communication program. 2. See Address Selection section below. 3. Seat board properly in slot. Apply pressure directly over area with gold card edge fingers.
Computer will not boot with AT6n50 installed.	1. Interrupt conflict. 2. See problem: Communication Errors.	1.a. Turn interrupt DIP switches OFF. 1.b. See Interrupt Selection section below.
Direction is reversed. (stable servo response)	1. Command output (CMD) connections and feedback device connections or mounting are reversed.	1. Switch CMD- with the CMD+ connection to the drive (if your 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.
Direction is reversed. (unstable servo response)	1. Not tuned properly. 2. Phase of encoder reversed or mounting of ANI input is such that it counts in the opposite direction as the commanded direction.	1. Refer to the tuning instructions in the Servo Tuner User Guide or in Appendix A. 2. If encoder feedback, swap the A+ and A- connections to the AT6n50. If ANI feedback, change the mounting so that the counting direction is reversed.
Distance, velocity, and accel are incorrect as programmed.	1. Incorrect encoder resolution setting.	1. Match the `ERES` command setting (default `ERES` setting is 4,000 counts/rev) to match the post-quadrature resolution of the encoder. `ERES` values for Compumotor encoders: E Series: `ERES4000` SM Series Servo Motors: SMxxxxD-xxxx: `ERES2000` SMxxxxE-xxxx: `ERES4000` OEM Series motors (servo): OEM2300E05A-MO: `ERES2000` OEM2303E05A-MO: `ERES2000` OEM3400E05A-MO: `ERES2000` OEM3401E10A-MO: `ERES2000` OEM2300E05A-MO: `ERES4000` OEM2303E10A-MO: `ERES4000` OEM3400E10A-MO: `ERES4000` OEM3401E10A-MO: `ERES4000` OEM2300E20A-MO: `ERES8000` OEM2303E20A-MO: `ERES8000` OEM3400E20A-MO: `ERES8000` OEM3401E20A-MO: `ERES8000`
Encoder counts missing.	1. Improper wiring. 2. Encoder slipping. 3. Encoder too hot. 4. Electrical noise. 5. Encoder frequency too high.	1. Check wiring. 2. Check and tighten encoder coupling. 3. Reduce encoder temperature with heatsink, thermal insulator, etc. 4a. Shield wiring. 4b. Use encoder with differential outputs. 5. Peak encoder frequency must be below 1.6MHz post-quadrature. Peak frequency must account for velocity ripple.
Erratic operation.	1. Electrical Noise and/or improper shielding. 2. Improper wiring.	1.a. Reduce electrical noise or move AT6n50 AUX board away from noise source. 1.b.. Refer to the Electrical Noise portion of the Technical Reference section in the Compumotor catalog. 2. Check wiring for opens, shorts, & mis-wired connections.
LEDs	See Diagnostic LEDs above (page 28)	
Motion does not occur.	1. Check LEDs. 2. End-of-travel limits are active. 3. ENBL (enable) input not grounded. 4. Drive fault detected. 5. Improper wiring. 6. Load is jammed. 7. No torque from motor. 8. Max. allowable position error (`SMPER` value) exceeded.	1. See Diagnostic LEDs above. 2.a. Hardware limit switches: Move load off of limits or disable limits by sending the `LHØ` command to the appropriate axis. 2.b. Software limits: Set `LSPOS` to a value greater than `LSNEG`. 3. Ground the ENBL connection. 4.a. Check status with `TASXF` command (see bit #4). 4.b. Verify correct drive fault level setting (`DRFLVL` command value). 5. Check command (CMD), shutdown (SHTNC or SHTNO), drive fault (DFT), and end-of-travel limit connections. 6. Remove power and clear jam. 7. See problem: Torque, loss of. 8. Check status with `TASF` report (see bit #23), and issue the `DRIVE1` command to the affected axis.

Problem/Cause/Solution Table (continued)

Problem	Cause	Solution
Motion does not occur 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.
Mouse stops working or serial ports affected (after AT6n50 is installed).	1. Interrupt conflict. 2. Address conflict.	1. See Interrupt Selection section below. 2. See Address Selection section below.
Operating system will not download, or download stops part way through.	1. Address conflict. 2. Download error.	1. See Address Selection section below. 2. See Downloading Errors table below.
Programmable inputs not working.	1. IN-P (input pull-up) on AUX board not connected to a power supply. 2. If external power supply is used, the grounds must be connected together. 3. Improper wiring.	1a. When inputs will be pulled down to 0V by an external device, connect IN-P to the +5V terminal or to an external 5-24V supply (<u>but not to both</u>). 1b. When inputs will be pulled to 5-24V or higher by an external device, connect IN-P to 0V. 2. Connect external power supply's ground to AT6n50's 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 (output-pull-up) on AUX board not connected to a 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 the +5V supply on the AUX board or to an external supply of up to 24V (but not to both). 3. Connect the external power supply's ground to the AT6n50's ground (GND). 4. Check wiring for opens, shorts, and mis-wired connections.
Runaway (if encoder counts positive when turned clockwise).	1. Direction connections reversed. 2. Improper tuning.	1. Switch CMD– with the CMD+ connection to drive. 2. Retune the AT6n50 and/or the drive. Refer to the tuning instructions in the Servo Tuner User Guide or in Appendix A.
Torque, loss of.	1. Improper wiring. 2. No power. 3. Drive failed. 4. Drive shutdown.	1. Check wiring to the motor, as well as other system wiring. 2. Check power connection and STATUS LEDs (see Diagnostic LEDs). 3.a. Check the drive fault `TASXF` report (see bit #4). 3.b. Check the drive condition. 4. Enable drive by sending the `DRIVE1` command to the affected axis.

Interrupt & Address Selection

CAUTION

Refer to your PC-AT's documentation to avoid interrupt and address conflicts, and the resulting possibility of system damage.

If you are using Motion Architect and you change these settings, be sure to configure the same address and interrupt settings in Motion Architect's Terminal or Panel modules.

Interrupts

The AT6n50 is factory configured with no interrupts selected. The AT6n50 does not need interrupts to function properly. If you want to use interrupts, select an interrupt in your system that is not already used in the system (refer to instructions on page 4).

Interrupts that are unassigned on the AT bus are IRQ10, 11,12, and 15. If serial port COM1 is not used, IRQ4 may be available. If serial port COM2 is not used, IRQ3 may be available. IRQ5 and 7 are defined for parallel printer ports and one or both of these may be available if your system has one or no parallel printer ports.

Address

The AT6n50 is factory-set to address 300H. If another card in your system uses address 300H (default setting), select a different address (refer to instructions on page 4).

If you are unsure which addresses are used, the following address are likely to be available: 308H, 310H, 318H, 380H, 388H, 3A0H, or 3A8H.

Downloading Errors (downloading the operating system)

> **NOTE**
> These are the potential error messages you could receive while trying to download the AT6n50 operating system using a terminal in Motion Architect, or using the AT6250.EXE or AT6450.EXE programs installed from the "AT6n50 Operating System and DOS Support" diskette found in the product ship kit.

Error	Description	Reason/Corrective Action
1	Operating System File Not Found	The operating system specified, or the default operating system (if unspecified) could not be found by the operating system downloader. Put the `AT6n50.OPS` file in the Windows\System directory.
2	Invalid Operating System File	The operating system specified, or the default operating system (if unspecified) is not a valid operating system or is corrupted. Re-install the operating system from the original "Operating System and DOS Support" disk provided in your ship kit.
3	Unexpected EOF	An EOF (end-of-file) character was received during the download. Re-install the operating system from the original "Operating System and DOS Support" disk provided in your ship kit.
4	Invalid Port Address	The port address specified while downloading is invalid. Use another address setting (768 ≤ port ≤ 1024 in increments of 8). See page 4 for address options.
5	Unknown Option	An unknown option was specified on the `AT6n50.EXE` command line.
6	Base Port Address Greater than 1024	The base port address is too high. Specify an address between 768 and 1024 decimal (use the `/PORT=` parameter if using the AT6n50.EXE, or use the Settings menu if using Motion Architect).
7	Base Port Address Less than 255	The base port address is too low. Specify an address between 768 and 1024 decimal (use the `/PORT=` parameter if using the AT6n50.EXE, or use the Settings menu if using Motion Architect).
8	Base Port Address Not a Multiple of 8	The base port address is not a multiple of 8. Specify a valid address (use the `/PORT=` parameter if using the AT6n50.EXE, or use the Settings menu if using Motion Architect).
9	Modified Download Requested	A partial download was requested on the command line.
10	Card Controller Error	The card controller did not respond as expected. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address.
11	Card Not found	The card did not respond as expected. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address).
12	Reading Card Rev	The card appeared to be working as expected until the revision was requested. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address (see page 4).
13	Waiting for Data Ready	The card did not respond when expected. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address.
14	Purging Data Out Buffer	The card output buffer could not be emptied. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address.
15	Waiting for Data Input Buffer Empty	The card did not respond to the data sent to it. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address.
16	Time-out Waiting for Processor Startup	The card did not respond as expected. The green LED on the back of the PC-card should be on for this error to occur. Verify that you are downloading to the correct address. Make sure there are no other peripheral cards (network adapters, bus mouse, etc.) at the same address. Try changing the card address. Use a fresh copy of the operating system from the disk that was shipped with the card. If the green LED on the back of the card flashes briefly during download of the operating system, the card may need repair.
17	CRC Error	The CRC value calculated during download is not the same as stored with the operating system. Either the file is corrupted on disk, or was corrupted during download. Try a fresh copy of the operating system. If your computer has a Turbo switch, switch it to low speed because some computers violate ISA bus timing specifications at high speed.
18	Operating System Rev not Compatible with Loader Rev	The operating system being downloaded is not compatible with the downloader (Motion Architect or AT6n50.EXE). Install the latest downloader from the disks in your AT6n50 ship kit.
19	Incompatible Card ROM rev	The card ROMS and the operating system downloader (Motion Architect or AT6n50.EXE) are incompatible. If you are using a new downloader, obtain a new set of ROMS from the factory.
20	Card Read Error (bad compare)	The downloader is unable to communicate reliably with the card. Try switching to 8-bit mode on the card, switching out of Turbo mode on your PC, or a different address.
21	Card Read Error (outbuf)	The downloader is unable to empty the output buffer. There may be an address conflict with another board. Try a different address.
22	Card ROMS - Command Line Parameter Passing Not Supported	The card ROMS are an old revision that do not support command line arguments. Obtain a ROM update from the factory.
23	Card ROMS - Unsupported Option Requested	The card ROMS do not support the option specified on the command line. Obtain a ROM update from the factory.
24	NULL Error	

Product Return Procedure

Step 1 Obtain the serial number and the model number of the defective unit, and secure a purchase order number to cover repair costs in the event the unit is determined by the manufacturers to be out of warranty.

Step 2 Before you return the unit, have someone from your organization with a technical understanding of the AT6n50 system and its application include answers to the following questions:

- What is the extent of the failure/reason for return?
- How long did it operate?
- Did any other items fail at the same time?
- What was happening when the unit failed (e.g., installing the unit, cycling power, starting other equipment, etc.)?
- How was the product configured (in detail)?
- Which, if any, cables were modified and how?
- With what equipment is the unit interfaced?
- What was the application?
- What was the system environment (temperature, enclosure, spacing, contaminants, etc.)?
- What upgrades, if any, are required (hardware, software, user guide)?

Step 3 Call for return authorization. Refer to the *Technical Assistance* phone numbers provided on the inside front cover of this document. The support personnel will also provide shipping guidelines.

Appendix A

Tuning

In this appendix:

- Servo control terminology
- Servo control techniques
- Servo tuning procedures
  (These procedures are based on empirical techniques. If you are using Servo Tuner™, refer to the Servo Tuner User Guide for instructions.)

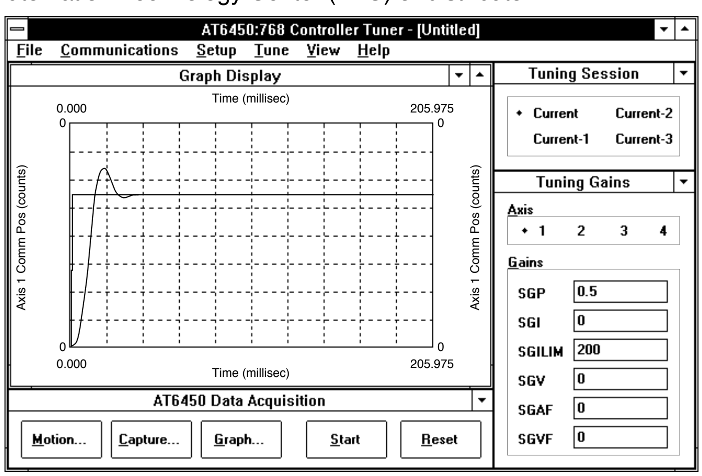
You should tune the AT6n50 before attempting to execute any motion functions. At a minimum, complete this chapter's Tuning Setup Procedure and Controller Tuning Procedures until you have found a proportional feedback gain that can give a stable response for your system. (The Drive Tuning Procedure below is for use with velocity drive systems only.) Then you can proceed to execute your motion functions. To gain a full understanding of tuning, you should read through this entire appendix and follow its procedures to ensure your system is properly tuned.

Servo Tuning Software Available

To effectively tune the AT6n50 (and any velocity drives you may be using), use the interactive tuning features in the Servo Tuner™. It greatly improves your efficiency and gives you powerful graphical tools to measure the performance of the system.

Servo Tuner is included as an integral element of Motion Builder™, an optional icon-based programming tool. Servo Tuner is also available as an optional add-on module to Motion Architect (it does not automatically come with the basic Motion Architect software package). Instructions for using Servo Tuner are provided in the Servo Tuner User Guide and in Motion Builder's online Help system and Motion Builder Startup Guide & Tutorial.

To order Motion Builder or the Servo Tuner add-on module to Motion Architect, contact your local Automation Technology Center (ATC) or distributor.



Servo System Terminology

This section gives you an overall understanding of the principles and the terminology used in tuning your AT6n50.

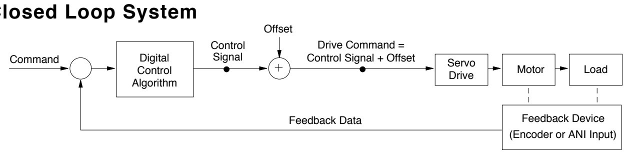
Servo Tuning Terminology

The AT6n50 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 appendix). **Tuning** is the process of selecting and adjusting these gains to achieve optimal servo performance.

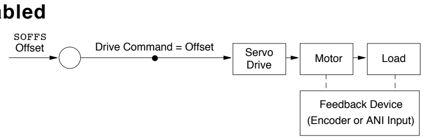
When this control algorithm is used, the whole servo system is a **closed-loop** system (see diagram below). It is called closed loop because the control algorithm accounts for both the **command** (position, velocity, tension, etc.) and the **feedback data** (from the encoder or ANI input); therefore,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

When all gains are set to zero, the digital control algorithm is disabled. During system setup or troubleshooting, it is desirable to disable the algorithm (by setting all the gains to zero) and use the `SOFFS` command to directly control the analog output.

Closed Loop System



Servo Algorithm Disabled



The controller has the capability of providing an analog voltage output of ±10V for commanding the 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. 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.

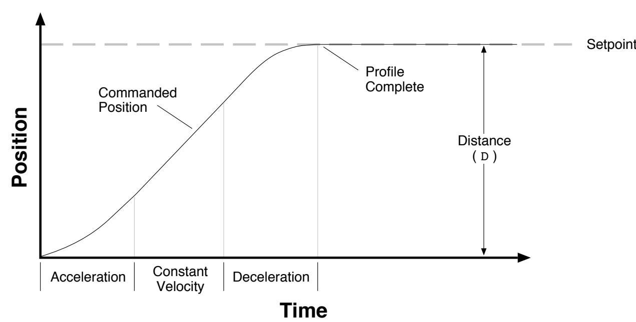
Position Variable Terminology

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 appendix refers to the *commanded position*, it means this calculated time-varying commanded position, not the distance (`D`) command. Conversely, when this appendix 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) measured with the feedback device (encoder or ANI input). Since this is the position achieved when the drive 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 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 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 motor'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 motor 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.

Servo Response Terminology

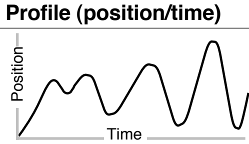
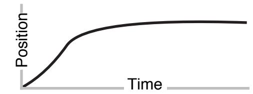
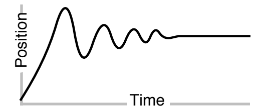
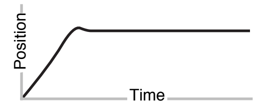
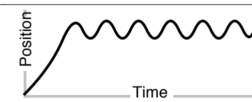
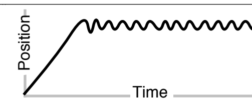
Stability

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.	

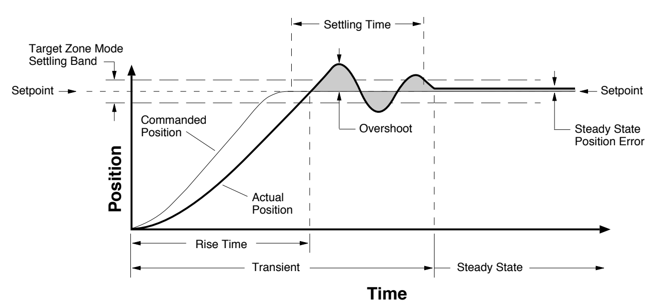
Performance Measurements

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 motor 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.



Tuning-Related Commands

More detailed information on each 6000 Series command can be found in the *6000 Series Software Reference*.

Tuning Gains:

`SGP` .... Sets the proportional gain in the **P**IV&F servo algorithm.

`SGI` .... Sets the integral gain in the P**I**V&F servo algorithm.

`SGV` .... Sets the velocity gain in the PI**V**&F servo algorithm.

`SGAF` .... Sets the acceleration feedforward gain in the PIV&$\mathbf{F_a}$ algorithm.

`SGVF` .... Sets the velocity feedforward gain in the PIV&$\mathbf{F_v}$ algorithm.

`SGILIM` .... 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.

`SGENB` .... 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.

`SGSET` .... Saves the presently-defined set of PIV&F gains as a 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.

Servo Performance:

`INDAX` .... Selects the number of available axes to use.

`SSFR` .... Sets the ratio between the update rate of the move trajectory and the update rate of the servo action. Affects the servo sampling update, the motion trajectory update, and the system update.

Feedback Setup:

`SFB` .... Selects the servo feedback device. Options (depending on the product) are: encoder or ANI input. IMPORTANT: Parameters for scaling, tuning gains, max. position error (`SMPER`), and position offset (`PSET`) are specific to the feedback device selected (with the `SFB` command) at the time the parameters are entered.

`ERES` .... Encoder resolution.

`SMPER` .... Sets the maximum allowable error between the commanded position and the actual position as measured by the feedback device. If the error exceeds this limit, the controller activates the Shutdown output and sets the DAC output to zero (plus any `SOFFS` offset). If there is no offset, the motor will freewheel to a stop. You can enable the `ERROR` command to continually check for this error condition (`ERROR.12-1`), and when it occurs to branch to a programmed response defined in the `ERRORP` program.

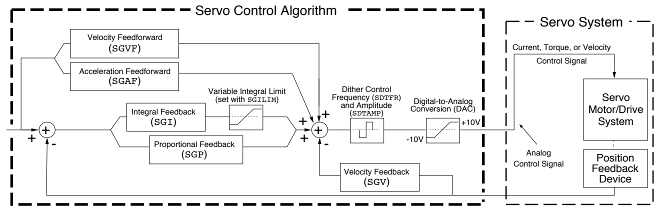
Servo Control Techniques

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

The AT6n50 employs a *PIV&F* servo control algorithm. The control techniques available in this system are:

P ..... Proportional Feedback (control with `SGP` command)
I ..... Integral Feedback (control with `SGI` command)
V .... Velocity Feedback (control with `SGV` command)
F ..... Velocity and Acceleration Feedforward (control with the `SGVF` and `SGAF` 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`)	Improve	Improve	Improve	Improve	Improve
Integral (`SGI`)	Degrade	Degrade	Improve	Improve	Improve
Velocity Feedback (`SGV`)	Improve	Improve	----------	----------	Degrade
Velocity Feedforward (`SGVF`)	----------	----------	----------	----------	Improve
Acceleration Feedforward (`SGAF`)	----------	----------	----------	----------	Improve

Proportional Feedback Control (`SGP`)

Proportional feedback **is the most important feedback for stabilizing a servo system.** When the controller 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.

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

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.

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.*

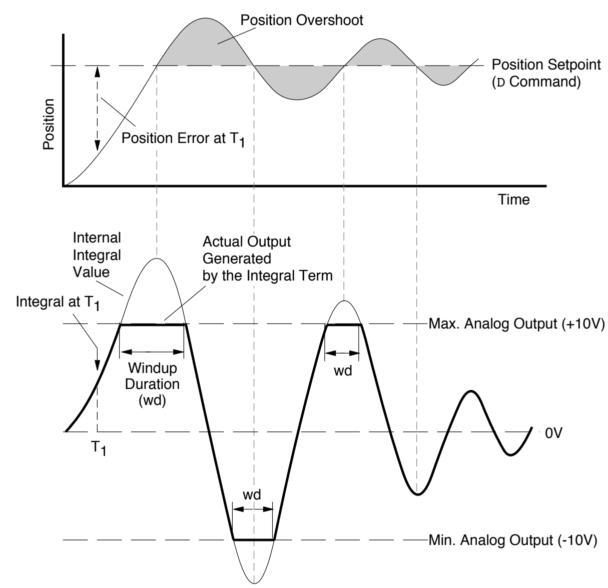
In the task of reducing position error, the integral gain (SGI) works differently than the proportional gain (SGP); this is because the magnitude of its control signal is not dependent on the magnitude of the position error as in the case of proportional feedback. If any position error persists, then 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

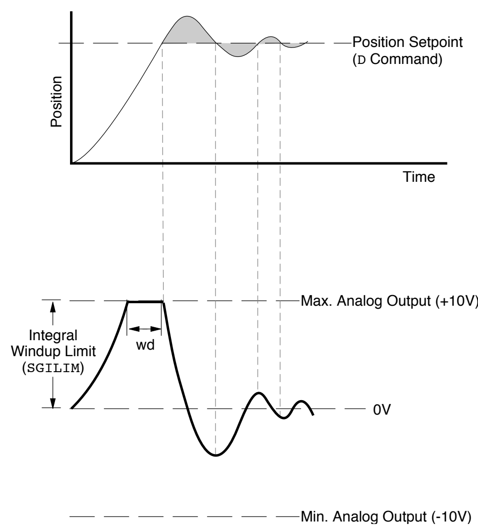
If integral control (SGI) is used and an appreciable position error has persisted long enough during the transient period (time taken to reach the setpoint), the control signal generated by the integral action can end up too high and saturate to the maximum level of the controller's analog control signal 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 5 of the *Controller Tuning Procedure* below.

Without SGILIM



With SGILIM



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 controller essentially anticipates the commanded position and initiates a control signal ahead of time to more closely follow (*track*) the commanded position.

Applications requiring contouring or 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 controller's analog control signal range of ±10V).

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 controller essentially anticipates the velocity of the commanded position and initiates a control signal ahead of time to more closely follow (*track*) the commanded position.

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.

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 offset (SOFFS), and maximum position error (SMPER) setting, to a specified gain set (see list below).

SGP..........Proportional Gain
SGV..........Velocity Gain
SGI..........Integral Gain
SGVF........Velocity Feedforward Gain
SGAF........Acceleration Feedforward Gain
SGILIM.....Integral Windup Limit
SOFFS......Servo Control Signal Offset
SMPER......Maximum Allowable Position Error

The gain set saved with the SGSET command can be enabled/recalled later with the SGENB 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).

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.

Tuning Setup Procedure

Use the following procedure to set up your servo system before completing the tuning procedures. You can perform this procedure for all 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 test procedures provided in Chapter 1:
>
> - Connect the drive (especially the drive's shutdown output).
> - Connect and test the feedback devices.
> - Connect and test the end-of-travel limits.
> - Test the AT6n50'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 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 AT6n50's analog output signal. An alternative is to issue the `@DRIVEØ` command to the AT6n50 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 drive's power supply.

Step 1

Remove power to the drive.

Step 2

Apply power to the AT6n50 only and issue the `DRIVE1111` command. Measure the AT6n50'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:

```
SGP0,0   ; Set proportional feedback gain to zero
SGV0,0   ; Set velocity feedback gain to zero
SGI0,0   ; Set integral feedback gain to zero
SGVF0,0  ; Set velocity feedforward gain to zero
SGAF0,0  ; Set acceleration feedforward gain to
         ; zero
```

Step 5

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.*

Step 6

Observe the AT6n50'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.

If the noise level is acceptable, proceed to Step 7. If the noise level is too high, refer to the guidelines on page 2.

Step 7

The purpose of this step is to ensure that a positive voltage on the AT6n50'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 rev (e.g., if using 1000-line encoders and no scaling, use the `@SMPER4ØØØ` command [same value as the `ERES` command]).

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 motor move slowly in the positive (clockwise) 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 AT6n50 and the drive and swap the **CMD+** and **CMD-** connections either at the AT6n50 or at the drive, whichever is more accessible (this will not work for servo drives that do not accept differential input). Then turn on the AT6n50 again, enter the DRIVE1111 command, and repeat Steps 4 through 7.d. before proceeding to Step 8.

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 AT6n50's analog output with the DVM to determine if there is any offset caused by the electrical interconnections between the AT6n50 and the drive.

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.

Step 9

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, 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 AT6n50.
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 AT6n50:

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 AT6n50's output voltage to its maximum level, 10.0 volts (SOFFS1Ø for axis 1, or SOFFS,1Ø for axis 2, etc.).

c. Adjust the drive gain factor (sometimes called the *tach gain*) such that when the AT6n50'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 AT6n50'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 preset values selectable with jumpers or DIP switches.

Step 2

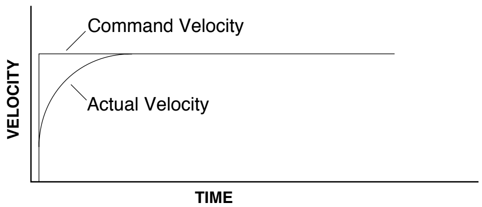
Tune the drive (iteratively) to achieve the desired response:

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

```
DEF STEPS ; Begin program definition for STEPS
@SGP0     ; Set the SGP gain to zero
@SGI0     ; Set the SGI gain to zero
@SGV0     ; Set the SGV gain to zero
@SGAF0    ; Set the SGAF gain to zero
@SGVF0    ; Set the SGVF gain to zero
@SMPER0   ; Disable checking the maximum
          ; allowable position error
@SOFFS0.5 ; Set command output to 0.5 volts
T1        ; Wait for 1 second
@SOFFS0   ; Set command output to zero volts
          ; (stopping the motor)
@SMPER1   ; Re-enable checking the maximum
          ; allowable position error
END       ; End definition of the program
STEPS     ; Execute the program called STEPS
          ; (the motor will move for 1 second
          ; and then stop)
```

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 AT6n50.

Controller Tuning Procedure

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

1. Setup up for tuning.
2. Select the AT6n50'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 AT6n50:

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).

NOTE: The default setting (SSFR4) is adequate for most applications.

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.)

SELECTING THE SSFR VALUE

Refer to the SSFR command description in the 6000 Series Software Reference. Use the table to determine the appropriate setting based on your desired servo update rates.

As an example, if your application requires a closed-loop bandwidth of 450 Hz, and you 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 3600 Hz. If two axes are running (INDAX2), then you should try using the SSFR4 setting.

For more in-depth discussion on the different update parameters (servo, motion and system), refer to the SSFR command description in the *6000 Series Software Reference*.

CAUTION

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.

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 on next page for tuning process*).

a. Enter the following commands to create a step input profile *(use a comma in the data fields for the axes you are not tuning—e.g.,* D,1ØØ *for tuning axis 2)*:

```
A999    ; Set accel to 999 units/sec/sec
AD999   ; Set decel to 999 units/sec/sec
V30     ; Set velocity to 30 units/sec
D100    ; Set distance to 100 units
```

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

c. Send the GO1 command to the axis that is being tuned at the time (e.g., GO,1 for axis 2).

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 (e.g., SGV,,Ø.1 for axis 3).

f. Repeat Step 4.c.

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.*

h. The flow diagram (next page) 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. (Refer to the *Tuning Scenario* section later in this chapter for a case example.) 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.

Step 5

Use the Integral Feedback Gain (SGI) to reduce steady state error. (Steady state position error is described earlier in the *Performance Measurements* section on page 35.)

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.

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**. The use of the Target Zone Settling Mode (STRGTE) is recommended.

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.

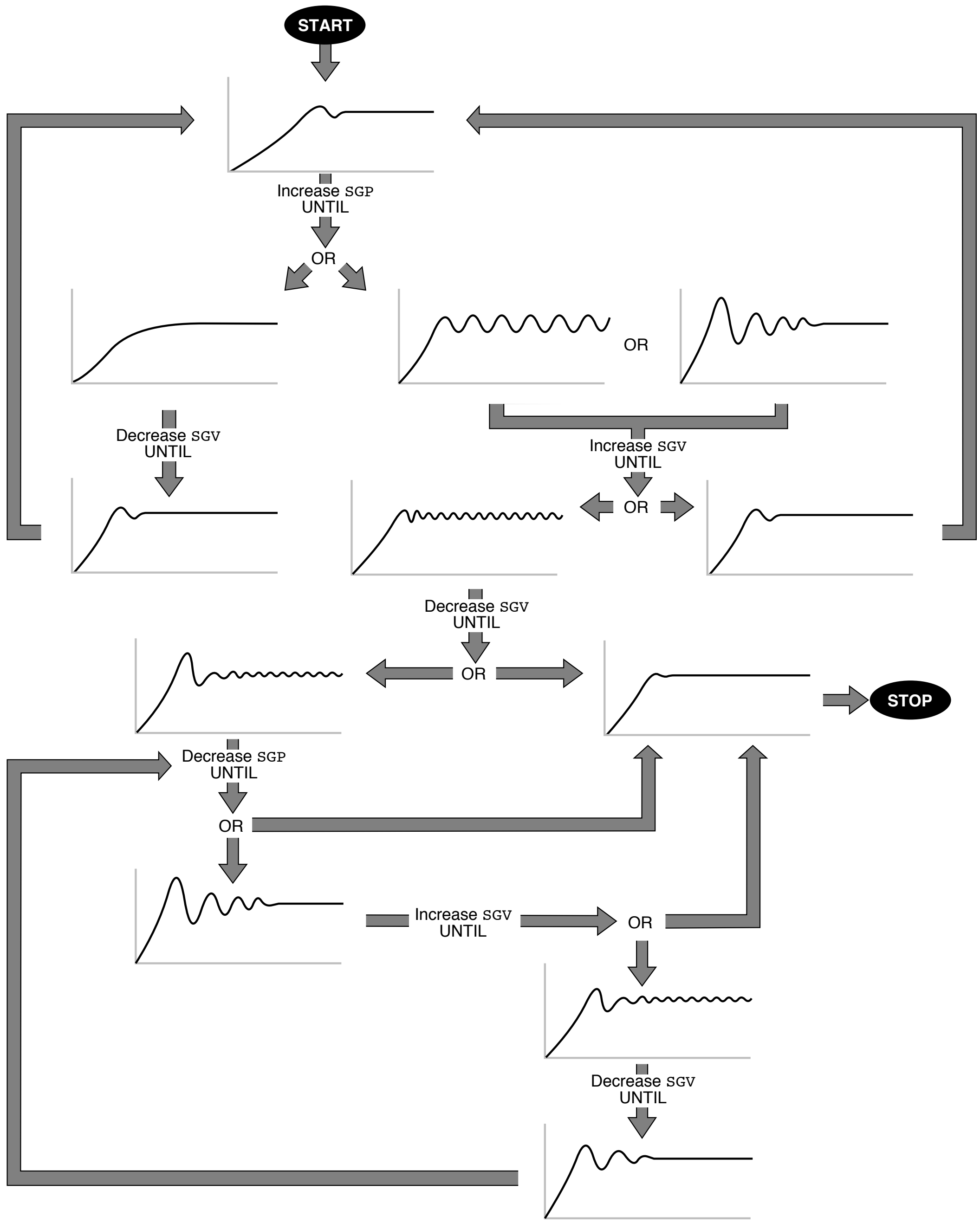
d. There are three things you can do at this point (If these three things do not work, that means the integral gain is too high and you have to lower it.):

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

2nd Check whether the AT6n50'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.*

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).

Tuning Process Flow Diagram (using proportional and velocity gains)



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).

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*.

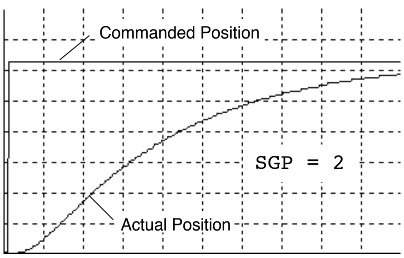
> **NOTE**
> The steps shown below (steps 1 - 11) represent the major steps of the process; the actual progression between these steps usually requires several iterations.

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.

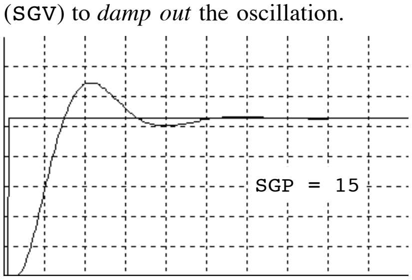
In the next step, we should increase SGP until the response is slightly underdamped.



Step 2

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

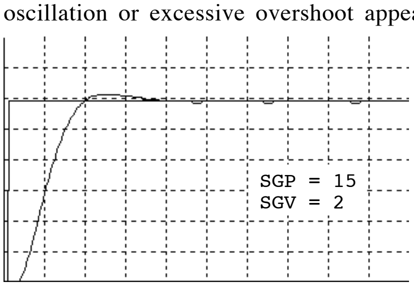
Therefore, we should introduce the velocity feedback gain (SGV) to *damp out* the oscillation.



Step 3

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

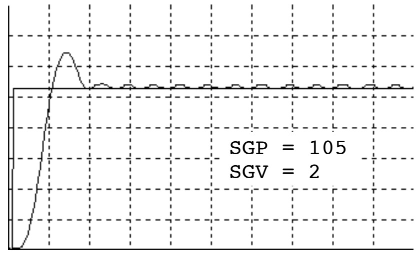
At this point, the SGP should be raised again until oscillation or excessive overshoot appears.



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.

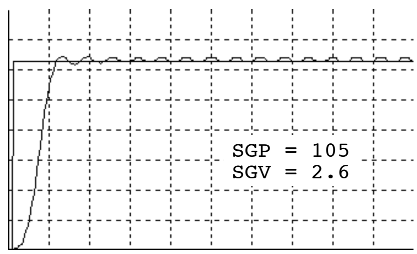
Next, we should try raising the SGV gain to see if it could damp out the overshoot and chattering.



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.

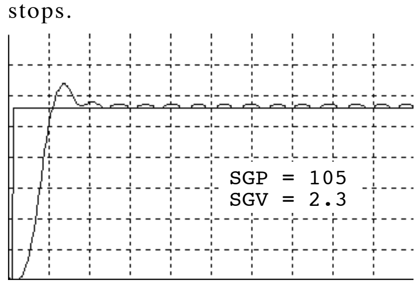
The next step should be to lower the SGV gain first.



Step 6

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

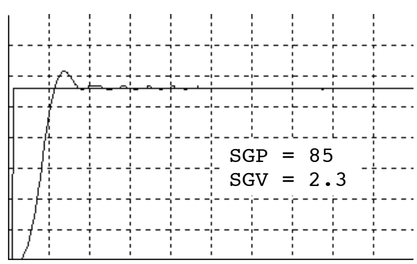
Therefore, we should lower the SGP gain until chattering stops.



Step 7

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

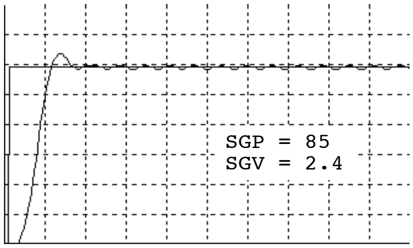
The next step should be to try raising the SGV to damp out the overshoot.



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.

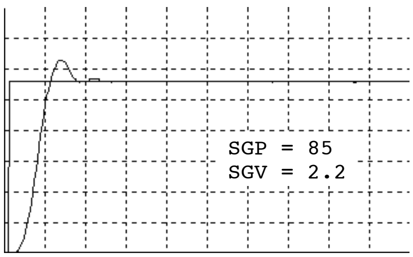
Next, we should lower the SGV gain until chattering stops.



Step 9

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

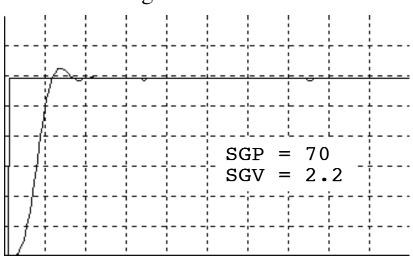
Next we should lower the SGP gain.



Step 10

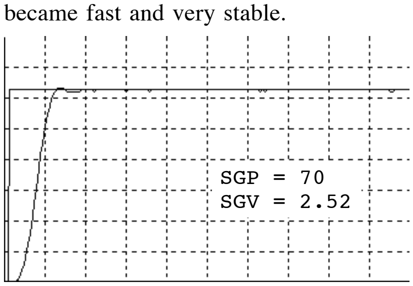
Overshoot is reduced very little after lowering the SGP gain to 70. (The SGV gain might have been lowered too much in Step 9.)

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.



Step 12

Now that we have determined the appropriate SGP and SGV gains, we can include them in the AT6n50's setup program. We put the gains in the setup program (to be run before other programs) because we want the AT6n50 to be in a "ready state" for motion. For more information on creating a setup program, refer to the *6000 Series Programmer's Guide*.

Example Setup Program:

```
DEF SETUP       ; Begin def. of "setup" program
@DRIVEØ         ; Disable all drives
INDAX4          ; Place all axes in use
SSFR4           ; Servo sampling frequency ratio
@DRFLVL1        ; Set drive fault level to
                ; "active high" for all axes
@KDRIVE1        ; Enable DISABLE ON KILL feature

; *********************************************
; * Setup for encoders (will need to switch   *
: * between encoder and ANI feedback)         *
; *********************************************
@SFB1           ; Select encoder feedback for
                ; all axes (subsequent scaling,
                ; gains, servo offset, PSET,
                ; and SMPER commands are specific
                ; to encoder feedback)
@ERES4000       ; Set encoder resolution to
                ; 4,000 counts/rev, all axes
@SCLA4000       ; Set scaling for programming
                ; accel/decel in revs/sec/sec
@SCLV4000       ; Set scaling for programming
                ; velocity in revs/sec
@SCLD4000       ; Set scaling for programming
                ; distances in revs
@SGP70          ; Set proportional feedback gain
@SGV2.52        ; Set velocity feedback gain
@SMPER0.001     ; Set max. position error to
                ; 1/1000 of a rev (4 encoder counts)
@PSET0          ; Set current position as
                ; absolute position zero

; *********************************************
; * Setup for ANI feedback (ANI OPTION ONLY)  *
; *********************************************
@SFB2           ; Select ANI feedback for all
                ; axes (subsequent scaling,
                ; gains, servo offset, PSET,
                ; and SMPER parameters are
                ; specific to ANI feedback)
@SCLA819        ; Set scaling for programming
                ; accel/decel in volts/sec/sec
@SCLV819        ; Set scaling for programming
                ; velocity in volts/sec
@CLD819         ; Set scaling for programming
                ; distances in volts
@SGP1           ; Set proportional feedback gain
@SGI0           ; Set integral feedback gain
@SGV0.5         ; Set velocity feedback gain
@SMPER.01       ; Set max. position error to
                ; 1/100 of a volt (8 ANI counts)
@PSET5          ; Set current position as
                ; absolute position 5

@SFB1           ; Select encoder feedback for
                ; start of main program

; *********************************************
; * Insert other appropriate commands in the  *
; * setup program (e.g., custom power-up mesg,*
; * scaling factors, input function assignmts,*
; * output function assignments, input and    *
; * output active levels, etc.). See Prog.    *
; *  Guide, Chapter 3, for more information.  *
; *********************************************

END             ; End definition of "setup" prog
STARTP SETUP    ; Assign the program named setup
                ; as the program to be executed
                ; on power up & reset
```

I N D E X

T

U-Z

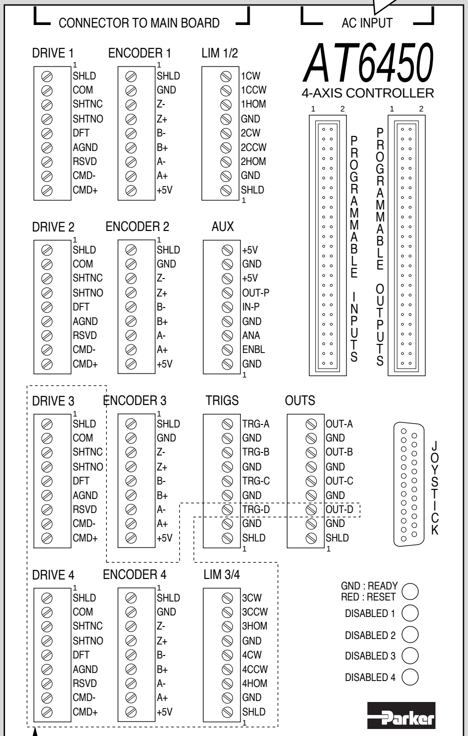
AT6250 & AT6450 Servo Controllers

Parker Automation

Connections

See also pages 7-23

The DC input version (AT6n50-DC) has a 5-pin screw terminal connector for DC power input (see page 20 for connections).



Dashed line denotes features not available on the AT6250.

PIN OUTS NOT LABELED

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

Even numbered pins are connected to logic ground.
MSB = Most Significant Bit; LSB = Least Significant Bit

JOYSTICK

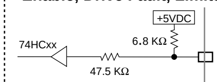
Pin	Function
1-4	Analog Channels 1-4
8	Shield (chassis gnd)
14	Isolated Logic gnd
15	Axes Select Input
16	Velocity Select Input
17	Release Input
18	Trigger Input
19	Auxiliary Input
23	+5VDC Output

Pins 5-7, 9-13, 20-21, 24-25 are reserved

I/O SPECIFICATIONS & INTERNAL SCHEMATICS

Power (see page 20 for connection instructions):
PC Card..............5VDC @ 2.5A from the PC-AT bus.
120VAC version...90-132VAC, 50/60Hz, 1.5A @ 120VAC, single-phase. (or connect 5VDC ±10% @ 1.6A to a +5V terminal)
240VAC version...90-264VAC, 50/60Hz, 0.75A @ 240VAC, single-phase. (or connect 5VDC ±10% @ 1.6A to a +5V terminal)
DC version...........+5VDC @ 1.6A; ±15VDC @ 50mA.

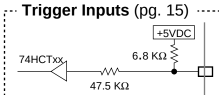
Enable, Drive Fault, Limits, and Joystick Inputs (pg. 7,8,12,&14)



ENBL, DFT, CW, CCW, HOM, or Joystick digital inputs (pins 15-19)

Specs: HCMOS-compatible*; voltage range = 0-24VDC.

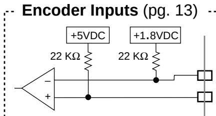
Trigger Inputs (pg. 15)



TRG-n inputs

Specs: TTL (low ≤ 0.4V, high ≥ 2.4V); voltage range = 0-24VDC.

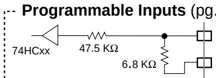
Encoder Inputs (pg. 13)



A–, B–, or Z–
A+, B+, or Z+

Specs: Differential comparator. Use 2-phase quadrature encoders; max. frequency = 1.6 MHz; min. time between transitions = 625 ns. TTL levels (Low ≤ 0.4V, High ≥ 2.4V); range = 0-5VDC.

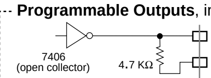
Programmable Inputs (pg. 16)



General-Purpose Programmable Input
IN-P. Connect to the +5V terminal on the AUX connector, **or** to an external 5-24VDC supply (**not to both**). To sink current, connect to a GND terminal. **

Specs: HCMOS-compatible*; voltage range = 0-24VDC.

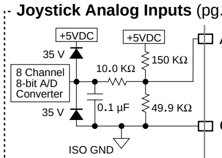
Programmable Outputs, includes OUT-A through OUT-D (pg. 16)



General-Purpose Prog. Output & OUT-n
OUT-P. Connect to the +5V terminal on the AUX connector, **or** to an external supply of up to 24VDC (**not to both**). **

Specs: Open collector output. Outputs will sink up to 30mA, or source up to 5mA at 5-24VDC.

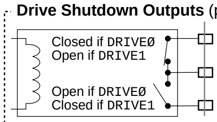
Joystick Analog Inputs (pg. 14)



Analog Channel Inputs (pins 1-4)
Ground (pin 14)

Specs: Voltage range = 0-2.5VDC, 8-bit. Must not exceed 5VDC.

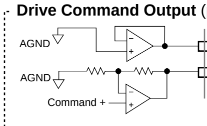
Drive Shutdown Outputs (pg. 8-11)



SHTNC (normally closed)
COM (signal common for shutdown)
SHTNO (normally open)

Specs: Solid state relay. Max. rating = 175VDC, 0.25A, 3W.

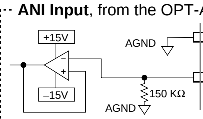
Drive Command Output (pg. 8-11)



CMD– (command signal return)
CMD+ (command signal output)

Specs: ±10V analog output; 12-bit DAC. Load should be > 2KΩ impedance.

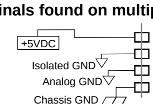
ANI Input, from the OPT-AT6N50-A analog input card (pg. 7)



AGND
Analog Input Terminal

Specs: ±10V analog input; 14-bit ADC.

Terminals found on multiple connectors



+5V
GND
AGND
SHLD

5V terminal found on multiple connectors. Total load limit for all connections is 1.4A for AT6n50-120 and 0.4A for AT6n50-240. Load limit for AT6n50-DC depends on your external +5VDC supply.

ANA Output (supplemental ±10V output): see details on page 8.

* *HCMOS-compatible levels: Low ≤ 1.00V, High ≥ 3.25V.*
** *Disconnect from +5V terminal BEFORE connecting an external 5-24VDC supply.*

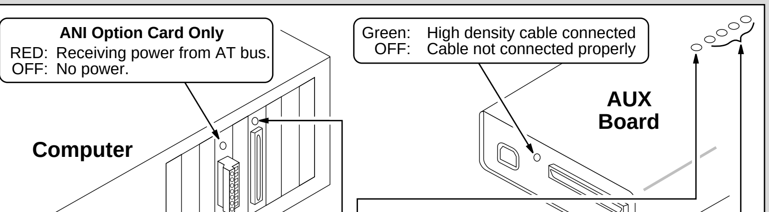
Troubleshooting

See also pages 27-32

- LEDs – see drawing below.
- Status information (see command descriptions in *6000 Series Software Reference*):
  - General status information.....................`TASF`, `TSSF`, `TSTAT`
  - Limits (end-of-travel, home)..................`TASF`, `TLIM`
  - ENBL input..`TINOF` (bit #6)
  - Programmable inputs and TRG-n...........`TIN`, `INFNC`
  - Programmable outputs and OUT-n........`TOUT`, `OUTFNC`
  - Gains (currently active)..........................`TGAIN`
- ENBL input must be grounded to GND terminal to allow motion.
- CW & CCW inputs must be grounded to GND terminal to allow motion (or disable with `LHØ` command).
- Connecting OUT-P or IN-P to the +5V terminal **and** to an external supply will damage the AT6n50.
- To help prevent electrical noise, shield all connections at one end only.
- Error messages while programming or executing programs – see *6000 Series Programmer's Guide.*
- Error messages while downloading the operating system – see error table on page 31.
- Tuning – see pages 24 & 33-45 (if you own Servo Tuner, refer to the *Servo Tuner User Guide*).
- PC card address and interrupt conflicts – see page 4 & 30.
- Technical support – see phone numbers on inside of front cover, and the `HELP` command response.

ANI Option Card Only
RED: Receiving power from AT bus.
OFF: No power.

Green: High density cable connected
OFF: Cable not connected properly



When power is applied, the status LED on the AT6n50 card will be off and the status LED on the AUX board should be red. If the LED on the AUX board is off, the AUX board does not have AC power. After the operating system is downloaded, the status LEDs on the AUX board and the AT6n50 card will turn green indicating the system is ready for operation. If both status LEDs do not turn green after downloading the operating system, an error has occurred. The operating system downloader (Motion Architect or the AT6250.EXE or AT6450.EXE program from your DOS support disk) issues an error message if the download operation is not successful (refer to the Downloading Error table on page 31).

After power up, the DISABLED status LEDs will be red until you issue the `DRIVE1111` command to enable the drives, at which time the LEDs will turn off. These LEDs will again turn red if the drives are shut down—possible causes are:
- Shutdown commanded with `DRIVE` command.
- Drive fault input (DFT) is active.
- Enable input (ENBL) is not grounded.
- Maximum allowable position error (set with the `SMPER` command value) is exceeded.
- Kill command (`K`, `!K` or `<CTRL>K`) was issued, or a Kill input or user fault input was activated while the Disable Drive on Kill feature was enabled (refer to the `KDRIVE` command).

We welcome your feedback on our products and user guides. Please send your comments to our email address: `6000user@cmotor.com`
Direct your technical questions to your local ATC or distributor, or to the numbers printed on the inside front cover of this document.

email