

---

**THE UNIDEX® 600 SERIES UTILITIES  
MANUAL, OS/2 MANUAL**

**P/N: EDU145(V1.1)**

---



AEROTECH, Inc. • 101 Zeta Drive • Pittsburgh, PA. 15238-2897 • USA  
Phone (412) 963-7470 • Fax (412) 963-7459  
Product Service: (412) 967-6440; (412) 967-6870 (Fax); (412) 967-6427 (BBS)

UNIDEX 30 is a product of Aerotech, Inc.  
UNIDEX 31 is a registered trademark of Aerotech, Inc. and is not to be used in publication without the written consent of Aerotech, Inc.  
UNIDEX 600 is a product of Aerotech, Inc.  
Inductosyn is a registered trademark of Farrand Industries, Inc.  
OS/2 Operating System is a registered trademark of the IBM Corporation in the U.S. and other countries.  
MS-DOS and Windows are products of Microsoft Corporation.

The UNIDEX 30/31/600 Series Motion Controller  
Operations and Technical Manual Revision History:

Rev 1.0	August 1994
Rev 1.1	June 1997

## PREFACE

This section gives you an overview of topics covered in each of the sections of this manual as well as conventions used in this manual. The UNIDEX 30/31/600 System Utilities Manual contains information on the following topics:

### CHAPTER 1: OVERVIEW

This chapter contains an overview of the UNIDEX 30/31/600 Motion Controllers. This chapter also defines the utility programs common to the UNIDEX 30, UNIDEX 31, and UNIDEX 600 Motion Controllers which will be discussed in greater detail in the remaining chapters.

### CHAPTER 2: DEBUG960

This chapter contains information relative to how you can manipulate axis parameters, carry out motion tasks, configure axes and perform system diagnostic tests. Also in this chapter is information regarding startup, brushless motor commutation and parameter definitions.

### CHAPTER 3: STATUS

This chapter covers information related to the Axis Status & Fault Utility. This chapter provides information on how to monitor the Axis Status, Fault Status, Servo Status and Motion Status of the axis processor. It explains how the user can obtain the current value of the masks that define the activity of the axis processor upon fault detection.

### CHAPTER 4: PLOT

This chapter supplies information that defines the various functions used to measure, adjust and record the servo performance of each axis of motion (necessary when tuning the axis). Typically, this information is in graphic form.

### APPENDIX A: PARAMETER SUMMARY

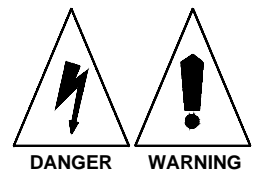
Appendix A touches on the parameters associated with each of the utilities. The summary includes information such as the parameter default values, the minimum values and maximum values for each parameter and informs you if the parameter is read, write or read/write.

### APPENDIX B: WARRANTY AND FIELD SERVICE

Appendix B contains the warranty and field service policy for Aerotech products.

Throughout this manual the following conventions are used:

- The terms UNIDEX 30, UNIDEX 31, UNIDEX 600 and U30, U31, U600 are used interchangeably throughout this manual
- Danger and/or Warning symbols (see left) appear in the outer margins next to important precautions. Failure to observe these precautions could result in serious injury and/or damage to the equipment.



- Keys such as Shift, Ctrl, Alt and Enter are bold faced and enclosed in brackets (e.g., <Shift>, <Ctrl>, <Alt> and <Enter>) to distinguish them from individual keystrokes
- Hexadecimal numbers are listed using a preceding "0x" (for example, 0x300, 0x12F, 0x01EA, etc.) to distinguish them from decimal numbers
- The terms <Enter> and <Return> (appear as bold text in brackets, as shown) are used interchangeably throughout this document when referring to the keyboard
- Within the index, a bold locator page number (e.g., Components, **1-1**) indicates that the reference is part of an illustration. An italic locator page number (e.g., CCWEOT, *4-7*) indicates that the reference is part of a table, while a bold italic page number indicates that the reference is within a table. Text references are shown in a standard serif font (e.g., Software Setup, 3-1).
- This manual uses the symbol "▽ ▽ ▽" to indicate the end of a chapter.

### **APPENDIX C: GLOSSARY OF TERMS**

Appendix C contains a list of definitions of terms used throughout this manual.

▽ ▽ ▽

## TABLE OF CONTENTS

<b>Chapter 1: INTRODUCTION</b> .....	1-1
1.1. Introduction .....	1-1
1.2. Unidex 30 Motion Controller .....	1-1
1.3. Unidex 31 Motion Controller .....	1-1
1.4. Unidex 600 Motion Controller .....	1-2
<b>Chapter 2: DEBUG960</b> .....	2-1
2.1. Overview .....	2-1
2.2. Getting Started .....	2-1
2.3. Startup Considerations .....	2-2
2.4. Basic Commands .....	2-2
2.4.1. INFO - Revision Level and Axis Configuration .....	2-3
2.4.2. STAT - Axis Status .....	2-3
2.4.3. HELP - Command Syntax .....	2-5
2.5. Configuring Axes .....	2-7
2.5.1. CONFIG - Resolver/Inductosyn Axis .....	2-8
2.5.2. CONFIGENCODER - Encoder Axis .....	2-8
2.5.3. CONFIGHALL - Hall Effect .....	2-9
2.6. Parameter Operation Commands .....	2-9
2.6.1. Viewing Parameters .....	2-9
2.6.1.1. STAT/PARM - Display Axis Parameters & Servo States .....	2-9
2.6.1.2. GET - Displaying Parameter Values .....	2-9
2.6.1.3. M - Monitor .....	2-10
2.6.2. SET - Editing Parameters .....	2-11
2.7. Motion Commands .....	2-11
2.7.1. HOME - Homing Axis .....	2-12
2.7.2. INDEX - Incremental Motion .....	2-12
2.7.3. MOVETO - Absolute Motion .....	2-13
2.7.4. START - Continuous Motion .....	2-13
2.7.5. HALT/ABORT - Halting/Aborting Motion .....	2-13
2.8. Analog and Digital I/O .....	2-14
2.8.1. Xycom Digital I/O .....	2-14
2.8.1.1. INTXYCOM - Configuration .....	2-14
2.8.1.2. READXYCOM - Reading Digital Inputs .....	2-15
2.8.1.3. WRITEXYCOM - Writing Digital Outputs .....	2-15
2.8.2. Matrix Analog Inputs .....	2-16
2.8.2.1. INITMATRIX - Channel Operation for Analog Inputs .....	2-16
2.8.2.2. READMATRIX - Reading Analog Inputs .....	2-16
2.9. System Diagnostics .....	2-16
2.9.1. COMM - Communication Testing .....	2-16
2.10. Fault Handling .....	2-18
2.10.1. FAULTMASK .....	2-20
2.10.2. DISABLEMASK .....	2-22
2.10.3. HALTMASK .....	2-22
2.10.4. AUXMASK .....	2-22
2.10.5. ABORTMASK .....	2-22
2.10.6. INTMASK .....	2-22

2.10.7.	Acknowledging Faults.....	2-23
2.11.	Programming Errors .....	2-23
2.12.	Parameter Descriptions.....	2-24
2.12.1.	ABORTMASK.....	2-24
2.12.2.	ACCEL.....	2-25
2.12.3.	ACCELMODE.....	2-25
2.12.4.	ACCELRATE .....	2-25
2.12.5.	AFFGAIN .....	2-25
2.12.6.	ALT_STATUS.....	2-25
2.12.7.	AUX.....	2-25
2.12.8.	AUXMASK.....	2-25
2.12.9.	AUXOFFSET.....	2-26
2.12.10.	AVGVEL .....	2-26
2.12.11.	AVGVELTIME.....	2-26
2.12.12.	BASE_SPEED .....	2-26
2.12.13.	BLOCKMOTION .....	2-27
2.12.14.	CAMOFFSET .....	2-27
2.12.15.	CAMPOINT.....	2-27
2.12.16.	CAMPOSITION .....	2-27
2.12.17.	CCWEOT.....	2-27
2.12.18.	CLOCK .....	2-27
2.12.19.	CWEOT .....	2-27
2.12.20.	DECEL.....	2-28
2.12.21.	DECELMODE .....	2-28
2.12.22.	DECELRATE .....	2-28
2.12.23.	DISABLEMASK .....	2-28
2.12.24.	DRIVE .....	2-28
2.12.25.	ECHO.....	2-28
2.12.26.	FAULT.....	2-29
2.12.27.	FAULTMASK .....	2-29
2.12.28.	FBWINDOW .....	2-29
2.12.29.	FEEDRATEMODE.....	2-29
2.12.30.	HALTMASK.....	2-29
2.12.31.	HOMESWITCHPOS .....	2-29
2.12.32.	HOMESWICHTOL.....	2-30
2.12.33.	HOMEVELMULT .....	2-30
2.12.34.	I AVG.....	2-30
2.12.35.	I AVGLIMIT.....	2-30
2.12.36.	I AVGTIME.....	2-30
2.12.37.	ICMD .....	2-30
2.12.38.	IMAX .....	2-31
2.12.39.	INPOSLIMIT .....	2-31
2.12.40.	INTMASK .....	2-31
2.12.41.	IOLEVEL.....	2-31
2.12.42.	IVEL .....	2-31
2.12.43.	KI .....	2-31
2.12.44.	KP .....	2-32
2.12.45.	MASTERLEN.....	2-32
2.12.46.	MASTERPOS .....	2-32
2.12.47.	MASTERABS .....	2-32
2.12.48.	MASTERRES .....	2-32

2.12.49.	MAX_PHASE.....	2-32
2.12.50.	MOTIONSTATUS .....	2-33
2.12.51.	MOVEQDEPTH.....	2-37
2.12.52.	MOVEQSIZE .....	2-37
2.12.53.	PGAIN .....	2-37
2.12.54.	PHASE_SPEED.....	2-37
2.12.55.	POS .....	2-37
2.12.56.	POSCMD.....	2-37
2.12.57.	POSERR .....	2-37
2.12.58.	POSERRLIMIT .....	2-37
2.12.59.	POSTARGET .....	2-38
2.12.60.	POSTOGO.....	2-38
2.12.61.	PROFQDEPTH.....	2-38
2.12.62.	PROFQSIZE .....	2-38
2.12.63.	RAWPOS.....	2-38
2.12.64.	RESOLVER.....	2-38
2.12.65.	REVERSALMODE .....	2-39
2.12.66.	REVERSALVALUE.....	2-39
2.12.67.	SAFEZONECCW .....	2-39
2.12.68.	SAFEZONECW .....	2-39
2.12.69.	SAFEZONEMODE.....	2-39
2.12.70.	SERVOSTATUS .....	2-40
2.12.71.	SIMULATION.....	2-41
2.12.72.	SOFTLIMITMODE.....	2-41
2.12.73.	STATUS .....	2-44
2.12.74.	SYNCSPEED.....	2-47
2.12.75.	VELCMDTRAP.....	2-47
2.12.76.	VELPOSITION.....	2-47
2.12.77.	VELTRAP.....	2-48
2.12.78.	VEL_SHIFT.....	2-48
2.12.79.	VFF .....	2-48
2.13.	Brushless Motor Commutation.....	2-48
2.13.1.	Aligning Resolver Based Brushless Motors.....	2-48
2.13.2.	Working With Hall Effect Motors .....	2-49
<b>Chapter 3:</b>	<b>STATUS.....</b>	<b>3-1</b>
3.1.	Overview .....	3-1
3.2.	Display Pull-down Menu .....	3-2
3.2.1.	Axis Status Display .....	3-3
3.2.1.1.	Drive Enabled .....	3-4
3.2.1.2.	Aux Output Enable.....	3-4
3.2.1.3.	CW Input .....	3-5
3.2.1.4.	CCW Input.....	3-5
3.2.1.5.	Home Input .....	3-5
3.2.1.6.	Drive Fault Input.....	3-5
3.2.1.7.	At Home.....	3-5
3.2.1.8.	Done.....	3-6
3.2.1.9.	In-position .....	3-6
3.2.1.10.	Faulted .....	3-6
3.2.1.11.	Probe Input.....	3-6
3.2.1.12.	Marker Input .....	3-6

3.2.1.13.	Hall Input #1, #2, #3, and #4 .....	3-7
3.2.1.14	Move Direction .....	3-7
3.2.1.15.	Moving .....	3-7
3.2.1.16.	Accel Phase .....	3-7
3.2.1.17.	Decel Phase .....	3-8
3.2.1.18.	Homing .....	3-8
3.2.1.19.	Feedrate Override.....	3-8
3.2.1.20.	Profile Mode .....	3-8
3.2.1.21.	Sync Mode .....	3-9
3.2.1.22.	Cam Table Enable.....	3-9
3.2.1.23.	Homing Direction.....	3-9
3.2.1.24.	Continuous Move.....	3-9
3.2.1.25.	Queued Command.....	3-9
3.2.1.26.	Hold Active .....	3-10
3.2.1.27.	Aux Mode .....	3-10
3.2.1.28.	Block Motion .....	3-10
3.2.1.29.	Hold Queue .....	3-10
3.2.2.	Servo Status Display .....	3-11
3.2.2.1.	Drive Enable .....	3-11
3.2.2.2.	Aux Output Enable.....	3-11
3.2.2.3.	CW Input.....	3-11
3.2.2.4.	CCW Input.....	3-11
3.2.2.5.	Home Input .....	3-12
3.2.2.6.	Drive Fault Input.....	3-12
3.2.2.7.	At Home .....	3-12
3.2.2.8.	Done.....	3-12
3.2.2.9.	In-position .....	3-12
3.2.2.10.	Faulted .....	3-12
3.2.2.11.	Probe Input.....	3-12
3.2.2.12.	Marker Input .....	3-12
3.2.2.13.	Hall Input #1, #2, #3, and #4 .....	3-12
3.2.2.14.	Integral -clamped .....	3-12
3.2.2.15.	Integral + clamped .....	3-13
3.2.2.16.	VFF Enable .....	3-13
3.2.2.17.	AFF Enable .....	3-13
3.2.2.18.	Alive.....	3-13
3.2.2.19.	VFF or Zero Pgain .....	3-13
3.2.2.20.	FB Failure Input.....	3-14
3.2.2.21.	Master FB Failure .....	3-14
3.2.2.22.	HP VME Laser.....	3-14
3.2.2.23.	AC Mode.....	3-14
3.2.2.24.	Msetting .....	3-14
3.2.2.25.	Homed.....	3-15
3.2.2.26.	Encoder Feedback .....	3-15
3.2.2.27.	Errormapping Enable .....	3-15
3.2.2.28.	Position Loop Only .....	3-15
3.2.3.	Motion Status Display .....	3-16
3.2.3.1.	Move Direction .....	3-16
3.2.3.2.	Moving.....	3-16
3.2.3.3.	Accel Phase .....	3-16
3.2.3.4.	Decel Phase.....	3-16



3.2.3.5.	Homing .....	3-17
3.2.3.6.	Feedrate Override .....	3-17
3.2.3.7.	Profile Mode .....	3-17
3.2.3.8.	Sync Mode .....	3-17
3.2.3.9.	Cam Table Enable.....	3-17
3.2.3.10.	Homing Direction .....	3-17
3.2.3.11.	Continuous Move.....	3-17
3.2.3.12.	Queued Command.....	3-17
3.2.3.13.	Hold Active.....	3-17
3.2.3.14.	Aux Mode .....	3-17
3.2.3.15.	Block Motion .....	3-17
3.2.3.16.	Hold Queue .....	3-17
3.2.3.17.	Disable Queued.....	3-18
3.2.3.18.	Halt Queued.....	3-18
3.2.3.19.	Abort Queued.....	3-18
3.2.3.20.	Accel Enable .....	3-18
3.2.3.21.	Decel Enable .....	3-19
3.2.3.22.	Accel Sign.....	3-19
3.2.3.23.	Accel Mode.....	3-19
3.2.3.24.	Decel Mode.....	3-19
3.2.3.25.	Bounded.....	3-20
3.2.3.26.	Setup Pending .....	3-20
3.2.3.27.	Checker Flag.....	3-20
3.2.3.28.	Quick Home .....	3-20
3.2.3.29.	Interrupt Pending .....	3-20
3.2.3.30.	Pendant Jog.....	3-21
3.2.3.31.	Marker Armed.....	3-21
3.2.3.32.	Jog Enable.....	3-21
3.2.4.	Fault Status Display .....	3-22
3.2.4.1.	Position Error Limit .....	3-22
3.2.4.2.	RMS Current Limit .....	3-23
3.2.4.3.	CW Hardware EOT .....	3-23
3.2.4.4.	CCW Hardware EOT .....	3-23
3.2.4.5.	CW Software EOT .....	3-23
3.2.4.6.	CCW Software EOT .....	3-23
3.2.4.7.	Drive .....	3-24
3.2.4.8.	Feedback .....	3-24
3.2.4.9.	Programming.....	3-24
3.2.4.10.	Master Feedback.....	3-24
3.2.4.11.	Homing .....	3-24
3.2.4.12.	User.....	3-25
3.2.4.13.	Velocity Trap .....	3-25
3.2.4.14.	Velocity Command Trap.....	3-25
3.2.4.15.	Home Switch Tolerance.....	3-25
3.2.4.16.	Probe Input.....	3-26
3.2.4.17.	CNC .....	3-26
3.2.4.18.	External Feedback.....	3-26
3.2.4.19.	Safe Zone .....	3-26
3.2.5.	Fault Mask Display .....	3-27
3.2.6.	Disable Mask Display .....	3-28
3.2.7.	Interrupt Mask Display .....	3-29

3.2.8.	Halt Mask Display.....	3-30
3.2.9.	Aux Mask Display.....	3-31
3.3.	Time Pull-down Menu .....	3-32
3.4.	Exit Pull-down Menu.....	3-33
3.4.1.	Exit.....	3-33
3.4.2.	Resume.....	3-33
3.5.	Fault_Control Pull-down Menu .....	3-34
3.5.1.	Faults Latched .....	3-34
3.5.2.	Faults Not Latched .....	3-34
3.5.3.	Clear Faults .....	3-34
3.6.	Help Pull-down Menu.....	3-35
3.6.1.	About.....	3-36

## Chapter 4: PLOT ..... 4-1

4.1.	Overview .....	4-1
4.2.	File Pull-down Menu .....	4-2
4.2.1.	Save.../Load.....	4-3
4.2.2.	About.....	4-4
4.2.3.	Exit .....	4-4
4.3.	Plot Pull-down Menu.....	4-5
4.3.1.	Actual Velocity.....	4-5
4.3.2.	Commanded Velocity.....	4-5
4.3.3.	Velocity Error.....	4-6
4.3.4.	Actual Position .....	4-6
4.3.5.	Commanded Position.....	4-6
4.3.6.	Position Error .....	4-6
4.3.7.	Torque .....	4-6
4.3.8.	Spare.....	4-6
4.4.	Trigger Pull-down Menu .....	4-7
4.4.1.	Acquire 1 Set of Data .....	4-7
4.4.2.	Acquire Data Continuously .....	4-7
4.4.3.	Stop Data Acquisition .....	4-7
4.4.4.	Set Trap Mode.....	4-8
4.4.5.	Set Data Acquisition Mode... ..	4-9
4.5.	Collect Pull-down Menu .....	4-10
4.6.	Display Pull-down Menu .....	4-11
4.6.1.	Samples .....	4-11
4.6.2.	Common Axis.....	4-11
4.6.3.	Zero Line .....	4-11
4.6.4.	Cursor.....	4-12
4.7.	Axis Pull-down Menu .....	4-13
4.7.1.	Single Step.....	4-13
4.7.2.	Auto Step.....	4-13
4.7.3.	Halt Motion .....	4-14
4.7.4.	Start Continuous Motion .....	4-14
4.7.5.	Tune... ..	4-14
4.7.6.	Set Speed/Distance.....	4-15
4.7.7.	Select Axis.....	4-16

- 4.8. Tuning servo Loops ..... 4-16
  - 4.8.1. Kp - Proportional Gain ..... 4-17
  - 4.8.2. Ki - Integral Gain ..... 4-18
  - 4.8.3. PGain - Position Gain..... 4-18
  - 4.8.4. Vff - Velocity Feedforward Gain ..... 4-18
  - 4.8.5. AffGain - Acceleration Feedforward Gain ..... 4-18
  - 4.8.6. Alpha - AffGain Filter ..... 4-18
  - 4.8.7. VGain - Constant Velocity Gain ..... 4-18
  - 4.8.6. Tuning Procedure for Servo Loops ..... 4-18
- 4.9. AC Brushless Motor Tuning Tip ..... 4-29
- 4.10. Tuning With Tachometer Feedback ..... 4-30
  - 4.10.1. Vff - Velocity Feed Forward ..... 4-30
  - 4.10.2. Vgain - Constant Velocity Gain ..... 4-30
  - 4.10.3. Servo Parameter Setup for Tachometer Feedback ..... 4-30
  - 4.10.4. The Servo Loop Parameters for Tachometer based Systems ..... 4-31
    - 4.10.3.1. PGain - Position Gain ..... 4-31
    - 4.10.3.2. Vff - Velocity Feedforward Gain ..... 4-31
    - 4.10.3.3. Kp - Proportional Gain..... 4-31
    - 4.10.3.4. Ki - Integral Gain ..... 4-32
    - 4.10.3.5. AffGain - Acceleration Feedforward Gain ..... 4-32
    - 4.10.3.6. VGain - Constant Velocity Gain ..... 4-32
- 4.11. Tuning Tachometer Loops..... 4-32

**Appendix A: PARAMETER SUMMARY ..... A-1**

**Appendix B: WARRANTY AND FIELD SERVICE POLICY.....B-1**

**Appendix C: GLOSSARY OF TERMS .....C-1**

**Index**





**LIST OF FIGURES**

Figure 2-1.	Debug960 General Information Screen.....	2-3
Figure 2-2.	Debug960 Parameter/Status Screen #1 .....	2-4
Figure 2-3.	Debug960 Parameter/Status Screen #2 .....	2-5
Figure 2-4.	Debug960 Help Screen #1 .....	2-6
Figure 2-5.	Debug960 Help Screen #2 .....	2-7
Figure 2-6.	Defined Slope of a Velocity Curve at a Specified Angle .....	2-26
Figure 3-1.	Status Utility Initialization Screen.....	3-2
Figure 3-2.	Display Pull-down Menu.....	3-3
Figure 3-3.	Aerotech Axis Status Display.....	3-4
Figure 3-4.	Aerotech Servo Status Display.....	3-11
Figure 3-5.	Aerotech Motion Status Display .....	3-16
Figure 3-6.	Aerotech Fault Status Display.....	3-22
Figure 3-7.	Aerotech Fault Mask Display.....	3-27
Figure 3-8.	Aerotech Disable Mask Display.....	3-28
Figure 3-9.	Aerotech Interrupt Mask Display .....	3-29
Figure 3-10.	Aerotech Halt Mask Display .....	3-30
Figure 3-11.	Aerotech Aux Mask Display .....	3-31
Figure 3-12.	Time Pull-down Menu .....	3-32
Figure 3-13.	Exit Pull-down Menu .....	3-33
Figure 3-14.	Fault_Control Pull-down Menu.....	3-34
Figure 3-15.	Help Pull-down Menu .....	3-35
Figure 3-16.	About Dialog Box .....	3-36
Figure 4-1.	Plot Mainmenu Dialog Box .....	4-2
Figure 4-2.	File Pull-down Menu .....	4-3
Figure 4-3.	File Selection Dialog Box.....	4-4
Figure 4-4.	Plot Pull-down Menu.....	4-5
Figure 4-5.	Trigger Pull-down Menu .....	4-7
Figure 4-6.	Set Trap Mode Dialog Box .....	4-8
Figure 4-7.	Set Data Acquisition Mode Dialog Box .....	4-9
Figure 4-8.	Collect Pull-down Menu.....	4-10
Figure 4-9.	Display Pull-down Menu .....	4-11
Figure 4-10.	Cursor Measurement Tool Dialog Box.....	4-12
Figure 4-11.	Axis Pull-down Menu.....	4-13
Figure 4-12.	Axis Tuning Dialog Box.....	4-14
Figure 4-13.	Speed/Distance Setup Dialog Box.....	4-15
Figure 4-14.	Axis Selection Screen.....	4-16
Figure 4-15.	Servo Loop Diagram .....	4-17
Figure 4-16.	Flowchart of Overall Tuning Process .....	4-19
Figure 4-17.	Unacceptable Velocity Error .....	4-22
Figure 4-18.	Acceptable Velocity Error (While Adjusting Kp) .....	4-22
Figure 4-19.	Unacceptable Position Error (While Adjusting Ki) .....	4-23
Figure 4-20.	Acceptable Position Error (While Adjusting Ki).....	4-23
Figure 4-21.	Plot Showing an Appropriate Value for PGain.....	4-24
Figure 4-22.	Plot Showing Overall Effects When PGain is High.....	4-25
Figure 4-23.	Plot Showing Velocity Feedforward Enabled ( Vff=1) .....	4-25

Figure 4-24. Plot Showing Optimal AffGain Adjustment ..... 4-26  
Figure 4-25. Plot Showing Final Performance of ATS3220140P X axis table,  
with a BM130 motor and an AS32030 amplifier. .... 4-27  
Figure 4-26. Torque Ripple Plot of an AC Brushless Motor..... 4-29  
Figure 4-27. Flowchart of Overall Tachometer Tuning Process ..... 4-33  
Figure 4-28. Cross-Section of the DS16020/16030 Amplifier ..... 4-34  
Figure 4-29. Amplifier Potentiometer Layout ..... 4-35  
Figure 4-30. Oscilloscope Showing Current Feedback for One Move..... 4-37

▽ ▽ ▽

**LIST OF TABLES**

Table 2-1. General Mask Bits..... 2-18  
Table 2-2. Status Bit Map of 32 Bit Conditions ..... 2-20  
Table 2-3. Motion Status Bits..... 2-33  
Table 2-4. Servo Status Bits ..... 2-40  
Table 2-5. Status Bits ..... 2-44

Table 4-1. Data Acquisition Modes ..... 4-9  
Table 4-2. Data Acquisition Parameters..... 4-10  
Table 4-3. Initial Torque Mode Servo Loop Parameter Values ..... 4-20  
Table 4-4. Servo Loop Axis Parameters for Tachometer based systems ..... 4-31  
Table 4-5. Initial Servo Parameter Values - Tachometer Tuning..... 4-34

▽ ▽ ▽





## CHAPTER 1: INTRODUCTION

**In This Section:**

- Introduction ..... 1-1
- UNIDEX 30 Motion Controller..... 1-1
- UNIDEX 31 Motion Controller..... 1-2
- UNIDEX 600 Motion Controller..... 1-2

### 1.1. Introduction

The U600 Series Utilities Manual describes three basic utilities. These utilities include Debug960, Status and Plot. Each utility comes standard with the UNIDEX 30, UNIDEX 31 and UNIDEX 600 Motion Controller. The utilities as well as the motion controllers are defined throughout the following sections.

#### 1.1.1. Debug960

The Debug960 Utility is a low level diagnostic tool that permits the user to manipulate axis parameters, carry out simple motion tasks and configure axes. The utility is typically used during the initial phases of machine startup to test I/O and exercise the axis as well as during normal machine operation to verify parameter settings.

#### 1.1.2. Status

The Status Utility provides the user with information about what is occurring on the axis processor. This information includes categories such as Axis Status, Fault Status, Servo Status and Motion Status. Though status provides the user with information on the current value of the masks, it will not explain or permit the user to modify these masks.

#### 1.1.3. Plot

The Plot Utility is a data acquisition application that measures, adjusts and records the servo performance of a particular axis in motion. The axis processor card passes servo information to the AT front end through the VME backplane. This passing of information is what causes plot to process the information. The system (U30, U31 or U600) presents this information in a graphical form that resembles an oscilloscope or strip chart recording. In addition to showing servo performance in real time, this utility can save its data collections for later recall.

### 1.2. UNIDEX 30 Motion Controller

The UNIDEX 30 is a 12 slot VME only version of the UNIDEX 31. This controller uses a 80960 RISC based servo axis processor that is a full-length 16 bit ISA expansion card. It is software compatible with the UNIDEX 31 with the exception of support for VME I/O used with the UNIDEX 600.

**1.3. UNIDEX 31 Motion Controller**

The UNIDEX 31 structure consists of 16 open VME bus slots and 5 open PC/AT bus slots that permit third party function cards to be added. The UNIDEX 31 uses the 32-bit Intel i960 RISC processor that can control up to 16 axes.

This system includes preloaded software that includes the OS/2 operating system, the motion toolkit function library and the system utility software programs such as Debug960, Status and Plot.

**1.4. UNIDEX 600 Motion Controller**

The UNIDEX 600 is a PC-AT bus motion control system based on the UNIDEX 31 architecture. This controller uses a third party 80960 RISC based servo axis processor and contains 12 VME bus slots only. The U600 is software compatible with the UNIDEX 31 with the exception of support of the AT bus I/O used with the UNIDEX 30.

▽ ▽ ▽

## CHAPTER 2: DEBUG 960

### In This Section:

- Overview ..... 2-1
- Getting Started..... 2-1
- Startup Considerations..... 2-2
- Basic Commands ..... 2-2
- Configuring Axes..... 2-7
- Parameter Operation Commands ..... 2-9
- Motion Commands ..... 2-11
- Analog and Digital I/O ..... 2-14
- System Diagnostics..... 2-16
- Fault Handling ..... 2-18
- Parameter Descriptions..... 2-24
- Brushless Motor Commutation ..... 2-48

### 2.1. Overview

Debug960 is a low level diagnostic tool, built from the original UNIDEX 31 library commands, that allows you to manipulate axis parameters, perform simple motion tasks and configure axes. The utility is typically used during the initial phases of a machine startup to test inputs/outputs (I/O) and exercise the axis. Another common use is for verifying parameter settings during normal machine operation.

### 2.2. Getting Started

The Debug960 Utility is installed on the U30, U31 and the U600 Motion Control System in the C:\U31\UTILITY sub directory and can also be found on the U31 System Utilities distribution diskette. Debug960 can be started in two ways:

1. Depending on which motion controller the has, double click on the U30, U31 or U600 folder located on the OS/2 desktop. Selecting the folder displays the icons for each installed utility and application. Execute the Debug960 program by double clicking, with the left mouse button, on the Debug960 program icon.
2. Open an OS/2 full screen or window and type Debug960 followed by pressing the <Enter> key. This step starts the Debug960 program.

Debug960 displays the following axis prompt:

```
AXIS 01>
```

All commands executed within Debug960 effect the axis denoted by the prompt. To select a new axis, enter the following command:

```
AXIS ??
```

where ?? is the number, from 1 to 16, of the desired axis.

**2.3. Startup Considerations**

*Setting Feedback Devices*

If Debug960 is being used to communicate to an axis for the first time, then the axis must be configured for the type of feedback device present. This is done with either one of the following commands:

CONFIG	for Resolver type feedback
CONFIGENCODER	for encoder feedback
CONFIGHALL	for encoder feedback with the addition of hall sensors required for brushless motor operation.

An axis does not accept a CONFIG command if the drive is enabled. (For information on the DRIVE parameter refer to Section 2.12.24).

*Monitor*

The M command (monitor) in Debug960 is very useful in troubleshooting both the hardware and software areas of a new system. For axes that are configured, the user can use the monitor command to observe axis position (M POS). This can help verify that the feedback for that axis is working by moving the axis by hand and observing the position changes (note that the U30, U31 and U600 expects positive velocity output for positive polarity torque commands). The resolution of an axis can also be confirmed using this method. To do this, move the axis a known amount (one revolution, one inch, etc.) and take note in the change of the monitored position.

To enable an axis it is necessary to use the SET DRIVE command. If the DRIVE parameter is set to zero (0) then the axis is disabled. In order to enable an axis, the DRIVE parameter must be set to a one (1).

*Set Drive*

Before enabling the drive, appropriately set the FAULTMASK, DISABLEMASK, HALTMASK, ABORTMASK and INTMASK parameters. Please refer to the Fault Handling section of this document (Section 2.10).

Limiting the current command to the drive is recommended when operating a system for the first time. This can be done using the IMAX parameter. The default for the IMAX parameter is set to the maximum allowable current command. Setting IMAX to a value less than the default limits the current command to the drive. IMAX is further explained in the Parameter section of this chapter (Section 2.11.38).

*Current Command*

Once an axis is configured and the operation of the feedback is confirmed, the axis can then be tuned for proper servo operation using the Plot Utility provided with the user's system. Consult Chapter 4 (Plot Utility) of this manual for assistance in tuning motors.

**2.4. Basic Commands**

*Axis Tuning*

There are a series of commands available to the user that display fundamental axis information. These commands include current DMR.IMG revision levels and on-line help for all the commands within a program.

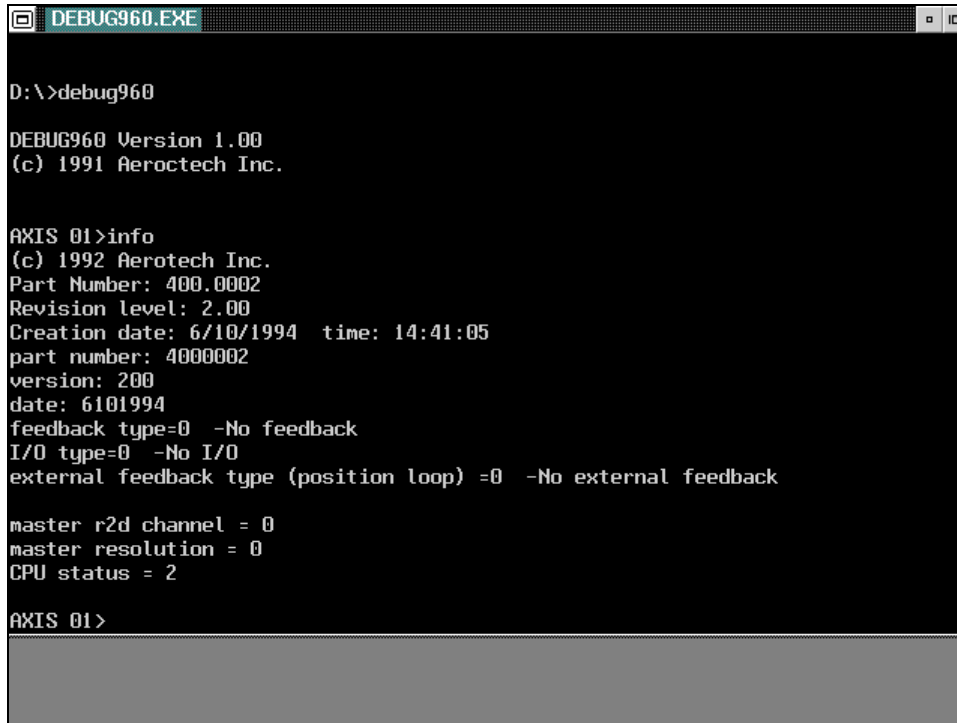
**2.4.1. INFO - Revision Level and Axis Configuration**

*Information Screen*

The installed firmware revision level and the configuration of the axis, denoted by the axis prompt, can be displayed by entering the following command:

INFO

A typical INFO screen appears in Figure 2-1. This shows the axis firmware level, the feedback type, what channel the feedback is setup for, the resolution of the feedback device, commutation information (for brushless motors), the type and channel for the Digital-to-Analog (D-to-A) card and external feedback type (if employed).



```

DEBUG960 Version 1.00
(c) 1991 Aerotech Inc.

AXIS 01>info
(c) 1992 Aerotech Inc.
Part Number: 400.0002
Revision level: 2.00
Creation date: 6/10/1994   time: 14:41:05
part number: 4000002
version: 200
date: 6101994
feedback type=0  -No feedback
I/O type=0  -No I/O
external feedback type (position loop) =0  -No external feedback

master r2d channel = 0
master resolution = 0
CPU status = 2

AXIS 01>

```

Figure 2-1. Debug960 General Information Screen

### 2.4.2. STAT - Axis Status

Typing STAT discloses the status of all axis parameters. These parameters are presented on two separate screens. The first screen appears immediately. By hitting the <Enter> key, the second status screen is displayed. The values that appear are the current values when STAT was requested and may not represent the current values if the servo is operating. In other words, some variables may change if motion is occurring or if other applications are running simultaneously. Typical status screens are illustrated in Figure 2-2 and Figure 2-3. Note that by typing PARM the user can achieve the same function as typing STAT.

*Axis Parameter  
Status*

```

DEBUG960.EXE
master resolution = 0
CPU status = 2

AXIS 01>STAT
STATUS      =      48512 0000bd80  POS      =      0 00000000
ECHO       =      0 00000000  CLOCK    =     365698 00059482
RESOLVER   =      0 00000000  RAWPOS   =      0 00000000
POSCMD     =      0 00000000  POSERR   =      0 00000000
POSTOGO    =      0 00000000  POSTARGET =      0 00000000
IVEL       =      0 00000000  AVGVEL   =      0 00000000
AVGVELTIME =     1000 000003e8  KI       =     2000 000007d0
KP         =     10000 00002710  PGAIN    =     10 0000000a
VFF        =      0 00000000  DRIVE    =      0 00000000
AUX        =      0 00000000  AFFGAIN  =      0 00000000
BLOCKMOTION =      0 00000000  REVERSALMODE =      0 00000000
REVERSALVALUE =      0 00000000  IAVG     =      0 00000000
IMAX       =     32767 00007fff  IAVGLIMIT =     32767 00007fff
IAVGTIME   =     1000 000003e8  ICMD     =      0 00000000
POSERRLIMIT =     65536 00010000  INOSLIMIT =     65 00000041
CWEOT      =     2147483647 7fffffff  CCWEOT   =    -2147483647 80000001
VELTRAP    =      0 00000000  VELCMDTRAP =      0 00000000
VELPOSITION =      0 00000000  FBWINDOW =      0 00000000
SAFEZONECW =      8 00000008  SAFEZONECCW =      0 00000000
SAFEZONEMODE =      0 00000000  SIMULATION =      0 00000000
more...

```

Figure 2-2. Debug960 Parameter/Status Screen #1

```

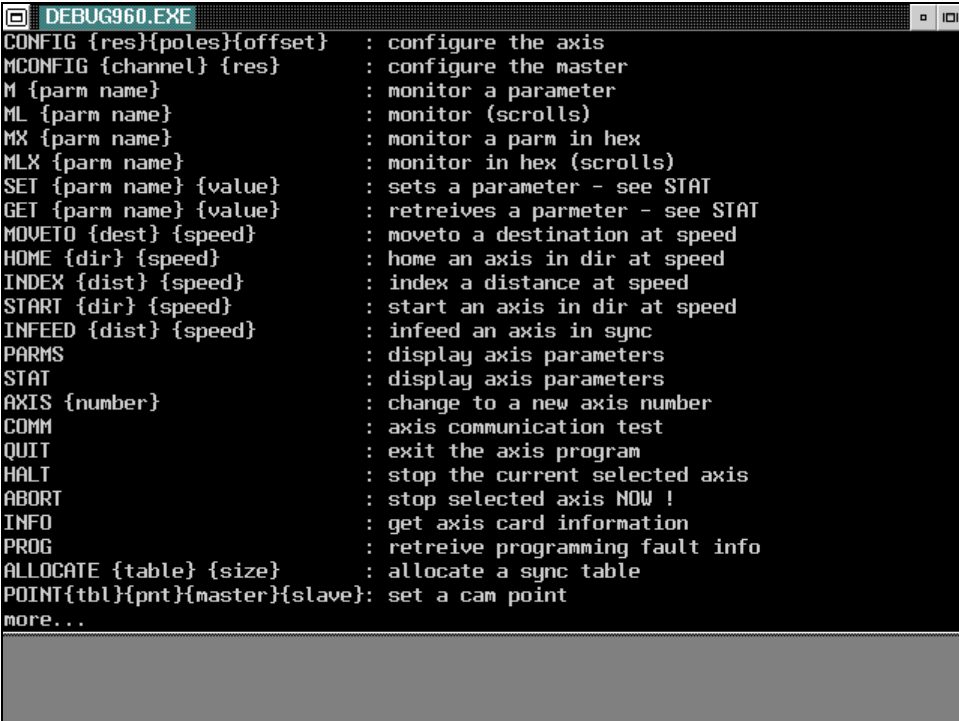
DEBUG960.EXE
VELTRAP = 0 00000000 VELCMDTRAP = 0 00000000
VELPOSITION = 0 00000000 FBWINDOW = 0 00000000
SAFEZONECW = 8 00000008 SAFEZONECCW = 0 00000000
SAFEZONEMODE = 0 00000000 SIMULATION = 0 00000000
ACCEL = 0 00000000 DECEL = 0 00000000
ACCELMODE = 0 00000000 DECELMODE = 0 00000000
FEEDRATEMODE = 0 00000000 ACCELRATE = 100000 000186a0
DECELRATE = 100000 000186a0 HOMESWITCHPOS = 0 00000000
HOMESWITCHTOL = 0 00000000 FAULT = 0 00000000
FAULTMASK = -1 ffffffff DISABLEMASK = -1 ffffffff
INTMASK = 0 00000000 AUXMASK = 0 00000000
HALTMASK = 0 00000000 IOLEVEL = 63 0000003f
AUXOFFSET = 0 00000000 ABORTMASK = 0 00000000
CAMPOSITION = 0 00000000 CAMPOINT = 0 00000000
MASTERPOS = 0 00000000 MASTERRES = 0 00000000
MASTERLEN = 0 00000000 MASTERABS = 0 00000000
CAMOFFSET = 0 00000000 SYNCSPD = 1000 000003e8
PROFQDEPTH = 0 00000000 PROFQSIZE = 32 00000020
MOVEQDEPTH = 0 00000000 MOVEQSIZE = 16 00000010
MOTIONSTATUS = 0 00000000 SERVOSTATUS = 48512 0000bd80
VEL_SHIFT = 0 00000000 BASE_SPEED = 0 00000000
MAX_PHASE = 0 00000000 PHASE_SPEED = 0 00000000
ALT_STATUS = 0 00000000 SOFTLIMITMODE = 0 00000000
HOMEVELMULT = 100 00000064
AXIS 01>
    
```

Figure 2-3. Debug960 Parameter/Status Screen #2

### 2.4.3. HELP - Command Syntax

Typing HELP displays a series of screens that list all the active Debug960 commands and their arguments. Help displays information in the same manner as the STAT and PARM commands. Only the first screen of information is displayed and subsequent screens can be viewed by hitting the <Enter> key. Figure 2-4 and Figure 2-5 reflect the help screens at the time this document was created.

*Status of Active  
Commands/Arguments*



```
DEBUG960.EXE
CONFIG {res}{poles}{offset} : configure the axis
MCONFIG {channel} {res} : configure the master
M {parm name} : monitor a parameter
ML {parm name} : monitor (scrolls)
MX {parm name} : monitor a parm in hex
MLX {parm name} : monitor in hex (scrolls)
SET {parm name} {value} : sets a parameter - see STAT
GET {parm name} {value} : retrieves a parameter - see STAT
MOVETO {dest} {speed} : moveto a destination at speed
HOME {dir} {speed} : home an axis in dir at speed
INDEX {dist} {speed} : index a distance at speed
START {dir} {speed} : start an axis in dir at speed
INFEED {dist} {speed} : infeed an axis in sync
PARMS : display axis parameters
STAT : display axis parameters
AXIS {number} : change to a new axis number
COMM : axis communication test
QUIT : exit the axis program
HALT : stop the current selected axis
ABORT : stop selected axis NOW !
INFO : get axis card information
PROG : retrieve programming fault info
ALLOCATE {table} {size} : allocate a sync table
POINT{tbl}{pnt}{master}{slave}: set a cam point
more...
```

Figure 2-4. Debug960 Help Screen #1



```

DEBUG960.EXE
PROG                : retrieve programming fault info
ALLOCATE {table} {size} : allocate a sync table
POINT{tbl}{pnt}{master}{slave}: set a cam point
UMP {table}         : dump points from table to screen
CALC {table}        : calculate coefficients for table
CAN                 : do a 'canned' dump of sync points
SYNC {table} {mode} : put the given table into sync
GO {table} {mode}   : put the given table into sync
LINEAR              : test the linear move function on axes 1,2,3
HOLD                : feedhold an axis that is in motion
RELEASE             : release an axis that is in feedhold
HOLDQ               : stop but retain any queued moves
RELEASEQ            : allow queued moves to be executed after HOLDQ
LACE                : dump a canned profile into table 0
DUMPTTEST          : test of cam table allocation-deallocation
FAULT               : cause a user interrupt on the selected axis
FREECAM {table}    : de-allocate the specified cam table
STRIP               : display status of strip chart info
FLUSH               : flush the move queue for the current axis
MEM                 : get free memory available on axis card
HELP                : display this screen
?                   : display this screen
^R                  : repeat last command

AXIS 01>

```

Figure 2-5. Debug960 Help Screen #2

## 2.5. Configuring Axes

On initial power up or reset of the axis processor of the system, all axis parameters and configuration data are cleared and reset to their internal defaults (a summary of parameters and their default values appear in Appendix A). To re-initialize the axis card, the user must specify axis configuration parameters such as axis type, feedback channel and D/A channel. Fault masks and other axis parameters are also important for re-initializing the axis card.

Drives must be disabled before the user can configure an axis. To do this, enter SET DRIVE 0. This combination of commands disables the axis.



Configuring the axis feedback type can be achieved by issuing one of the commands explained in the sections that follow.

All the arguments presented for the following commands are optional. By typing the command name followed by pressing the <Enter> key, the default values for the arguments are loaded.



**Motors having  
Resolver/Inductosyn****2.5.1. CONFIG - Resolver/Inductosyn Axis**

To configure a DC brush or an AC brushless motor using Resolver/Inductosyn feedback, the following syntax applies:

```
CONFIG resolution poles offset bound R2Dchannel
```

where

<i>resolution</i>	Resolver to digital conversion resolution (10, 12, 14, or 16 bits). The default is 14 bits.
<i>poles</i>	Number of electrical poles for a brushless motor (32 max), set to zero for DC axis. The default is zero (0).
<i>offset</i>	Commutation offset for brushless motors (1,024 counts per 360° offset). The default is zero (0).
<i>bound</i>	Enable (bound=1), disable (bound=0) software limits. The default is zero (0).
<i>R2Dchannel</i>	Resolver-to-digital (R/D) conversion channel (1 to 16 depending on number of installed R/D cards) for position feedback. The default is the current axis.



If specifying the number of bits for the resolution, it is necessary to also specify a value for the number of poles. Omission of any or all of the above arguments will indicate that the default value is to be used.

**Encoder  
Configuration****2.5.2. CONFIGENCODER - Encoder Axis**

To configure an axis as DC brush with encoder feedback, the following syntax applies:

```
CONFIGENCODER lines encchannel D2Achannel bound
```

where

<i>lines</i>	Encoder counts per motor rev. The default is 2,000.
<i>encchannel</i>	Encoder channel for servo position feedback. The default is the current axis.
<i>D2Achannel</i>	Digital-to-analog (D/A) converter channel. The default is the current axis.
<i>bound</i>	Enable (bound=1), disable (bound=0) software limits. The default is zero (0).

### 2.5.3. CONFIGHALL - Hall effect

To configure a DC or AC brushless motor using Hall effect and encoder feedback for commutation, the following syntax applies:

```
CONFIGHALL lines cyclelines offset commchan encchannel
```

where

<i>lines</i>	Encoder counts per motor revolution. The default is 6,000.
<i>cyclelines</i>	Encoder counts per electrical cycle. If set to zero (0), commutation defaults to six step. The default is 1,500.
<i>offset</i>	Commutation offset for CONFIGHALL is specified in degrees ( $-360^{\circ} \leq \text{offset} \leq 360^{\circ}$ ). The units are equal to 0 through 1,024 or 0 to $360^{\circ}$ .
<i>commchan</i>	Encoder channel used to provide Hall effect and encoder commutation data. The default is the current axis.
<i>encchannel</i>	Encoder channel for servo position feedback. The default is the current axis.

*Setting Current*

## 2.6. Parameter Operation Commands

There are four basic commands within the Debug960 Utility that allow the user to get, set and monitor axis parameters and servo states. These commands are outlined in the following sections.

### 2.6.1. Viewing Parameters

Three commands exist that allow the user to display the current value of an axis parameter or servo state. These parameters appear below.

#### 2.6.1.1. STAT/PARM - Display Axis Parameters and Servo States

The STAT or PARM command, covered under Axis Status of the Basic Command section earlier in this manual, causes all of the axis parameters and servo states to be displayed on the screen (please see Figure 2-2 and Figure 2-3).

*List of Parameters*

#### 2.6.1.2. GET - Displaying Parameter Values

The GET command permits the user to display the current setting of an individual parameter. To use this command type GET followed by any valid parameter name. Note that all valid parameter names are listed with the STAT or PARM command.

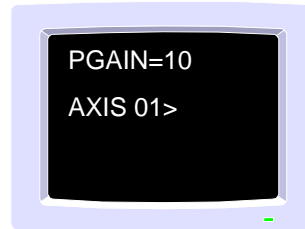
*Single Parameter*

**Example**

To display the current setting of the PGAIN parameter, type the following command at the axis prompt and hit the <Enter> key:

```
GET PGAIN
```

Debug960 responds with the following display:



After executing the GET command, the current axis prompt is displayed and further commands may be entered.

**Bad Parameter Name**

The PGAIN value shown is for demonstrative purposes only, since the display always reflects the current system setting.

Specifying an invalid parameter name when using the GET command causes Debug960 to respond with a line of question marks (?????) followed by the current axis prompt. This indicates that the given parameter name was not recognized.

**2.6.1.3. M - Monitor**

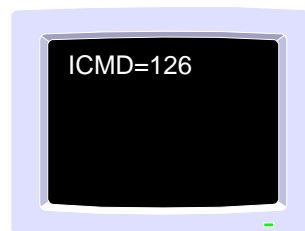
The M command allows the user to watch dynamic axis parameters and servo states such as axis position, position error, torque command, etc. To invoke this command, type the letter M followed by any valid parameter name and hit the <Enter> key. The value of the parameter is updated on the screen until the user strikes another key, at which point the axis prompt returns. Once again, Debug960 is ready to accept additional commands.

**Example**

To monitor the current value of the torque command, type the following command at the axis prompt and hit the <Enter> key:

```
M ICMD
```

Debug960 responds with the following display:



To terminate the monitoring of the torque command, strike any key. Debug960 responds with the current axis prompt so that the user may continue to enter commands.

*End Torque  
Command*

The ICMD value shown is for demonstrative purposes only, the display will reflect the current system setting.



Adding an X to the M monitor command (e.g., MX) allows the user to monitor parameters. This command displays the variable in a hexadecimal format. This command is useful for checking bit mask type variables (e.g., auxiliary mask, disable mask, fault mask, interrupt mask, abort mask and halt mask).

*Monitor  
Parameters*

Specifying an invalid parameter name when using the GET command causes Debug960 to respond with a line of question marks (?????) followed by the current axis prompt. This indicates that the given parameter name was not recognized.

### 2.6.2. SET - Editing Parameters

Only the SET command permits the user to change system parameters. Note that while all parameters and servo states are readable with the STAT, PARM, GET and M commands, they can not all be modified via the SET command. Parameters such as the servo status (STATUS) and torque command (ICMD) cannot be changed via the SET command. If a read-only parameter is specified on the command line with the SET command, Debug960 responds with a line of question marks (?????) followed by the current axis prompt.

*Change  
Parameters*

To invoke the SET command, type SET followed by any valid axis parameter name and a parameter value followed by hitting the <Enter> key.

*Example*

To disable the axis drive, enter the following command:

```
SET DRIVE 0
```

Debug960 then responds with the current axis prompt and is ready to accept further commands.

### 2.7. Motion Commands

Basic axis motion can be commanded from the axis prompt within Debug960. The purpose for commands is to send an axis home, start continuous motion and to make absolute and relative moves. Commands also exist that abort or halt all motion in progress (see halting/aborting motion).

*Command Axis  
Motion*

### 2.7.1. HOME - Homing Axes

#### *Axes Home*

Two types of axis homing are available from Debug960. The first home routine moves the axis at a user specified feedrate and a user specified direction until it encounters the home limit. The home position is then determined as the first Resolver/Inductosyn null or encoder marker found with the home limit active. The syntax for this command is as follows:

HOME *dir feedrate*

where

if *dir* ≥ 0                    home in CW direction

if *dir* < 0                    home in CCW direction

*feedrate*                    occurs in counts/sec.

An alternate home routine defines the home position as the first Resolver/Inductosyn null or encoder marker out of the limit. This routine moves at a user specified feedrate and a user specified direction until it reaches the home limit. At this point, the direction of motion is reversed and the home position defined as the first Resolver/Inductosyn null or encoder marker out of the limit. The syntax for this command follows.

HOMEALT *dir feedrate*

See the above HOME command for a complete description of the applicable arguments.

### 2.7.2. INDEX - Incremental Motion

#### *Increment Steps of Motion*

Incremental motion is a result of using the INDEX command. Position displacements relative to the current axis position are made by specifying a displacement and feedrate. Note that the feedrate is always positive. Negative velocities are attained by specifying negative displacements. The syntax for this command is as follows:

INDEX *displacement feedrate*

where

*displacement*                is specified in counts

*feedrate*                    occurs in counts/sec.

#### *Example*

INDEX 10000 2000                    ; move 10,000 counts at 2,000 counts per sec.

INDEX -5000 4000                    ; move -5,000 counts at 4,000 counts per sec.

#### *Example*

### 2.7.3. MOVETO - Absolute Motion

Absolute moves are a result of using the MOVETO command. Displacements to positions relative to the home position are made by specifying an absolute position and feedrate. Note that the feedrate is always positive. Negative velocities are obtained by specifying an absolute target position which is less than the current absolute position. The syntax for this command is as follows:

```
MOVETO position feedrate
```

where

*position* specifies an absolute target in counts  
*feedrate* occurs in counts/sec.

MOVETO 12345 2000 ; move to absolute position 12,345 at 2,000 counts per sec.

*Target Position*

*Example*

### 2.7.4. START - Continuous Motion

Continuous motion, where an axis is commanded to run at a user specified velocity and direction, can be commanded via the START command. Start commands take as arguments both the desired feedrate and direction. Feedrates are always positive with negative directions being specified via the direction (see below) argument. The syntax for this command is as follows:

```
START dir feedrate
```

where

if  $dir \geq 0$  move in CW (+) direction  
 if  $dir < 0$  move in CCW (-) direction  
*feedrate* occurs in counts/sec.

START-1 10000 ; move in the CCW direction at 10,000 counts/sec.

*Move  
Continuously*

*Example*

### 2.7.5. HALT/ABORT - Halting/Aborting Motion

Debug960 uses two commands to stop commanded motion: HALT and ABORT. The HALT command causes the axis to come to a controlled stop using the deceleration ramp time specified in the parameter list. The ABORT command causes an abrupt (uncontrolled) and immediate stop to motion. Each of these commands are invoked by entering their names at the axis prompt.

*Controlled and  
Uncontrolled  
Stops*

## 2.8. Analog and Digital I/O

### *Inputs/Outputs*

Several commands manipulate the optional digital and analog I/O cards available with the U30, U31 and U600. These commands permit the configuration, retrieval and setting of the I/O for both the analog and digital option cards. Make sure that the desired option card is present within the system. Attempting to access optional VME cards which are not in the systems card cage results in a VME bus (32 bit) error. A VME bus error requires resetting the axis card and reloading the firmware. To reset the system, press the RESET button on the 960 card and reinstall any previously set parameters.

### 2.8.1. Xycom Digital I/O

#### *Allowable Xycom Cards*

The Xycom Digital I/O card consists of 32 bits of digital I/O that the user can configure as inputs or outputs on blocks of 8 bits (one byte). The axis firmware supports up to four (4) Xycom cards in a system for a maximum of 128 bits of digital I/O. Configuration of the card(s) and the setting and reading of the digital I/O can all be performed via Debug960.

#### 2.8.1.1. INITXYCOM - Configuration

#### *Configure Xycom Cards*

The Xycom card can be initialized by two methods. The first method is an attempt to read or write from the card via the commands described in the following section. If a read or write occurs to an unconfigured card, the axis firmware automatically configures the card for 16 inputs and 16 outputs. With the automatic configuration, the inputs are located on bits 0 to 15 and the outputs on bits 16 to 31. Note that this method only configures one card in the system. This method will not configure multiple cards, therefore any attempt to read or write to unconfigured digital I/O above bit number 31 will yield unpredictable results.

The card(s) may also be configured using the INITXYCOM command. This command accepts, as arguments, the number of cards in the system and the number of input bytes. The syntax follows.

```
INITXYCOM InputBytes NumberCards
```

#### *Example*

All inputs are mapped to the lowest bit locations. For example, if four (4) cards are present in the system and three (3) bytes (24 bits) are configured as inputs, they will be mapped into bits 0 through 23.

```
INITXYCOM 3 4
```

The result of this configuration is three (3) input bytes (24 bits) and 13 output bytes (104 bits). The number of output bytes and bits is calculated by subtracting the specified number of inputs from the total number of bytes (in this example 4x4 bytes = 16 bytes). Therefore, 16 total bytes equal three (3) input bytes which equal 13 output bytes.



### 2.8.1.2. READXYCOM - Reading Digital Inputs

Digital inputs can be read through the READXYCOM command. This command accepts, as an argument, the bit location to be read. An example appears below.

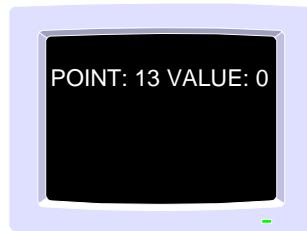
*Reading Inputs*

```
READXYCOM BitNumber
```

Assuming the configuration of the card matches the configuration shown in the previous example, bit location 13 could be read as follows:

```
READXYCOM 13
```

Debug960 returns the following data:



The actual value returned is a reflection of the current status of the digital input.

The user may also apply the m (monitor) command to monitor bit zero (0). To do this, enter m xycom 0 followed by hitting the <Enter> key.

### 2.8.1.3. WRITEXYCOM - Writing Digital Outputs

Digital outputs can be set through this command. This command accepts, as arguments, the bit location (Point) and the value (Value) to be output (0 or 1).

*Writing to Outputs*

```
WRITEXYCOM Point Value
```

Assuming that the configuration of the card matches the configuration shown in the previous example, bit location 32 could be set to a logical one (1) as follows:

```
WRITEXYCOM 32 1
```

Debug960 returns with the axis prompt.

Digital outputs can be read via the READXYCOM value to verify their current state.



## 2.8.2 Matrix Analog Inputs

### *Optional Input Card*

The optional Matrix analog input card supports eight (8) differential or 16 single ended 12 bit analog inputs.

### 2.8.2.1. INITMATRIX - Channel Operation for Analog Inputs

### *Channel Operation*

The Matrix card can be configured for 16 channel operation via the INITMATRIX16 command or for eight (8) channel operation via the INITMATRIX command. The card must also be physically set for 16 or eight (8) channel mode via the jumpers located on the card. Consult the Matrix card documentation for appropriate jumper configuration.

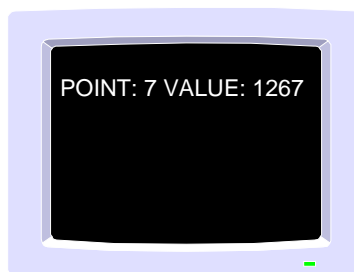
### 2.8.2.2. READMATRIX - Reading Analog Inputs

### *Analog Input Reading*

The READMATRIX command permits the reading of analog inputs. This command accepts, as arguments, the analog channel to be read (0 to 15 or 0 to 7 depending on the configuration). Data is returned as a signed 16 bit quantity with zero (0) fill for the lower four (4) undefined bits of the 12 bit quantity. By entering the following command,

```
READMATRIX 7
```

Debug960 returns the following data:



The actual returned value is a reflection of the current status of analog input.

## 2.9. System Diagnostics

These commands inform the user of the activity occurring on the axis processor.

### 2.9.1. COMM - Communication Testing

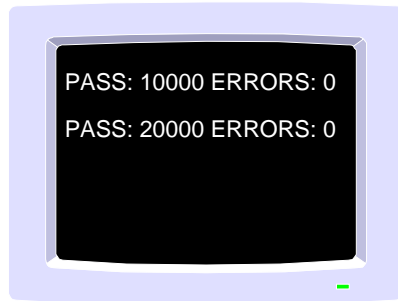
### *Testing Channel Communication*

Debug960 offers a utility to test the integrity of the communication channel across the VMEbus backplane. This utility transmits packets of data across the VME with an accompanying checksum. This checksum verifies the integrity of the received data. Any errors encountered during the transmission are reported on the screen. To initiate the test the user must type COMM at the axis prompt. For example,

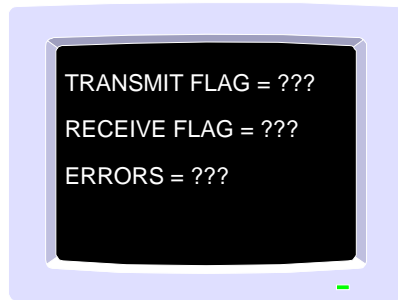
### *Example*

```
COMM
```

Debug960 responds by displaying the following message for every successful 10,000 iterations of the test routine:



Should an error occur, Debug960 will display the following message:



If the communication fails completely, the above message would scroll the screen for the duration of the test and all the test iterations would be failures.



Every error identified increments the error count that is displayed every 10,000 iterations. If ten (10) errors had occurred, the following message would appear on the display.



*Example*

*Stop Channel  
Communication  
Test*

To stop the communication test strike any key and the test will halt after the completion of the current 10,000 cycle iteration.

## 2.10. Fault Handling

### *Enable/Disable Faults*

Each system uses a series of "masks" to enable or disable faults and to invoke protective measures (e.g., drive disabling or halting axis motion). These masks are FAULTMASK, HALTMASK, DISABLEMASK, INTMASK, AUXMASK and ABORTMASK.

Note that the occurrence of axis faults and programming errors are not immediately reported to the user through Debug960. Notification does not occur because Debug960 would have to set the INTMASK to enable interrupts and claim that interrupt under OS/2. Enabling and claiming the interrupt may conflict with a user's application causing undesirable and unpredictable results.

**Table 2-1. General Mask Bits**

Bit	Fault Type	Description
0	Position Error Limit	difference between instantaneous commanded position and actual position exceeds the amount specified in the POSERRLIMIT parameter.
1	RMS Current Limit	average current exceeds the amount specified in the IAVGLIMIT parameter.
2	CW Hard Limit	clockwise EOT fault occurred when the system encountered the CW limit switch.
3	CCW Hard Limit	counter-clockwise EOT fault occurred when the system encountered a CCW limit switch.
4	CW Soft Limit	clockwise software EOT fault occurred when the user commanded an axis to move beyond the position specified in the CWEOT parameter.
5	CCW Soft Limit	counter-clockwise software EOT fault occurred when the user commanded the axis to move beyond the position specified in the CCWEOT parameter.
6	Drive Fault	system asserts the drive fault input for the axis. However, after clearing the drive fault input, this bit continues to reflect the fact that the fault occurred.
7	Feedback Fault	system asserted the feedback failure input from the feedback associated with the axis. This typically occurs when the feedback device is not functioning properly.
8	Programming Fault	axis processor received an invalid command from the CPU card. Any type of programming error causes this fault to occur.

Table 2-1. General Mask Bits (Continued from Page 2-18)

Bit	Fault Type	Description
9	Master Feedback Fault	system asserted feedback failure input from the feedback channel associated with the axis configured as a master. This typically occurs when the feedback device on the master axis is not functioning properly.
10	Home Fault	system encountered a homing fault. This typically occurs for two reasons: while executing a home cycle upon detection of the home limit switch input and when the system encounters an end-of-travel limit switch before the first Resolver null or marker pulse.
11	User Fault	application has requested a fault be generated with the DMRSetUserFault( ) command. It provides a way to generate a VMEbus interrupt from the application program.
12	Velocity Trap	actual velocity exceeded the value specified in the VELTRAP parameter.
13	Velocity Command Trap	instantaneous commanded velocity exceeded the value specified in the VELCMDTRAP parameter.
14	Home Tolerance Fault	distance traveled from the time the system detected the marker pulse or the Resolver null, and the time at which the system encountered the home limit switch is less than the value specified in the HOMESWITCHTOL parameter. Typically, this occurs during a homing sequence.
15	Probe Fault	occurs each time the probe trigger causes the position to latch. This is useful for notifying the application program that position information is available.
16	CNC Fault	error occurred in the CNC process to which this axis is dedicated. This fault indicator is inclusive of all faults that the CNC processor might generate.
17	External Feedback Fault	difference between the integration of the velocity command and velocity feedback is greater than the FBWINDOW parameter.
18	Safe Zone	SAFEZONE is active and the axis has violated the defined safe zone.
19-31		Unused

*Enable/Disable  
Faults*

**2.10.1. FAULTMASK**

The faultmask shows if a fault is enabled or disabled on a particular axis. Setting a bit that corresponds to a given fault to one (1) enables the fault. Likewise, changing a bit of a related fault to zero (0) clears or disables that fault. Table 2-2 lists the supported faults and their corresponding bit locations.

To enable the position error limits, CW and CCW faults, set bits 0, 2 and 3. This corresponds to a binary 1101, hexadecimal D or decimal 13. Note that SET only accepts decimal values as inputs, therefore decimal 13 would be used with the SET command:

```
SET FAULTMASK 13
```

Once a set of faults are enabled, the remaining masks can be set to determine the action taken when a fault occurs. Table 2-2 explains each mask in detail.

**Table 2-2. Status Bit Map of 32 Bit Conditions**

Fault Type	Bit	Description
Position Error Limit	0	0 = normal operation, no sign of position error 1 = exceeded value of POSERRLIMIT parameter
RMS Current Limit	1	0 = normal operation, no current error 1 = exceeded value in IAVGLIMIT parameter
CW Hard Limit	2	0 = normal operation, no CW limit switch detected 1 = encountered CW limit switch
CCW Hard Limit	3	0 = normal operation, no CCW limit switch found 1 = encountered CCW limit switch
CW Soft Limit	4	0 = normal operation 1 = exceeded position specified in CWEOT parameter
CCW Soft Limit	5	0 = normal operation 1 = exceeded position specified in CCWEOT parameter
Drive Fault	6	0 = normal operation, no drive fault input on axis 1 = drive fault input occurred on the axis
Feedback Fault	7	0 = normal operation, no feedback failure 1 = asserted feedback failure input from the feedback channel associated with the axis
Programming Fault	8	0 = normal operation, no invalid commands were received 1 = axis processor received an invalid command from CPU processor

**Table 2-2. Status Bit Map of 32 Bit Conditions (Continued from Page 20)**

Fault Type	Bit	Description
Master Feedback Fault	9	0 = normal operation, no master feedback failure 1 = asserted feedback failure input from the R/D channel associated with the master axis
Home Fault	10	0 = normal operation, no homing fault encountered 1 = system encountered a homing fault
User Fault	11	0 = normal operation, no interrupt occurred 1 = generated a VMEbus interrupt on the program
Velocity Trap	12	0 = normal operation, no velocity setting exceeded 1 = exceeded value in VELTRAP parameter
Velocity Command Trap	13	0 = normal, no command velocity exceeded 1 = exceeded value in VELCMDTRAP parameter
Home Tolerance Fault	14	0 = normal operation, no difficulty encountered during homing sequence 1 = distance is less than value specified in HOMESWITCHTOL parameter
Probe Fault	15	0 = normal operation, no latch on probe trigger 1 = probe trigger caused position to latch
CNC Fault	16	0 = normal operation of the CNC process 1 = error occurred in CNC process
External Feedback Fault	17	0 = normal operation 1 = integration of velocity command and velocity feedback exceeds value of FBWINDOW parameter
Safe Zone	18	0 = normal operation, no violation of safe zone 1 = axis violated the defined safe zone
19-31		Unused

### 2.10.2. DISABLEMASK

Setting the DISABLEMASK bits causes the drives to be disabled when the fault occurs. The disable mask bits get “ANDed” with the fault mask bits. If any of the resultant bits test true (match), the U30, U31 or U600 disables the drive. The bit descriptions are the same as that listed in Table 2-2.

### 2.10.3. HALTMASK

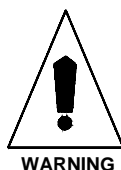
Setting the bits in the HALTMASK causes the axis in motion to decelerate to a stop in the time indicated by the DECEL parameter (see Section 2.12.20). The halt mask bits get “ANDed” with the fault mask bits. If any of the resultant bits test true (match), the controller disables the drive. The bit descriptions are listed in Table 2-2.

### 2.10.4. AUXMASK

Setting the bits in AUXMASK causes the auxiliary output to be set when a fault occurs. This output helps to engage brakes on a vertical axis. The auxiliary mask bits get “ANDed” with the fault status bits. If any of the resultant bits test true, the U30, U31 or U600 sets the auxiliary output. The auxiliary mask bit descriptions are the same as those defined in Table 2-2.

### 2.10.5. ABORTMASK

Setting the bits in the ABORTMASK causes the axis to come to an abrupt (uncontrolled) stop without any programmed decel. The abort mask bits get “ANDed” with the fault status bits. If any of the resultant bits test true (match) the U30, U31 or U600 executes a “fast” feedhold.



When setting these bits be especially careful in that a direct stop to motion may cause physical harm to the user and/or damage to the equipment.

### 2.10.6. INTMASK

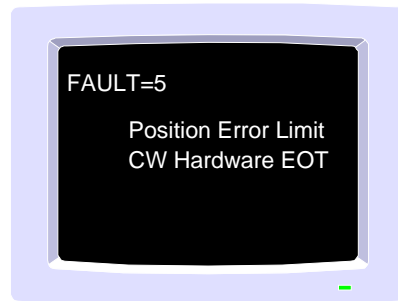
Setting the bits in the INTMASK causes an interrupt to be sent to the OS/2 operating system. Typically, this interrupt is trapped by an application that runs under OS/2 which is used to notify the user that a fault occurred. The interrupt mask bits get “ANDed” with the fault status bits. If any of the resultant bits test true (match), the U30, U31 or U600 sends an interrupt to the OS/2 operating system.



### 2.10.7. Acknowledging Faults

Debug960 does not inform the user when the generated fault occurred. The user must query the system with the GET FAULT command. GET FAULT will return any active faults. Faults are acknowledged or cleared by using the SET FAULT command.

If a position error and CCW limit fault are active, GET FAULT displays the following.



To acknowledge and clear both of these faults, at the axis prompt, the user would enter:

```
SET FAULT 5
```

Once a fault is cleared, there is no longer a record that it ever occurred.

The only way to remove fault conditions from the GET FAULT display is to acknowledge them.



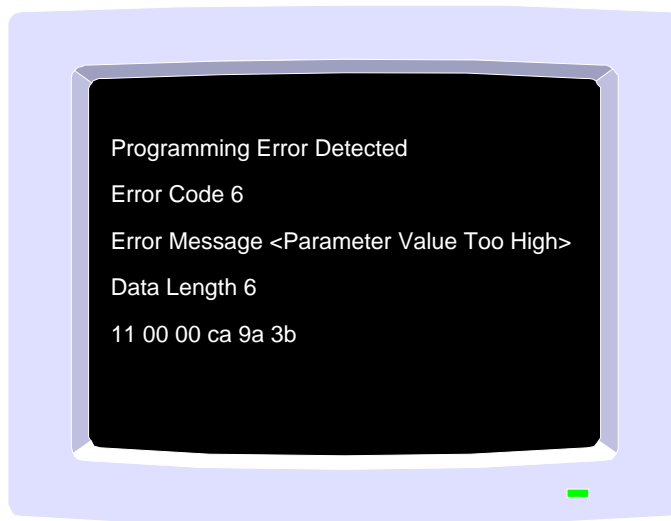
The user can also monitor a fault using the m (monitor) command.

### 2.11. Programming Errors

Programming errors result when parameters are set outside their minimum or maximum allowable limits. They also occur when incorrect arguments are specified in configuration commands.

To display programming errors, type PROG at the axis command prompt and hit the <Enter> key. If an error is present, Debug960 responds with diagnostic information that pertains to the type of error and the data received by the axis card.

Assume a programming error has been generated from setting a proportional gain (KP) parameter beyond its maximum limit. Typing PROG at the axis prompt and hitting the <Enter> key will reveal the following message.



Only the error message string provides the user with useful information. The error code, data length and data string are used for internal diagnostic purposes only.

Typing PROG also clears and acknowledges any programming errors present. Note that if multiple programming errors occurred, only the last error will be displayed by the PROG command. To allow all errors to be recorded it is up to the user to include a statement in their program that (e.g., setting the bit masks) allows the user to stop motion, do a status inquiry and view each error as it occurs.

## 2.12. Parameter Descriptions

The following list reflects the active parameters at the time this document was created. Newer revisions of axis firmware may include more parameters. Consult Aerotech's Systems Engineering division for additional information on any parameters not defined within this list.

### 2.12.1. ABORTMASK {Read/Write}

The Abort Mask parameter controls which faults will cause the axis to abort all motion. When motion is aborted, the axis is stopped immediately without regard to the DECEL parameter setting. The current position error is also set to zero (0).

The value specified represents a bit mask where each bit corresponds to a specific fault. Setting a bit to one (1) causes the axis to abort when that particular fault is generated (assuming the corresponding bit in the FAULTMASK parameter is set). See Table 2-2 for a description of the associated bits.

**2.12.2. ACCEL {Read/Write}**

The Acceleration parameter controls the time that it takes to change from the current velocity to a greater velocity during the execution of a motion command. The accepted values can range between 0 and 100,000 milliseconds. The default value is zero (0) msec.

**2.12.3. ACCELMODE {Read/Write}**

The Acceleration Mode parameter permits the selection of either linear or sinusoidal (1-cosine) ramping during the execution of a motion command(s). A value of zero (0) selects sinusoidal ramping and a value of one (1) selects a linear ramp. The default value is zero (0).

**2.12.4. ACCELRATE {Read/Write}**

The Acceleration Rate parameter sets the rate of acceleration, while the ACCELMODE parameter specifies the rate-based ramping. The valid range is between 1 and 10,000,000 machine counts/msec<sup>2</sup>. The default is 100,000 counts/msec<sup>2</sup>.

**2.12.5. AFFGAIN {Read/Write}**

This parameter allows the user to specify the Acceleration Feedforward Gain used in the acceleration loop of the selected axis. The valid programming range is 0 to 1,000. The default value is zero (0).

**2.12.6. ALT\_STATUS {Read Only}**

This parameter is for internal use only.

**2.12.7. AUX {Read/Write}**

The Auxiliary parameter allows the user to enable and disable the auxiliary output of a selected axis. This output is commonly used for a motor brake. It can be used in conjunction with the AUXMASK parameter to allow the output to change state on an axis fault. A value of zero (0) disables the output, while a one (1) enables it. The default value is zero (0).

**2.12.8. AUXMASK {Read/Write}**

The Auxiliary Mask parameter determines which faults will turn off the auxiliary output. AUXMASK is a bit mask in which each bit corresponds to a specific fault. Setting a bit to one (1) turns the auxiliary output off when that particular fault is generated (assuming the corresponding bit in the FAULTMASK parameter is set). See Table 2-2 for a description of the associated bits.

**2.12.9. AUXOFFSET {Read/Write}**

The Auxiliary Offset parameter refers to an offset that is applied to the master position of the auxiliary output table associated with an axis. The point at which the table begins and ends may be shifted forward or backwards by setting this value. The valid range for programming this parameter is all signed 32 bit numbers. The default value is zero (0) machine counts.

**2.12.10. AVGVEL {Read Only}**

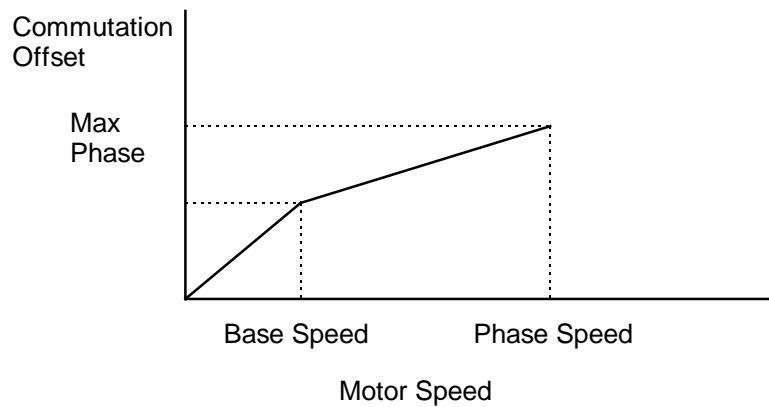
The Average Velocity parameter is the velocity averaged over an AVGVELTIME interval. This parameter is continuously updated according to the setting of the AVGVELTIME parameter.

**2.12.11. AVGVELTIME {Read/Write}**

The Average Velocity Time parameter specifies the time over which the velocity is averaged for the AVGVEL parameter (above). The valid range for programming this parameter is between 1 and 1,000 milliseconds (in 10 msec increments). The default value is 1,000.

**2.12.12. BASE\_SPEED {Read/Write}**

This parameter, as well as the MAX\_PHASE and PHASE\_SPEED parameters, describes the desired commutation angle generated each time the user aligns the feedback device with an AC brushless motor. As can be seen from the following graph, these parameters define the slope of the commutation angle at various motor speeds. The valid range for programming this parameter is between 0 and 10,000,000; however, this is only valid for internal use.



**Figure 2-6. Defined Slope of Commutation offset at Motor Speed**

**2.12.13. BLOCKMOTION {Read/Write}**

The Block Motion parameter causes the axis to ignore any commands that initiate motion. The only exception is when the axis is already under control of a sync table. While blocking motion, the axis accepts commands that stop motion so that an axis already in motion may be stopped. A value of zero (0) causes motion commands to be processed, while a value of one (1) causes them to be ignored. The default value is equal to zero (0).

**2.12.14. CAMOFFSET {Read/Write}**

The Cam Offset parameter applies an offset to the master position without modifying the master position. This parameter moves the master portion of the cam table forward or backwards. The valid range for this parameter is 0x80000001 to 0x7fffffff (all signed 32 bit integers). The default value is equal to zero (0).

**2.12.15. CAMPOINT {Read Only}**

The Cam Point parameter represents the point in the cam table that is used to access the cam point.

**2.12.16. CAMPOSITION {Read Only}**

The Cam Position parameter is the value (along the master axis) of the current position (in what units ??) within the cam table.

**2.12.17. CCWEOT {Read/Write}**

This parameter corresponds with the counterclockwise end-of-travel limit. The axis card will not generate a position that exceeds this value. The CW\_FAULT fault will be generated if the user attempts to command a position beyond this value. The valid range for this parameter is between 0x80000001 and 0x7fffffff machine counts (all signed 32 bit integers). The default value is 07FFFFFFFH ( $2^{31-1}$ ) counts (-2147483647).

**2.12.18. CLOCK {Read/Write}**

The Clock parameter allows the user to get or set a one millisecond clock. The clock continues to count from the specified time.

**2.12.19. CWEOT {Read/Write}**

This parameter corresponds with the clockwise end-of-travel limit. The axis card will not generate a position that exceeds this value. A CW\_FAULT is generated if the user attempt to command a position beyond this value. The valid range for this parameter is 0x80000001 to 0x7fffffff machine (all signed 32 bit integers). The default value is 07FFFFFFFH ( $2^{31-1}$ ) counts (2147483647).

**2.12.20.                    DECEL {Read/Write}**

The Deceleration parameter controls the time that it takes to change from the current velocity to a slower velocity during the execution of a motion command. The valid range for this parameter is between 0 and 100,000 milliseconds. The default value is zero (0) msec.

**2.12.21.                    DECELMODE {Read/Write}**

The Deceleration Mode parameter permits the user to select either linear or sinusoidal (1-cosine) ramping during the execution of a motion command(s). A value of zero (0) selects sinusoidal ramping, while a value of one (1) selects a linear ramp. The default value is zero (0).

**2.12.22.                    DECELRATE {Read/Write}**

The Deceleration Rate parameter sets the rate of deceleration, while the DECELMODE parameter specifies the rate-based ramping. The valid range for programming this parameter is between 1 and 10,000,000 machine counts per second squared. The default value is 100,000 count/msec<sup>2</sup>.



The user may also specify deceleration parameters from within a parts program.

**2.12.23.                    DISABLEMASK {Read/Write}**

The Disable Mask parameter determines which faults disable the axis. This parameter represents a bit mask where each bit corresponds to a specific fault. Setting a bit to one (1) disables the axis on that particular type of fault (assuming the corresponding bit in the FAULTMASK parameter is set). See Table 2-2 for a description of the associated bit masks.

**2.12.24.                    DRIVE {Read/Write}**

The Drive parameter enables and disables a selected axis. A value of zero (0) disables the drive, while a value of one (1) enables it. The default value is zero (0).

**2.12.25.                    ECHO {Read/Write}**

The Echo parameter Echo parameter allows the user to set a “dummy” parameter. It has no effect on the operation of the controller, but may be used to test communications with the axis processor or as a temporary holding area.

**2.12.26. FAULT {Read/Write}**

The Fault parameter allows the user to view axis faults and clear those that no longer exist. The parameter is a bit mask where each bit corresponds to a specific fault. When writing to this parameter, the axis card attempts to clear all faults corresponding to the bits set in this mask. Bits set to zero (0) have no effect on the system. See Table 2-1 and Table 2-2 for a description of the associated bits.

**2.12.27. FAULTMASK {Read/Write}**

The Fault Mask parameter determines which faults are active. This parameter is a bit mask where each bit corresponds to a specific fault. Setting a bit to one (1) makes that particular fault active. Conversely, clearing a bit causes that fault to be ignored.

**2.12.28. FBWINDOW {Read/Write}**

While processing motion commands, the axis processor integrates both the position command and the velocity command. The Feedback Window parameter permits the user to specify a maximum amount by which these two integrations may differ. If this difference exceeds the amount specified by this parameter, a feedback fault is generated. The valid range for this parameter is between 0 and 1,000,000 machine counts. The default value is zero (0) and disables the feedback fault monitoring.

**2.12.29. FEEDRATEMODE {Read/Write}**

The Feedrate Mode parameter specifies if the axis is subject to feedrate override controls. A value of zero (0) disables feedrate override controls, while a value of one (1) makes the axis subject to feedrate override. The default value is zero (0).

**2.12.30. HALTMASK {Read/Write}**

The Halt Mask parameter controls which faults should cause an axis to halt. When motion is halted, the axis gracefully decelerates to zero velocity in the amount of time specified in the DECEL parameter. Position error tracking is unaffected.

The value specified for this parameter is a bit mask where each bit corresponds to a specific fault. Setting a bit to one (1) causes the axis to halt when that particular fault is generated (assuming the corresponding bit in the FAULTMASK parameter is set).

**2.12.31. HOMESWITCHPOS {Read Only}**

The Home Switch Position parameter is the Resolver home position.

**2.12.32. HOMESWITCHTOL {Read/Write}**

To ensure the accuracy of a homing sequence, there must be some minimum distance between the home limit switch and the home marker pulse. Otherwise, the axis processor may miss the first marker pulse and use the second marker as the home position (before home offset). The distance required depends upon such factors as feedback resolution and home feedrate.

The Home Switch Tolerance parameter specifies the minimum distance, in machine counts, that must exist between the home limit and the marker pulse. If this distance is not present, the HOME\_SWITCH\_TOLERANCE fault occurs. This parameter has a valid range between 0 and 16,384 machine counts. The default value is zero (0).

**2.12.33. HOMEVELMULT {Read/Write}**

The Home Velocity Multiplier is the scale factor used to scale the speed for the homing position. The valid range for programming this parameter is between 0 and 1,000. The default value is 100. For example:

*Example*

100 = 100% speed for homing.

*Example*

10 = 10% speed for homing.

**2.12.34. IAVG {Read Only}**

The Average Current is the current averaged over the IAVGTIME interval. This parameter is continuously updated according to the IAVGTIME parameter setting.

**2.12.35. IAVGLIMIT {Read/Write}**

This parameter detects over-current operation. The instantaneous current command is averaged over a time specified by IAVGTIME (below) parameter. If the RMS average exceeds the value of this parameter an IAVG fault is generated. The valid range for programming this parameter is between 0 and 0x7fff. The default value is 32,767 (00007fff).

**2.12.36. IAVGTIME {Read/Write}**

This parameter defines the time over which the instantaneous current command is averaged. This parameter is used in conjunction with the IAVGLIMIT parameter to detect over-current operation. The valid range for programming this parameter is between 10 and 4,000 milliseconds. The default value is 1,000 msec.

**2.12.37. ICMD {Read Only}**

This Instantaneous Current Command parameter is the instantaneous current command.



**2.12.38.                   IMAX {Read/Write}**

This parameter sets the maximum value of current command outputted by the D-to-A card. This is also referred to as current limit. The user units are (10 volts/32,767 counts). The valid range for this parameter is between 0 and 32,767 (0x7fff). The default value is 32,767.

**2.12.39.                   INPOSLIMIT {Read/Write}**

The In-position Limit parameter defines the in-position band. If the observed position plus or minus the position command is within the in-position band and the axis is done, the axis in-position status bit becomes active. The valid range for this parameter is between 0 and 65,536 machine counts. The default value is 65,535 counts.

**2.12.40.                   INTMASK {Read/Write}**

The Interrupt Mask parameter determines which faults cause the system interrupts. It is a bit mask where each bit corresponds to a specific fault. Setting a bit to one (1) causes an interrupt to be generated when that particular fault is generated (assuming the corresponding bit in the FAULTMASK parameter is set). See Table 2-1 and Table 2-2 for descriptions of the associated bit masks.

**2.12.41.                   IOLEVEL {Read/Write}**

The Input/Output Level parameter specifies the on or off state of each of the inputs into the axis processor card and outputs from the axis processor card. IOLEVEL allows the user to normally open/closed switches and active high/low fault signals. This value is a bit mask where only the six least significant bits are valid. These bits correspond with the least significant six (6) bits of the STATUS parameter. A one (1) in a particular bit means that the input/output is active high and a zero (0) means it is active low. The valid range for this parameter is 0 to 0x3f. The default value is 63 (03FH), corresponding to all active high signals.

**2.12.42.                   IVEL {Read Only}**

The Instantaneous Velocity parameter is the instantaneous velocity.

**2.12.43.                   KI {Read/Write}**

The Integral Gain parameter gets and sets the integral gain of the velocity loop for a selected axis. Refer to the *Unidex 31 Hardware Manual* for a description of how this parameter functions in the servo. The valid range for this parameter is between 0 and 5,000,000. The default value is 2,000.

**2.12.44. KP {Read/Write}**

This Proportional Gain parameter shows the current proportional gain of the velocity loop for a selected axis. Refer to the *Unidex 31 Hardware Manual* for a description of how this parameter functions in the servo. The valid range of this parameter is between 0 and 10,000,000. The default value is 10,000.

**2.12.45. MASTERLEN {Read/Write}**

The Master Length parameter provides a means of implementing a user-specified rollover of the master position. It does not necessarily have to match the length of the master portion of a cam table for a slave axis. The valid range for programming this parameter is between 0 and 0x7ffffff (must be positive). The default value is zero (0).

**2.12.46. MASTERPOS {Read/Write}**

The Master Position parameter corresponds to the observed master position of the master feedback channel associated with an axis. It should be noted that an axis has both its own position and a master position when the DMRConfigMaster has been issued to an axis. The valid programming range is 0x80000001 to 0x7ffffff (all signed 32 bit integers).

**2.12.47. MASTERABS {Read Only}**

The Master Absolute Position parameter corresponds to the observed master position of the master feedback channel associated with an axis. It should be noted that an axis has both its own position and a master position when the DMRConfigMaster has been issued to an axis. The valid range for programming this parameter is between 0x80000001 and 0x7ffffff (all signed 32 bit integers).

**2.12.48. MASTERRES {Read Only}**

The Master Resolver parameter is the specified (not incremental) Resolver position for the master axis at the given time.

**2.12.49. MAX\_PHASE {Read/Write}**

This parameter, as well as the BASE\_SPEED and PHASE\_SPEED parameters, describes the desired commutation profile generated each time the user aligns the feedback device with an AC brushless motor. As can be seen from the graph, in Figure 2-6, these parameters define the slope of the commutation curve at various motor offset speeds. The valid programming range is between 0 and 360; however, this can only be set at the factory.

**2.12.50. MOTIONSTATUS {Read Only}**

The Motion Status parameter is read only and returns a value that the user can decode to determine what bits are active (on) relative to the operational mode of the trajectory generator. To access the value of this parameter enter GET MOTIONSTATUS followed by pressing the <Enter> key. Table 2-2, on the following page, lists the supported faults and their corresponding bit locations.

**Table 2-3. Motion Status Bits**

Bit	Status Type	Description
0	Move Direction	indicates the direction an axis is currently moving. ON corresponds to the clockwise direction and OFF signifies counter-clockwise. The state of this bit is undetermined unless the axis is in motion.
1	Moving	alerts the user when the associated axis is currently in motion. ON signifies the axis is in motion. OFF implies that axis trajectory generation is complete (upon completion the bit gets set).
2	Accel Phase	determines if the axis is currently within the acceleration portion of the motion command being executed. The acceleration phase is the period of time when the trajectory generator is changing the velocity from the current velocity to the commanded velocity. Therefore, if the current velocity is 10 counts/msec and the commanded velocity is 5 counts/msec, the value of the accel phase flag changes while the axis slows to the new velocity (5 counts/msec). Through the run constant phase or the deceleration phase, the user can terminate this phase. Note that this bit is of no use if the corresponding axis is currently being controlled via a cam table.
3	Decel Phase	determines if the axis is currently within the deceleration portion of the motion command being executed. This phase is the time period in which the axis is slowing from the commanded velocity to a complete stop. ON appears while the axis is decelerating. OFF appears if the axis trajectory generation is complete (when the done bit is set). When controlling the corresponding axis through a cam table, this bit has no significance.
4	Homing	determines if the associated axis is currently executing a homing sequence. ON indicates a home cycle is in progress.

**Table 2-3. Motion Status Bits (Continued from Page 33)**

Bit	Status Type	Description
5	Feedrate Override	signifies if manual feedrate override (MFO) controls are active on an axis. ON signifies that these controls are enabled for the axis.
6	Profile Mode	shows if the associated axis is currently executing moves based on the data found in the profile queue for a particular axis. ON indicates that profile queue data is being executed.
7	Sync Mode	indicates if the associated axis is being controlled by another (master) axis. The synchronization may be cam mode or ratio mode. ON signifies synchronization is enabled.
8	CAM Table Enable	shows if the associated axis is synchronized with another axis by means of a cam table. This bit is of no significance if the axis is not being controlled by another axis (sync mode bit above). ON shows that a cam table has been activated with this axis acting as the slave axis.
9	Homing Direction	indicates the direction the associated axis is currently traveling. It is only valid while the axis is executing its homing sequence (homing bit). ON corresponds to CW direction and OFF signifies CCW motion.
10	Continuous Move	ON implies that the axis is currently executing a continuous motion command. OFF appears in the display at all other times.
11	Queued Command	shows if the associated axis is currently executing moves based on the data found in the motion queue for this axis. ON signifies that motion queue data is being executed.
12	Hold Active	designates if the associated axis is currently being held from movement. ON signifies the axis is being held.
13	Aux Mode	signifies if there is currently an active synchronized auxiliary output table associated with an axis. ON appears when an auxiliary output table is active.
14	Block Motion	indicates whether all commands initiating motion are blocked from execution on an axis. ON signifies that the system must ignore all motion commands.
15	Hold Queue	indicates whether execution of queued motion commands has been temporarily suspended. ON signifies suspended execution.

Table 2-3. Motion Status Bits (Continued from Page 34)

Bit	Status Type	Description
16	Disable Queued	provides the current status of queued motion command execution. ON implies that the system is to ignore all queued motion commands. Otherwise, the system places the commands into the queue and executes as soon as possible.
17	Halt Queued	provides data regarding the current state of the motion queue. ON indicates that the system halted execution of the motion queue. Upon halting the motion queue, the axis gracefully decelerates to zero (0) velocity in the amount of time specified in the DECEL parameter. This has no effect on the position error tracking. The motion queue is flushed when the halt queue field is ON. The halt process is caused by an active fault which is enabled in the HALTMASK parameter.
18	Abort Queued	similar to the halt queued bit. However, this indicator notifies the user that the system aborted execution of the motion queue. The motion queue is not flushed when this field is ON. The abort process is caused by an active fault which is enabled in the ABORTMASK parameter.
19	Accel Enable	indicates the amount of time spent during the accel phase of trajectory generation. ON indicates that the system is to use a non-zero time period.
20	Decel Enable	indicates the amount of time spent within the decel phase of trajectory generation. ON indicates that the system is to use a non-zero time period.
21	Accel Sign	indicates the direction of the change in velocity between subsequent move commands. ON signifies that the axis must increase in speed and OFF signifies that the axis speed must decrease. The system does not consider the sign of the velocity. Note that this field is only valid if the accel phase bit is ON.
22	Accel Mode	indicates the type of acceleration currently being used. ON signifies that the system is to use sinusoidal acceleration and OFF signifies linear ramping.
23	Decel Mode	indicates the type of deceleration currently being used. ON signifies that the system is to use sinusoidal deceleration and OFF signifies linear ramping.

**Table 2-3. Motion Status Bits (Continued from Page 35)**

Bit	Status Type	Description
24	Bounded	tells if the mechanics connected to the axis have a limited amount of travel. Spindle axes, for example, are “un-bounded” since they may travel an infinite amount without encountering a limit. ON specifies that there is a limit on the distance that the axis can travel.
25	Setup Pending	the axis processor uses this field for indicating a pending setup. This field is cleared after the setup is complete. ON appears while a setup is pending. Under normal operating conditions, the user may ignore the state of this field.
26	Checker Flag	the axis processor uses these fields for synchronization between the 4kHz interrupt and the various foreground modules. Under normal operating conditions, the user may ignore the state of this field.
27	Quick Home	indicates when the axis is seeking the home limit switch due to the result produced by the DMRMoveQuickHome() command. ON indicates execution of this command.
28	Interrupt Pending	provides internal processing to the axis processor and therefore the user may ignore it. It signifies the occurrence of a fault and requires servicing.
29	Pendant Jog	indicates the current operational mode of the teach pendant. ON indicates that the axis processor responds to the joystick inputs on the teach pendant.
30	Marker Armed	shows, during a homing sequence, when the axis processor is actively searching for an encoder marker pulse. It turns ON when the system encounters the appropriate limit switch and clears upon detection of the encoder marker pulse.
31	Jog Enable	indicates that the axis processor is currently in one of the various jog modes available.

**2.12.51. MOVEQDEPTH {Read Only}**

The Move Queue Depth parameter indicates how many commands, waiting to be executed, are contained in the queue.

**2.12.52. MOVEQSIZE {Read Only}**

The Move Queue Size parameter indicates the maximum number of motion commands that are permitted in the queue.

**2.12.53. PGAIN {Read/Write}**

The Proportional Gain parameter gets and sets the position gain of the position loop for a selected axis. Refer to the *Unidex 31 Hardware Manual* for a description of how this parameter functions in the servo. The valid range for programming this parameter is between 0 and 1,000. The default value is ten (10).

**2.12.54. PHASE\_SPEED {Read/Write}**

The Phase Speed parameter describes the desired commutation offset generated each time the user aligns the feedback device with an AC brushless motor (see Figure 2-6).

Where the commutation angle begins to advance, as indicated in Figure 2-6, the BASE\_SPEED parameter specifies the motor's velocity. The MAX\_PHASE parameter specifies what maximum commutation angle to use while computing the commutation offset that yields the best result. The PHASE\_SPEED parameter specifies the motor velocity as the commutation offset reaches MAX\_PHASE degrees. The valid range for programming this parameter is between 1 and 10,000,000; however, this can only be set at the factory.

**2.12.55. POS {Read/Write}**

The Position parameter shows the observed position of a selected axis. The valid range for programming this parameter is between 0x80000001 and 0x7fffffff (all signed 32 bit numbers).

**2.12.56. POSCMD {Read Only}**

The Position Command parameter is the current value of the position command.

**2.12.57. POSERR {Read Only}**

The Position Error is the difference, in counts, between the actual position and the desired position.

**2.12.58. POSERRLIMIT {Read/Write}**

The Position Error Limit parameter corresponds to the maximum position error which can be incurred without generating a position error limit fault. The valid range for programming this parameter is 0 to 10,000,000 machine counts. The default value is 65,536 counts.

**2.12.59. POSTARGET {Read Only}**

The Position Target parameter is the position that is targeted when a MOVETO command is sent. For example, a specified move to 10,000 would position the target at 10,000.

**2.12.60. POSTOGO {Read Only}**

The Position-to-go parameter indicates the number of counts remaining for the current move.

*Example*

For example, if the position is POS=0 and the move is MOVETO 10000, the position to go would be 10,000. Additionally, if the position is currently POS=1000 the position to go POSTOGO would equal 9,000.

**2.12.61. PROFQDEPTH {Read Only}**

The Profile Queue Depth parameter is the number of profile commands in the queue that are waiting to be executed.

**2.12.62. PROFQSIZE {Read Only}**

The Profile Queue Size parameter indicates the maximum number of profile commands permitted in the queue at one time.

**2.12.63. RAWPOS {Read Only}**

The Raw Position parameter is the actual feedback position before any axis correction (error mapping) is incorporated.

**2.12.64. RESOLVER {Read Only}**

The Resolver parameter is the Resolver absolute position of the normal axis (not the position of the master axis). The valid range for programming this parameter is between 0 and 10,000,000 machine counts. The value returned can be related to 0 through 360° mechanical of resolver.



**2.12.65. REVERSALMODE {Read/Write}**

To provide greater position accuracy, this parameter allows the user to specify a number of machine steps to compensate for any backlash present in the system. Backlash occurs when the ball screw changes direction and moves a fixed distance before the stage begins moving in the new direction. This parameter specifies the length of this distance in machine steps. The valid range for programming this parameter is between 0 and 1,000. The default value is zero (0) machine steps and does not provide backlash compensation.

**2.12.66. REVERSALVALUE {Read Only}**

This parameter is the current correction factor (in counts) for the reversal mode. The valid range for this parameter is between 0 and 1,000. The default is zero (0).

**2.12.67. SAFEZONECCW {Read/Write}**

This parameter allows the user to specify the counterclockwise boundary of the safe zone associated with the specified axis. Safe zones may be used to designate an area in which the axis is permitted to travel or one in which the axis is not permitted to travel. The determination is made based upon the SAFEZONEMODE parameter. The valid range for programming this parameter is 0x80000001 to 0x7fffffff machine steps (all 32 bit signed integers). The default value is zero (0).

**2.12.68. SAFEZONECW {Read/Write}**

The Safe Zone Clockwise parameter allows the user to specify the clockwise boundary of the safe zone associated with the specified axis. Safe zones may be used to designate an area in which the axis is permitted to travel or one in which the axis is not permitted to travel. The determination is made based upon the SAFEZONEMODE parameter. The valid range for programming this parameter is between 0x80000001 and 0x7fffffff machine steps (all 32 bit signed integers). The default value is zero (0).

**2.12.69. SAFEZONEMODE {Read/Write}**

The Safe Zone Mode parameter determines how to interpret the SAFEZONECW and SAFEZONECCW parameters (above). If the value of this parameter is zero (0), the safe zone is disabled. If this parameter is specified as one (1), the safe zone specified defines the area in which the axis is not permitted to enter. If specified as two (2), the safe zone defines the area in which the axis is not permitted to exit. The default value is zero (0).

If the associated axis is commanded to move into an area that violates an active safe zone, a safe zone fault is generated.

**2.12.70. SERVOSTATUS {Read Only}**

The Servo Status parameter returns a value that the user can decode to determine what bits are active (on) relative to the current operational state of the servo. To access the value of this parameter enter GET SERVOSTATUS followed by pressing the <Enter> key. Table 2-4, on the following page, lists the supported faults and their corresponding bit locations.

**Table 2-4. Servo Status Bits**

Bit	Status Type	Description
0	Drive Enable	shows if commutation is being performed on the associated axis. When ON, commutation is currently being performed.
1	Aux Output Enable	specifies the current state of the auxiliary output associated with an axis. When ON, the output is currently asserted.
2	CW Input	indicates the current state of the clockwise limit input to the axis processor for the associated axis. When ON, the limit is currently asserted.
3	CCW Input	specifies the current state of the counter-clockwise limit input to the axis processor for the associated axis. When ON, the limit is currently asserted.
4	Home Input	specifies the current state of the home limit switch input to the axis processor for the associated axis. When ON, the limit is currently asserted.
5	Drive Fault Input	specifies the current state of the drive fault input to the axis processor. When ON, a fault condition is present on the drive associated with the axis.
6	At Home	specifies if the current position of the associated axis corresponds with the position upon completion of a home cycle. This does not imply, however, that a home cycle has just been completed. When ON, the current position is equal to the home position.
7	Done	indicates when motion commands have completed execution. OFF signifies that trajectory generation exists on the associated axis. ON appears on the display after injecting the last velocity command into the servo loop.

Table 2-4. Servo Status Bits (Continued from Page 40)

Bit	Status Type	Description
8	In-position	specifies if the difference between the commanded position and the actual position of the associated axis is less than the amount specified in the INPOSLIMIT parameter. ON signifies that this difference is less than the specified amount.
9	Faulted	alerts the user if a fault condition exists on the associated axis. If so, look to the Fault Status Screen to verify the present type of fault. ON signifies that a fault is present.
10	Probe Input	shows the current state of the axis processor input dedicated for the touch probe. ON signifies that the input is asserted.
11	Marker Input	shows when the marker pulse from the encoder associated with an axis is active. ON signifies the marker pulse is present.
12	Hall Input #1	shows the current state of the Hall effect input associated with Axis #1. ON signifies that the input is currently asserted.
13	Hall Input #2	shows the current state of the Hall effect input associated with Axis #2. ON signifies that the input is currently asserted.
14	Hall Input #3	shows the current state of the Hall effect input associated with Axis #3. ON signifies that the input is currently asserted.
15	Hall Input #4	shows the current state of the Hall effect input associated with Axis #4. ON signifies that the input is currently asserted.
16	Integral - clamped	states that the system put a clamp on the integral error of the system at the maximum negative value.
17	Integral + clamped	states that the system put a clamp on the integral error of the system at the maximum positive value.
18	VFF enable	indicates enabled velocity feedforward for the axis. The system uses velocity feedforward during velocity contouring to minimize velocity error. ON signifies that VFF is enabled.

Table 2-4. Servo Status Bits (Continued from Page 41)

Bit	Status Type	Description
19	AFF enable	indicates enabled acceleration feedforward for an axis. An AFGAIN of zero (0) disables this field. ON signifies that AFFGAIN is non-zero.
20	alive	determines if the user enabled and configured the associated axis. This remains ON as long as these two conditions exist.
21	VFF or zero PGAIN	indicates whether to operate the servo in the velocity feedforward mode or to disable the position loop (by setting the PGAIN to zero).
22	FB Failure input	corresponds with the current state of the feedback failure input on the R/D channel associated with an axis.
23	master fb failure	corresponds with the current state of the feedback failure input on the R/D channel associated with the axis configured as a master to this axis.
24	HP VME Laser	specifies that the type of feedback device associated with the axis is an HP VME Laser. This bit gets set at configuration time and remains set until resetting the axis processor or re-configuring the axis.
25	AC mode	indicates the configuration of the axis to determine what type of motor to use. ON signifies that the setup is for an AC brushless motor. OFF indicates that the configuration requires the axis to servo with a DC motor.  Assume that all axes configured with Resolver or Hall effect feedback types require use of an AC brushless motor. Axes configured with encoder feedback are assumed to be DC motors.
26	msetting	indicates that the axis processor is computing the commutation offset for use with the associated axis. Debug960 permits entry into this mode.
27	homed	specifies that the associated axis has completed a home cycle. This field is cleared when the home sequence is initiated and set when it is completed. This does not imply, however, that a home cycle has just been completed. When ON, the axis has been homed.
28	encoder feedback	indicates the type of feedback device associated with the axis is an encoder. This bit gets set at configuration time and remains set until resetting the axis processor or re-configuring the axis.

**Table 2-4. Servo Status Bits (Continued from Page 42)**

Bit	Status Type	Description
29	error mapping enable	notifies the user that at least one axis correction table is active on the associated axis. This bit turns ON upon activation of the correction table and remains ON as long as the table is in effect.
30	position loop only	indicates a disabled position loop by setting the PGAIN axis parameter to zero (0). This bit, used in conjunction with the VFF or zero PGAIN bit, determines if the system has the velocity feedforward enabled.
31	Unused	

**2.12.71. SIMULATION {Read/Write}**

This parameter helps to facilitate easy debugging of parts programs under the UNIDEX 30, UNIDEX 31 and UNIDEX 600 CNC application. The Simulation parameter allows the user to place an axis into simulation mode. While in this mode, torque is maintained on the motors but no motion can occur. While executing a parts program on a simulated axis, the axis processor will perform all calculations normally, but the velocity command is not sent to the motor. Instead, it is used as the feedback for the defined axis, effectively creating a system without velocity error. All other features, such as data acquisition, function normally. Setting this parameter to zero (0) disables the simulation mode, while a value of one (1) enables the mode. The default value is zero (0).

**2.12.72. SOFTLIMITMODE {Read/Write}**

The Soft Limit Mode parameter sets the mode of activation for the software limits (defined by the CWEOT and CCWEOT parameters), as well as the safe zones (defined by the SAFEZONECW, SAFEZONECCW and SAFEZONEMODE parameters). In many systems, the current absolute position of an axis is unknown until after the axis reaches its home position. Therefore, the user should not enforce a software limit or a safe zone until the system successfully completes the homing process.

However, the mechanics of some systems do not permit execution of a normal homing sequence. Therefore, the user must use an alternate method to determine the absolute position. In this case, it may be logical to permit software limits and safe zones to be active at all times.

Setting this parameter to a one (1) causes the features to become active after successfully homing the axis. The default value is zero (0) and causes these features to always be active.

**2.12.73. STATUS {Read Only}**

The Status parameter is read only and returns a value that the user can decode to determine what bits are active (on) relative to the current state of the associated axis. To access the value of this parameter enter GET STATUS followed by pressing the <Enter> key. The information returned pertains to the hardware status (drive fault present, limit switch active, etc.), as well as the current firmware operational mode (homing, sync mode, enabled, etc.). Table 2-4, on the following page, lists the supported faults and their corresponding bit locations.

**Table 2-5. Status Bits**

Bit	Status Type	Description
0	Drive Enable	shows if commutation is being performed on the associated axis. When ON, commutation is currently being performed.
1	Aux Output Enable	specifies the current state of the auxiliary output associated with an axis. When ON, the output is currently asserted.
2	CW Input	indicates the current state of the clockwise limit input to the axis processor for the associated axis. When ON, the limit is currently asserted.
3	CCW Input	specifies the current state of the counter-clockwise limit input to the axis processor for the associated axis. When ON, the limit is currently asserted.
4	Home Input	specifies the current state of the home limit switch input to the axis processor for the associated axis. When ON, the limit is currently asserted.
5	Drive Fault Input	specifies the current state of the drive fault input to the axis processor. When ON, a fault condition is present on the drive associated with the axis.
6	At Home	specifies if the current position of the associated axis corresponds with the position upon completion of a home cycle. This does not imply, however, that a home cycle has just been completed. When ON, the current position is equal to the home position.
\7	Done	indicates when motion commands have completed execution. Turning this field OFF signifies that trajectory generation exists on the associated axis. ON appears on the display after injecting the last velocity command into the servo loop.

Table 2-5. Status Bits (Continued from Page 44)

Bit	Status Type	Description
8	In-position	specifies if the difference between the commanded position and the actual position of the associated axis is less than the amount specified in the INPOSLIMIT parameter. ON signifies that this difference is less than the specified amount.
9	Faulted	alerts the user if a fault condition exists on the associated axis. If so, look to the Fault Status Screen to verify the present type of fault. ON signifies that a fault is present.
10	Probe Input	shows the current state of the axis processor input dedicated for the touch probe. ON signifies that the input is asserted.
11	Marker Input	shows when the marker pulse from the encoder associated with the axis is active. ON signifies the marker pulse is present.
12	Hall Input #1	shows the current state of the Hall effect input associated with Axis #1. ON signifies that the input is currently asserted.
13	Hall Input #2	shows the current state of the Hall effect input associated with Axis #2. ON signifies that the input is currently asserted.
14	Hall Input #3	shows the current state of the Hall effect input associated with Axis #3. ON signifies that the input is currently asserted.
15	Hall Input #4	shows the current state of the Hall effect input associated with Axis #4. ON signifies that the input is currently asserted.
16	Move Direction	indicates the direction an axis is currently moving. ON corresponds with the clockwise direction and OFF signifies counter-clockwise. The state of this bit is undetermined unless the axis is in motion.
17	Moving	alerts the user when the associated axis is currently in motion. ON signifies the axis is in motion. OFF appears on the display when the axis trajectory generation is complete (upon completion the bit gets set).

Table 2-5. Status Bits (Continued from Page 45)

Bit	Status Type	Description
18	Accel Phase	determines if the axis is currently within the acceleration portion of the motion command being executed. The acceleration phase is the period of time when the trajectory generator is changing the velocity from the current velocity to the commanded velocity. Therefore, if the current velocity is 10 counts/msec and the commanded velocity is 5 counts/msec, the value of the accel phase flag changes while the axis slows to the new velocity (5 counts/msec). Through the run constant phase or the deceleration phase, the user can terminate this phase. Note that this bit is of no use if the corresponding axis is currently being controlled via a cam table.
19	Decel Phase	determines if the axis is currently within the deceleration portion of the motion command being executed. The U31 definition for this phase is the time period in which the axis is slowing from the commanded velocity to a complete stop. ON appears while the axis is decelerating. OFF appears if the axis trajectory generation is complete (when the done bit is set). When controlling the corresponding axis through a cam table this bit has no significance.
20	Homing	determines if the associated axis is currently executing a homing sequence. ON indicates a home cycle is in progress.
21	Feedrate Override	signifies if manual feedrate override (MFO) controls are active on this axis. ON signifies that these controls are enabled for this axis.
22	Profile Mode	shows if the associated axis is currently executing moves based on the data found in the profile queue for an axis. ON indicates that profile queue data is being executed.
23	Sync Mode	indicates if the associated axis is being controlled by another (master) axis. The synchronization may be cam mode or ratio mode. ON signifies synchronization is enabled.
24	CAM Table Enable	shows if the associated axis is synchronized with another axis by means of a cam table. This bit is of no significance if the axis is not being controlled by another axis (sync mode bit above). ON shows that a cam table has been activated with this axis acting as the slave axis.



Table 2-5. Status Bits (Continued from Page 46)

Bit	Status Type	Description
25	Homing Direction	indicates the direction the associated axis is currently traveling. It is only valid while the axis is executing its homing sequence (homing bit). ON corresponds to CW direction and OFF signifies CCW motion.
26	Continuous Move	ON implies that the axis is currently executing a continuous motion command. OFF appears in the display at all other times.
27	Queued Command	shows if the associated axis is currently executing moves based on the data found in the motion queue for this axis. ON signifies that motion queue data is being executed.
28	Hold Active	designates if the associated axis is currently being held from movement. ON signifies the axis is being held.
29	Aux Mode	signifies if there is currently an active synchronized auxiliary output table associated with an axis. ON appears when an auxiliary output table is active.
30	Block Motion	indicates whether all commands initiating motion are blocked from execution on an axis. ON signifies that the system will ignore all motion commands.
31	Hold Queue	shows if execution of queued motion commands has been temporarily suspended. ON signifies suspended execution.

**2.12.74. SYNCSPPEED {Read/Write}**

When sync with a mode of two (2) is called, the slave axis assumes that its present position is the start of the cam table. It then reads the master position and determines what offset it must move to match its position with that of the master in the table. Then, it uses the speed defined in this parameter to index the appropriate amount. The valid programming range is between 1 and 0x7ffffff. The default value is 1,000.

**2.12.75. VELCMDTRAP {Read/Write}**

The Velocity Command Trap parameter specifies the maximum commanded velocity that the system must tolerate. If the commanded velocity exceeds the amount specified, a velocity command trap is generated. The valid range for programming this parameter is between 0 and 65,536,000 machine counts per second. A value of zero (0) disables commanded velocity trap detection. The default value is zero (0).

**2.12.76. VELPOSITION {Read Only}**

The Velocity Position parameter is the axis position from the secondary feedback.

**2.12.77. VELTRAP {Read/Write}**

The Velocity Trap parameter specifies the maximum instantaneous speed that the system must tolerate. If the actual velocity exceeds the amount specified, a velocity trap is generated. The valid range for programming this parameter is between 0 and 65,536,000 machine counts per second. A value of zero (0) disables the velocity trap detection. The default value is zero (0).

**2.12.78. VEL\_SHIFT {Read/Write}**

This parameter is reserved for internal use.

**2.12.79. VFF {Read/Write}**

The Velocity Feedforward parameter enables or disables the velocity feedforward function. Enabling this function minimizes position following error. A value of one (1) enables this function and a value of zero (0) disables it. The default value is zero (0).

**2.13. Brushless Motor Commutation**

Brushless motors require additional setup beyond verifying proper phasing of encoder or Resolver/Inductosyn feedback.

A test routine for checking the phasing of a brushless servo is provided with Debug960. This command, MSETTEST, generates a rotating current command that should cause the motor to increment in a positive direction. If the motor does not rotate or does not rotate in a positive direction, change the axis phasing at the motor amplifier. The IMAX value should also be set to a low value to prevent unnecessary motor heating or damage. After MSETTEST has completed, it is necessary to set the DRIVE to zero (0) to turn off this routine.

**2.13.1. Aligning Resolver Based Brushless Motors**

Once the proper phasing has been established between the current commands and the motor, MSET is used to insure that the feedback device is aligned with the motor. MSET generates a current command that is defined as zero (0) angle. MSET also displays the absolute position feedback. The position display should indicate a zero (0) or near zero (0) position. If the displayed position is not at zero (0), then a commutation offset can be installed with the CONFIG command. This commutation offset should be the negative of the position displayed with MSET. As with MSETTEST, IMAX should be set to a low value to prevent motor heating or damage. A carriage return exits MSET.

### **2.13.2. Working With Hall Effect Motors**

Hall effect motors can be commutated using only the Hall effect feedback transducers for commutation. In this mode of commutation a wave form is presented to the motor which has six discrete steps ( +1, +1, 0, -1, -1, 0), modulated by the magnitude of the torque command. Since there are only six steps per electrical cycle, the commutation offset should only be in increments of 60 degrees. Finding the commutation offset for a Hall effect motor is therefore a trial and error sequence where commutation offsets of 0,60,120,180,240,300 are specified with the CONFIGHALL command.

▽ ▽ ▽



## CHAPTER 3: STATUS

**In This Section:**

- Overview ..... 3-1
- Display Pull-down Menu ..... 3-2
- Axis Status Display..... 3-3
- Servo Status Display..... 3-11
- Motion Status Display ..... 3-16
- Fault Status Display ..... 3-22
- Fault Mask Display ..... 3-27
- Disable Mask Display..... 3-28
- Interrupt Mask Display ..... 3-29
- Halt Mask Display ..... 3-30
- Aux Mask Display ..... 3-31
- Time Pull-down Menu ..... 3-32
- Exit Pull-down Menu..... 3-33
- Fault\_Control Pull-down Menu ..... 3-34
- Help Pull-down Menu ..... 3-35

### 3.1. Overview

This diagnostic utility monitors the status of the axis processor for the UNIDEX 30, UNIDEX 31 and UNIDEX 600 controller. STATUS displays the status of the various bits of axis control information for any axis that has been configured. To access status, select the type of status to be monitored from the Display Pull-down Menu. This utility polls the axis processor for that status and displays it several times per second.

All of the status bits available from within this screen are reflective of some condition (i.e., drive enabled, at home, faulted, etc.). This utility notifies the user if a condition exists on each axis. As can be seen from Figure 3-1, each specific type of status within a group appears vertically on the left side of the screen and the axes names appear horizontally across the upper part of the screen. The user may view undisplayed axis and status types using the scroll bars at the bottom and right hand side of the display.

*Axis Processor Status*

*Status Bits*

Aerotech Axis Status Display																
Display	Time	Exit	Fault_Control													Help
Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00 drive enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
01 aux output enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
02 CW input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
03 CCW input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
04 home input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
05 drive fault input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
06 at home	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
07 done	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
08 in-position	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
09 faulted	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10 probe input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
11 marker input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
12 hall input #1	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
13 hall input #2	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
14 hall input #3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15 hall input #4	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
16 move direction	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17 moving	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18 accel phase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19 decel phase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20 homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21 feedrate override	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22 profile mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23 sync mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24 cam table enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25 homing direction	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26 continuous move	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27 queued command	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28 hold active	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 3-1. Status Utility Initialization Screen

### 3.2. Display Pull-down Menu

*Selecting Status Types to Monitor*

Through the Display Pull-down Menu the user may select or change the group of status types currently being monitored. These groups include axis status, servo status, motion status and fault status. The user may also look up the current value of the masks that define the actions taken by the axis processor upon detection of the various types of faults. However, the user cannot modify these masks from within this screen. These masks include the fault masks, disable masks, interrupt masks, halt masks and aux masks.

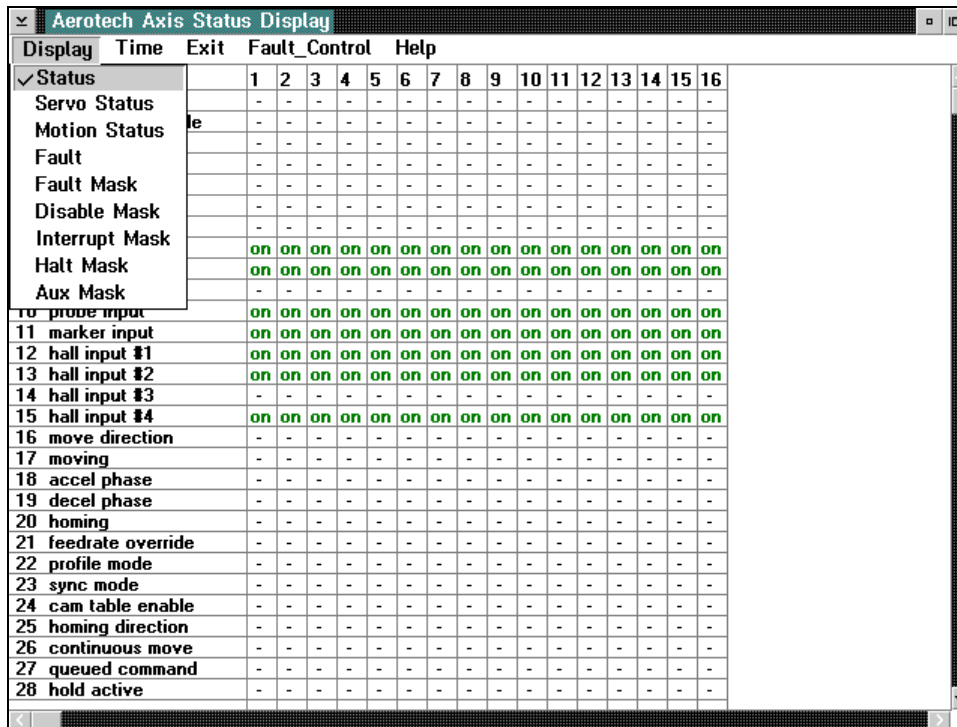


Figure 3-2. Display Pull-down Menu

### 3.2.1. Axis Status Display

The information found in the Aerotech Axis Status Display conveys information regarding the current state of the associated axis. It conveys information pertaining to the hardware status (drive fault present, limit switch active, etc.) and the current firmware operational mode (homing, sync mode enabled, etc.). This display sets the update time for all the information displayed in the Axis Status Screen. The following sections provide a brief description of the meaning associated with each field. Figure 3-3 illustrates the Aerotech Axis Status Display.

*Status of a Selected Axis*

Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00 drive enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
01 aux output enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
02 CW input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
03 CCW input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
04 home input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
05 drive fault input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
06 at home	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
07 done	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
08 in-position	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
09 faulted	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10 probe input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
11 marker input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
12 hall input #1	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
13 hall input #2	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
14 hall input #3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15 hall input #4	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on
16 move direction	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17 moving	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18 accel phase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19 decel phase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20 homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21 feedrate override	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22 profile mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23 sync mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24 cam table enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25 homing direction	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26 continuous move	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27 queued command	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28 hold active	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 3-3. Aerotech Axis Status Display

**3.2.1.1. drive enabled**

*Commutation Status of an Axis*

This field shows if commutation is being performed on the associated axis. When ON, commutation is currently being performed.

Related Parameter(s): DRIVE  
IOLEVEL  
Related Command(s): DMRGetDrive( )  
DMRSetDrive( )

**3.2.1.2. aux output enable**

*Auxiliary Output Status of an Axis*

This field specifies the current state of the auxiliary output associated with the specified axis. When ON, the output is currently asserted.

Related Parameter(s): AUX  
IOLEVEL  
Related Command(s): DMRGetAux( )  
DMRSetAux( )



### 3.2.1.3. CW input

This field specifies the current state of the clockwise limit input to the axis processor for the specified axis. When ON, the limit is currently asserted.

*Clockwise Limit  
Input of an Axis*

Related Parameter(s): IOLEVEL

Related Command(s): NONE

### 3.2.1.4. CCW input

This field specifies the current state of the counterclockwise limit input to the axis processor for the associated axis. When ON, the limit is currently asserted.

*Counterclockwise  
Limit Input Status of  
an Axis*

Related Parameter(s): IOLEVEL

Related Command(s): NONE

### 3.2.1.5. home input

This field specifies the current state of the home limit switch input to the axis processor for the associated axis. When ON, the limit is currently asserted.

*Home Limit Switch  
Input Status of Axis*

Related Parameter(s): IOLEVEL

Related Command(s): DMRMoveHome()  
DMRMoveHomeRev()  
DMRMoveQuickHome()

### 3.2.1.6. drive fault input

This field specifies the current state of the drive fault input to the axis processor. When ON, a fault condition is present on the drive associated with the specified axis.

*Drive Fault Input  
Status of an Axis*

Related Parameter(s): IOLEVEL

Related Command(s): NONE

### 3.2.1.7. at home

This field specifies if the current position of the selected axis corresponds with the position upon completion of a home cycle. This does not imply, however, that a home cycle has just been completed. When ON, the current position is equal to the home position.

*Current Position Vs  
Home Position Status*

Related Parameter(s): NONE

Related Command(s): DMRMoveHome()  
DMRMoveHomeRev()  
DMRMoveQuickHome()

**3.2.1.8. done**

*Motion Command Completed Status*

This field indicates when motion commands have completed execution. Turning this field OFF signifies that trajectory generation exists on the selected axis. ON appears on the display after injecting the last velocity command into the servo loop.

Related Parameter(s): NONE

Related Command(s): All commands initiating motion

**3.2.1.9. in-position**

*Commanded Position Vs Actual Position*

This field specifies if the difference between the commanded position and the actual position of the chosen axis is less than the amount specified by the in-position limit parameter. Turning this field ON signifies that this difference is less than the specified amount.

Related Parameter(s): INPOSLIMIT

Related Command(s): DMRGetInPositionBand()

DMRSetInPositionBand()

All commands initiating motion

**3.2.1.10. faulted**

*Axis Fault Status*

This field alerts the user if a fault condition exists on the selected axis. If so, look to the fault status screen to verify the present type of fault. ON signifies that a fault is present.

Related Parameter(s): FAULT

Related Command(s): DMRGetFault()()

DMRSetFault()()

**3.2.1.11. probe input**

*Touch Probe Status*

This field shows the current state of the axis processor input dedicated for the touch probe. ON signifies that the input is asserted.

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.1.12. marker input**

*Encoder Marker Pulse Status*

This field shows when the marker pulse from the encoder associated with the axis is active. ON signifies the marker pulse is present.

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.1.13. hall input #1, #2, #3 and #4**

These fields show the current state of each of the Hall effect inputs associated with an axis. ON signifies that the input is currently asserted.

*Hall Effect Input Status*

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.1.14. move direction**

This field shows the direction the chosen axis is currently moving. ON corresponds with the clockwise direction and OFF signifies counterclockwise. The state of this bit is undetermined unless the axis is in motion.

*CW or CCW Direction of Motion*

Related Parameter(s): NONE

Related Command(s): All commands initiating motion

**3.2.1.15. moving**

This field alerts the user when the associated (picked) axis is currently in motion. ON signifies the axis is in motion. OFF appears on the display when the axis trajectory generation is complete (upon completion the bit gets set).

*Motion Status*

Related Parameter(s): NONE

Related Command(s): All commands initiating motion

**3.2.1.16. accel phase**

This field determines if the selected axis is currently within the acceleration portion of the motion command being executed. The acceleration phase is the period of time when the trajectory generator is changing the velocity from the current velocity to the commanded velocity. Therefore if the current velocity is 10 counts/msec and the commanded velocity is 5 counts/msec, the value of the accel phase flag changes while the axis slows to the new velocity (5 counts/msec). The user can terminate this phase through the run constant phase or the deceleration phase.

*Change in the Current Velocity to Commanded Velocity*

This bit has no significance if the corresponding axis is currently being controlled via a cam table.



Related Parameter(s): NONE

Related Command(s): All commands initiating motion

*Commanded Velocity Slows to Complete Stop*



**3.2.1.17. decel phase**

This field determines if the chosen axis is currently within the deceleration portion of the motion command being executed. The system's definition for this phase is the time period in which the axis is slowing from the commanded velocity to a complete stop. ON appears on the screen while the axis is decelerating. OFF appears if the axis trajectory generation is complete (when this done bit is set).

If the user is controlling the corresponding axis through a Cam table, this bit has no significance.

Related Parameter(s): NONE  
 Related Command(s): All commands initiating motion

*Execution of a Home Cycle Status*

**3.2.1.18. homing**

This field determines if the associated axis is currently executing a homing sequence. ON signifies the home cycle is in progress.

Related Parameter(s): NONE  
 Related Command(s): DMRMoveHome( )  
 DMRMoveHomeRev( )  
 DMRMoveQuickHome( )

*MFO Controls*

**3.2.1.19. feedrate override**

This field signifies if manual feedrate override (MFO) controls are active on the selected axis. ON signifies that these controls are enabled for the axis.

Related Parameter(s): NONE  
 Related Command(s): NONE

*Execution of Profile Queue Data*

**3.2.1.20. profile mode**

This field shows if the associated axis is currently executing moves based on the data found in the profile queue for this axis. ON signifies that profile queue data is being executed.

Related Parameter(s): NONE  
 Related Command(s): DMRStartProfileQueue( )  
 DMRStopProfileQueue( )

**3.2.1.21. sync mode**

This field shows if the associated axis is being controlled by another (master) axis. This synchronization may be cam mode or ratio mode. ON signifies that synchronization is enabled.

*Master Axis Control*

Related Parameter(s): NONE  
 Related Command(s): DMRGetSyncMode( )

**3.2.1.22. cam table enable**

This field shows if the associated axis is synchronized with another axis by means of a cam table. This bit has no significance if the axis is not being controlled by another axis (sync mode bit, above). ON shows that a Cam table has been activated with the selected axis acting as the slave axis.

*Axis Synchronization via Cam Table*

Related Parameter(s): NONE  
 Related Command(s): DMRSetSyncMode( )

**3.2.1.23. homing direction**

This field indicates the direction the associated axis is currently traveling. It is only valid while the axis is executing its homing sequence (homing bit). ON corresponds to CW direction and OFF signifies CCW motion.

*Direction of Travel*

Related Parameter(s): HOME DIRECTION  
 Related Command(s): DMRMoveHome( )  
 DMRMoveHomeRev( )  
 DMRMoveQuickHome( )

**3.2.1.24. continuous move**

ON appears on the screen in this field when the selected axis is currently executing a continuous motion command. OFF appears in the display at all other times.

*Continuous Motion*

Related Parameter(s): NONE  
 Related Command(s): DMRStartMotion( )  
 DMRStopMotion( )

**3.2.1.25. queued command**

This field shows if the associated axis is currently executing moves based on the data found in the motion queue for the axis. ON signifies that motion queue data is being executed.

*Execution of Motion Queue Data*

Related Parameter(s): NONE  
 Related Command(s): DMRQueueMoveIncremental( )  
 DMRQueueMoveAbsolute( )

**3.2.1.26. hold active*****Axis Movement  
Halted Status***

This field designates if the associated axis is currently being held from movement. ON signifies the axis is being held.

Related Parameter(s): NONE  
Related Command(s): DMRHoldQueue( )  
DMRReleaseQueue( )  
DMRHoldMotion( )  
DMRRelease Motion( )

**3.2.1.27. aux mode*****Auxiliary Output  
Table Status***

This field signifies if there is currently an active synchronized auxiliary output table associated with the selected axis. ON appears on the display when an auxiliary output table is active.

Related Parameter(s): NONE  
Related Command(s): DMRGetAuxMode( )  
DMRSetAuxMode( )

**3.2.1.28. block motion*****Prevent All Motion  
Command Execution***

This field indicates whether all commands initiating motion are blocked from execution on a selected axis. ON signifies that the system is to ignore all motion commands.

Related Parameter(s): NONE  
Related Command(s): DMRGetBlockMotion( )  
DMRSetBlockMotion( )

**3.2.1.29. hold queue*****Queued Motion  
Command Execution***

This field shows if execution of queued motion commands has been temporarily suspended. ON signifies suspended execution.

Related Parameter(s): NONE  
Related Command(s): DMRHoldQueue( )  
DMRReleaseQueue( )

### 3.2.2. Servo Status Display

Much of the information found on the Servo Status Display is identical to the information found on the Axis Status screen. However, the upper 16 fields of this display convey information that relates to the current operational state of the servo. This screen helps to determine if the servo is enabled, if there are any end-of-travel limits on and other conditions that effect the operation or state of the servo.

*Operational State of the Servo*

Display	Time	Exit	Fault_Control																Help
Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
00 drive enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
01 aux output enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
02 CW input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
03 CCW input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
04 home input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
05 drive fault input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
06 at home	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
07 done	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
08 in-position	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
09 faulted	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
10 probe input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
11 marker input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
12 hall input #1	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
13 hall input #2	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
14 hall input #3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
15 hall input #4	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
16 integral - clamped	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
17 integral + clamped	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
18 VFF enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
19 AFF enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
20 alive	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
21 VFF or zero Pgain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
22 fb failure input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
23 master fb failure	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
24 HP VME Laser	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
26 AC mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
27 msetting	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
28 homed	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			

Figure 3-4. Aerotech Servo Status Display

#### 3.2.2.1. drive enable

Refer to Section 3.2.1.1.

#### 3.2.2.2. aux output enable

Refer to Section 3.2.1.2.

#### 3.2.2.3. CW input

Refer to Section 3.2.1.3.

#### 3.2.2.4. CCW input

Refer to Section 3.2.1.4.

**3.2.2.5. home input**

Refer to Section 3.2.1.5.

**3.2.2.6. drive fault input**

Refer to Section 3.2.1.6.

**3.2.2.7. at home**

Refer to Section 3.2.1.7.

**3.2.2.8. done**

Refer to Section 3.2.1.8.

**3.2.2.9. in-position**

Refer to Section 3.2.1.9.

**3.2.2.10. faulted**

Refer to Section 3.2.1.10.

**3.2.2.11. probe input**

Refer to Section 3.2.1.11.

**3.2.2.12. marker input**

Refer to Section 3.2.1.12.

**3.2.2.13. Hall input #1, #2, #3 and #4**

These information fields are synonymous with the corresponding field in the axis status group.

**3.2.2.14. integral - clamped**

This field states that the system put a clamp on the integral error of the system at the maximum negative (-) value.

Related Parameter(s): NONE

Related Command(s): NONE

*Clamp on Integral  
Error Status*



**3.2.2.15. integral + clamped**

This field states that the system put a clamp on the integral error of the system at the maximum positive (+) value.

*Clamp on Integral  
Error Status*

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.2.16. VFF enable**

This field indicates enabled velocity feed forward for the selected axis. The system uses velocity feed forward during velocity contouring to minimize velocity error. ON signifies that VFF is enabled.

*Velocity Feed  
Forward Status*

Related Parameter(s): VFF

Related Command(s): DMRGetVelocityFeedForward( )

DMRSetVelocityFeedForward( )

**3.2.2.17. AFF enable**

This field indicates enabled acceleration feed forward for a chosen axis. An AFGAIN of zero disables the field. ON signifies that AFFGAIN is non-zero.

*Acceleration Feed  
Forward Status*

Related Parameter(s): AFFGAIN

Related Command(s): DMRGetAFFGain( )

DMRSetAFFGain( )

**3.2.2.18. alive**

This field determines if the user enabled and configured the associated axis. This bit remains ON as long as these two conditions exist.

*Verifying Configured  
and Enabled Axes*

Related Parameter(s): NONE

Related Command(s): DMRConfigAxis( )

DMRConfigEncoder( )

DMRConfigHallEffect( )

DMRSetDrive( )

**3.2.2.19. VFF or zero Pgain**

This bit indicates whether to operate the servo in the velocity feed forward mode or to disable the position loop (by setting PGAIN equal to zero).

*Servo Mode  
Operation*

Related Parameter(s): PGAIN

Related Command(s): DMRGetVelocityFeedForward( )

DMRSetVelocityFeedForward( )

**3.2.2.20. fb failure input**

*Feedback Failure Input*

This field corresponds with the current state of the feedback failure input on the Resolver-to-digital (R/D) channel associated with the chosen axis.

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.2.21. master fb failure**

*Master Feedback Failure Input*

This field corresponds with the current state of the feedback failure input on the Resolver-to-digital (R/D) channel associated with the axis configured as a master to this axis.

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.2.22. HP VME Laser**

*HP VME Laser Feedback*

This bit specifies that the type of feedback device associated with the selected axis is an HP VME laser. This bit gets set at configuration time and remains set until resetting the axis processor or re-configuring the axis.

Related Parameter(s): NONE

Related Command(s): NONE

**3.2.2.23. AC mode**

*AC/DC Motor Configuration*

This field indicates the configuration of the associated axis to determine what type of motor to use. ON signifies that the setup is for an AC brushless motor. OFF indicates that the configuration requires that the axis is a servo with a DC motor.

Assume that all axes configured with Resolver or Hall effect feedback types require use of an AC brushless motor. Axes configured with encoder feedback are assumed to be DC motors.

Related Parameter(s): NONE

Related Command(s):  
DMRConfigAxis()  
DMRConfigEncoder()  
DMRConfigHallEffect()

**3.2.2.24. msetting**

*Commutation Offset*

This field indicates that the axis processor is computing the commutation offset for use with the associated axis. Several Aerotech diagnostic utilities (such as debug960.exe) permit entry into this mode.

Related Parameter(s): Commutation Offset

Related Command(s): DMRConfigAxis()

**3.2.2.25. homed**

This field specifies that the associated axis completed a home cycle. This field is cleared when the home sequence is initiated and set when it is completed. This does not imply, however, that a home cycle has just been completed. When ON, the axis has been homed.

*Completion of a Home Cycle*

Related Parameter(s): NONE  
 Related Command(s): DMRMoveHome()  
 DMRMoveHomeRev()  
 DMRMoveQuickHome()

**3.2.2.26. encoder feedback**

This bit specifies that the type of feedback device associated with the selected axis is an encoder. This bit gets set at configuration time and remains set until resetting the axis processor or re-configuring the axis.

*Feedback Device, Encoder*

Related Parameter(s): NONE  
 Related Command(s): DMRConfigEncoder()  
 DMRConfigAxis()  
 DMRConfigHallEffect()

**3.2.2.27. errormapping enable**

This notifies the user that at least one axis correction table is active on the associated axis. This bit turns ON upon activation of the correction table and remains ON as long as the table is in effect.

*Axis Correction Table*

Related Parameter(s): NONE  
 Related Command(s): DMRsetErrorMap()

**3.2.2.28. position loop only**

This field indicates a disabled position loop by setting the PGAIN axis parameter to zero. This bit, used in conjunction with the VFF or zero PGAIN bit, determines if the system has velocity feed forward enabled.

*Disabled Position Loop*

Related Parameter(s): PGAIN  
 VFF  
 Related Command(s): DMRSetVelocityFeedForward()  
 DMRSetPgain()

**Enabled Motion Routines**

**3.2.3. Motion Status Display**

Once again, many of the fields available on the Motion Status Display dialog box are also duplicated on other screens. However, the upper 16 fields contain unique information regarding the current state of the operational mode of the trajectory generator. This screen indicates which motion routines are currently enabled.

Display	Time	Exit	Fault_Control	Help												
Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00 move direction	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
01 moving	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
02 accel phase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
03 decel phase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
04 homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
05 feedrate override	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
06 profile mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
07 sync mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
08 cam table enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
09 homing direction	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10 continuous move	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11 queued command	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12 hold active	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13 aux mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14 block motion	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15 hold queue	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16 disable queued	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17 halt queued	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18 abort queued	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19 accel enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20 decel enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21 accel sign	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22 accel mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23 decel mode	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24 bounded	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25 setup pending	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26 checker flag	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27 quick home	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28 interrupt pending	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 3-5. Aerotech Motion Status Display

**3.2.3.1. move direction**

Refer to Section 3.2.1.14.

**3.2.3.2. moving**

Refer to Section 3.2.1.15.

**3.2.3.3. accel phase**

Refer to Section 3.2.1.16.

**3.2.3.4. decel phase**

Refer to Section 3.2.1.17.

**3.2.3.5. homing**

Refer to Section 3.2.1.18.

**3.2.3.6. feedrate override**

Refer to Section 3.2.1.19.

**3.2.3.7. profile mode**

Refer to Section 3.2.1.20.

**3.2.3.8. sync mode**

Refer to Section 3.2.1.21.

**3.2.3.9. cam table enable**

Refer to Section 3.2.1.22.

**3.2.3.10. homing direction**

Refer to Section 3.2.1.23.

**3.2.3.11. continuous move**

Refer to Section 3.2.1.24.

**3.2.3.12. queued command**

Refer to Section 3.2.1.25.

**3.2.3.13. hold active**

Refer to Section 3.2.1.26.

**3.2.3.14. aux mode**

Refer to Section 3.2.1.27.

**3.2.3.15. block motion**

Refer to Section 3.2.1.28.

**3.2.3.16. hold queue**

This information field is synonymous with the corresponding field in the Axis Status group.

**3.2.3.17.      disable queued**

*Queued Motion  
Command Status*

This field provides an indication of the current state of queued motion command execution. When ON appears on the display, the system ignores all queued motion commands. Otherwise, the system places the commands into the queue and executes as soon as possible.

Related Parameter(s):      NONE  
 Related Command(s):      DMRQueueMoveIncremental( )  
    DMRQueueMoveAbsolute( )

**3.2.3.18.      halt queued**

*Motion Queue Status*

This field also provides information about the current state of the motion queue. ON appears on the display when the system halts execution of the motion queue. Upon halting the motion queue, the axis gracefully decelerates to zero velocity in the amount of time specified in the deceleration parameter. This has no effect on the position error tracking. The motion queue is flushed when the halt queue field is ON. The halt process is caused by an active fault that is enabled in the HALTMASK parameter.

Related Parameter(s):      HALTMASK  
 Related Command(s):      All fault & mask related commands

**3.2.3.19.      abort queued**

*Motion Queue  
Aborted*

This field is similar to the halt queued indicator above. However, this indicator notifies the user that the system aborted execution of the motion queue. The motion queue is not flushed when this field is ON. The abort process is caused by an active fault that is enabled in the ABORTMASK parameter.

Related Parameter(s):      ABORTMASK  
 Related Command(s):      All fault & mask related commands

**3.2.3.20.      accel enable**

*Acceleration Time of  
Trajectory Generator*

This field provides an indication of the amount of time spent during the accel phase of trajectory generation. ON indicates that the system is to use a non-zero time period.

Related Parameter(s):      ACCEL  
 Related Command(s):      DMRGetAccelTime( )  
    DMRSetAccelTime( )

**3.2.3.21. decel enable**

This field provides an indication of the amount of time spent within the decel phase of trajectory generation. ON indicates that the system is to use a non-zero time period.

*Deceleration Time of Trajectory Generator*

Related Parameter(s):        DECEL  
 Related Command(s):        DMRGetDecelTime( )  
                                   DMRSetDecelTime( )

**3.2.3.22. accel sign**

This field indicates the direction of the change in velocity between subsequent move commands. ON signifies that the axis must increase in speed and OFF signifies that the axis speed must decrease. The system does not consider the sign of the velocity.

*Velocity Changes Between Move Commands*

This field is only valid if the accel phase bit is ON.



Related Parameter(s):        NONE  
 Related Command(s):        All commands initiating motion

**3.2.3.23. accel mode**

This field specifies the type of acceleration currently being used. ON signifies that the system is to use sinusoidal acceleration and OFF signifies linear ramping.

*Sinusoidal/Linear Acceleration*

Related Parameter(s):        ACCELMODE  
 Related Command(s):        DMRGetAccelMode( )  
                                   DMRSetAccelMode( )

**3.2.3.24. decel mode**

This field specifies the type of deceleration currently being used. ON signifies that the system is to use sinusoidal deceleration and OFF signifies linear ramping.

*Sinusoidal/Linear Deceleration*

Related Parameter(s):        DECELMODE  
 Related Command(s):        DMRGetDecelMode( )  
                                   DMRSetDecelMode( )

**3.2.3.25. bounded*****Limited Travel of an Axis***

This field tells if the mechanics connected to the selected axis have a limited amount of travel. Spindle axes, for example, are “un-bounded” since they may travel an infinite amount without encountering a limit. ON specifies that there is a limit on the distance that the selected axis can travel.

Related Parameter(s): NONE  
 Related Command(s): DMRConfigAxis( )  
 DMRConfigEncoder( )  
 DMRConfigHallEffect( )

**3.2.3.26. setup pending*****Pending Setup***

The axis processor uses this field for indicating a pending setup. The field is cleared after the setup is complete. ON appears while a setup is pending. Under normal operating conditions the user may ignore the state of this field.

Related Parameter(s): NONE  
 Related Command(s): NONE

**3.2.3.27. checker flag*****Synchronization***

The axis processor uses these fields for synchronization between the 4kHz interrupt and the various foreground modules. Under normal operating conditions the user may ignore the state of this field.

Related Parameter(s): NONE  
 Related Command(s): NONE

**3.2.3.28. quick home*****Locating the Home Limit Switch***

This field provides an indication of when the chosen axis is seeking the home limit switch due to the result produced by the DMRMoveQuickHome( ) command. ON appears on the display upon execution of this command.

Related Parameter(s): NONE  
 Related Command(s): DMRMoveQuickHome( )

**3.2.3.29. interrupt pending*****Pending Interrupt***

This field provides internal processing to the axis processor and therefore the user may ignore the state of this field. It signifies the occurrence of a fault and requires servicing.

Related Parameter(s): NONE  
 Related Command(s): NONE



**3.2.3.30. pendant jog**

This field specifies the current operational mode of the U31 Teach Pendant. ON appears on the display when the axis processor responds to the joystick inputs and Teach Pendant inputs.

*Operating Mode of  
Teach Pendant*

Related Parameter(s): NONE  
Related Command(s): DMREnablePendant()  
DMRDisablePendant()

**3.2.3.31. marker armed**

During a homing sequence this field shows when the axis processor is actively searching for an encoder marker pulse. This field turns ON when the system encounters the appropriate limit switch and gets cleared upon detection of the encoder pulse.

*Search of Encoder  
Marker Pulse*

Related Parameter(s): NONE  
Related Command(s): DMRMoveHome()  
DMRMoveHomeRev()  
DMRMoveQuickHome()

**3.2.3.32. jog enable**

This field signifies that the axis processor is currently in one of the various jog modes available.

*Jog Mode*

Related Parameter(s): PGAIN  
Related Command(s): DMRSetJogMode()  
DMRSetPendantJogMode()  
DMRSetMouseJogMode()  
DMRVmeJogMode()

### 3.2.4. Fault Status Display

*Monitor Faults*

This display monitors the faults present on the axis processor. These faults remain active until explicitly cleared by the user. This menu displays ON in any field that has an active fault associated with the selected axis. To activate a fault it is necessary to enable the corresponding bit in the FAULTMASK parameter. Figure 3-6 illustrates this display.

Display	Time	Exit	Fault_Control																Help
Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
00 Position error limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
01 RMS current limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
02 CW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
03 CCW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
04 CW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
05 CCW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
06 Drive	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
07 Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
08 Programming	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
09 Master Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
10 Homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
11 User	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
12 Velocity trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
13 Velocity command trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
14 Home switch tolerance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
15 probe input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
16 CNC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
17 External Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
18 Safe Zone	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
19 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
20 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
21 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
22 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
23 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
24 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
25 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
26 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
27 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
28 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			

Figure 3-6. Aerotech Fault Status Display

#### 3.2.4.1. Position error limit

*Commanded Vs Actual Position Error*

This error occurs when the difference between the instantaneous commanded position and the actual position is greater than the amount specified as the position error limit.

- Related Parameter(s): POSERRLIMIT
- Related Command(s): DMRGetPositionErrorLimit()  
DMRSetPositionErrorLimit()  
All commands initiating motion

**3.2.4.2. RMS current limit**

This fault occurs when the average instantaneous current exceeds the amount specified in the current fault limit parameter.

*Average  
Instantaneous  
Current Overrun*

Related Parameter(s): IAVGLIMIT  
 Related Command(s): DMRGetCurrentFaultLimit()  
 DMRSetCurrentFaultLimit()

**3.2.4.3. CW Hardware EOT**

This is a clockwise end-of-travel fault that occurs when the system encounters the clockwise limit switch.

*Clockwise Limit  
Switch Located*

Related Parameter(s): IOLEVEL  
 Related Command(s): NONE

**3.2.4.4. CCW Hardware EOT**

This is a counterclockwise end-of-travel fault that occurs when the system encounters a counterclockwise limit switch.

*Counterclockwise  
Limit Switch Found*

Related Parameter(s): IOLEVEL  
 Related Command(s): NONE

**3.2.4.5. CW Software EOT**

A clockwise software end-of-travel fault occurs when the user commands an axis to move beyond the position specified in the clockwise end-of-travel parameter.

*Clockwise EOT  
Parameter Exceeded*

Related Parameter(s): CWEOT  
 Related Command(s): DMRGetCwEot()  
 DMRSetCwEot()  
 All commands initiating CW motion

**3.2.4.6. CCW Software EOT**

A counterclockwise software end-of-travel fault occurs when the user commands the chosen axis to move beyond the position specified in the counterclockwise end-of-travel parameter.

*Counterclockwise  
EOT Parameter  
Exceeded*

Related Parameter(s): CCWEOT  
 Related Command(s): DMRGetCcwEot()  
 DMRSetCcwEot()  
 All commands initiating CCW motion

**3.2.4.7. Drive*****Drive Fault Input Asserted***

A fault of this type occurs when the system asserts the drive fault input for the selected axis. However, after clearing the drive fault input, this bit continues to reflect the fact that the fault occurred.

Related Parameter(s): IOLEVEL  
 Related Command(s): NONE

**3.2.4.8. Feedback*****Feedback Device Functioning Bad***

This fault signifies that the system asserted the feedback failure input from the Resolver-to-digital (R/D) channel associated with the selected axis. This typically occurs when the feedback device is not functioning properly.

Related Parameter(s): NONE  
 Related Command(s): NONE

**3.2.4.9. Programming*****Invalid Command Received by CPU***

This fault is present when the axis processor receives an invalid command from the CPU card. Any type of programming error causes this fault to occur.

Related Parameter(s): NONE  
 Related Command(s): ALL

**3.2.4.10. Master Feedback*****Feedback Failure Input on Master***

This fault signifies that the system asserted feedback failure input from the R/D channel associated with the axis configured as a master. This typically occurs when the feedback device on the master axis is not functioning properly.

Related Parameter(s): NONE  
 Related Command(s): NONE

**3.2.4.11. Homing*****Homing Fault Encountered***

This field provides an indication that the system encountered a homing fault. This typically occurs:

- while executing a home cycle,
- upon detection of the home limit switch input, or
- if the system encounters an EOT limit switch before the first resolver null.

Related Parameter(s): NONE  
 Related Command(s): DMRMoveHome()  
 DMRMoveHomeRev()  
 DMRMoveQuickHome()

**3.2.4.12. User**

This fault signifies that an application has requested a fault be generated with the DMRSetUserFault( ) command. It provides a method of generating a VMEbus interrupt from the application program.

*Request for Fault*

Related Parameter(s): NONE  
 Related Command(s): DMRSetUserFault( )

**3.2.4.13. Velocity trap**

This type of fault occurs when the actual velocity exceeds the value specified in the maximum velocity trap parameter.

*Velocity Exceeded  
Velocity Trap  
Parameter*

Related Parameter(s): VELTRAP  
 Related Command(s): DMRGetVelocityTrap( )  
 DMRSetVelocityTrap( )  
 All commands initiating motion

**3.2.4.14. Velocity command trap**

This type of fault occurs when the instantaneous commanded velocity exceeds the value specified in the maximum commanded velocity parameter.

*Instantaneous  
Commanded Velocity  
Exceeded Associated  
Parameter*

Related Parameter(s): VELCMDTRAP  
 Related Command(s): DMRGetVelocityCmdTrap( )  
 DMRSetVelocityCmdTrap( )  
 DMRMoveQuick Home( )

**3.2.4.15. Home switch tolerance**

This fault occurs during a homing sequence when the distance traveled from the time the system detects the marker pulse and the time at which the system encounters the home limit switch is less than the value specified in the HOMESWITCHTOL parameter.

*HOMESWITCHTOL  
Parameter Exceeded*

Related Parameter(s): HOMESWITCHTOL  
 Related Command(s): DMRGetHomeSwitchTolerance( )  
 DMRSetHomeSwitchTolerance( )  
 DMRMoveHome( )  
 DMRMoveHomeRev( )  
 DMRMoveQuickHome( )

**3.2.4.16. Probe input*****Probe Trigger  
Latches Position***

This fault occurs each time the probe trigger causes the position to latch. This is useful for notifying the application program that the position information is available.

Related Parameter(s): NONE

Related Command(s): DMREnableProbe( )

**3.2.4.17. CNC*****Error in CNC  
Process***

This fault indicates occurrence of an error in the CNC process to which the selected axis is dedicated. This fault indicator is inclusive of all faults that the CNC processor might generate.

Related Parameter(s): All CNC related parameters

Related Command(s): All CNC related commands

**3.2.4.18. External Feedback*****FBWINDOW  
Parameter Exceeded***

This fault occurs if the difference between the integration of the velocity command and the velocity feedback is greater than the FBWINDOW parameter.

Related Parameter(s): FBWINDOW

Related Command(s): NONE

**3.2.4.19. Safe Zone*****Violation of Safe  
Zone***

This fault occurs when the SAFEZONEMODE is active and the selected axis has violated the defined safe zone.

Related Parameter(s): SAFEZONECW  
SAFEZONECCW  
SAFEZONEMODE

Related Command(s): NONE

### 3.2.5. Fault Mask Display

The Fault Mask Display (shown in Figure 3-7) is used to display which faults have been enabled (determined by the FAULTMASK parameter setting) for each of the selected axes. Note, however, that this display only permits the user to view the fault settings and does not permit the user to make changes to the status of the bits. The fields associated with this screen match that of the Aerotech Fault Status Display shown in Figure 3-6.

*Enabled Fault Status*

Display	Time	Exit	Fault_Control																Help
Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
00 Position error limit	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
01 RMS current limit	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
02 CW Hardware EOT	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
03 CCW Hardware EOT	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
04 CW Software EOT	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
05 CCW Software EOT	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
06 Drive	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
07 Feedback	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
08 Programming	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
09 Master Feedback	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
10 Homing	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
11 User	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
12 Velocity trap	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
13 Velocity command trap	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
14 Home switch tolerance	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
15 probe input	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
16 CNC	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
17 External Feedback	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
18 Safe Zone	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
19 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
20 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
21 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
22 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
23 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
24 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
25 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
26 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
27 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			
28 Unknown	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on			

Figure 3-7. Aerotech Fault Mask Display

*Faults That Disable Axes*

**3.2.6. Disable Mask Display**

The Disable Mask Display (shown in Figure 3-8) indicates which faults should disable the axis (determined by the DISABLEMASK and FAULTMASK parameter setting). Note, however, that this display only permits the viewing of the fault settings and does not permit the user to make changes to the status of the bits. The fields associated with this screen match that of the Aerotech Fault Status Display shown in Figure 3-6.

Display	Time	Exit	Fault_Control																Help
Axis Number:			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
00	Position error limit		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
01	RMS current limit		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
02	CW Hardware EOT		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
03	CCW Hardware EOT		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
04	CW Software EOT		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
05	CCW Software EOT		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
06	Drive		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
07	Feedback		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
08	Programming		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
09	Master Feedback		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
10	Homing		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
11	User		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
12	Velocity trap		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
13	Velocity command trap		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
14	Home switch tolerance		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
15	probe input		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
16	CNC		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
17	External Feedback		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
18	Safe Zone		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
19	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
20	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
21	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
22	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
23	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
24	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
25	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
26	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
27	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	
28	Unknown		on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	on	

Figure 3-8. Aerotech Disable Mask Display



### 3.2.7. Interrupt Mask Display

The Interrupt Mask Display (shown in Figure 3-9) shows which faults will cause the system to interrupt. (determined by the INTMASK and FAULTMASK parameter setting). Note, however, that this display only permits the user to view the fault settings and does not permit the user to make changes to the status of the bits. The fields associated with this screen match that of the Aerotech Fault Status Display shown in Figure 3-6.

*Faults That Interrupt System*

Display	Time	Exit	Fault_Control																Help
Axis Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
00 Position error limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
01 RMS current limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
02 CW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
03 CCW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
04 CW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
05 CCW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
06 Drive	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
07 Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
08 Programming	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
09 Master Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
10 Homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
11 User	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
12 Velocity trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
13 Velocity command trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
14 Home switch tolerance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
15 probe input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
16 CNC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
17 External Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
18 Safe Zone	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
19 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
20 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
21 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
22 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
23 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
24 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
25 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
26 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
27 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
28 Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			

Figure 3-9. Aerotech Interrupt Mask Display

**Faults That Halt  
Axes**

**3.2.8. Halt Mask Display**

The Halt Mask Display (shown in Figure 3-10) is used to display which faults will cause the axis to halt (determined by the HALTMASK and FAULTMASK parameter setting). Note, however, that this display only permits the user to view the fault settings and does not permit the user to make changes to the status of the bits. The fields associated with this screen match that of the Aerotech Fault Status Display shown in Figure 3-6.

Display	Time	Exit	Fault_Control																Help
Axis Number:			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
00	Position error limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
01	RMS current limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
02	CW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
03	CCW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
04	CW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
05	CCW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
06	Drive	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
07	Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
08	Programming	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
09	Master Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	Homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
11	User	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
12	Velocity trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
13	Velocity command trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
14	Home switch tolerance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
15	probe input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
16	CNC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
17	External Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
18	Safe Zone	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
19	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
20	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
21	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
22	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
23	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
24	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
25	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
26	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
27	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
28	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Figure 3-10. Aerotech Halt Mask Display

### 3.2.9. Aux Mask Display

The Auxiliary Mask Display (shown in Figure 3-11) shows which faults will turn off the auxiliary output (determined by the AUXMASK and FAULTMASK parameter setting) associated with an axis. Note, however, that this display only permits the user to view the fault settings and does not permit the user to make changes to the status of the bits. The fields associated with this screen match that of the Aerotech Fault Status Display shown in Figure 3-6.

*Faults That Turn Off  
Auxiliary Output*

Display	Time	Exit	Fault_Control																Help
Axis Number:			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
00	Position error limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
01	RMS current limit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
02	CW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
03	CCW Hardware EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
04	CW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
05	CCW Software EOT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
06	Drive	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
07	Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
08	Programming	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
09	Master Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	Homing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
11	User	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
12	Velocity trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
13	Velocity command trap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
14	Home switch tolerance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
15	probe input	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
16	CNC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
17	External Feedback	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
18	Safe Zone	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
19	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
20	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
21	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
22	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
23	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
24	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
25	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
26	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
27	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
28	Unknown	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Figure 3-11. Aerotech Aux Mask Display

Status Screen Update  
Time

3.3. Time Pull-down Menu

The Time Pull-down Menu (shown in Figure 3-12) displays and sets the update time for all information in the status screens. The purpose for having a low (50 ms) update time is to allow the system to run at a faster speed. High speeds cause the system to slow down. However, an update time of 50 ms will cause multitasking to slow down and outputs to be updated slowly.

Display	Time	Exit	Fault_Control	Help
Axis Num	50 ms.			
00 drive e	100 ms.			
01 aux ou	250 ms.			
02 CW in	500 ms.			
03 CCW i				
04 home input				
05 drive fault input				
06 at home				
07 done		on	on	on
08 in-position		on	on	on
09 faulted		-	-	-
10 probe input		on	on	on
11 marker input		on	on	on
12 hall input #1		on	on	on
13 hall input #2		on	on	on
14 hall input #3		-	-	-
15 hall input #4		on	on	on
16 move direction		-	-	-
17 moving		-	-	-
18 accel phase		-	-	-
19 decel phase		-	-	-
20 homing		-	-	-
21 feedrate override		-	-	-
22 profile mode		-	-	-
23 sync mode		-	-	-
24 cam table enable		-	-	-
25 homing direction		-	-	-
26 continuous move		-	-	-
27 queued command		-	-	-
28 hold active		-	-	-

Figure 3-12. Time Pull-down Menu

### 3.4. Exit Pull-down Menu

The Exit Pull-down Menu (shown in Figure 3-13) is used to exit the Status Utility and return to the OS2 window.

*Quit The Status Utility*

Aerotech Axis Status Display																	
Display	Time	Exit	Fault_Control														Help
Axis Number:		Exit	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
00	drive enable	Resume	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
01	aux output enable		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
02	CW input		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
03	CCW input		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
04	home input		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
05	drive fault input		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
06	at home		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
07	done		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
08	in-position		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
09	faulted		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	probe input		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
11	marker input		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
12	hall input #1		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
13	hall input #2		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
14	hall input #3		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
15	hall input #4		on	on	on	on	on	on	on	on	on	on	on	on	on	on	
16	move direction		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
17	moving		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
18	accel phase		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
19	decel phase		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
20	homing		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
21	feedrate override		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
22	profile mode		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
23	sync mode		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
24	cam table enable		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
25	homing direction		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
26	continuous move		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
27	queued command		-	-	-	-	-	-	-	-	-	-	-	-	-	-	
28	hold active		-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Figure 3-13. Exit Pull-down Menu

#### 3.4.1. Exit

This option has not been implemented at this time.

#### 3.4.2. Resume

This option has not been implemented at this time.

### 3.5. Fault\_Control Pull-down Menu

*Latched and Unlatched Faults*

The Fault Control menu (shown in Figure 3-14) is used to set the fault display (see Figure 3-6) as either latching or not and also is used to clear any faults that are displayed. When a fault is latched status will read and display the current fault status, but it is up to the user to select the Clear Faults option to clear the fault(s). For faults that are not latched, the system will automatically cleared them.

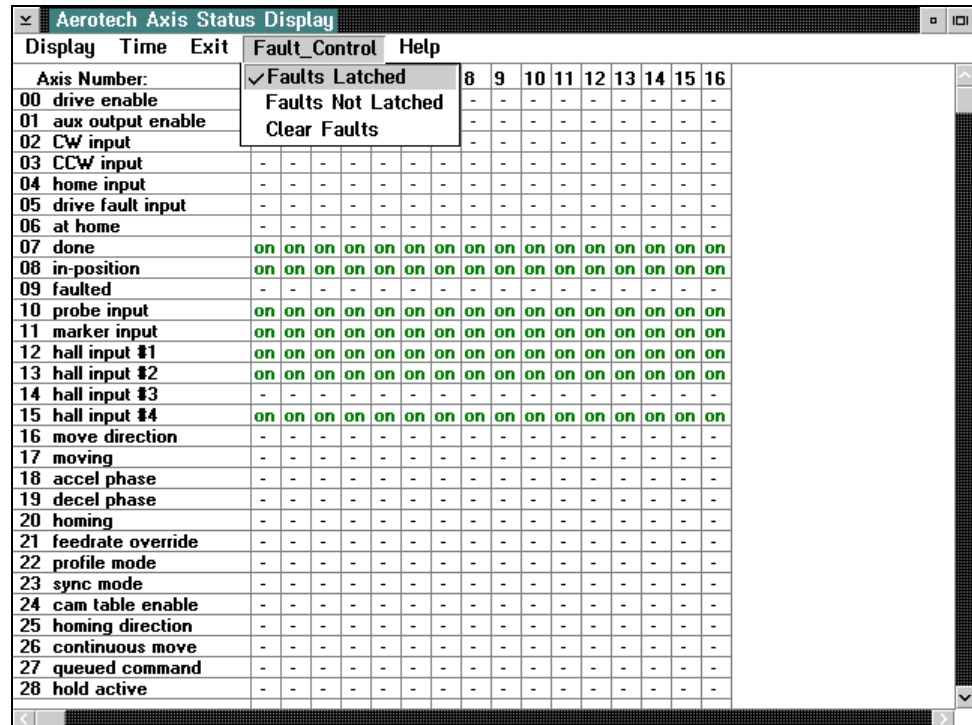


Figure 3-14. Fault\_Control Pull-down Menu

#### 3.5.1. Faults Latched

*Latched Faults*

This option indicates that the faults are latched. To clear the faults, select Clear Faults.

#### 3.5.2. Faults Not Latched

*Un-latched Faults*

This option indicates that the faults are not latched. The faults are cleared continuously as they occur.

#### 3.5.3. Clear Faults

*Clearing Faults*

Selecting the Clear Faults option will clear all faults. Faults that are cleared will appear on the display with a dash (-) while those that continue to occur will be displayed as ON.

### 3.6. Help Pull-down Menu

The Help Pull-down Menu (shown in Figure 3-15) contains a single option - the Abort option. The Abort option is explained in the next section. Future revisions of this utility may include additional help topics.

*Help Topics*

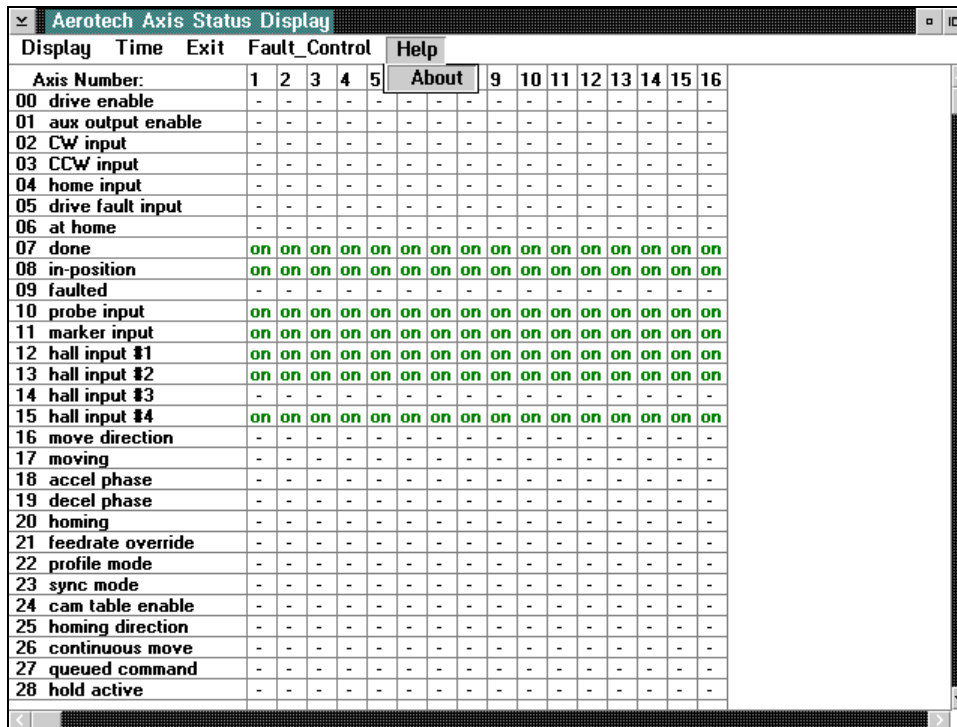


Figure 3-15. Help Pull-down Menu

### 3.6.1. About

*Version of Software*

This option, located within the Help Pull-down Menu, provides information relative to the version of software, the date of origin of the Status Utility and copyright information.

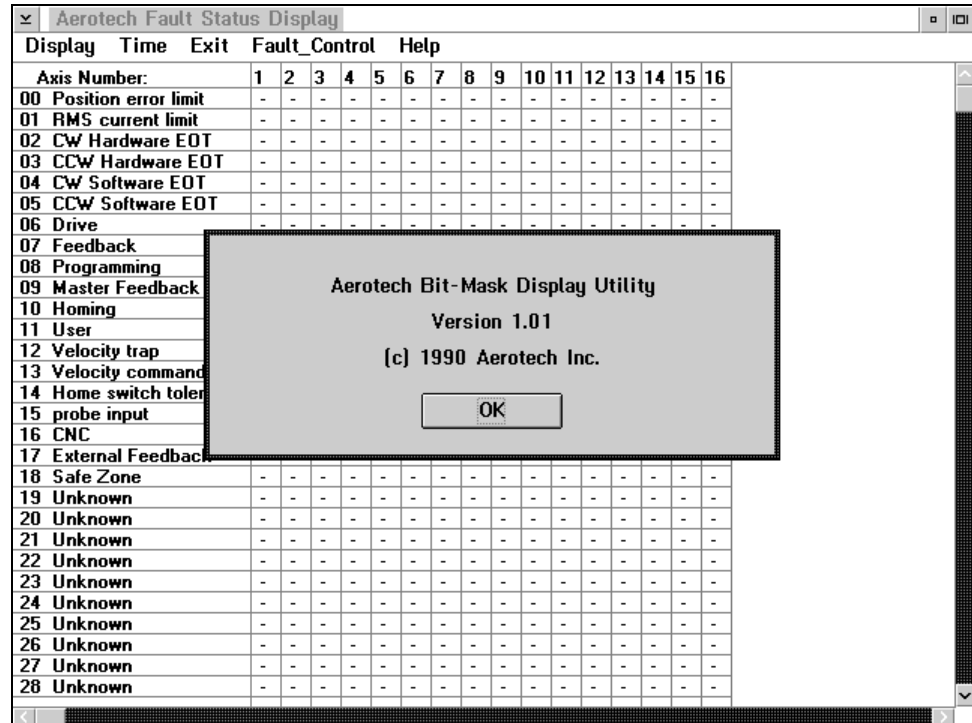


Figure 3-16. About Dialog Box





**CHAPTER 4: PLOT**

<b>In This Section:</b>	
• Overview .....	4-1
• File Pull-down Menu .....	4-2
• Plot Pull-down Menu .....	4-5
• Trigger Pull-down Menu .....	4-7
• Collect Pull-down Menu .....	4-10
• Display Pull-down Menu .....	4-11
• Axis Pull-down Menu .....	4-13
• Tuning servo Loops .....	4-16
• Tuning Procedure for Servo Loops .....	4-18
• AC Brushless Motor Tuning Tip .....	4-29
• Tuning With Tachometer Feedback .....	4-30
• Tuning Tachometer Loops .....	4-32

**4.1. Overview**

Plot is a data acquisition application used to measure, adjust and record the servo performance of each axis of motion. The axis processor card passes servo information to the AT front end through the VME backplane that causes the Plot Utility to process the information. The system presents this information in a graphical form that resembles an oscilloscope or strip chart recording. In addition to showing servo performance in real time, the Plot Utility can save the data collections for later recall.

Upon execution of the Plot Utility, a dialog box (shown in Figure 4-1) is displayed showing the current axis (Axis 1) and plot status (idle or done).

*Plot in Definition*

*Axis/Plot Status*

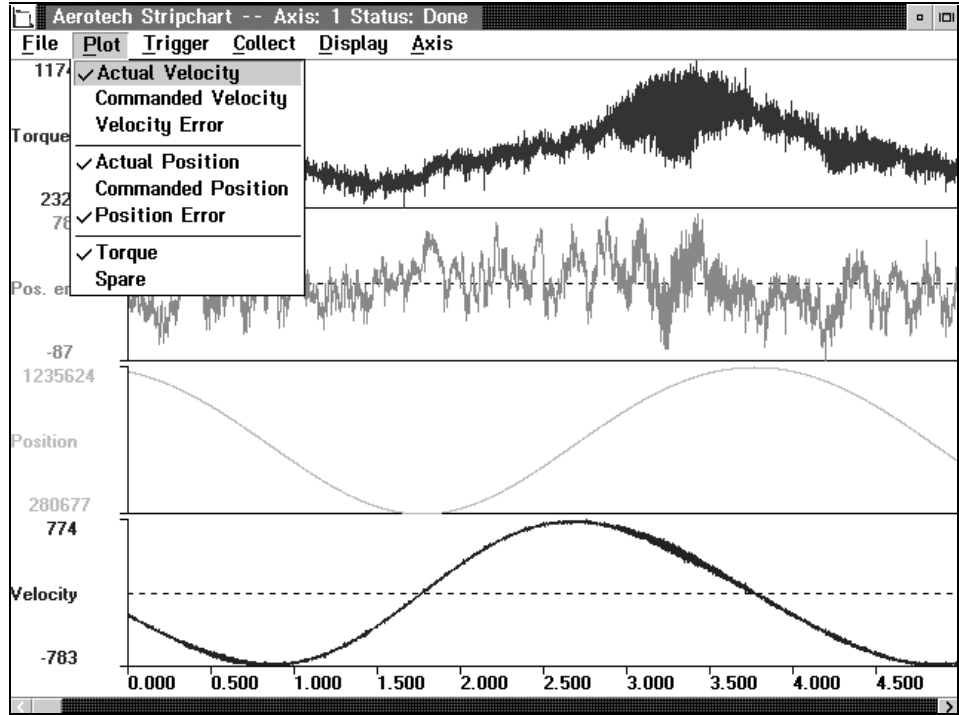


Figure 4-1. Plot Mainmenu Dialog Box

### 4.2. File Pull-down Menu

#### *Load/Save Files*

The File Pull-down Menu permits the user to load previously saved files, save the current data collection to a specific file and inquire about the version of plot currently running.

To identify the above selections, refer to Figure 4-2.

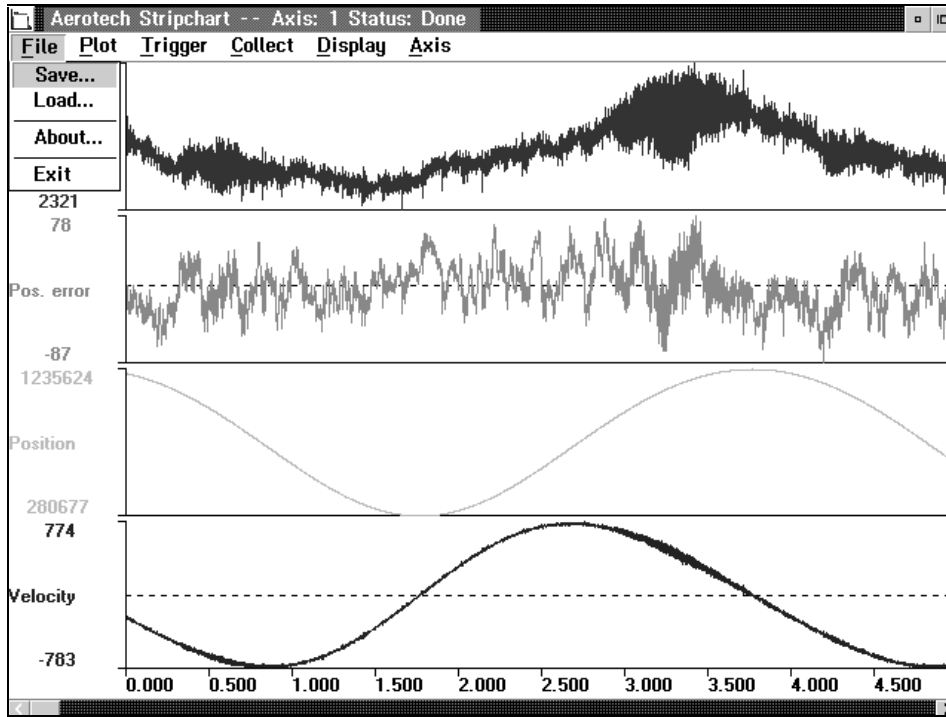


Figure 4-2. File Pull-down Menu

#### 4.2.1. Save.../Load...

Choosing the Load or Save option from the File Pull-down Menu causes a File Selection Dialog Box to appear on the screen (see Figure 4-3). To enter the file name to load or save, position the cursor in the Enter Filename: entry field and manually enter the file name. To initiate this command, press the <Enter> key.

*Opening and Storing Files*

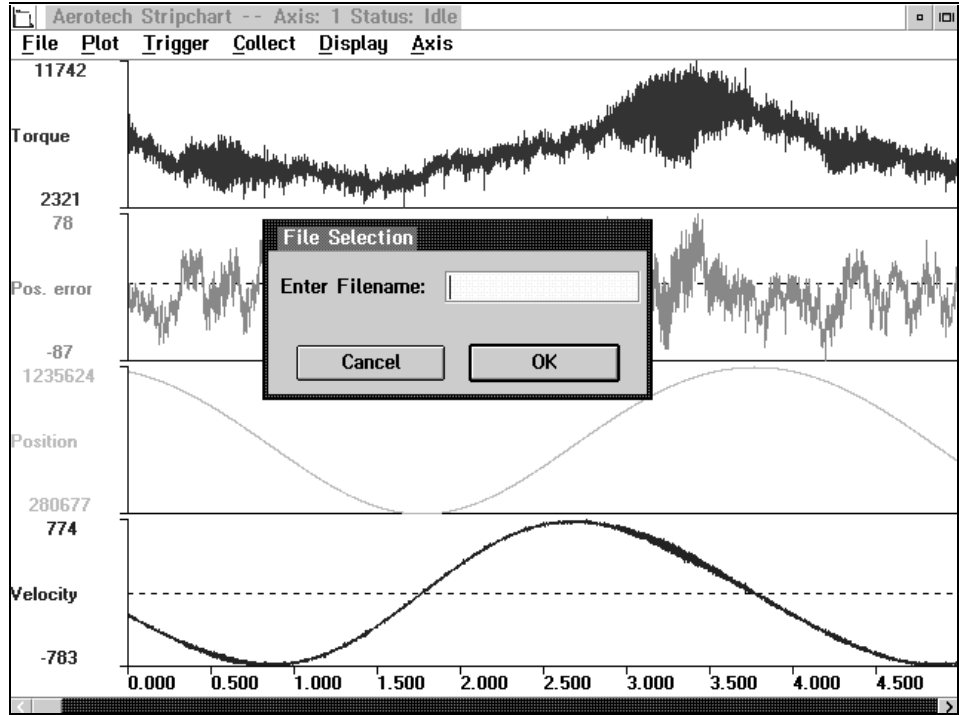


Figure 4-3. File Selection Dialog Box

### 4.2.2. About...

*Level of Software*

This option provides information pertaining to the Plot Utility version level and copyright information.

### 4.2.3. Exit

*Quitting the Software*

The Exit option permits the user to exit the Plot Utility.

### 4.3. Plot Pull-down Menu

This menu toggles which channels of servo information to display (ON or OFF). To determine the types of choices that exist, refer to Figure 4-4. These channels include Actual Velocity, Commanded Velocity, Velocity Error, Actual Position, Commanded Position, Position Error, Torque and a Spare channel. Selecting one of these channels causes the requested information to appear on the display. The information for channels not selected remains in memory, therefore the user can select it anytime data collection takes place. Each time the user save a file, the system saves all other additional information linked to that file.

*Displaying Servo Data*

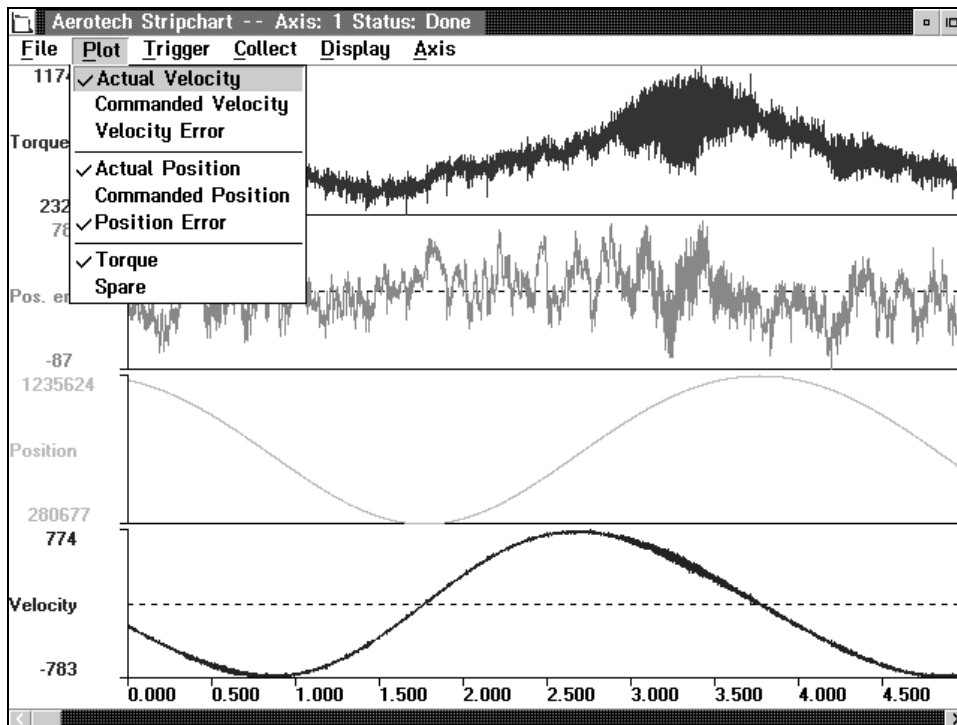


Figure 4-4. Plot Pull-down Menu

#### 4.3.1. Actual Velocity

This velocity is the actual speed that the motor reached with respect to velocity error. Certain parameters in Debug960 can affect this velocity.

*Velocity, Actual*

#### 4.3.2. Commanded Velocity

This is the velocity commanded by the user. Certain parameters in Debug960 can affect the commanded velocity.

*Velocity, Commanded*

<i>Velocity, Error</i>	<b>4.3.3. Velocity Error</b> The velocity error is the difference between the commanded velocity and the actual velocity.
<i>Position, Exact</i>	<b>4.3.4. Actual Position</b> This position is the exact position that was reached with respect to position error. Certain parameters in Debug960 can affect this position.
<i>Position, Specified</i>	<b>4.3.5. Commanded Position</b> The command position is the position that the user specifies. Certain parameters in Debug960 can affect the commanded position.
<i>Position, Error</i>	<b>4.3.6. Position Error</b> Position error is the difference between the actual position and the commanded position.
<i>Motor Rotation</i>	<b>4.3.7. Torque</b> The torque option allows the user to control the force of motor rotation.
<i>Displaying Unique Information</i>	<b>4.3.8. Spare</b> Spare is a custom input channel used to display unique information. This input requires custom optional axis firmware not provided with a typical system.

### 4.4. Trigger Pull-down Menu

There are five choices offered through this menu. For a list of these choices, refer to Figure 4-5. This menu behaves similar to the trigger control on an oscilloscope.

*Trigger*

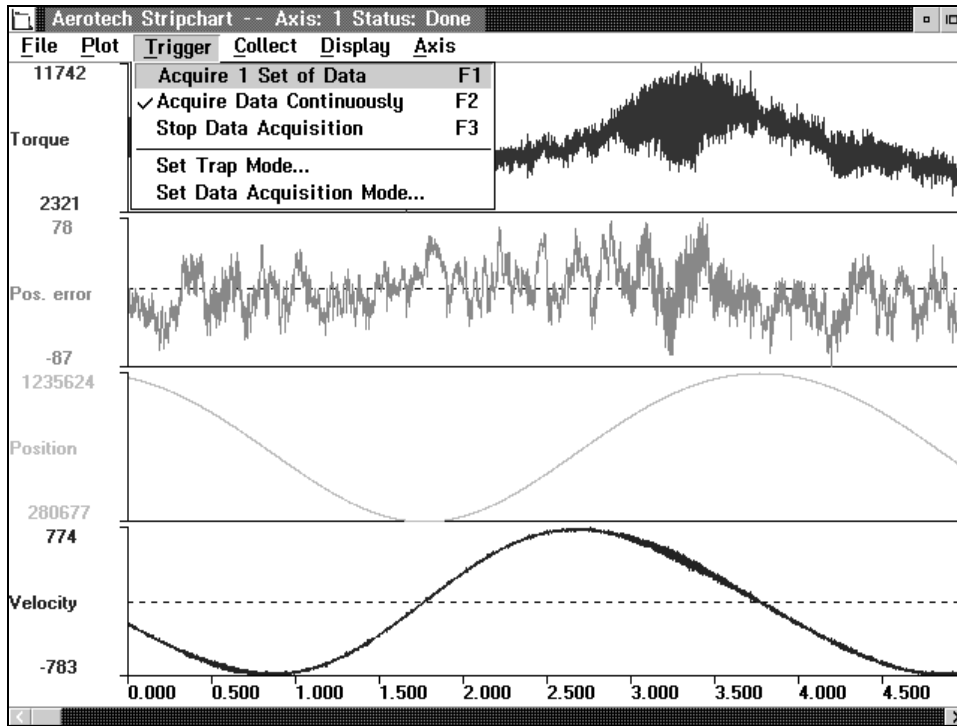


Figure 4-5. Trigger Pull-down Menu

#### 4.4.1. Acquire 1 Set of Data

The Acquire 1 Set of Data option performs the same function as the single sweep control on a scope. Using this control allows one collection of data only. Note that the user may select this option with the F1 function key.

*Collect One Data Set*

#### 4.4.2. Acquire Data Continuously

The Acquire Data Continuously option collects data on a continuous basis and is the equivalent of putting an oscilloscope in the auto or freerun modes. Note that the user may select this option with the F2 option key.

*Collect Data Continuously*

#### 4.4.3. Stop Data Acquisition

The Stop Data Acquisition feature halts the data collection at the end of the current sample period. Note that the user may select this option with the F3 function key.

*Stop Data Collection*

#### 4.4.4. Set Trap Mode...

##### *Servo PerformanceTraps*

The Set Trap Mode Dialog Box, shown in Figure 4-6, allows the user to set servo performance traps. The trap mode works in conjunction with the trigger control. After sending the trap, all triggering and data collection stops unless another motion command exists or the user selects either of the two available options: Acquire 1 set of Data or Acquire Data Continuously. When using a trap to capture an event generated by an application outside of plot, the user must then set the trigger at Acquire Data Continuously.

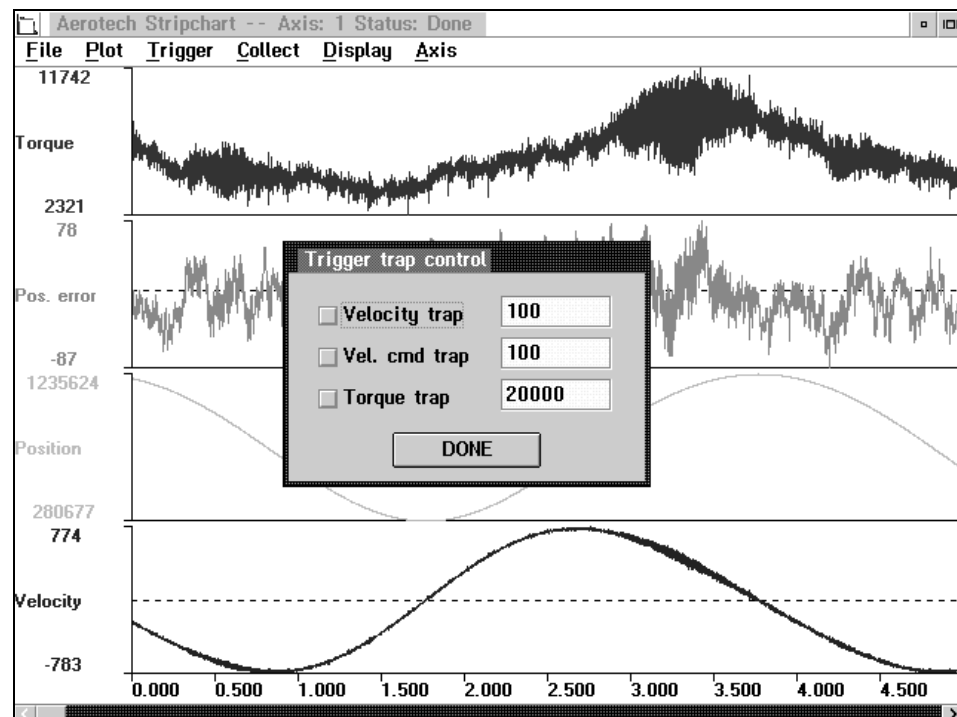


Figure 4-6. Set Trap Mode Dialog Box

##### *Setting Traps*

There are three choices to choose from when in the trap menu. These choices include: Velocity trap, Velocity command trap and Torque trap. To set a trap it is first necessary to put a value in the box for the trap chosen and then check that trap. As an example, a value of 1,000 in the Torque trap entry field sets this trap if the system exceeds that value while collecting data. The value put in the Torque trap entry field does not carry a sign and causes a trap to get set regardless of the sign of the actual velocity, velocity command or the torque command. If the Plot Utility generates the motion profile, the system halts the motion, but the enable line remains on. The data collection stops upon detection of a set trap and a message indicates the sample point at which the system exceeded the trap value. If the generation of a motion profile occurred outside the Plot Utility, the data collection stops, indicating the sample point at which the trap got set, but does not halt the motion.



### 4.4.5. Set Data Acquisition Mode...

The final choice under the Trigger Pull-down Menu is the Set Data Acquisition Mode. Figure 4-7 shows an illustration of the screen for this mode.

*Modes of Data Acquisition*

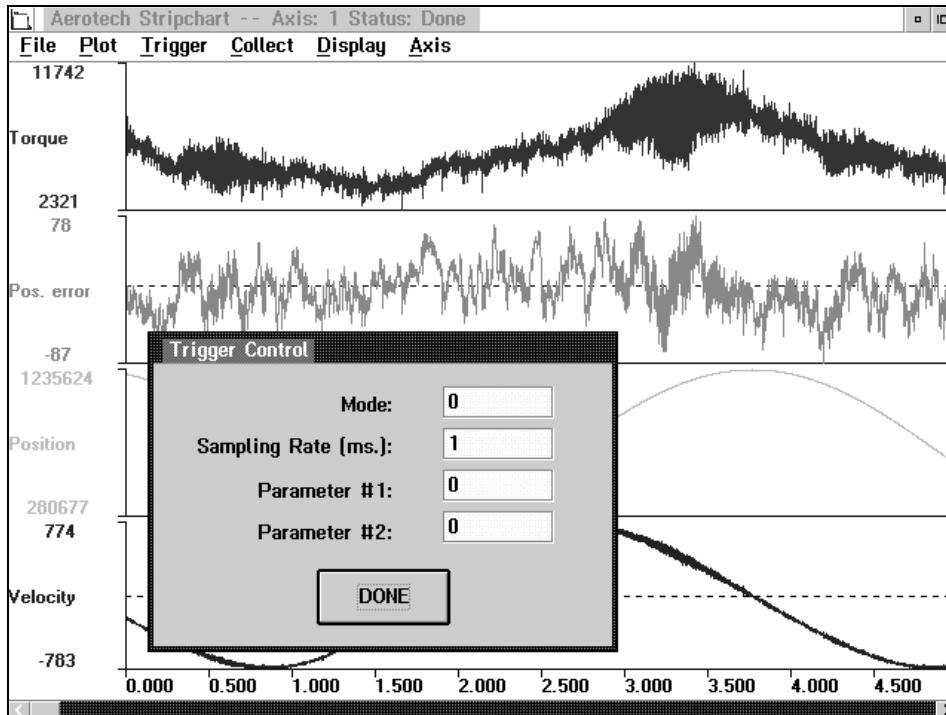


Figure 4-7. Set Data Acquisition Mode Dialog Box

This screen determines when to start data collections. There are five available modes for setting data acquisition (see Table 4-1).

Table 4-1. Data Acquisition Modes

Mode	Description
0	starts data collection at the same point as when motion profile started in plot.
1	determines that the master position information is used to control the trigger.
2	uses the beginning of the cam table to control the trigger.
3	uses the torque command information to control the trigger.
4	used for an external software trigger source such as an application program.

*Setting Data Acquisition*

Also in this screen are two entry fields that allow the user to associate a particular parameter with each trigger mode. The following information (see Table 4-2) describes the two parameters. These parameters are Parameter #1 and Parameter #2.

*Associating Parameters with Modes in Data Acquisition*

**Table 4-2. Data Acquisition Parameters**

Parameter	Description
1	works along with mode 1 or mode 3, and determines when to set the value of either the position or the torque.
2	Upon selecting mode 1 or mode 3, parameter 2 determines when to start the trigger for data collection. A value of zero (0) initializes the trigger process when the Position or Torque entry field is less than or equal to the value installed in the Parameter 1 entry field. A value of one (1) in this field permits the trigger to begin when the position or torque is greater than the value installed in the Parameter 1 entry field.

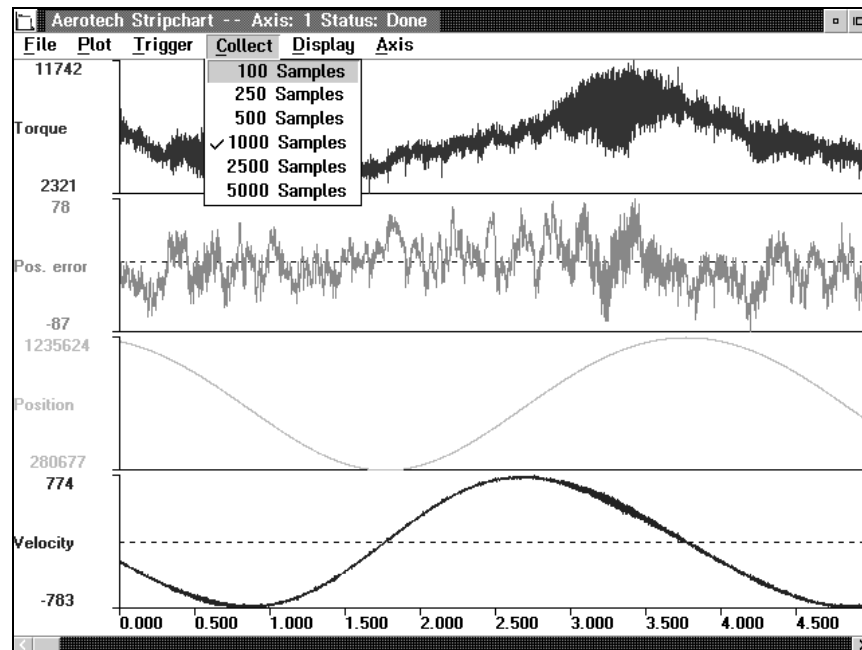
*Data Collection Sampling Rate*

The remaining choice in the Set Data Acquisition Mode is the Sampling Rate. This value determines the sampling rate (in msec) for the data collection. The default value and the fastest sampling rate is at one (1) millisecond. This setting, along with the value chosen in the Collect Pull-down Menu, determines the total sample time and the resolution of the sample.

**4.5. Collect Pull-down Menu**

*Sample Time*

Through this menu the user can set the number of samples to be taken. The number selected can be any one of the choices in Figure 4-8. The number of chosen samples multiplied by the selected Sampling Rate determines the total time that a given sample takes.



**Figure 4-8. Collect Pull-down Menu**

### 4.6. Display Pull-down Menu

This menu allows the user to choose the resolution for the displayed data. This is similar to adjusting the time base on a digital storage scope with a captured waveform. The user can choose a display time base that is much smaller than a collect time, allowing for a more detailed image of the collected data. The user can also control the position along the expanded data collection by using the horizontal scroll box located at the bottom of the screen. Figure 4-9 indicates the options available from within the Display Pull-down Menu.

*Displayed Data Resolution*

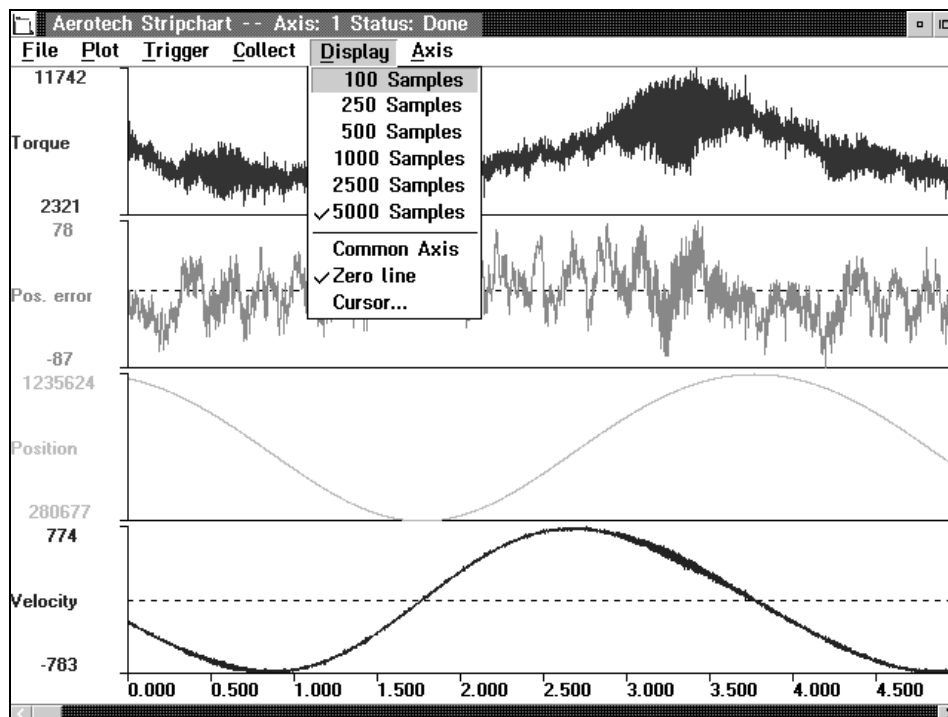


Figure 4-9. Display Pull-down Menu

#### 4.6.1. Samples

This represents the number of points for a display. This is equivalent to adjusting the time base on a digital storage scope with a stored waveform.

#### 4.6.2. Common Axis

This places all selected items plotted on the screen.

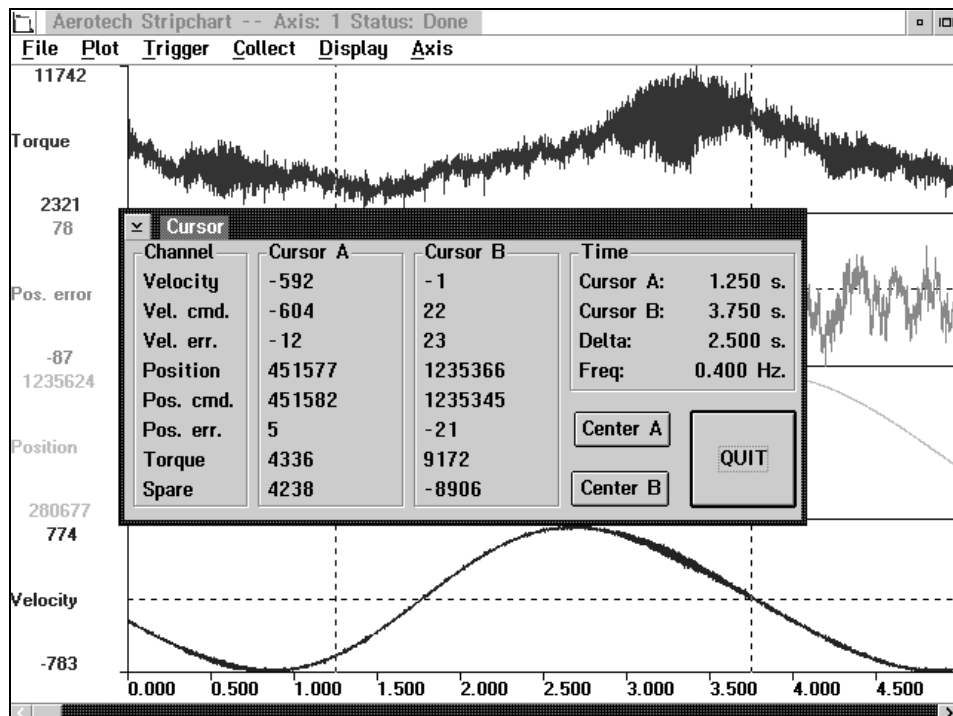
#### 4.6.3. Zero Line

Also located under the Display Pull-down Menu is an option that allows the user to place a Zero Line on the display with the displayed data. Additionally, the user can toggle the screen format between a common axis and separate axis.

**4.6.4. Cursor...**

*Cursor Measurement Tool*

Selecting the Cursor option from the Display Pull-down Menu causes a Cursor Measurement Tool Dialog Box to appear in the lower right-hand corner of the screen (see Figure 4-10).



**Figure 4-10. Cursor Measurement Tool Dialog Box**

*Positioning the Cursor*

This tool measures the values of the sampled parameters at both cursor points. It is possible to move the cursors to a specific point in the display by clicking and dragging with the mouse. Also in this window are sample points that show the location of each cursor, the time difference between the two cursors and the frequency (one cycle) between the two cursors. Center A and Center B boxes allow the user to place either cursor at the center of the displayed screen. A very good use for this utility involves the study of a periodic disturbance.

### 4.7. Axis Pull-down Menu

Selecting the Axis Pull-down Menu provides the user with many options (see Figure 4-11). Some of these options include: Single Step, Auto Step, Halt Motion, Start Continuous Motion, Tune, Set Speed/Distance and Select Axis. Each of these options are defined throughout the following sections.

*Configuring the Axis*

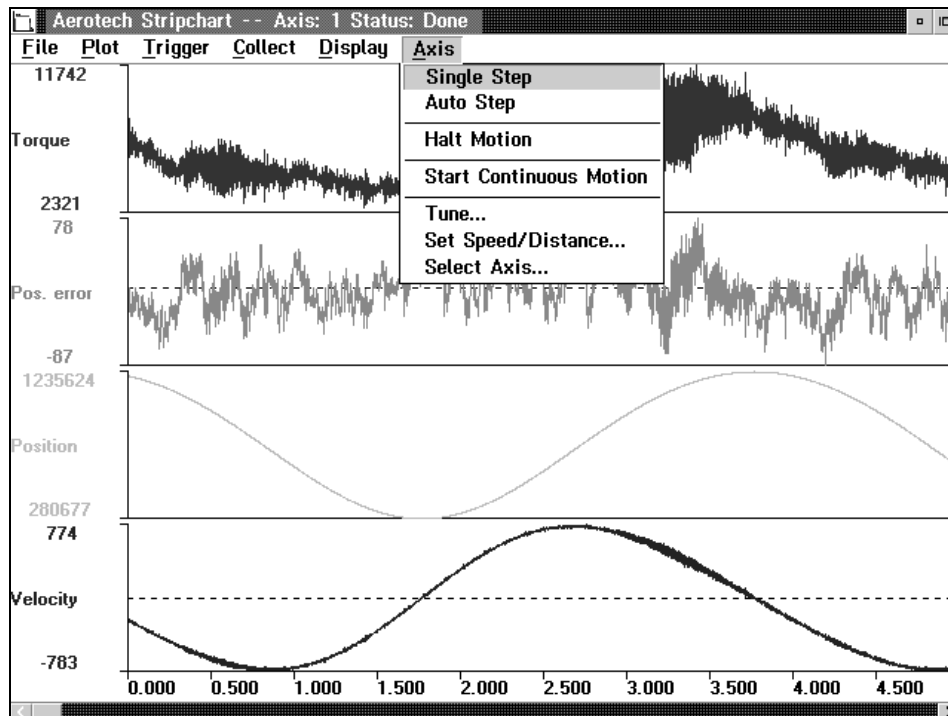


Figure 4-11. Axis Pull-down Menu

#### 4.7.1. Single Step

The Single Step option executes a single step using the lengths and speed values setup in the Continuous Motion Control Menu. A step starts when the data collection sample point is at zero. To ensure this, set the Trigger Control to Acquire 1 Set of Data position.

*One Step Motion*

#### 4.7.2. Auto Step

Auto Step executes a positive and negative motion profile using the length and speed values established in the Continuous Motion Control Menu. The direction of motion alternates between positive and negative with each sample period. The speeds and lengths should be checked to make sure the motion profile does not take longer than the sampling time or any mechanical restrictions that the axis may have.

*Motion Profile in Two Directions*

### 4.7.3. Halt Motion

#### *Motion Profile Stop*

The Halt Motion option stops any motion profile immediately.

### 4.7.4. Start Continuous Motion

#### *Motion Profile Begin*

Start Continuous Motion starts motion in the direction and the speed setup in the Continuous Motion Control Menu.

### 4.7.5. Tune...

#### *Servo Gain Settings*

Selecting the Tune option causes a window that shows the selected axis servo gain settings to appear on the screen (see Figure 4-14).

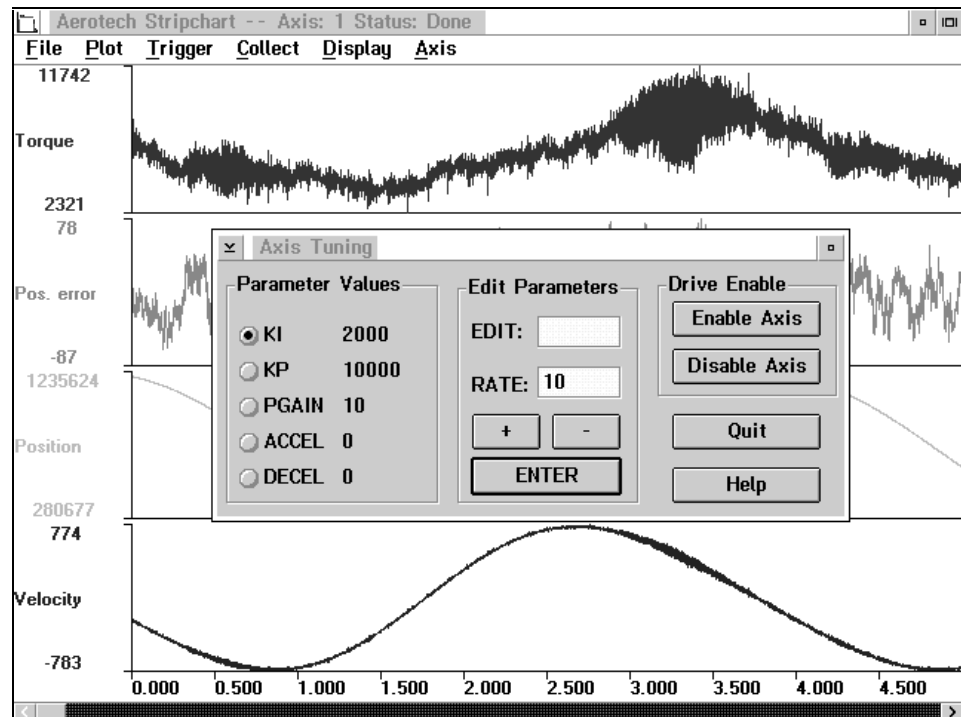


Figure 4-12. Axis Tuning Dialog Box

#### *Gain Adjustments*

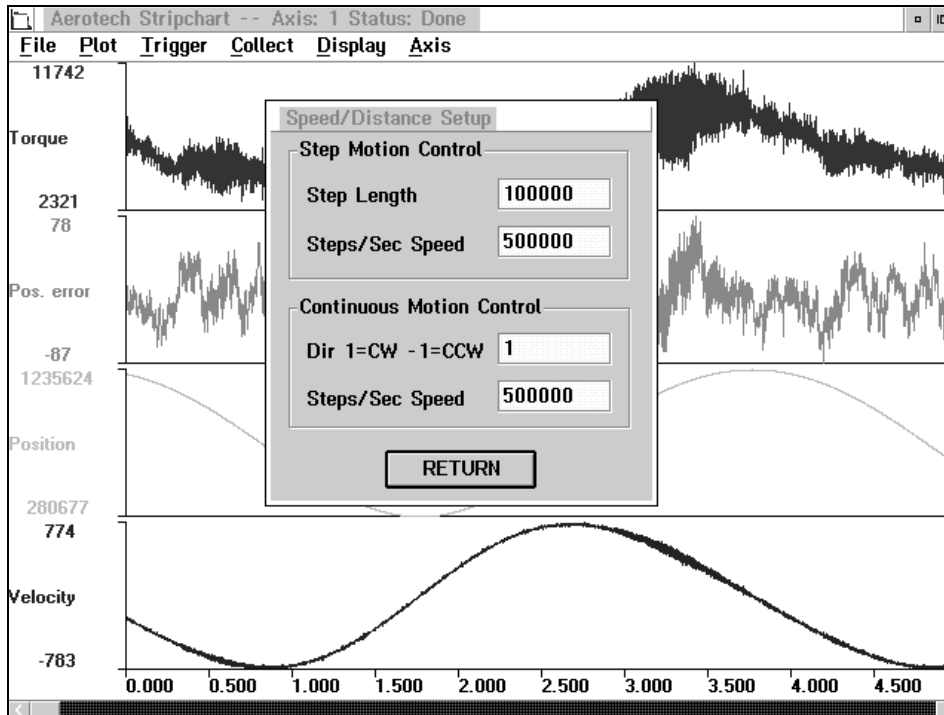
Included in this window are the axis enable control boxes. It is possible to alter the servo gains in this screen while observing the effect on the motion profile on the main screen. The user can change the gains by using the edit box or the +/- increment boxes.

For an explanation of the motion commands (e.g., single step, auto step, halt motion and start continuous motion), shown in the Axis Tuning Dialog Box, refer to sections 4.7.1 through 4.7.4).

**4.7.6. Set Speed/Distance...**

Choosing the Set Speed/Distance option from within the Axis Pull-down Menu brings about the dialog box shown in Figure 4-13.

*Axis Speed  
Axis Distance*



**Figure 4-13. Speed/Distance Setup Dialog Box**

Through the Set Speed/Distance window the user can setup the step length and the step speed for the stop motion control and also the continuous motion direction and speed for the continuous motion control. The units of measure for this screen are the feedback devices resolution for the axis selected.

*Stop Motion Based  
on Length and Speed  
of a Step*

### Changing and Selecting Axes

#### 4.7.7. Select Axis...

Selecting the Select Axis option permits the user to select or change an available axis (see Figure 4-12). Only one (1) axis may be selected at a time.

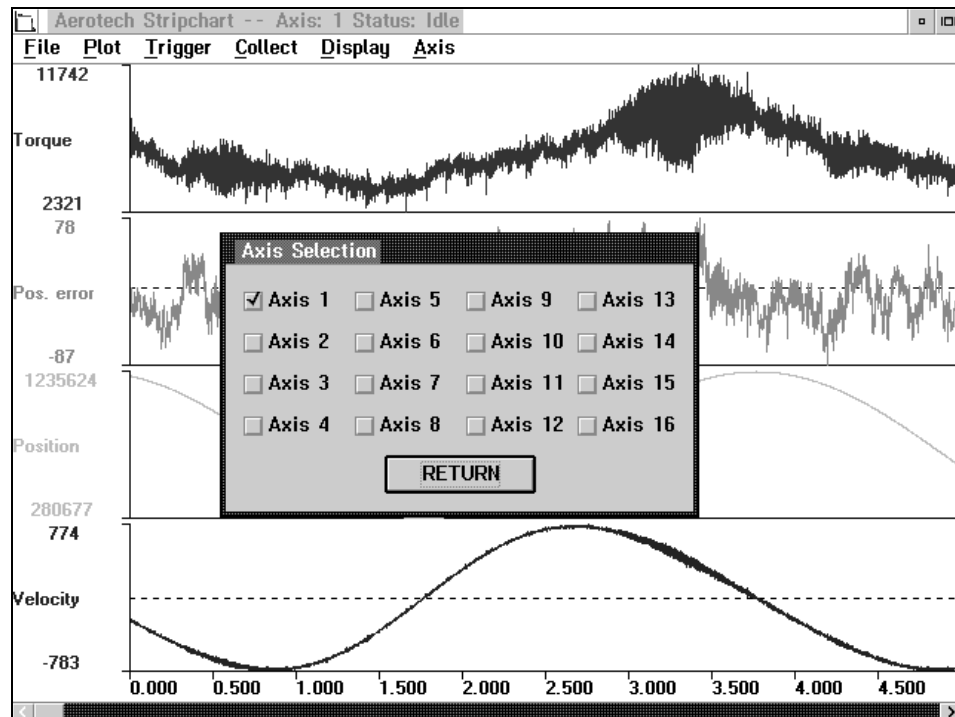


Figure 4-14. Axis Selection Screen

#### 4.8. Tuning servo Loops

This section explains the procedures for tuning a UNIDEX 600 Series controller servo loop with and without tachometer feedback using the Plot.exe utility. The UNIDEX 600 firmware/utility software package (UTIL600-OS/2) contains a graphical tool that can be used to display the effects of the servo loop gain settings.

Included in this section are step-by-step procedures for tuning motors controlled by the UNIDEX 600 to yield optimal performance. Optimal performance can be defined two ways. First, it may be defined as the smallest amount of allowable position error according to user-defined tolerances while still having smooth motion. Alternatively, it may be defined as the smallest amount of position error that is allowable by user-defined tolerances while having minimal settling time without concern for smoothness of motion.

To produce smooth motion, the Velocity Loop takes precedence, and an attempt will be made to have a tight Velocity Loop and a tight Position Loop. To have very little settling



time, the Position Loop takes precedence and an attempt will be made to have a tight Position Loop. The UNIDEX 600 uses a dual control loop having an inner velocity loop and an outer position loop. The servo loops update time is defined by the G130/G131 G-code, which allows either a 1 kilohertz or 4 kilohertz servo loop update rate. Refer to Figure 4-15 for a block diagram of the Servo Loop.

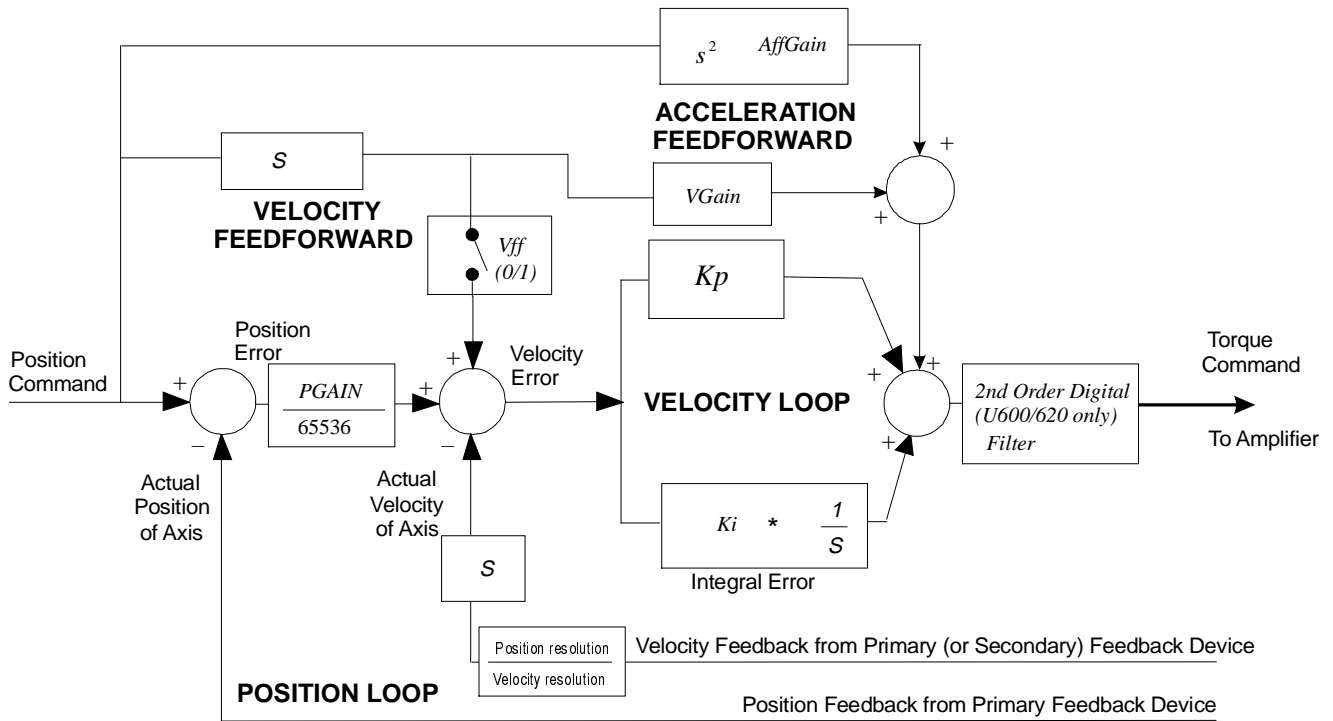


Figure 4-15. Servo Loop Diagram

Before tuning an axis, the motor and feedback device must be properly configured.

For additional information, see the following documentation:  
UNIDEX 600 or 630/631 Hardware Manual



### 4.8.1. Kp - Proportional Gain

This is the Proportional Gain. It is part of the Velocity Loop in the UNIDEX 600's Servo Loop. This parameter reduces the amount of velocity error. Also, this gain has a dampening, or stabilizing effect in the servo loop. This is the first servo loop parameter to adjust.

#### **4.8.2. Ki - Integral Gain**

This is the Integral Gain. It is part of the Velocity Loop in the UNIDEX 600's Servo Loop. This parameter reduces the amount of velocity error. Moreover, it removes steady-state position errors at the end of a move. This is the second servo loop parameter to adjust.

#### **4.8.3. PGain - Position Gain**

This is the Position Gain. It is the only parameter in the Position Loop of the UNIDEX 600's Servo Loop. This parameter reduces the amount of position error and decreases the settling time. This is the third servo loop parameter to adjust.

#### **4.8.4. Vff - Velocity Feedforward Gain**

This is the Velocity Feedforward Gain. It is the only parameter in UNIDEX 600's velocity feedforward loop. This parameter is either 1, enabling velocity feedforward, or 0 to disable it. This parameter is used to minimize position errors proportional to velocity.

#### **4.8.5. AffGain - Acceleration Feedforward Gain**

This is the Acceleration Feedforward Gain. It is the only parameter in the acceleration feedforward loop in the UNIDEX 600's servo loop. This parameter is used to remove position error during the acceleration and deceleration phase of a move. Normally this parameter has a magnitude of less than 200. The user may not have to set this servo loop parameter greater than zero.

#### **4.8.6. Alpha - AffGain Filter**

The Alpha parameter is used within the acceleration feedforward portion of the servo loop. It is responsible for filtering the effect of the AffGain parameter reducing the noise introduced into the servo loop by the rapid velocity changes the AffGain produces. The Alpha parameter has inverse scaling, 65,536 produces minimum filtering and 1, maximum filtering

#### **4.8.7. VGain - Constant Velocity Gain**

This parameter is used primarily for tachometer based (velocity loop) systems to reduce the position error (following error) during constant velocity. It can also be used in torque mode systems with large frictional loads. The VGain parameter is multiplied by the commanded velocity to produce a voltage proportional to velocity that is added to the DAC output value to the servo amplifier that minimizes the following error.

#### **4.8.8. Tuning Procedure for Servo Loops**

The following procedure can be used as a guide when tuning the UNIDEX 600 servo loop. This procedure does not apply to motors with tachometers. Figure 4-16 shows the overall tuning process with the Plot utility. The tuning process discussed in this section was performed using the "X" (lower) Axis of an Aerotech ATS3220140P X-Y open frame table, with a BM130 AC brushless motor and an AS32030 amplifier at 160VDC. The table had no load mounted to the table, except for that of the Y axis that is physically part of the table. The user's system may behave differently and have different values for servo loop gains. However, the overall process is the same and the same process can be repeated for other axes. When adjusting each of the servo loop gains, the user will essentially be following the procedure below:

1. Exercise the axis through a move profile typical to your application.
2. Observe the servo loop performance with the Plot utility.
3. Make a decision on whether to increase or decrease the value of the servo loop gain parameter or proceed to the next servo loop parameter.
4. Repeat.

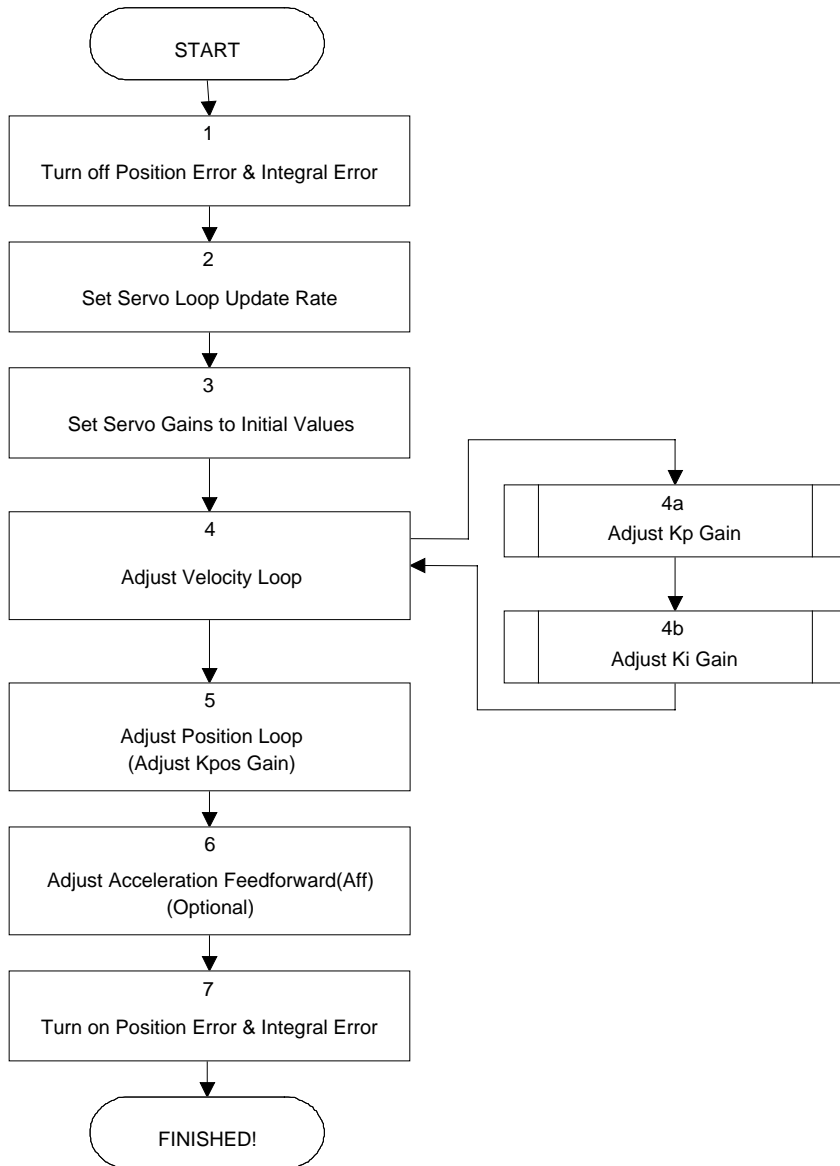


Figure 4-16. Flowchart of Overall Tuning Process

The following is a step-by-step procedure for tuning motors without tachometers.



Please read each step thoroughly before performing the task.

1. Turn off the “Position Error” and “Velocity Error” bits in the FaultMask axis parameter, by starting the Zsid960 utility. Download the axis firmware if this has not been done. Select the axis that you wish to tune, with the AXIS command. Record the current value of the FaultMask axis parameter by using the GET command as follows:

**GET FaultMask**

A value will be displayed after entering the preceding command. Record this value to restore it after tuning the axis.

To disable these faults set the FaultMask axis parameter to 8398 as follows:

**SET FaultMask 8398**

2. Set the servo loop parameters to the initial values indicated in the chart below.

**Table 4-3. Initial Torque Mode Servo Loop Parameter Values**

PGain	Ki	Kp	Vff	VGain	AffGain	DACOffset	Alpha
0	0	1000	0	0	0	0	65536

3. Prepare the Plot utility for tuning by performing the following functions:
  - a. Press the maximize button on the title bar so that the window fills the entire screen.
  - b. Use the ‘Axis’ menu select the ‘Select Axis’ menu selection to select the axis to be tuned.
  - c. Use the ‘Plot’ menu, to select only Velocity Command, Velocity Error, and Position Error.
  - d. Select the ‘Axis’ menu to select the “Set speed/distance’ menu selection to set the Distance and Speed entry fields (in steps & steps/second) for a typical move profile, define the desired AccelMode and DecelModes and enter the appropriate Accel/Decel Rate/Times into the dialog boxes, found in the ‘Tune’ box that may be selected from the ‘Axis’ menu.

When the “Single+/-” button’s are selected, the Axis will move the specified distance in the direction determined by the Single+/- button selected.

4. Adjust the Velocity Loop using Kp. The PGain and Ki have been set to zero (0) to eliminate the Position Loop. Thus, the only servo loop gain having any effect is Kp.

Even though the user may only be concerned with how well the axis positions, the Velocity Loop cannot be overlooked as it is the basis for positioning, because

$$\text{Distance} = \text{Velocity} * \text{Time}$$

The better an axis tracks velocity, the closer it will be to its commanded position.



The objective while adjusting  $K_p$ , is to minimize the velocity error, typically the error may be typically reduced to roughly 5-20 machine counts. The velocity error will not be eliminated completely,  $K_i$  will help to accomplish this also. Also, most users desire the axis to have a high degree of stiffness. As  $K_p$  is increased, observe that the motor shaft (or drive screw) becoming increasingly stiff, or resistant to external forces. The user may CAUTIOUSLY attempt to turn the shaft manually, and it will become increasingly harder to turn, even though it will not return to its original position (because the position loop gain (PGain) is zero).

No screeching or squealing should be heard from the motor when it is stationary. Noise indicates that  $K_p$  is set too high causing an oscillation. It may screech a little during the move, but not while at a rest.

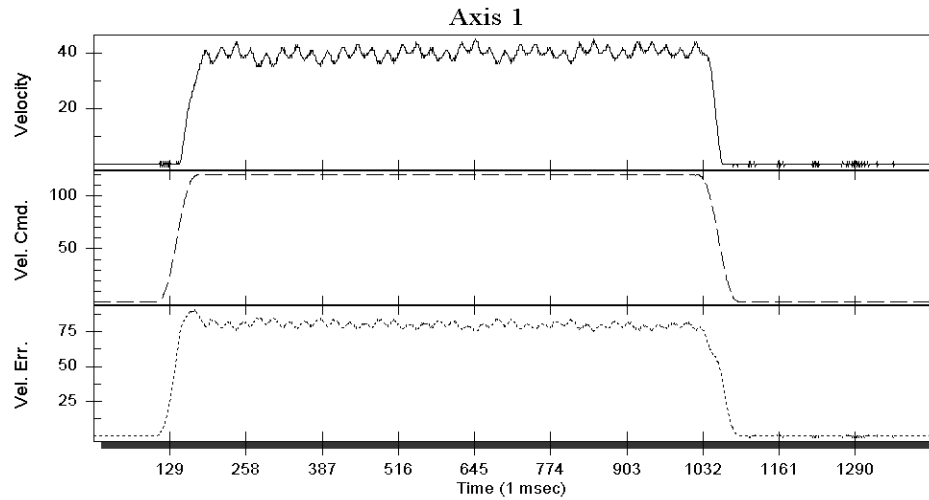
If the motor doesn't move,  $K_p$  is too low. Increase the value of  $K_p$  and try again by pressing the Single +/- buttons.



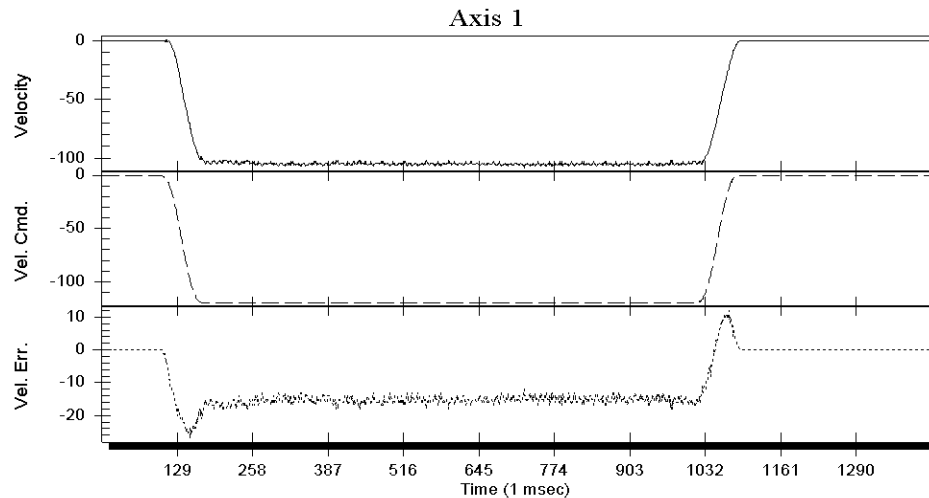
If the user is fine tuning the servo loop gains that Aerotech has provided for the system, use the existing  $K_p$  as the starting point.



Once the motor has begun cycling, a plot will display similar to Figure 4-17. From the graph, it can be seen that there are 75 to 85 machine counts of velocity error. " $K_p$ " should be increased to reduce the amount of velocity error. After repeating this process a few times, the velocity error will look similar to Figure 4-18. From this graph the user can observe that the average velocity error during the move is about 20 machine counts. Likewise, the axis does not oscillate when it is stationary.

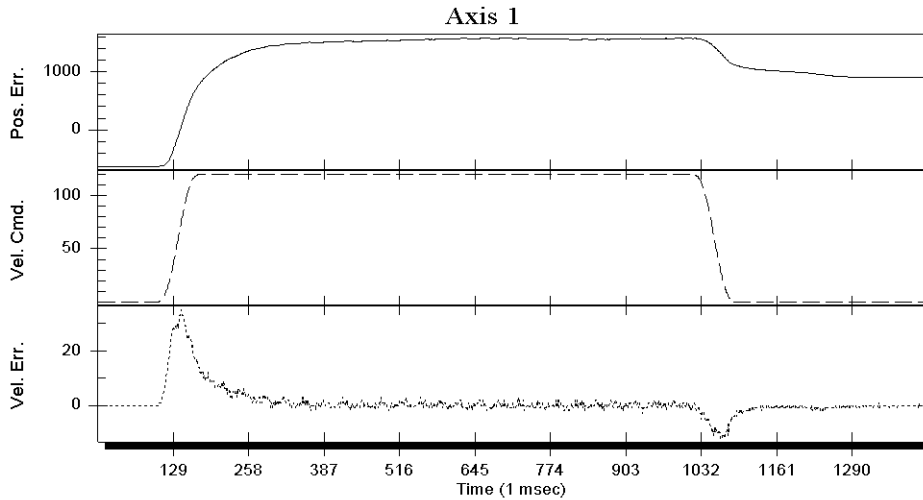


**Figure 4-17. Unacceptable Velocity Error**

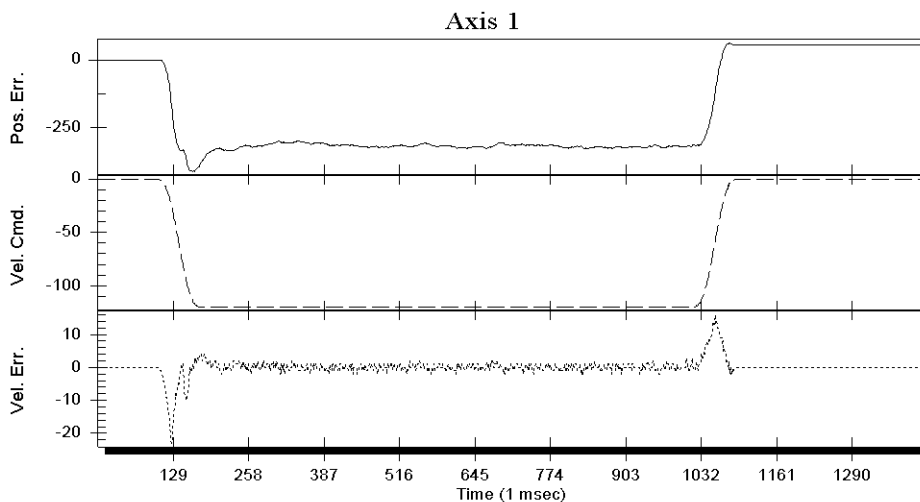


**Figure 4-18. Acceptable Velocity Error (While Adjusting Kp)**

Begin adjusting  $K_i$  with an initial value of 25. The main objective while increasing  $K_i$  is to reduce velocity error and position error. As  $K_i$  is increased, the error is reduced. However, the position error should not cross the “zero line”, indicating position overshoot, otherwise it will increase the settling time of the axis. At this point,  $K_i$  has reached its optimal value. Also, a very large  $K_i$  will introduce a low frequency oscillation in the position error. From the perspective of the load, this is an unwanted vibration, that may be unacceptable to the user. Refer to Figure 4-19.



**Figure 4-19. Unacceptable Position Error (While Adjusting Ki)**



**Figure 4-20. Acceptable Position Error (While Adjusting Ki)**

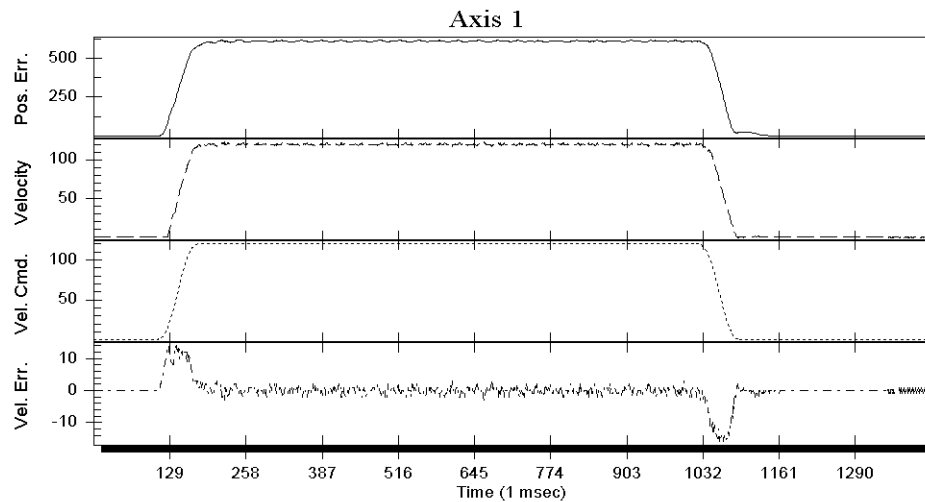
Depending on the application, the user does not need to be too concerned with vibration in the axis. However, when the position error crosses through zero, it indicates the velocity loop gain is too high, especially, if maintaining position is important to the user. This zero crossing (or position overshoot) causes the settling time to increase and the PGain within the position loop to be increased to keep the motor on its desired trajectory. The two servo loop gains are then in effect working against each other instead of complimenting each other.

Shown in Figure 4-20 is a graph with Ki adjusted more optimally. As can be observed from the position error, it is smoother and the position error does not cross through zero indicating positional overshoot. Likewise, the velocity error has been reduced.

- Adjust the servo loops position gain by varying the PGain. Use an initial value of 1-4 for PGain. As PGain is increased, it will be observed that the position error is reduced. The objective is to adjust PGain until the position error is within an acceptable range for the user or an oscillation occurs.

As previously mentioned, if PGain is too high, the user will encounter a high frequency oscillation (axis vibrates strongly). This will cause the UNIDEX 600 to generate an *IAvgLimit* fault, which indicates that the continuous current rating of the motor has been exceeded (the *IAvgLimit* fault is essentially a software fuse).

Shown in Figure 4-21 is a plot of an axis with PGain adjusted optimally. From the plot it can be seen that settling time is minimal. In other words, there is no damped oscillation at the end of the commanded move; so the axis is “in position” at the end of the commanded move. For comparison, Figure 4-22 illustrates a plot where PGain is too high.



**Figure 4-21. Plot Showing an Appropriate Value for PGain**



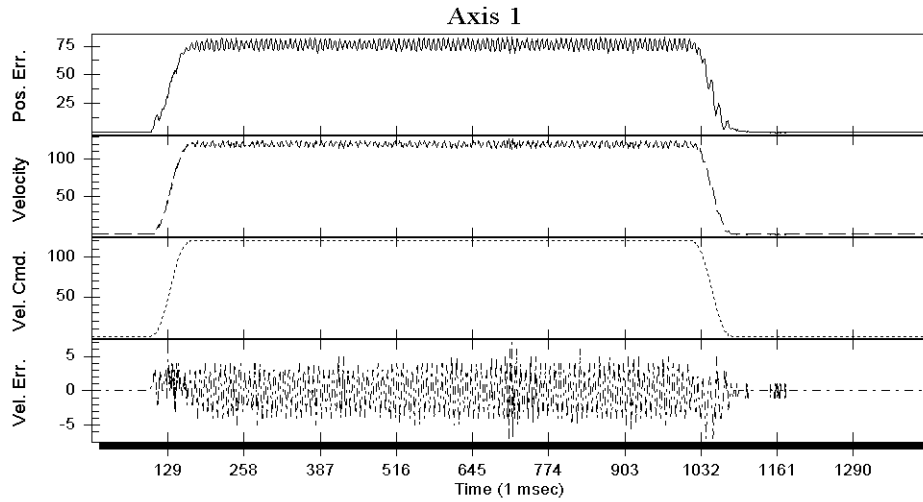


Figure 4-22. Plot Showing Overall Effects When PGain is High

6. Set  $V_{ff} = 1$  to enable velocity feedforward. Note, in Figure 4-23 that the position error has been minimized during the constant velocity portion of the move.

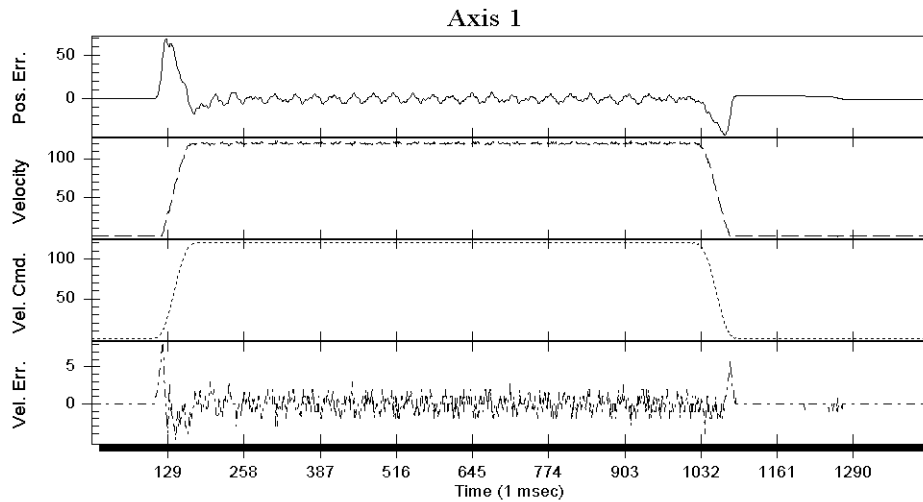


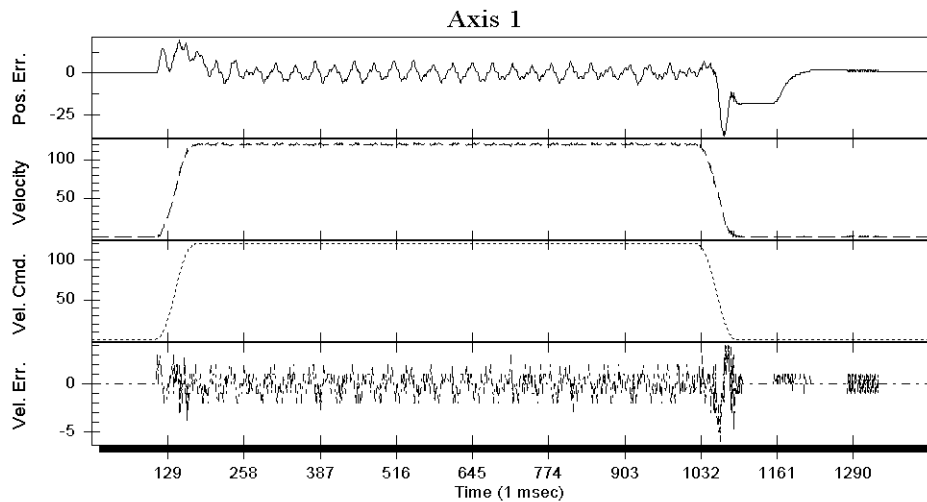
Figure 4-23. Plot Showing Velocity Feedforward Enabled ( $V_{ff}=1$ )

7. Minimize position error during acceleration and deceleration by increasing the AffGain servo loop parameter.

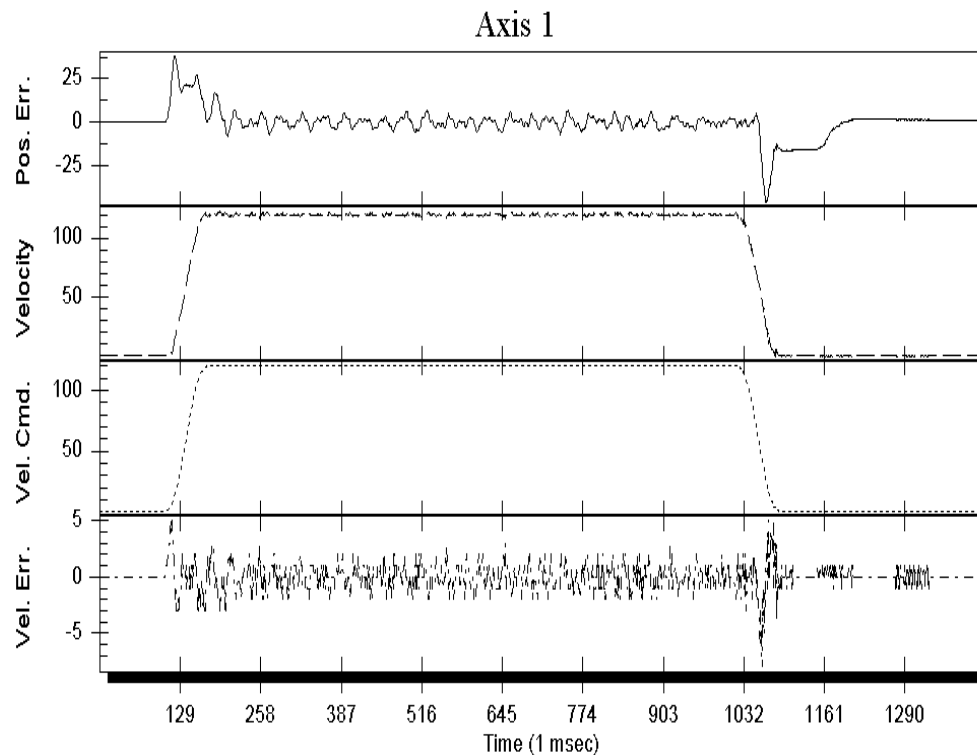
A normal value for AffGain is typically less than 200. The AffGain servo loop parameter will reduce the large position error visible during acceleration and deceleration. Increasing the AffGain will increase the noise produced by the axis due to the attempt of the servo loop to compensate for velocity changes. This noise may be reduced by varying the Alpha parameter. The Alpha parameter will filter the effect of the AffGain parameter minimizing the noise created by the velocity changes. The Alpha parameter is inversely scaled. No filtering is produced by setting the parameter to 65,536 and maximum filtering is produced by setting the parameter to 1. Figure 4-24 shows the optimal AffGain setting for our motor/drive/table combination. The Alpha servo loop parameter was not modified from its default setting of 65,536 providing no filtering of the AffGain servo loop parameter.



Adjusting the AffGain servo loop parameter is optional. The user's application may not require it.



**Figure 4-24. Plot Showing Optimal AffGain Adjustment**



**Figure 4-25. Plot Showing Final Performance of ATS3220140P X axis table, with a BM130 motor and an AS32030 amplifier.**

The final performance of the X axis of the ATS3220140P open frame table is shown in Figure 4-25. The distance of the move in the plot is 110,000 machine steps (110mm). The velocity of the move in the plot is 120,000 machine steps per second (120mm, or 4.7 inches per second). In summary, the points that should be noted include, the actual velocity (and position) of the axis following the commanded velocity (and position) within several machine counts during the constant velocity portion of the move, as well as during acceleration and deceleration of the axis, no ringing or position overshoot following the end of the move, and lastly, the time at which it takes for the axis to settle to the desired positional accuracy when the velocity command reaches zero at the end of the move. Should smoothness of motion be a performance criteria, it may be desired to have a more constant velocity (and position) error, indicating less fluctuation in the velocity of the axis providing for smoother motion of the axis. The servo loop gain parameters used are as follows:

Kp = 3000	Distance = 110,000
Ki = 250	Speed = 120,000
PGain = 3	AcclMode = 0
AffGain = 60	DecelMode = 0
Alpha = 0	AccelRate = 1,000,000

Vff = 1                      DecelRate = 1,000,000  
VGain = 0                    Accel = 80  
                                  Decel = 80

8. Turn the Position Error and Integral Error Traps on by returning to the Zsid960.exe utility. Turn the "Position Error" and the "Velocity Trap" back on by using the SET command as follows:

**SET FaultMask #####**

**(Note: ##### represents the number returned by the GET command in step 2 of this tuning procedure)**

This will re-enable these fault conditions.

#### 4.9. AC Brushless Motor Tuning Tip

Shown in Figure 4-26 is a tuning plot of an AC brushless motor. Note the ripple effect during the move. This is normal since AC brushless motors usually have a larger amount of torque ripple than DC brush motors. This torque ripple has been observed to be approximately 10 machine steps peak to peak with an unloaded AC brushless motor. While tuning an axis driving a brushless motor, torque ripple can be verified to be the cause of the disturbance by comparing one electrical cycle of the disturbance to one electrical cycle of the brushless motor. One electrical cycle of the brushless motor can be calculated in machine steps by dividing the number of machine counts per revolution of the motor by the number of electrical cycles of the motor. Aerotechs BM Series of brushless motor are all 8 pole motors, with 4 electrical cycles per revolution of the motor. A standard BM Series motor has a 1,000 line encoder mounted on the motor, which the controller does a times 4 multiplication on, producing 4,000 machine steps per revolution of the motor. Therefore, one electrical cycle of the motor is equal to:

$$4000 / 4 = 1000 \text{ machine steps per electrical cycle of a standard BM Series motor}$$

Aerotechs BLM Series of brushless linear motor all have an electrical cycle equal to 2.4 inches. Various linear encoders may be used with the BLM Series motors. However, one electrical cycle of the motor is equal to four times the number of encoder counts per 2.4 inches (the controller does a times 4 multiplication of the encoder signal).

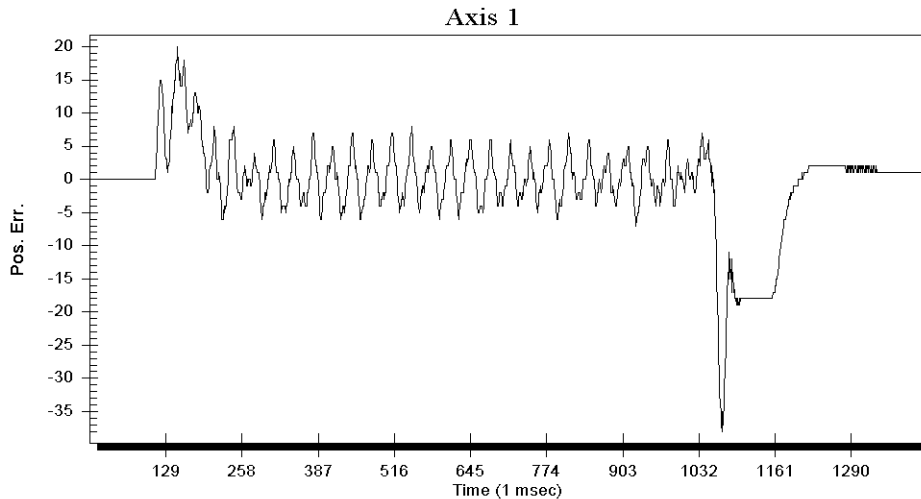


Figure 4-26. Torque Ripple Plot of an AC Brushless Motor

#### **4.10. Tuning With Tachometer Feedback**

The UNIDEX 600 Series controller is easily configured for controlling motors with external tachometers providing velocity feedback. To configure the controller for an external tachometer based Velocity Loop, the inherent digital Velocity Loop operation within the controller must be disabled. This is done by setting the digital servo loop proportional gain (Kp) and the integral gain (Ki) to zero. The servo system's Velocity Loop then must be adjusted by the potentiometers on the amplifier for the particular motor/tachometer/amplifier/load combination.

When configured this way, the analog outputs of the UNIDEX 600 which normally provide current commands to amplifiers will now deliver velocity commands to amplifiers accepting tachometer feedback.

In this configuration, the servo system has the following characteristics:

- The amplifier is configured to accept tachometer based velocity feedback.
- The amplifier regulates the Velocity Loop of the servo system. Velocity Loop regulation is accomplished by the Pre-amp section of the amplifier.
- The proportional (Kp) and integral (Ki) gain parameters in the UNIDEX 600 controller's servo loop have been set to zero (0), disabling its digital Velocity loop functionality.
- The controller is now commanding velocity to the amplifier instead of commanding torque.

##### **4.10.1. Vff - Velocity Feed Forward**

The Following Error (position error) that occurs while the axis is moving may be reduced significantly by setting the velocity feed forward gain to one. When the velocity feed-forward function is enabled (i.e., Vff = 1), an added voltage is summed with the velocity command to the amplifier. This signal is proportional to the Velocity Command.

##### **4.10.2. Vgain - Constant Velocity Gain**

The Following Error (position error) that occurs while the axis is moving at a constant velocity may be reduced by setting the VGain parameter to a non-zero value. This causes the velocity command to be increased proportionally by the commanded velocity which is scaled by the VGain servo loop parameter.

##### **4.10.3. Servo Parameter Setup for Tachometer Feedback**

When configuring a servo loop containing external velocity feedback from a tachometer, the servo gain values shown in Table 4-4 are used.

**Table 4-4. Servo Loop Axis Parameters for Tachometer based systems**

Parameter	Name	Value	Comments
Position Gain	PGain	Adjust per application	Should be maximized for servo stability and acceptable position error (following error) levels.
Integral Gain	Ki	Always zero	Unused.
Proportional Gain	Kp	Always zero	Unused.
Velocity Feed Forward	Vff	Optional	Minimizes following error (position error) of the servo system.
Acceleration Feed Forward	AffGain	Always Zero	Unused.
Acceleration Feed Forward Gain Filter	Alpha	Always Zero	Unused.
Constant Velocity Gain	VGain	Adjust per application	Should be maximized for acceptable position error (following error) levels during constant velocity.
Offset to Null Digital to Analog converter offset	DACOffset	Adjust per application	Should be set non-zero to null any offset in the velocity output command that will introduce a velocity offset into the system.

**4.10.3. The Servo Loop Parameters for Tachometer based Systems**

However, the servo loop parameters have slightly different meanings with tachometer feedback. The parameter definitions are as shown below.

**4.10.3.1. PGain - Position Gain**

The Position Gain is the only gain in the Position Loop in the UNIDEX 600's Servo Loop. This gain reduces the amount of position error and decreases the settling time. It is the first servo loop parameter to adjust.

**4.10.3.2. Vff - Velocity Feedforward Gain**

The Velocity Feedforward Gain is the only gain in the Velocity Feedforward Loop in the UNIDEX 600's Servo Loop. This gain reduces the amount of position error for systems with a tachometer. It is set to one after preliminary tuning is done.

**4.10.3.3. Kp - Proportional Gain**

Kp is the proportional gain used in systems with tachometers. It must be set to zero (0).

#### 4.10.3.4. Ki - Integral Gain

Ki is the integral gain used in systems with tachometers. It must be set to zero (0).

#### 4.10.3.5. AffGain - Acceleration Feedforward Gain

The Acceleration Feedforward Gain is the only gain in the Acceleration Feedforward Loop in the UNIDEX 600's Servo Loop. For systems with tachometers, it must be set to zero (0).

#### 4.10.3.6. VGain - Constant Velocity Gain

The VGain servo loop parameter is used to reduce the amount of position error during constant velocity mode.

### 4.11. Tuning Tachometer Loops

The following procedure is a guide for tuning motors with tachometers. Figure 4-27 shows the overall tuning process. The following procedure can be used as a guide when tuning the UNIDEX 600 servo loop. The tuning process discussed in this section was performed using the "X" (lower) Axis of an Aerotech ATS3220140P X-Y open frame table, with a 1035DC brush motor and an DS16020 amplifier at 40VDC. The user's system may behave differently and have different values for servo loop gains. However, the overall process is the same and the same process can be repeated for other axes. When adjusting each of the servo loop gains, the user will essentially be following the procedure below:

1. Exercise the axis through a move profile typical to your application.
2. Observe the servo loop performance with the Plot utility.
3. Make a decision on whether to increase or decrease the value of the servo loop gain parameter or proceed to the next servo loop parameter.
4. Repeat.



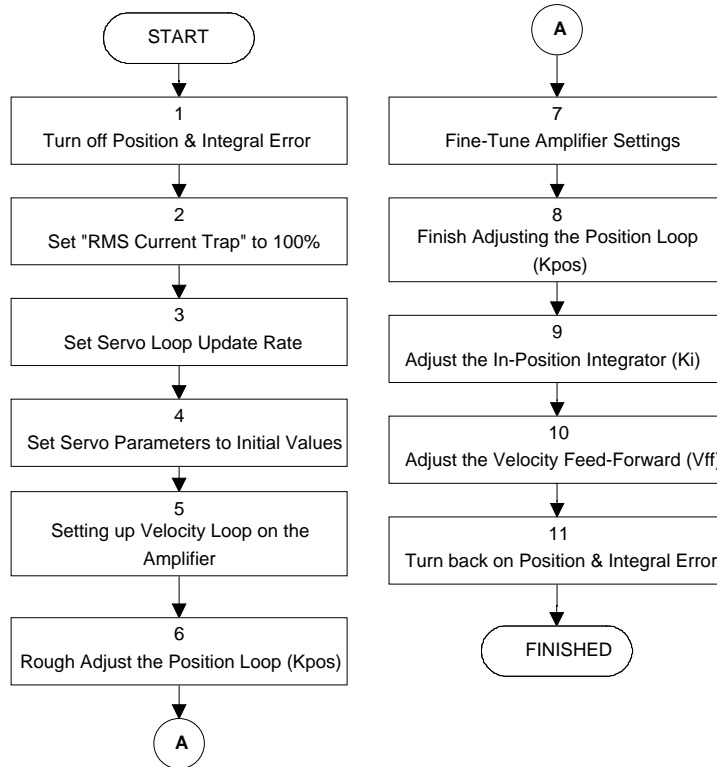


Figure 4-27. Flowchart of Overall Tachometer Tuning Process

The following is a step-by-step procedure for tuning motors with tachometers.

Please read each step thoroughly before performing the task.



1. Turn off the “Position Error” and “Velocity Error” bits in the FaultMask axis parameter. Start the Zsid960 utility. Download the axis firmware if this has not been done. Select the axis that you wish to tune, with the ‘AXIS’ command. Record the current value of the FaultMask axis parameter by using the GET command as follows:

**GET FaultMask**

A value will be displayed after entering the preceding command. Record this value to restore it after tuning the axis.

To disable these faults set the FaultMask axis parameter to 8398 as follows:

**SET FaultMask 8398**

2. Set the IAvgLimit parameter to 32,767 (100%)
3. Set servo loop parameters to the initial values indicated in the chart below.

**Table 4-5. Initial Servo Parameter Values - Tachometer Tuning**

VGain	PGain	Ki	Kp	Vff	DACOffset	Alpha	AffGain
0	0	Always 0	Always 0	0	0	Always 0	Always 0

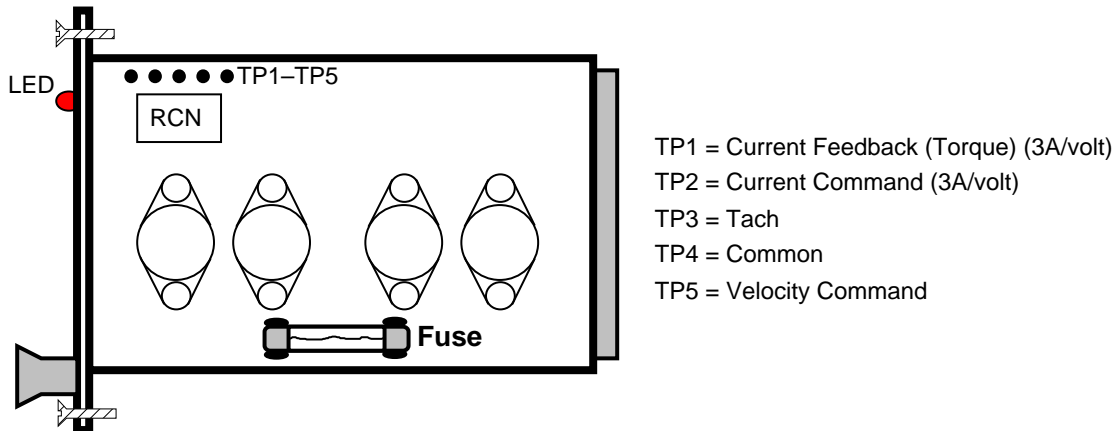
4. Adjust the Velocity Loop on the amplifier.



If the user has a non-Aerotech amplifier, the manufacturer should provide information for configuring the amplifier to accept a Velocity Command and explain how to optimize the Velocity Loop.

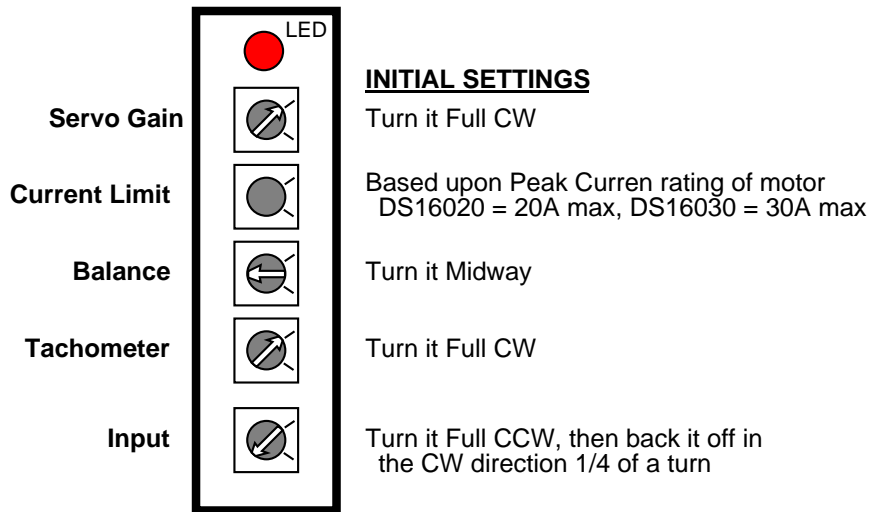
If the user has an Aerotech DS16020/DS16030 servo amplifier, the Velocity Loop is adjusted the following way:

- a. Select a fuse to protect the motor for the continuous current rating of the motor and insert it in the appropriate fuse holder of the amplifier. Refer to Figure 4-28 for the location of the fuse holder.



**Figure 4-28. Cross-Section of the DS16020/16030 Amplifier**

- b. Make rough adjustments to the potentiometers on the Aerotech DS16020/16030 servo amplifier as shown in Figure 4-29, then run the axis at its maximum speed. This maximum speed will be provided to you by Aerotech if you have purchased a complete system from Aerotech, otherwise, you will have to calculate the maximum speed yourself. While running the axis at maximum speed, adjust the input potentiometer on the amplifier such that the torque plot in Plot (torque actually indicates velocity when in the velocity command mode) indicates approximately 26,200 (8 volts)



**Figure 4-29. Amplifier Potentiometer Layout**

The initial setting of the Current Limit potentiometer is based upon the peak current rating of the motor. If the user has a motor with a 10A peak current rating and a DS16020 which has a maximum current output of 20A, set the Current Limit potentiometer to midway for a representation of 10A. Then back it off 1/8 turn in the CW direction. Full CW sets the minimum current and full CCW the maximum..



- c. Adjust the Servo Gain (AC gain) potentiometer on the amplifier by first enabling the axis and then turning the potentiometer CCW until the motor oscillates (i.e., the axis vibrates). The motor will produce a screeching sound when it oscillates. Back the gain off by turning it CW until the oscillation stops. Make another 1/8 turn CW from that position so its not on the borderline of having the motor oscillate.
5. Prepare the Plot utility for tuning by performing the following functions:
  - a. Press the maximize button on the Plot Window so that its window fills the entire screen.
  - b. Use the 'Axis' menu select the 'Select Axis' menu selection to select the axis to be tuned.
  - c. Use the 'Plot' menu, to select only Velocity Command, Velocity Error, and Position Error.
  - d. Select the 'Axis' menu to select the "Set speed/distance' menu selection to set the Distance and Speed entry fields (in steps & steps/second) for a typical move profile, define the desired AccelMode and DecelModes and enter the appropriate Accel/Decel Rate/Times into the dialog boxes, found in the 'Tune' box that may be selected from the 'Axis' menu.

When the “Single +/-” button’s are selected, the Axis will move the specified increment in the direction determined by the Single +/- button selected.

6. Adjust the PGain servo loop parameter such that the position error is at zero at or near the same time the Velocity Command is at zero. The adjustment to PGain is made by entering a value in the PGain box. If PGain is set too high, the position error will oscillate and the motor will vibrate. The user is not striving to reduce the position error, though that will happen. However the axis needs to be rough tuned because the following step will be to fine tune the potentiometers on the amplifier.



Since all Servo Gains are set to zero, the user must set the PGain to an initial value, otherwise the axis won’t move. A typical initial starting value is 5.

Once the axis begins moving, Pgain should be increased until the actual velocity tracks the commanded velocity, with minimal position overshoot.



If the motor doesn’t move then PGain is too low. Increase the value of PGain and try again by pressing the Single+/- buttons .

The axis may tend to drift away on its own when it is enabled. Adjusting the DACOffset axis parameter will null the offset causing the drift.



If you are fine tuning the servo loop gains that Aerotech has setup for your system, use the existing PGain as your starting point.

As PGain is increased, the position error will begin to be at zero at or near the end of the commanded move. The axis is now roughly tuned, so continue with the following step.

7. This step requires fine tuning the amplifier settings. First, adjust the Balance pot on the amplifier in order to remove any DC offset in the position error. Press the Auto button to cycle the axis. While the axis is moving, adjust the Balance pot and remove any DC offset in the position error. Press the Halt button when the task is done. Ideally the position error will be symmetrical in the positive and the negative directions, however, it is most important that the position error is at zero when the move is complete.

Second, the user will fine tune the Current Limit pot on the Aerotech DS16020/16030 amplifier after commanding the motor to move short, fast moves and observing the current feedback from TP1 on the amplifier with an oscilloscope. In order to do this, perform the following steps.

- a. Connect the Oscilloscope leads to TP1 (current feedback) and TP4 (common) on the amplifier.

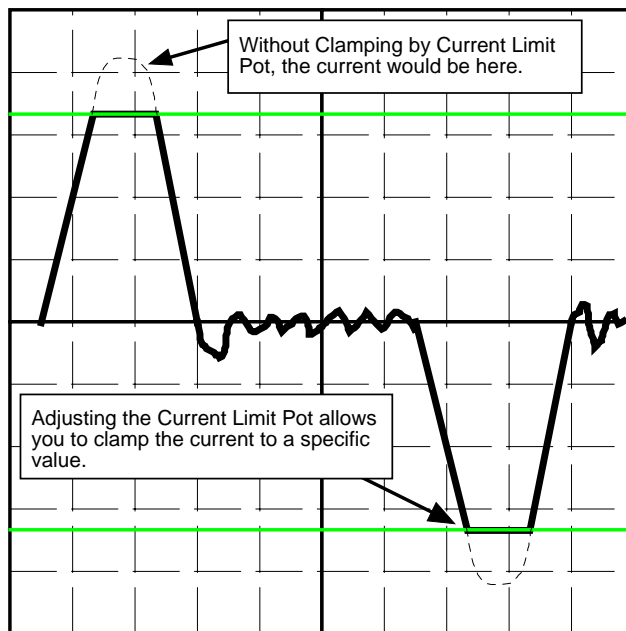
- b. Set the Distance and Speed entry fields to represent a typical short fast move. The idea here is to command the motor to move faster than it is capable of, so that the amplifier will saturate. This implies that the amplifier will be full on, that is commanding maximum current. This will allow you to adjust the current limit potentiometer which limits the maximum current that the amplifier can produce.
- c. Press the Auto button and allow the axis to cycle.

If the user cycles the axis too long at short fast speeds, the fuse will blow, so be prepared to make the adjustment fairly quickly. The user may desire to use the single buttons so there is a delay between moves preventing the fuse from blowing quickly.



- d. While the axis is moving, adjust the Current Limit pot to limit the current to either 4 times the continuous current rating of the motor or the peak current rating of the motor, whichever is less.

The current feedback on TP1 is 3 amps per volt, so a 2 volt signal on the Oscilloscope would represent 6 amps. Press the Halt button when complete. Figure 4-30 illustrates what the user will see after one move.



**Figure 4-30. Oscilloscope Showing Current Feedback for One Move**

Third, if necessary, the user may have to fine tune the Input pot if unable to achieve maximum speed for the motor. To fine tune the Input pot perform the following procedure.

- Connect the Oscilloscope to TP5 (velocity command) and TP4 (common) on the amplifier.
  - Set the Distance and Speed entry fields to represent a typical move at 1/2 of the maximum speed.
  - Press the Auto button and allow the axis to cycle.
  - While the axis is moving, adjust the Input pot so that when the motor is moving at 1/2 speed the Velocity Command on TP 5 is 4 volts.
  - Press the Halt button when completed.
8. Finish, by adjusting the Position Loop (PGain) where the main concern is to strive for a smooth variation in the position error and to have the position error reach zero at or near the same time the Velocity Command ends. After repeating the process of starting and stopping the axis and adjusting PGain the axis velocity should track the commanded velocity fairly well and the position error should be at zero at the end of the commanded move. Increasing the Pgain will lower the axis settling time.



If the PGain parameter is too high, the motor will oscillate.

9. Enable the Velocity feedforward parameter to reduce the position error (if desired). It is OK to allow some following error in the system, however, if multiple axes will be used for simultaneous contoured moves it is desirable that each axis will have the same following error.
10. Increase the VGain servo loop parameter to minimize the position error during the constant velocity portion of the move. Typically, VGain will be set less than 500.
11. Turn the Position Error and Integral Error Traps on by returning to the Zsid960.exe utility and using the SET command as follows:

**SET FaultMask #####**

**(Note: ##### represents the number returned by the GET command in step 2 of this tuning procedure)**

- . This will re-enable these fault conditions.

▽ ▽ ▽

**APPENDIX A: PARAMETER SUMMARY****In This Section:**

- Summarized List of Parameters

This appendix provides a summary of the parameters applicable to the UNIDEX 30, UNIDEX 31, and UNIDEX 600 Motion Controllers. Note that in the “Access” column the R/W represents a Read/Write parameter while a stand-alone R represents a Read Only parameter.

Parameter	Access	Min. Value	Max. Value	Default Value
ABORTMASK	R/W	0	0	-
ACCEL	R/W	0	100,000	0
ACCELMODE	R/W	0	3	0
ACCELRATE	R/W	1	10,000,000	100,000
AFFGAIN	R/W	-10,000	10,000	0
ALT_STATUS	R	0	0	-
AUX	R/W	0	1	0
AUXMASK	R/W	0	0	-
AUXOFFSET	R/W	0	0	-
AVGVEL	R	0	0	-
AVGVELTIME	R/W	1	1,000	1,000
BASE_SPEED	R/W	0	10,000,000	-
BLOCKMOTION	R/W	0	1	0
CAMOFFSET	R/W	0x80000001	0x7fffffff	0
CAMPOINT	R	0	0	-
CAMPOSITION	R	0	0	-
CCWEOT	R/W	0x80000001	0x7fffffff	-2147483647
CLOCK	R/W	0	0	-
CWEOT	R/W	0x80000001	0x7fffffff	2147483647
DECEL	R/W	0	100,000	0
DECELMODE	R/W	0	3	0
DECELRATE	R/W	1	10,000,000	100,000
DISABLEMASK	R/W	0	0	-

Parameter	Access	Min. Value	Max. Value	Default Value
DRIVE	R/W	0	1	0
ECHO	R/W	0	0	-
FAULT	R/W	0	0	-
FAULTMASK	R/W	0	0	-
FBWINDOW	R/W	0	0	-
FEEDRATEMODE	R/W	0	1	0
HALTMASK	R/W	0	0	-
HOMESWITCHPOS	R	0	0	-
HOMESWICHTOL	R/W	0	16,384	0
HOMEVELMULT	R/W	0	1,000	100
I AVG	R	0	0	-
I AVGLIMIT	R/W	0	0x7fff	32,767
I AVGTIME	R/W	0	4,000	1,000
ICMD	R	0	0	-
IMAX	R/W	0	0x7fff	32,767
INPOSLIMIT	R/W	0	65,536	65
INTMASK	R/W	0	0	-
IOLEVEL	R/W	0	0x3f	63
IVEL	R	0	0	-
KI	R/W	0	5,000,000	2,000
KP	R/W	0	10,000,000	10,000
MASTERLEN	R/W	0	0x7fffffff	0
MASTERPOS	R/W	0x80000001	0x7fffffff	-
MASTERABS	R	0x80000001	0x7fffffff	-
MASTERRES	R	0	0	-
MAX_PHASE	R/W	0	360	-
MOTIONSTATUS	R	0	0	-
MOVEQDEPTH	R	0	0	-
MOVEQSIZE	R	0	0	-
PGAIN	R/W	0	1,000	10
PHASE_SPEED	R/W	1	10,000,000	-
POS	R/W	0x80000001	0x7fffffff	-
POSCMD	R	0	0	-



Parameter	Access	Min. Value	Max. Value	Default Value
POSERR	R	0	0	-
POSERRLIMIT	R/W	0	10,000,000	65,536
POSTARGET	R	0	0	-
POSTOGO	R	0	0	-
PROFQDEPTH	R	0	0	-
PROFQSIZE	R	0	0	-
RAWPOS	R	0	0	-
RESOLVER	R	0	10,000,000	-
REVERSALMODE	R/W	0	1,000	-
REVERSALVALUE	R	0	1,000	0
SAFEZONECCW	R/W	0x80000001	0x7fffffff	0
SAFEZONECW	R/W	0x80000001	0x7fffffff	8
SAFEZONEMODE	R/W	0x0	0x2	0
SERVOSTATUS	R	0	0	-
SIMULATION	R/W	0x0	0x1	0
SOFTLIMITMODE	R/W	0	1	0
STATUS	R	0	0	-
SYNCSPEED	R/W	1	0x7fffffff	1,000
VELCMDTRAP	R/W	0	65,536,000	0
VELPOSITION	R	0	0	0
VELTRAP	R/W	0	65,536,000	0
VEL_SHIFT	R/W	-16	16	-
VFF	R/W	0	1	0

▽ ▽ ▽



**APPENDIX B: WARRANTY AND FIELD SERVICE****In This Section:**

- Laser Product Warranty
- Return Products Procedure
- Returned Product Warranty Determination
- Returned Product Non-warranty Determination
- Rush Service
- On-site Warranty Repair
- On-site Non-warranty Repair

Aerotech, Inc. warrants its products to be free from defects caused by faulty materials or poor workmanship for a minimum period of one year from date of shipment from Aerotech. Aerotech's liability is limited to replacing, repairing or issuing credit, at its option, for any products which are returned by the original purchaser during the warranty period. Aerotech makes no warranty that its products are fit for the use or purpose to which they may be put by the buyer, where or not such use or purpose has been disclosed to Aerotech in specifications or drawings previously or subsequently provided, or whether or not Aerotech's products are specifically designed and/or manufactured for buyer's use or purpose. Aerotech's liability or any claim for loss or damage arising out of the sale, resale or use of any of its products shall in no event exceed the selling price of the unit.

Aerotech, Inc. warrants its laser products to the original purchaser for a minimum period of one year from date of shipment. This warranty covers defects in workmanship and material and is voided for all laser power supplies, plasma tubes and laser systems subject to electrical or physical abuse, tampering (such as opening the housing or removal of the serial tag) or improper operation as determined by Aerotech. This warranty is also voided for failure to comply with Aerotech's return procedures.

***Laser Products***

Claims for shipment damage (evident or concealed) must be filed with the carrier by the buyer. Aerotech must be notified within (30) days of shipment of incorrect materials. No product may be returned, whether in warranty or out of warranty, without first obtaining approval from Aerotech. No credit will be given nor repairs made for products returned without such approval. Any returned product(s) must be accompanied by a return authorization number. The return authorization number may be obtained by calling an Aerotech service center. Products must be returned, prepaid, to an Aerotech service center (no C.O.D. or Collect Freight accepted). The status of any product returned later than (30) days after the issuance of a return authorization number will be subject to review.

***Return Procedure***

After Aerotech's examination, warranty or out-of-warranty status will be determined. If upon Aerotech's examination a warranted defect exists, then the product(s) will be repaired at no charge and shipped, prepaid, back to the buyer. If the buyer desires an air freight return, the product(s) will be shipped collect. Warranty repairs do not extend the original warranty period.

***Returned Product  
Warranty Determination***

**Returned Product Non-warranty Determination**

After Aerotech’s examination, the buyer shall be notified of the repair cost. At such time the buyer must issue a valid purchase order to cover the cost of the repair and freight, or authorize the product(s) to be shipped back as is, at the buyer’s expense. Failure to obtain a purchase order number or approval within (30) days of notification will result in the product(s) being returned as is, at the buyer’s expense. Repair work is warranted for (90) days from date of shipment. Replacement components are warranted for one year from date of shipment.

**Rush Service**

At times, the buyer may desire to expedite a repair. Regardless of warranty or out-of-warranty status, the buyer must issue a valid purchase order to cover the added rush service cost. Rush service is subject to Aerotech’s approval.

**On-site Warranty Repair**

If an Aerotech product cannot be made functional by telephone assistance or by sending and having the customer install replacement parts, and cannot be returned to the Aerotech service center for repair, and if Aerotech determines the problem could be warranty-related, then the following policy applies:

Aerotech will provide an on-site field service representative in a reasonable amount of time, provided that the customer issues a valid purchase order to Aerotech covering all transportation and subsistence costs. For warranty field repairs, the customer will not be charged for the cost of labor and material. If service is rendered at times other than normal work periods, then special service rates apply.

If during the on-site repair it is determined the problem is not warranty related, then the terms and conditions stated in the following "On-Site Non-Warranty Repair" section apply.

**On-site Non-warranty Repair**

If any Aerotech product cannot be made functional by telephone assistance or purchased replacement parts, and cannot be returned to the Aerotech service center for repair, then the following field service policy applies:

Aerotech will provide an on-site field service representative in a reasonable amount of time, provided that the customer issues a valid purchase order to Aerotech covering all transportation and subsistence costs and the prevailing labor cost, including travel time, necessary to complete the repair.

**Company Address**

Aerotech, Inc.  
101 Zeta Drive  
Pittsburgh, PA 15238-2897  
USA

Phone: (412) 963-7470  
Fax: (412) 963-7459



**APPENDIX C: GLOSSARY OF TERMS****In This Section:**

- Terms Used In This Manual
- Definitions

This appendix contains definitions of terms that are used throughout this manual.

**absolute positioning** - Absolute positioning is positioning that is done with respect to an initial starting position (typically referred to as the home position) and typically uses a standard coordinate system (using [X,Y] coordinates is an example of absolute positioning). In contrast, incremental (or relative) positioning is done using a series of relative moves. These moves are relative to the previous location rather than a single reference point (for example, relative changes in position [ $\Delta X, \Delta Y$ ] are examples of incremental positioning).

**acceleration feed forward** - Acceleration feed forward is a control strategy (represented as a dimensionless gain value that is sometimes used during the motor tuning process) in which current acceleration disturbances are converted into corrective actions now in order to minimize the future effects of the disturbances.

**accuracy** - Accuracy is the difference between an expected value and an actual value expressed as a percentage.

**axis** - An axis is a direction along which movement occurs.

**axis calibration** - Axis calibration is the process by which the current position of an axis is adjusted to match the actual position (as determined by a laser for example) of the axis.

**backlash** - Backlash is a movement that occurs between two or more interacting mechanical parts as a result of looseness.

**base address** - A base address is a number that represents the memory location in the computer where input/output (I/O) information can be stored. All devices within a computer must have unique I/O base addresses.

**brushless motor** - Aerotech brushless motors are three-phase, rare earth permanent magnet servo motors which generate a sinusoidal back EMF voltage and are usually referred to as AC brushless motors. Another type, usually referred to as the DC brushless motors, generate a trapezoidal back EMF and produce more torque ripple.

**C** - C is a high-level programming language (developed at Bell laboratories) that is able to manipulate a computer at a low level like assembly language. If a customized interface program is created using C, the appropriate C library must be included when the program is compiled.

**cam motion** - Digital cam motion refers to the electronic emulation of a mechanical cam. Digital cam motion is accomplished using a cam table - a list of positions that represent the profile of the cam. Typically, an axis will move through the cam table by indexing to the next position in a specified time interval.

**commutation** - Commutation refers to the process by which every other cycle of an alternating current is reversed so that a single unidirectional current is supplied. In the case of motors, commutation refers to the switching of current to motor windings which causes the motor to rotate. In a DC servo motor, this is done mechanically using brushes and a commutator. A brushless motor is electronically commutated using a position feedback device such as an encoder that is mounted to the rotor. Stepping motors are electronically commutated without feedback (in an open loop fashion).

**DOS** - DOS is an acronym for Disk Operating System--a master control program that runs a computer and acts as a scheduler, providing job, task, data and device management. DOS is a generic term that refers to an operating system. MS-DOS is a single-user operating system for PCs that was designed by Microsoft Corporation. Since the DOS and MS-DOS programs are very, the terms DOS and MS-DOS are frequently interchanged.

**encoder** - An encoder is a rotary device that transmits a pulsed signal based on the number of revolutions of the device.

**faults** - A fault is an error condition that occurs when a component of the UNIDEX 30/31/600 system operates outside certain parameters. Fault masks are used to allow the system to detect and act on any fault condition of the system. Examples of major fault conditions include position faults, velocity faults, integral faults, RMS over current faults and feedback faults.

**Hall effect switch** - A Hall effect switch is a solid state switch that is activated by a magnetic field. Some AC brushless motors use Hall effect switches.

**hexadecimal number format** - Hexadecimal number format is a method of representing large numbers using base 16 rather than the standard base 10. In base 16 or hexadecimal number format (often abbreviated "hex"), the number positions represent powers of 16 (rather than powers of 10 in decimal). The decimal number positions (1's, 10's, 100's, 1,000's, 10,000's, etc.) are replaced with hexadecimal number positions (1's, 16's, 256's, 4096's, etc.). Also, while the individual numerals for the decimal system are 0-9, the numerals for the hexadecimal number system (which requires 16 unique "numerals") are 0-9 then A-F (where  $A_{16}=10_{10}$ ,  $B_{16}=11_{10}$ ,  $C_{16}=12_{10}$ ,  $D_{16}=13_{10}$ ,  $E_{16}=14_{10}$ , and  $F_{16}=15_{10}$ ). For simplicity in this manual, hexadecimal numbers are written with a preceding "0x" rather than using the subscript 16. For example, the hexadecimal number 12A5 is written 0x12A5. Numbers without the preceding "0x" are assumed to be decimal unless otherwise indicated.

**home cycle** - The home cycle is series of motions that are used to move the specified axes to a hardware referenced position. There are two feedrates (in the form of parameters) associated with the home cycle: the Power On Feedrate and the Normal Home Feedrate. The power on home cycle (the first commanded home cycle following a power up) uses the Power On Feedrate parameter and the normal home cycle (all subsequent home cycles) uses the Normal Home Feedrate parameter.

**home marker option** - The home marker option is a type of encoder that can be used with stepper motors. This option provides an inexpensive way of establishing a very accurate home reference (usually within 0.1 microns, in most Aerotech equipment). The home marker is protected in a rugged housing that also provides terminal connections for the encoder, the motor and the limit switch.

**incremental positioning** - Incremental (or relative) positioning is done using a series of relative moves. These moves are relative to the previous location rather than a single reference point (for example, relative changes in position  $[\Delta X, \Delta Y]$  are examples of incremental positioning). In contrast, absolute positioning is positioning that is done with respect to an initial starting position (typically referred to as the home position) and typically uses a standard coordinate system (using  $[X, Y]$  coordinates is an example of absolute positioning).

**Inductosyn** - An Inductosyn is a rugged, very accurate, multi-pole electromagnetic transducer with an operating principle similar to that of a resolver.

**integral gain** - Integral gain is a dimensionless motor tuning parameter that serves to help remove steady-state position errors as well as reduce the effects of tachometer loop drift.

**jog move** - A jog move is a momentary movement of an servo drive to provide manual control of axis motion.

**joystick** - A joystick is manual input control device that digitizes a path using two axes. A joystick offers direct motion control for easy machine setup and testing.

**microstepping** - Microstepping is a technique for driving stepping motors more smoothly and with higher resolution than full step control. Current is divided in a sine-cosine fashion between motor phases to provide intermediate positions between full step positions.

**multitasking** - Multitasking is software technique that gives several functions (or tasks) the appearance of individually having sole access to the resources of the system (for example, the microprocessor). In its simplest form, a multitasking system assigns a small time slice to each task in a round robin fashion. Only one task at a time has access to the multitasking system's resources. When each successive task has had the opportunity to use the system resources (for a brief period), the cycle repeats.

**plane** - A plane is an axis or group of axes that can be coordinated (for example, a particular action of one plane can trigger an action on another plane) or independent (for example, one plane can be milling a part while another plane is etching circles). Planes can also be virtual panes, which are not linked to any particular axis, but act as queues or buffers.

**program** - A program is a set of instructions that are carried out in some predefined logical order. A UNIDEX 30, UNIDEX 31 or UNIDEX 600 program is a sequential list of UNIDEX 31 programming commands which tell the control board how to perform specific motions for a particular application. These programs may be created/edited on-line (from within the ???) or off-line (using any standard ASCII text editor). U30/U31/U600 program files use ??? as their extension.

**proportional gain** - Integral gain is a dimensionless motor tuning parameter that produces an output which is the summation of the velocity errors, thereby producing an increasing, counteracting force for a constant or increasing position error.

**quadrature** - Quadrature is the state of two signals that are displaced 90 degrees with respect to each other. In most rotary incremental optical encoders, light (from an LED, for example) is measured after it is passed through slits in a grating disk (which is attached to the axis being measured). Typically, two tracks on the disk have their gratings displaced 90 degrees with respect to each other (that is, the tracks are said to be in quadrature).

**resolver** - A resolver is a two-phase, rotary, electromagnetic transducer in which inductive coupling (between the rotor and stator windings) and trigonometric principles are employed to provide absolute position information over one electrical cycle (which is one revolution for "single-step" resolvers)

**resolver-to-digital card (R/D)** - The R/D card is an optional PC-based R/D card that is used to receive resolver or Inductosyn feedback. Resolution is selectable among 10-bit, 12-bit, 14-bit or 16-bit.

**servo control system** - A servo control system (servo loop) is a motion control system which continuously compares desired position/velocity to actual position/velocity and produces an error correction command. Servo systems use sensors to feedback actual position/velocity.

**software** - The term software refers to a computer program. Contrast software with hardware, the physical machinery, components and support peripherals through which the software runs.

**traps** -*See* faults.

**tuning** - Tuning is the process of optimizing the operation of a servo system.

**velocity feed forward** - Velocity feed forward is a control strategy (represented as a dimensionless gain value that is sometimes used during the motor tuning process) in which current velocity disturbances are converted into corrective actions now in order to minimize the future effects of the disturbances.

**velocity profiled motion** - Velocity profiled motion is a move of a programmed distance and speed from the current position. Velocity profiled motions are executed only after the previous motion has reached its deceleration point.

▽ ▽ ▽



## INDEX

Bold locator type faces in this index indicate that the index reference is part of a figure (e.g., Abort Button F12, **2-22**). Italic locator type faces indicate that the index reference is part of a table (e.g., Command Summary, *A-1*).

## Symbols

16 bit ISA Expansion Card, 1-1  
32-bit Intel i960 RISC Processor, 1-2  
80960 RISC processor, 1-1

### A

ABORT Command, 2-13  
Abort Motion Abruptly, 2-22  
abort queued bit, 3-18  
ABORTMASK, 2-22  
ABORTMASK Parameter, 2-24  
About, 3-36  
About Dialog Box, **3-36**  
Absolute Target Position, 2-13  
AC mode bit, 3-14  
accel enable bit, 3-18  
accel mode bit, 3-19  
ACCEL Parameter, 2-25  
accel phase bit, 3-7, 3-16  
accel sign bit, 3-19  
Acceleration feedforward gain - Aff, 4-31  
ACCELMODE Parameter, 2-25  
ACCELERATE Parameter, 2-25  
Acknowledging Faults, 2-23  
Acquire 1 Set of Data Option, 4-7, 4-8  
Acquire Data Continuously Option, 4-7, 4-8  
Actual Position, 4-5  
Actual Velocity, 4-5  
Aerotech Aux Mask Display, **3-31**  
Aerotech Axis Status Display, **3-4**  
Aerotech Disable Mask Display, **3-28**  
Aerotech Fault Mask Display, **3-27**  
Aerotech Fault Status Display, **3-22**  
Aerotech Halt Mask Display, **3-30**  
Aerotech Interrupt Mask Display, **3-29**  
Aerotech Motion Status Display, **3-16**  
Aerotech Servo Status Display, **3-11**  
Aff, 4-32  
Aff Acceleration feedforward gain, 4-18  
AFF enable bit, 3-13  
AFFGAIN Parameter, 2-25  
alive bit, 3-13  
ALT\_STATUS Parameter, 2-25  
Analog and Digital I/O, 2-14

Matrix Analog Inputs, 2-16  
Reading Digital Inputs, 2-15  
Writing Digital Outputs, 2-15  
Xycom Digital I/O, 2-14  
at home bit, 3-5, 3-12  
Auto Mode, 4-7  
Aux Mask Display, 3-31  
aux mode bit, 3-10, 3-17  
aux output enable bit, 3-4, 3-11  
AUX Parameter, 2-25  
Auxiliary Output Status of an Axis, 3-4  
AUXMASK, 2-22  
AUXMASK Parameter, 2-25  
AUXOFFSET Parameter, 2-26  
AVGVEL Parameter, 2-26  
AVGVELTIME Parameter, 2-26  
Axes Setup, 2-7  
Axis Configuration, 2-2  
Axis Configuration Commands, 2-2  
Axis Control Information, 3-1  
Axis Deceleration, 2-22  
Axis Drive Disabled, 2-11  
Axis Enable Control Boxes, 4-14  
Axis Exercising, 2-1  
Axis Feedback Type, 2-7  
Axis Feedback Verification, 2-2  
Axis Firmware Level, 2-3  
Axis Homing Routine, 2-12  
Axis Parameter Status, 2-3  
Axis Position, 2-2  
Axis Processor Status, 3-1  
Axis Pull-down Menu, 4-13  
Axis Resolution Verification, 2-2  
Axis Scope window, 4-18, 4-32  
Axis Selection, 2-1  
Axis Selection Screen, **4-16**  
Axis Servo Performance, 4-1  
Axis Status, 2-3  
Axis Status Display, 3-3  
Axis Tuning Dialog Box, **4-14**  
Axis Tuning for Servo Operation, 2-2

### B

Balance potentiometer, 4-36  
BASE\_SPEED Parameter, 2-26, 2-32  
Basic Commands, 2-2  
block motion bit, 3-10, 3-17  
BLOCKMOTION Parameter, 2-27  
*bound argument*, 2-8  
bounded bit, 3-20  
Brake Engagement, 2-22  
Brushless Motor Commutation, 2-48

### C

C:\U31\UTILITY Sub-directory, 2-1

- cam table enable bit, 3-9, 3-17
  - CAMOFFSET Parameter, 2-27
  - CAMPOINT Parameter, 2-27
  - CAMPOSITION Parameter, 2-27
  - CCW Hard Limit Fault, 2-18, 2-20
  - CCW Hardware EOT, **3-23**
  - CCW Home, 2-12, 2-13
  - CCW input bit, 3-5, 3-11
  - CCW Soft Limit Fault, 2-18, 2-20
  - CCW Software EOT, 3-23
  - CCWEOT Parameter, 2-27
  - Center A Check Box, 4-12
  - Center B Check Box, 4-12
  - Centering the Cursor, 4-12
  - Change System Parameters, 2-11
  - Channel Operation for Analog Inputs, 2-16
  - checker flag bit, 3-20
  - Checksum Verification of Data, 2-16
  - Choosing an Axis, 2-1
  - Choosing the Sample Rate, 4-10
  - Claiming Interrupts, 2-18
  - Clear Faults, 3-34
  - CLOCK Parameter, 2-27
  - CNC, 3-26
  - CNC Fault, 2-19, 2-21
  - Collect Pull-down Menu, 4-10
  - COMM Command, 2-16
  - Command Syntax, 2-5
  - Commanded Position, 4-5
  - Commanded Velocity, 4-5
  - Commanding Motion, 2-11
  - Commands
    - ABORT, 2-13
    - COMM, 2-16
    - CONFIG, 2-8
    - CONFIGENCODER, 2-8
    - CONFIGHALL, 2-9
    - GET, 2-9
    - HALT, 2-13
    - HALT/ABORT, 2-13
    - HELP, 2-5
    - HOME, 2-12
    - ICMD, 2-11
    - INDEX, 2-12
    - INFO, 2-2
    - INITMATRIX, 2-16
    - INITXYCOM, 2-14
    - M, 2-10
    - MOVETO, 2-13
    - PARM, 2-9
    - PARMS, 2-3
    - PROG, 2-23
    - READMATRIX, 2-16
    - READXYCOM, 2-15
    - SET, 2-11
    - START, 2-13
    - STAT, 2-3, 2-9
    - STATUS, 2-11
    - WRITEXYCOM, 2-15
  - commchan argument*, 2-9
  - Communication Testing, 2-16
  - Commutation Offset Setting, 2-8, 2-9
  - Commutation on an Axis, 3-4
  - CONFIG Command, 2-2, 2-8
  - CONFIGENCODER Command, 2-2, 2-8
  - CONFIGHALL Command, 2-2, 2-9
  - Configuration of Axis, 2-2
  - Configuring an Axis, 2-2
  - Configuring Axes, 2-7
  - Configuring Axis Hall Effect, 2-9
  - Configuring I/O, 2-14
  - Configuring Xycom Cards, 2-14
  - Continuous Data Collection, 4-7
  - Continuous Motion Control, 4-15
  - continuous move bit, 3-9, 3-17
  - Creating Servo Performance Traps, 4-8
  - Current Command Limiting, 2-2
  - Current limit potentiometer, 4-36
  - Cursor Measurement Tool Dialog Box, 4-12
  - Cursor Option, 4-12
  - CW Hard Limit Fault, 2-18, 2-20
  - CW Hardware EOT, **3-23**
  - CW Home, 2-12, 2-13
  - CW input bit, 3-5, 3-11
  - CW Soft Limit Fault, 2-18, 2-20
  - CW Software EOT, 3-23
  - CWEOT Parame XE "Parameters:CWEOT" ter, 2-27
  - cyclelines argument*, 2-9
- ## D
- D to A Card Channel Inquiry, 2-3
  - D/A Converter Channel Setting, 2-8
  - D2Achannel argument*, 2-8
  - Data
    - Saving, 4-2
  - Data Collection, 4-5
    - Mode 0, 4-9
    - Mode 1, 4-9
    - Mode 2, 4-9
    - Mode 4, 4-9
  - Data Collection Mode 3, 4-9
  - Data Collection Sampling Rate, 4-10
  - Data Resolution, 4-11
  - Debug960, 1-1
  - Debug960 General Information Screen, 2-3
  - Debug960 Help Screen #1, 2-6
  - Debug960 Help Screen #2, 2-7
  - Debug960 Initialization, 2-1
  - Debug960 Parameter/Status Screen #1, 2-4
  - Debug960 Parameter/Status Screen #2, 2-5
  - decel enable bit, 3-19
  - decel mode bit, 3-19
  - DECEL Parameter, 2-22, 2-28

decel phase bit, 3-8, 3-16  
 Decel Ramp Time, 2-13  
 Decelerate Axis to Stop, 2-22  
 DECELMODE Parameter, 2-28  
 DECELRATE Parameter, 2-28  
 Defined Slope of a Velocity Curve at a Specified Angle,  
   **2-26**  
 Digital and Analog I/O, 2-1, 2-14  
 Disable an Axis, 2-2, 2-7  
 Disable Mask Display, 3-28  
 disable queued bit, 3-18  
 Disable the Axis Drive, 2-11  
 DISABLEMASK, 2-22  
 DISABLEMASK Parameter, 2-28  
 Disabling Faults, 2-18  
 Disabling the Drives, 2-22  
   *displacement argument*, 2-12  
 Display Axis Parameter Value, 2-9  
 Display Axis Parameters, 2-9  
 Display Pull-down Menu, 3-2, **3-3**, 4-11  
 Display Servo States, 2-9  
 Displaying Servo Information, 4-5  
 done bit, 3-6, 3-12  
 Drive, 3-24  
 drive enable bit, 3-11  
 drive enabled bit, 3-4  
 Drive Fault, 2-18, 2-20  
 drive fault input bit, 3-5, 3-12  
 DRIVE Parameter, 2-2, 2-28

**E**

ECHO Parameter, 2-28  
 Edit System Parameters, 2-11  
 Editing Parameters, 2-11  
   SET, 2-11  
 Electrical Poles Setting, 2-8  
 Enable an Axis, 2-2  
 Enabling Faults, 2-18  
 Enabling Interrupts, 2-18  
   *encchan argument*, 2-9  
   *encchanne argument*, 2-8  
 Encoder Axis Configuration, 2-8  
 Encoder Channel/Hall Effect Setting, 2-9  
 Encoder Channel/Position Feedback Setting, 2-8, 2-9  
 Encoder Counts/Electrical Cycle Setting, 2-9  
 Encoder Counts/Motor Rev. Setting, 2-8, 2-9  
 Encoder Feedback, 2-2  
 encoder feedback bit, 3-15  
 Encoder Feedback/Hall Sensors, 2-2  
 Engaging Brakes, 2-22  
 Enter Filename  
   entry field, 4-3  
 Error Message String, 2-24  
 errormapping enable bit, 3-15  
 Establishing Servo Performance Traps, 4-8  
 Exceeded Trap Value Point, 4-8

Exceeding Parameter Limits, 2-23  
 Exercising an Axis, 2-1  
 Exit, 3-33  
 Exit Pull-down Menu, 3-33  
 Exit Torque Command, 2-11  
 External Feedback, 3-26  
 External Feedback Fault, 2-19, 2-21  
 External Feedback Inquiry, 2-3

**F**

Fault Acknowledgements, 2-23  
 Fault Handling, 2-18  
   FAULT TYPE, 2-18, 2-19, 2-20, 2-21  
     BIT LOCATION, 2-18, 2-19, 2-20, 2-21, 2-33, 2-34, 2-35, 2-36, 2-40, 2-41, 2-42, 2-43, 2-44, 2-45, 2-46, 2-47  
     Setting Fault Masks, 2-20  
 Fault Mask Display, 3-27  
 Fault Masks, 2-18, 2-19, 2-20, 2-21  
 FAULT Parameter, 2-29  
 Fault Status Display, 3-22  
 Fault\_Control Pull-down Menu, 3-34  
 faulted bit, 3-6, 3-12  
 FAULTMASK Parameter, 2-29  
 Faults Latched, 3-34  
 Faults Not Latched, 3-34  
 fb failure input bit, 3-14  
 FBWINDOW Parameter, 2-29  
 Feedback, 3-24  
 Feedback Channel Inquiry, 2-3  
 Feedback Device, 2-2  
 Feedback Fault, 2-18, 2-20  
 Feedback Resolution Inquiry, 2-3  
 Feedback Type Inquiry, 2-3  
 Feedback Types  
   encoder, 2-2  
   encoder/hall sensors, 2-2  
   resolver, 2-2  
   *feedrate argument*, 2-12, 2-13  
 feedrate override bit, 3-8, 3-17  
 FEEDRATEMODE Parameter, 2-29  
 File Load Option, 4-3  
 File Pull-down Menu, 4-2, 4-3  
 File Save Option, 4-3  
 File Selection Dialog Box, 4-3, **4-4**  
 Firmware Level, 2-2  
 Firmware Operational Mode, 3-3  
 Freerun Mode, 4-7

**G**

GET\_PGAIN Parameter, 2-10  
 GET\_FAULT Command, 2-23  
 GET Command, 2-9  
 Get/Set/Monitor Axis Parameters, 2-9  
 Get/Set/Monitor Servo States, 2-9

Getting Started, 2-1

## H

Hall Effect Setting, 2-9  
 hall input #1, #2, #3, and #4 bit, 3-7, 3-12  
 HALT Command, 2-13  
 Halt Data Acquisition, 4-7  
 Halt Mask Display, 3-30  
 halt queued bit, 3-18  
 HALT/ABORT Command, 2-13  
 Halting/Aborting Motion, 2-13  
 HALTMASK, 2-22  
 HALTMASK Parameter, 2-29  
 Hardware Status, 3-3  
 Hardware Trouble Shooting, 2-2  
 HELP Command, 2-5  
 Help Pull-down Menu, 3-35  
 hold active bit, 3-10, 3-17  
 hold queue bit, 3-10, 3-17  
 HOME Command, 2-12  
 Home Fault, 2-19, 2-21  
 home input bit, 3-5, 3-12  
 Home switch tolerance, 3-25  
 Home Tolerance Fault, 2-19, 2-21  
 HOMEALT, 2-12  
 homed bit, 3-15  
 HOMESWITCHPOS Parameter, 2-29  
 HOMESWITCHTOL Parameter, 2-30  
 HOMEVELMULT Parameter, 2-30  
 Homing, 3-24  
 Homing Axes, 2-12  
 homing bit, 3-8, 3-17  
 homing direction bit, 3-9, 3-17  
 HP VME Laser bit, 3-14

## I

I/O Testing, 2-1  
 IAVG Parameter, 2-30  
 IAVGLIMIT Parameter, 2-30  
 IAVGTIME Parameter, 2-30  
 ICMD Command, 2-11  
 ICMD Parameter, 2-30  
 Icons, 2-1  
 if  $dir \geq 0$  argument, 2-12, 2-13  
 if  $dir < 0$  argument, 2-12, 2-13  
 IMAX Parameter, 2-2, 2-31  
 Incremental Motion, 2-12  
 INDEX Command, 2-12  
 INFO Command, 2-2  
 INFO Screen, 2-3  
 Initial Power Up, 2-7  
 Initialization of Xycom Card, 2-14  
 Initiate Absolute Moves, 2-11  
 Initiate Relative Moves, 2-11  
 INITMATRIX Command, 2-16

INITXYCOM Command, 2-14  
 in-position bit, 3-6, 3-12  
 INOSLIMIT Parameter, 2-31  
 Input potentiometer, 4-38  
 integral + clamped bit, 3-13  
 integral -clamped bit, 3-12  
 Integral Error Trap, 4-20, 4-33  
 Integral gain - Ki, 4-31  
 Internal Defaults, 2-7  
 Interrupt Mask Display, 3-29  
 interrupt pending bit, 3-20  
 INTMASK, 2-18, 2-22  
 INTMASK Parameter, 2-31  
 Invalid Parameter Name, 2-10  
 IOLEVEL Parameter, 2-31  
 IVEL Parameter, 2-31

## J

jog enable bit, 3-21

## K

Ki Integral gain, 4-18, 4-22, 4-23  
 KI Parameter, 2-31  
 KP Parameter, 2-32  
 Kp Proportional gain, 4-17, 4-21, 4-31, 4-32  
 Kpos Position gain, 4-18, 4-24, 4-31, 4-32, 4-36

## L

*lines argument*, 2-8, 2-9

## M

M ICMD Command, 2-10  
 M Command, 2-2, 2-10  
 M POS Command, 2-2  
 marker armed bit, 3-21  
 marker input bit, 3-6, 3-12  
 master fb failure bit, 3-14  
 Master Feedback, 3-24  
 Master Feedback Fault, 2-19, 2-21  
 MASTERABS Parameter, 2-32  
 MASTERLEN Parameter, 2-32  
 MASTERPOS Parameter, 2-32  
 MASTERRES Parameter, 2-32  
 Matrix Analog Inputs, 2-16  
 MAX\_PHASE Parameter, 2-26, 2-32  
 Monitor, 2-10  
 Monitor Parameters, 2-11  
 Motion Commands, 2-11  
   Absolute Motion, 2-13  
   Continuous Motion, 2-13  
   HOMEALT, 2-12  
   Homing Axes, 2-12  
   Incremental Motion, 2-12

Motion Profile Generation, 4-8  
 Motion Profiles  
   Negative Profile, 4-13  
   Positive Profile, 4-13  
 Motion Status Display, 3-16  
 MOTIONSTATUS Parameter, 2-33  
 Motor Configuration  
   AC brushless, 2-8  
   DC brush, 2-8  
 Motor Tuning, 2-2  
 move direction bit, 3-7, 3-16  
 MOVEQDEPTH Parameter, 2-37  
 MOVEQSIZE Parameter, 2-37  
 MOVETO Command, 2-13  
 moving bit, 3-7, 3-16  
 Moving Cursors, 4-12  
 MSET, 2-48  
 MSETTEST, 2-48  
 msetting bit, 3-14  
 Multiple Data Collection, 4-7

**N**

Negative Motion Profile, 4-13

**O**

*offset argument*, 2-8, 2-9  
 OS/2 Desktop, 2-1  
 OS/2 Full Screen, 2-1  
 OS/2 Window, 2-1

**P**

Parameter #1, 4-10  
 Parameter #2, 4-10  
 Parameter Descriptions, 2-24  
 Parameter Limits, 2-23  
 Parameter Operation Commands, 2-9  
 Parameter Screening, 2-11  
 Parameters  
   ABORTMASK, 2-24  
   ACCEL, 2-25  
   ACCELMODE, 2-25  
   ACCELRATE, 2-25  
   AFFGAIN, 2-25  
   ALT\_STATUS, 2-25  
   AUX, 2-25  
   AUXMASK, 2-25  
   AUXOFFSET, 2-26  
   AVGVEL, 2-26  
   AVGVELTIME, 2-26  
   BASE\_SPEED, 2-26, 2-32  
   BLOCKMOTION, 2-27  
   CAMOFFSET, 2-27  
   CAMPOINT, 2-27  
   CAMPOSITION, 2-27

CCWEOT, 2-27  
 CLOCK, 2-27  
 CWEOT, 2-27  
 DECEL, 2-28  
 DECELMODE, 2-28  
 DECELRATE, 2-28  
 DISABLEMASK, 2-28  
 DRIVE, 2-28  
 ECHO, 2-28  
 FAULT, 2-29  
 FAULTMASK, 2-29  
 FBWINDOW, 2-29  
 FEEDRATEMODE, 2-29  
 HALTMASK, 2-29  
 HOMESWITCHPOS, 2-29  
 HOMESWITCHTOL, 2-30  
 HOMEVELMULT, 2-30  
**IAVG**, 2-30  
 IAVGLIMIT, 2-30  
 IAVGTIME, 2-30  
 ICMD, 2-30  
 IMAX, 2-31  
 INPOSLIMIT, 2-31  
 INTMASK, 2-31  
 IOLEVEL, 2-31  
 IVEL, 2-31  
 KI, 2-31  
 KP, 2-32  
 MASTERABS, 2-32  
 MASTERLEN, 2-32  
 MASTERPOS, 2-32  
 MASTERRES, 2-32  
 MAX\_PHASE, 2-26, 2-32  
 MOTIONSTATUS, 2-33  
 MOVEQDEPTH, 2-37  
 MOVEQSIZE, 2-37  
 PGAIN, 2-37  
 PHASE\_SPEED, 2-26, 2-32, 2-37  
 POS, 2-37  
 POSCMD, 2-37  
 POSERR, 2-37  
 POSERRLIMIT, 2-38  
 POSTARGET, 2-38  
 POSTOGO, 2-38  
 PROFQDEPTH, 2-38  
 PROFQSIZE, 2-38  
 RAWPOS, 2-38  
 RESOLVER, 2-38  
 REVERSALMODE, 2-39  
 REVERSALVALUE, 2-39  
 SAFEZONECCW, 2-39  
 SAFEZONECW, 2-39  
 SAFEZONEMODE, 2-39  
 SERVOSTATUS, 2-40  
 SOFTLIMITMODE, 2-43  
 STATUS, 2-44  
 SYNCSPD, 2-47

VEL\_SHIFT, 2-48  
 VELCMDTRAP, 2-47  
 VELPOSITION, 2-47  
 VELTRAP, 2-48  
 VFF, 2-48  
 PARMS Command, 2-3  
 pendant jog bit, 3-21  
 PGAIN Parameter, 2-10, 2-37  
 PHASE\_SPEED Parameter, 2-26, 2-32, 2-37  
 Picking an Axis, 2-1  
 Plot, 1-1  
 Plot Mainmenu Dialog Box, **4-2**  
 Plot Pull-down Menu, 4-5, **4-7**  
 Plot Utility, 2-2  
*poles argument*, 2-8  
 POS Parameter, 2-37  
 POSCMD Parameter, 2-37  
 POSERR Parameter, 2-37  
 POSERRLIMIT Parameter, 2-38  
*position argument*, 2-13  
 Position Displacements, 2-12  
 Position Error, 4-5, 4-22, 4-23  
 Position Error Fault, 2-18, 2-20  
 Position error limit, **3-22**  
 Position Error Trap, 4-20, 4-33  
 Position Gain - Kpos, 4-31  
 Position loop, 4-17, 4-23  
 position loop only bit, 3-15  
 Positive Motion Profile, 4-13  
 Positive Polarity Torque Commands, 2-2  
 Positive Velocity Output, 2-2  
 POSTARGET Parameter, 2-38  
 POSTOGO Parameter, 2-38  
 Preparation for Axis Configuration, 2-7  
 Preparation for Drive Enable, 2-2  
 Probe Fault, 2-19, 2-21  
 Probe input, 3-26  
 probe input bit, 3-6, 3-12  
 profile mode bit, 3-8, 3-17  
 PROFQDEPTH Parameter, 2-38  
 PROFQSIZE Parameter, 2-38  
 PROG Command, 2-23  
 Program Icons, 2-1  
 Programming, 3-24  
 Programming Errors, 2-23  
     Acknowledge, 2-24  
     Clear, 2-24  
 Programming Fault, 2-18, 2-20  
 Proportional gain - Kp, 4-31  
 Proportional gain parameter Kp, 4-30

## Q

QAxis Prompt, 2-1  
 queued command bit, 3-9, 3-17  
 quick home bit, 3-20

## R

R/D Conversion Channel Setting, 2-8  
 R/D Conversion Resolution, 2-8  
*R2Dchannel argument*, 2-8  
 RAWPOS Parameter, 2-38  
 Reading Analog Inputs, 2-16  
 Reading Digital Inputs, 2-15  
 READMATRIX Command, 2-16  
 READXYCOM Command, 2-15  
 Real-time Servo Performance, 4-1  
 Re-initializing the Axis Card, 2-7  
 Re-positioning Cursors, 4-12  
*resolution argument*, 2-8  
 Resolution of Displayed Data, 4-11  
 Resolver Feedback, 2-2  
 RESOLVER Parameters, 2-38  
 Resolver/Inductosyn Axis, 2-8  
 Resolver/Inductosyn Feedback, 2-8  
 Resume, 3-33  
 Retrieving I/O, 2-14  
 REVERSALMODE Parameter, 2-39  
 REVERSALVALUE Parameter, 2-39  
 Revision Level, 2-2  
 RMS current limit, 3-23  
 RMS Current Limit Fault, 2-18, 2-20

## S

Safe Zone, 3-26  
 SAFEZONECCW Parameter, 2-39  
 SAFEZONECW Parameter, 2-39  
 SAFEZONEMODE Parameter, 2-39  
 Sample Period, 4-13  
 Sampling Rate Option, 4-10  
 Saving Data, 4-2  
 Seating Up Brushless Motors  
     Working With Hall Effect Motors, 2-49  
 Select Axis Option, 4-16  
 Selecting an Axis, 2-1  
 Selecting Status Types to Monitor, 3-2  
 Send an Interrupt, 2-22  
 Sending Axes Home, 2-11  
 Servo Data Display Channels, 4-5  
     Actual Position, 4-5  
     Actual Velocity, 4-5  
     Commanded Position, 4-5  
     Commanded Velocity, 4-5  
     Position Error, 4-5  
     Spare, 4-5  
     Torque, 4-5  
     Velocity Error, 4-5  
 Servo gain potentiometer, 4-35  
 Servo Performance Traps, 4-8  
 Servo Status Display, 3-11  
 SERVOSTATUS Parameter, 2-40

SET DRIVE 0 Command, 2-11  
 SET DRIVE 0 Command, 2-7  
 SET FAULT Command, 2-23  
 SET Command, 2-11  
 Set Data Acquisition Mode, 4-9  
 Set Data Acquisition Mode Dialog Box, **4-9**  
 SET DRIVE Command, 2-2  
 Set Speed/Distance Option, 4-15  
 Set Trap Mode, 4-8  
 Set Trap Mode Dialog Box, **4-8**  
 Setting Auxiliary Output, 2-22  
 Setting Digital Outputs, 2-15  
 Setting Fault Masks, 2-20  
 Setting I/O for Analog/Digital Option Cards, 2-14  
 Setting the Sample Rate, 4-10  
 Setting Up Brushless Motor Commutation, 2-48  
     MSET, 2-48  
     MSETTEST, 2-48  
 setup pending bit, 3-20  
 Single button, 4-20, 4-36  
 Single Data Collection, 4-7  
 SOFTLIMITMODE Parameter, 2-43  
 Software Installation, 2-1  
 Software Limits Disabled, 2-8  
 Software Limits Enabled, 2-8  
 Software Trouble Shooting, 2-2  
 Spare, 4-5  
 Specifying Incorrect Arguments, 2-23  
 Speed/Distance Setup Dialog Box, **4-15**  
 START Command, 2-13  
 Starting Motion, 2-11  
 Startup, 2-1  
 Startup Considerations, 2-2  
 STAT Command, 2-3  
 STAT/PARM Command, 2-9  
 Status, 1-1  
 Status Bits, 2-33, 2-34, 2-35, 2-36  
 STATUS Command, 2-11  
 Status Errors  
     CCW Hardware EOT, 3-23  
     CCW Software EOT, 3-23  
     CNC, 3-26  
     CW Hardware EOT, 3-23  
     CW Software EOT, 3-23  
     Drive, 3-24  
     External Feedback, 3-26  
     Feedback, 3-24  
     Home switch tolerance, 3-25  
     Homing, 3-24  
     Master Feedback, 3-24  
     Position error limit, 3-22  
     Probe input, 3-26  
     Programming, 3-24  
     RMS current limit, **3-23**  
     Safe Zone, 3-26  
     User, 3-25  
     Velocity command trap, 3-25

Velocity trap, 3-25  
 STATUS Parameter, 2-44  
 Status Utility Initialization Screen, **3-2**  
 Step Length, 4-15  
 Step Speed, 4-15  
 Stop Data Acquisition, 4-7  
 Stop Motion Control, 4-15  
 Stopping a Communication Test, 2-18  
 Stopping Commanded Motion, 2-13  
 sync mode bit, 3-9, 3-17  
 SYNCSPD Parameter, 2-47  
 System Diagnostics, 2-16  
 System Reset, 2-7  
 System Utilities Diskette, 2-1

## T

Tachometer based velocity loop, 4-30  
 Time Pull-down Menu, 3-32  
 Tips on Startup, 2-2  
 Torque, 4-5  
 Torque trap entry field, 4-8  
 Torque Trap Mode, 4-8  
 Transmitting Data via VME Bus, 2-16  
 Trap Menu, 4-8  
 Trigger Pull-down Menu, 4-7  
 Trouble Shooting, 2-2  
 Tune Option, 4-14  
 Tuning procedures for Servo Loops, 4-18, 4-32  
 Tuning procedures for Tachometer Loops, 4-32  
 Tuning the Motor, 2-2  
 Tuning tips, 4-29  
 Tuning with tachometer feedback, 4-30

## U

U31 Library Commands, 2-1  
 U31 System Utilities Diskette, 2-1  
 Unidex 30, 1-1  
 Unidex 31, 1-2  
 Unidex 600, 1-2  
 Unrecognized Parameter Name, 2-11  
 User, 3-25  
 User Fault, 2-19, 2-21  
 Utility Icons, 2-1

## V

VEL\_SHIFT Parameter, 2-48  
 VELCMDTRAP Parameter, 2-47  
 Velocity command trap, 3-25  
 Velocity Command Trap Fault, 2-19, 2-21  
 Velocity Command Trap Mode, 4-8  
 Velocity Error, 4-5, 4-21  
 Velocity feedforward gain - Vff, 4-30, 4-31  
 Velocity loop, 4-16  
 Velocity loop adjustment, 4-20

Velocity trap, 3-25  
Velocity Trap Fault, 2-19, 2-21  
Velocity Trap Mode, 4-8  
VELPOSITION Parameter, 2-47  
VELTRAP Parameter, 2-48  
Vff, 4-31  
VFF enable bit, 3-13  
VFF or zero Pgain bit, 3-13  
VFF Parameter, 2-48  
Vff Velocity feedforward gain, 4-18  
Viewing Parameters, 2-9  
    GET, 2-9  
    Monitor, 2-10  
    PARAM, 2-9  
    STAT, 2-9  
VME, 1-1  
VME backplane, 4-1  
VME Bus Error, 2-14  
VME I/O, 1-1

**W**

Working With Hall Effect Motors, 2-49  
WRITEXYCOM Command, 2-15  
Writing Digital Outputs, 2-15

**X**

Xycom Digital I/O, 2-14

**Z**

Zero Line Option, 4-11

▽ ▽ ▽





## READER'S COMMENTS

**The UNIDEX® 600 Series Utilities Manual, OS/2 Manual  
P/N EDU 164, June, 1997**

Please answer the questions below and add any suggestions for improving this document. Is the information:

	<u>Yes</u> _____	<u>No</u> _____
Adequate to the subject?	_____	_____
Well organized?	_____	_____
Clearly presented?	_____	_____
Well illustrated?	_____	_____
Would you like to see more illustrations?	_____	_____
Would you like to see more text?	_____	_____

How do you use this document in your job? Does it meet your needs?  
What improvements, if any, would you like to see? Please be specific or cite examples.

---

---

---

---

---

Your name \_\_\_\_\_  
Your title \_\_\_\_\_  
Company name \_\_\_\_\_  
Address \_\_\_\_\_  
\_\_\_\_\_

Remove this page from the document and fax or mail your comments to the technical writing department of Aerotech.

AEROTECH, INC.  
Technical Writing Department  
101 Zeta Drive  
Pittsburgh, PA. 15238-2897 U.S.A.  
Fax number (412) 963-7009

