PACIFIC SCIENTIFIC

HIGH PERFORMANCE MOTORS & DRIVES

110 Fordham Road
Wilmington, MA 01887
(978) 988-9800
Fax (978) 988-9940

Part# MA930
List Price $30 U.S.
January, 1999
Rev E

M A 9 3 0

OC930 Hardware & Software Reference Manual

& SC900 Software Reference Manual

Firmware Version 1.5

WARRANTY AND LIMITATION OF LIABILITY

Includes software provided by Pacific Scientific

Pacific Scientific warrants its motors and controllers ("Product(s)") to the original purchaser (the "Customer"), and in the case of original equipment manufacturers or distributors, to their original consumer (the "Customer") to be free from defects in material and workmanship and to be made in accordance with Customer's specifications which have been accepted in writing by Pacific Scientific. In no event, however, shall Pacific Scientific be liable or have any responsibility under such warranty if the Products have been improperly stored, installed, used or maintained, or if customer has permitted any unauthorized modifications, adjustments, and/or repairs to such Products. Pacific Scientific's obligation hereunder is limited solely to repairing or replacing (at its option), at its factory any Products, or parts thereof, which prove to Pacific Scientific's satisfaction to be defective as a result of defective materials or workmanship, in accordance with Pacific Scientific's stated warranty, provided, however, that written notice of claimed defects shall have been given to Pacific Scientific within two (2) years after the date of the product date code that is affixed to the product, and within thirty (30) days from the date any such defect is first discovered. The products or parts claimed to be defective must be returned to Pacific Scientific, transportation prepaid by Customer, with written specifications of the claimed defect. Evidence acceptable to Pacific Scientific must be furnished that the claimed defects were not caused by misuse, abuse, or neglect by anyone other than Pacific Scientific.

Pacific Scientific also warrants that each of the Pacific Scientific Motion Control Software Programs ("Program(s)") will, when delivered, conform to the specifications therefore set forth in Pacific Scientific's specifications manual. Customer, however, acknowledges that these Programs are of such complexity and that the Programs are used in such diverse equipment and operating environments that defects unknown to Pacific Scientific may be discovered only after the Programs have been used by Customer. Customer agrees that as Pacific Scientific's sole liability, and as Customer's sole remedy, Pacific Scientific will correct documented failures of the Programs to conform to Pacific Scientific's specifications manual. PACIFIC SCIENTIFIC DOES NOT SEPARATELY WARRANT THE RESULTS OF ANY SUCH CORRECTION OR WARRANT THAT ANY OR ALL FAILURES OR ERRORS WILL BE CORRECTED OR WARRANT THAT THE FUNCTIONS CONTAINED IN PACIFIC SCIENTIFIC'S PROGRAMS WILL MEET CUSTOMER'S REQUIREMENTS OR WILL OPERATE IN THE COMBINATIONS SELECTED BY CUSTOMER. This warranty for Programs is contingent upon proper use of the Programs and shall not apply to defects or failure due to: (i) accident, neglect, or misuse; (ii) failure of Customer's equipment; (iii) the use of software or hardware not provided by Pacific Scientific; (iv) unusual stress caused by Customer's equipment; or (v) any party other than Pacific Scientific who modifies, adjusts, repairs, adds to, deletes from or services the Programs. This warranty for Programs is valid for a period of ninety (90) days from the date Pacific Scientific first delivers the Programs to Customer.

THE FOREGOING WARRANTIES ARE IN LIEU OF ALL OTHER WARRANTIES (EXCEPT AS TO TITLE), WHETHER EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR OF FITNESS FOR ANY PARTICULAR PURPOSE, AND ARE IN LIEU OF ALL OTHER OBLIGATIONS OR LIABILITIES ON THE PART OF PACIFIC SCIENTIFIC. PACIFIC SCIENTIFIC'S MAXIMUM LIABILITY WITH RESPECT TO THESE WARRANTIES, ARISING FROM ANY CAUSE WHATSOEVER, INCLUDING WITHOUT LIMITATION, BREACH OF CONTRACT, NEGLIGENCE, STRICT LIABILITY, TORT, WARRANTY, PATENT OR COPYRIGHT INFRINGEMENT, SHALL NOT EXCEED THE PRICE SPECIFIED OF THE PRODUCTS OR PROGRAMS GIVING RISE TO THE CLAIM, AND IN NO EVENT SHALL PACIFIC SCIENTIFIC BE LIABLE UNDER THESE WARRANTIES OR OTHERWISE, EVEN IF PACIFIC SCIENTIFIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, DAMAGE OR LOSS RESULTING FROM INABILITY TO USE THE PRODUCTS OR PROGRAMS, INCREASED OPERATING COSTS RESULTING FROM A LOSS OF THE PRODUCTS OR PROGRAMS, LOSS OF ANTICIPATED PROFITS, OR OTHER SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER SIMILAR OR DISSIMILAR, OF ANY NATURE ARISING OR RESULTING FROM THE PURCHASE, INSTALLATION, REMOVAL, REPAIR, OPERATION, USE OR BREAKDOWN OF THE PRODUCTS OR PROGRAMS, OR ANY OTHER CAUSE WHATSOEVER, INCLUDING NEGLIGENCE.

The foregoing shall also apply to Products, Programs, or parts for the same which have been repaired or replaced pursuant to such warranty, and within the period of time, in accordance with Pacific Scientific's date of warranty.

No person, including any agent, distributor, or representative of Pacific Scientific, is authorized to make any representation or warranty on behalf of Pacific Scientific concerning any Products or Programs manufactured by Pacific Scientific, except to refer purchasers to this warranty.

Table of Contents

1 Overview

Introduction

This manual covers the OC930 Serial Communication Option Card hardware and the SC930 drive firmware functionality. The SC930 drive firmware controls the base SC900 drive whether the OC930 Serial Option Card is installed or not.

1.1 OC930 Serial Communications Option Card Definition

Features

The Serial Communications Option Card (OC930) for your Pacific Scientific SC900 is used to communicate over a 9600 Baud serial link to a host PC. Combined with 930 Dialogue, a menu driven software support package which runs on your PC, the OC930 offers the following features:

- All digital setup of the current loop, velocity loop, and (when utilized) position loop. There are no pots, DIP switches, plug-on jumpers or components to alter when setting up the servo loops. All parameters are downloaded using an RS-232 or RS-485 port and can be saved in non-volatile memory in the SC900 or on the OC930.
- Automatic drive setup using 930 Dialogue.
- Automatic Analog Command Offset adjustment using 930 Dialogue.
- Simplified uploading, downloading, and disk storage of SC900 parameters for easy cloning and backup documentation.
- Precise readout of motor velocity, position, and other variables using the serial link and 930 Dialogue.
- On-board EEPROM which allows the OC930 to be used as a Personality Module with all non-volatile parameters stored on the OC930 (see Appendix B for additional information.)

Except for using the OC930 as a Personality Module, the OC930 is only required for set up and monitoring of SC900 drive operation. Once set up, the OC930 Option Card can be removed and an OC900 Blank Panel installed with the SC900 drive remaining fully functional.

1.2 How to use this manual

Chapter 2, “Getting Started”, is a step-by-step guide allowing you to configure an SC930 and run your motor within a few minutes. **It is strongly recommended that you go through Chapter 2 first.** This will give you a feel for using the SC930 and lay the framework for reading the other chapters. Chapters 3 through 7 should be read thoroughly to gain the most from the OC930. If your SC900 is to be used as a slave in electronic gearing or driven by step-and-direction inputs, Chapter 5 should also be read. Chapter 7 is an alphabetized listing of commands and variable with detailed descriptions which is a very useful reference during setup.

1.3 Warranty

The Pacific Scientific OC930 has a **two year warranty** against defects in material and assembly. Products that have been modified by the customer, physically mishandled, or otherwise abused through miswiring, and so on, are exempt from the warranty plan.

Warning

If the continuous current rating of the drive is greater than the continuous current rating of the motor that it is being used with, then it is possible to cause significant damage to the motor. Pacific Scientific may not honor the warranty of the motor if it is run under these conditions.

2 Getting Started

Introduction

This chapter provides a step-by-step introduction to setting up SC900s with the OC930. This procedure uses the minimum possible equipment to run an unloaded motor and sets motor speed from a PC's serial port. It is strongly recommended that all first time users go through this procedure to become familiar with the OC930 and the 930 Dialogue PC interface software before installing the servo system in a machine.

2.1 Setting Up the Hardware

What you will need

To go through this product introduction procedure you will need the following items.

- SC900 Base Servo Drive
- OC930 Serial Com Option Card
- Appropriate Brushless Motor with nothing attached to the shaft
- PC Running Windows 3.1 or higher
- 930 Dialogue Floppy Disk
- Motor Power and Feedback Cables (J2, J3)
- RS-232 Communications Cable (J31)
- DB-25 Connector Mate (J4)
- AC Power Line (J1)

If your OC930 is not already installed in your SC900, then use the following instructions to install it.

CAUTION

NEVER insert or remove an Option Card with the Control AC Power (J1-5,6) active. Damage to the base SC900 or the Option Card could occur.

Procedure

1. Remove Control AC Power from the SC900. The system status LED should be blank.
2. Loosen the two locking screws counter-clockwise on the existing face plate or existing Option Card and remove.
3. Position the new Option Card so that the silk screen reads the same as the base SC900.
4. Insert the Option Card by sliding it in all the way until it is flush with the base SC900.
5. Tighten the two locking screws by turning clockwise.

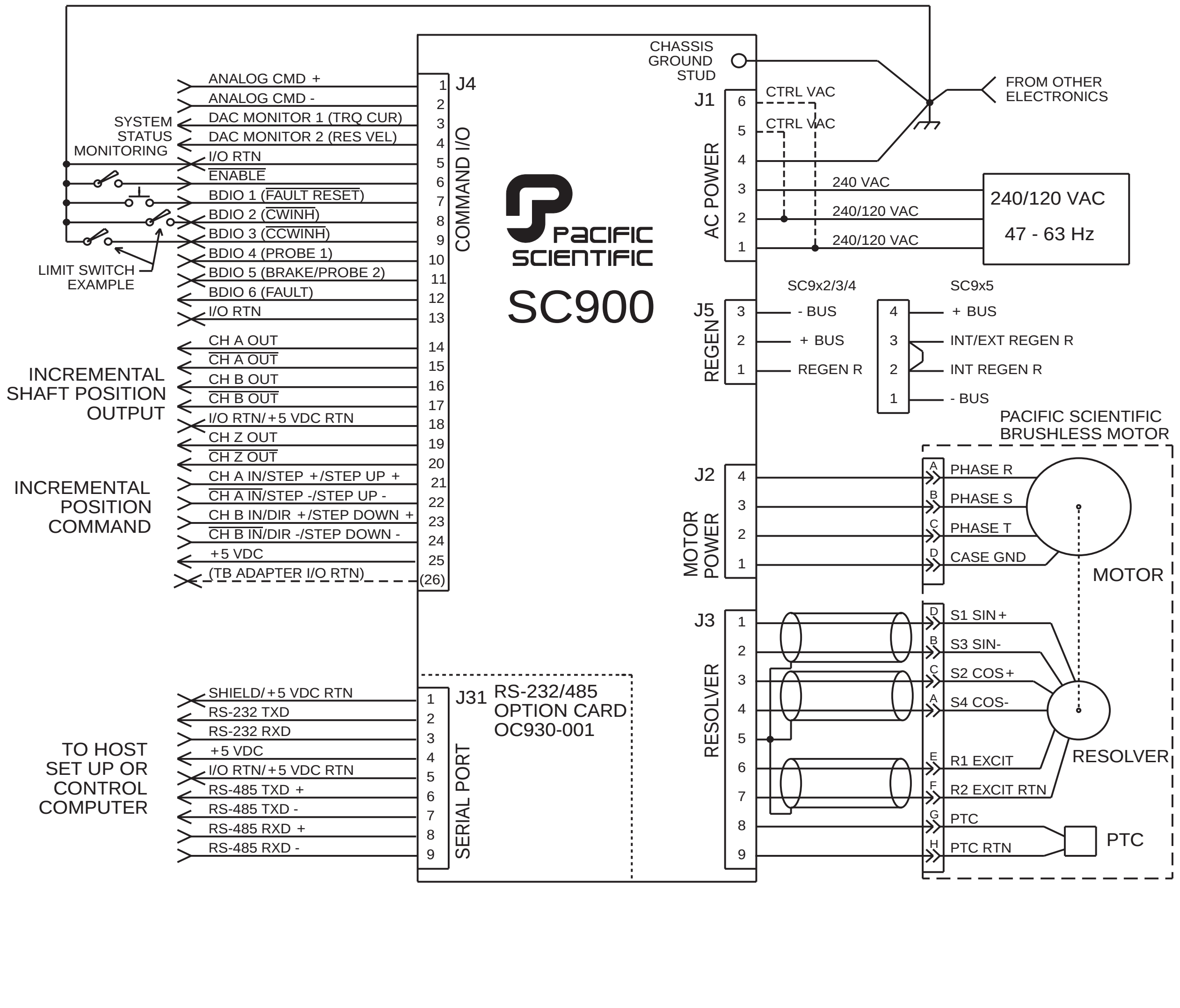
Wiring connections

Connect the motor, feedback, and AC Power cables as shown in the following Connection Diagram but **do not apply the AC Power at this time**. It is recommended that Pacific Scientific motor and feedback cables be used during setup since improper cabling is the number one cause of start up problems.

The RS-232 cable made by Pacific Scientific (order number CS-232-750) can be used to connect the 9 pin serial port socket on the OC930 to the PC. If this cable is unavailable, a simple 3 wire cable can be made using the wiring diagram shown on page 3-5.

The last connection needed is to provide the hardware enable to the SC900 via J4-6 and I/O RTN on J4-5. Preferably connect a toggle switch between J4-6 and J4-5. If a toggle switch is not available a clip lead that can connect or not connect J4-6 to J4-5 will do.

Connection diagram



2.2 Installing 930 Dialogue for Windows

Procedure

To install 930 Dialogue, perform the following:

1. Insert the 930 Dialogue for Windows diskette in your disk drive (A: or B:). Start Windows and choose **Run** from the **File Menu** of Program Manager. At the Command Line, type **A:\setup** (or B:\ setup) and **press the enter key (↵)** or **click on OK.**

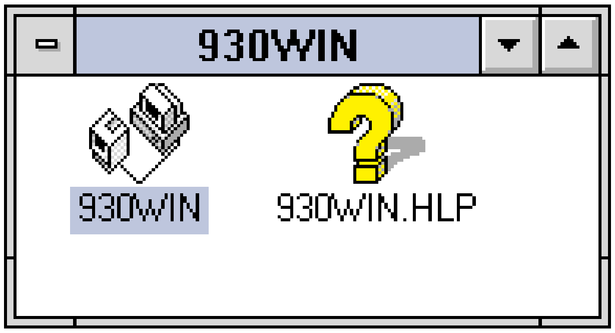
Note: *When finished, the 930 Dialogue disk should be removed from the drive and stored in a safe place.*

2.3 Starting 930 Dialogue

Procedure

To begin using 930 Dialogue for Windows, perform the following:

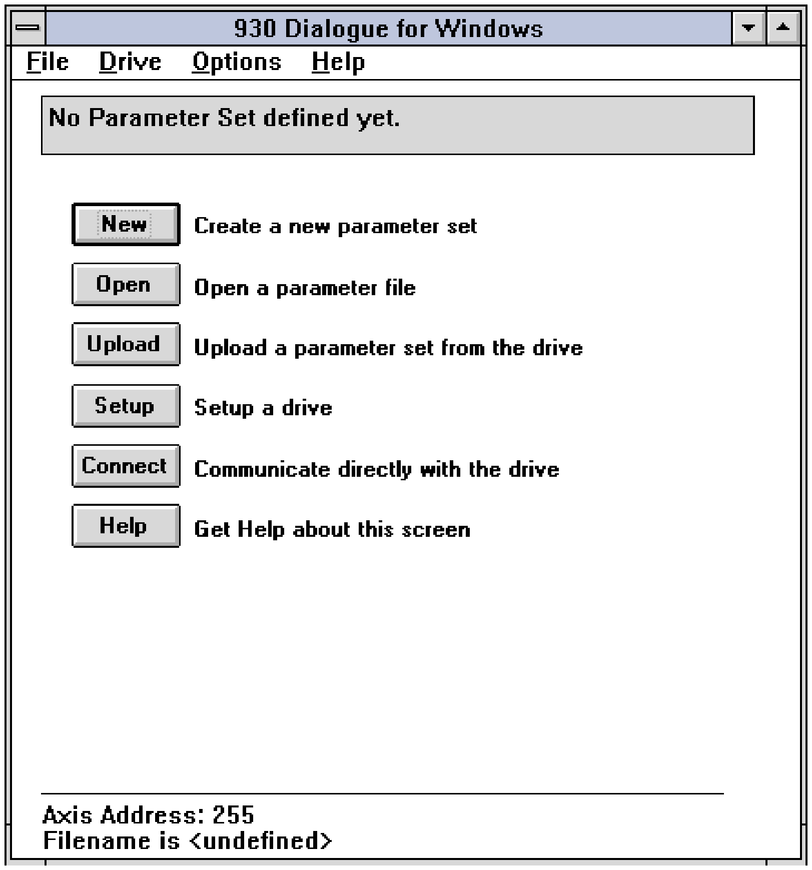
1. If not already open, **open the 930WIN Group** in Program Manager. Double click on the 930WIN icon.



2.4 Getting Around in 930 Dialogue

930 Dialogue main menu

Once you double-click on the 930WIN icon, the following window will appear:



No Parameter Set defined yet.

Create a new parameter set

Open a parameter file

Upload a parameter set from the drive

Setup a drive

Communicate directly with the drive

Get Help about this screen

Axis Address: 255
Filename is <undefined>

Movement keys

930 Dialogue for Windows is a standard Windows application and the normal cursor movement keys operate the same way as in all windows applications.

- **<F1>** gives context sensitive on-line help

2.5 Configuring Your System

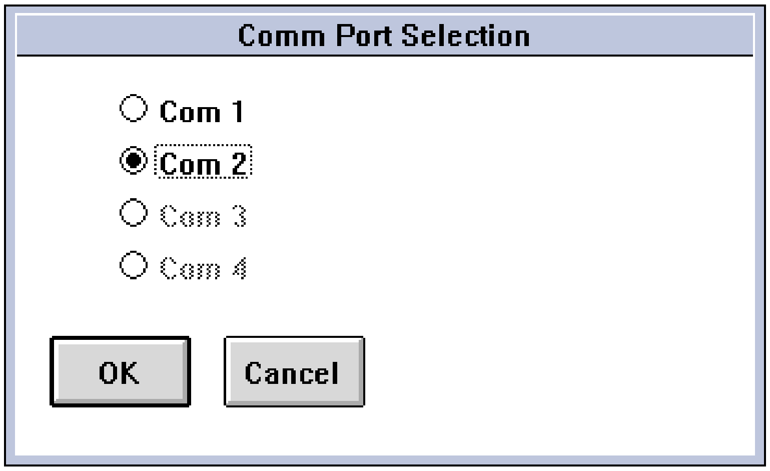
Applying AC Power

Carefully check all wiring connections and ensure that J4-6 is not connected to J4-5. Apply AC power to your controller. The drive status display LED should be alternately flashing U C (for unconfigured) after the power up message.

Serial Port

To specify the PC serial port that is connected to the OC930:

1. Select **Port Configuration** from the **Options** Menu and the following dialogue box will appear:


Comm Port Selection

○ Com 1 [✓ ● Com 2] ○ Com 3 ○ Com 4

2. Specify the serial port that you want to use and click on **OK**.

2.6 Configuring Your Drive

Procedure

- Select **Drive Set Up** from the **Drive** menu and press <**Enter**>. Select **Automatic** and press ↵. Enter the first four Digits of your Motor Part Number. For example, if the part number on your motor name plate is R32GENC-R2-NS-NV-00, type **R32G** and press ↵.
- Select the controller you're using (Example: SC933) and press ↵.
- Select the AC line voltage being used and press ↵**.**

  Note: *This option only appears if you are connected to an SC932 or SC933.*
- Select Velocity Block - Serial Command
- Select Medium
- Select a file name

You will see a message indicating that parameters, based upon your choices, are being downloaded to the controller. After the parameters have been downloaded, select **Yes** to the question "Do an NVSAVE now?" and press ↵. The drive status display should now show a steady 0 for configured and not enabled.

To verify that the set up procedure worked, cycle control AC power. The status LED should repeat its power up message and then return a steady 0 . If it is still alternating U C, repeat the set up procedure.

The SC930 has been configured as an serial port commanded controller. The current loop has been compensated properly for the selected motor and the servo parameters have been setup to give medium response (approximately 75 Hz velocity loop bandwidth) with the unloaded motor. Additional default settings have also been made.

Enabling Drive

The controller can be enabled at this time by closing the switch between the Enable/ input (J4-6) and I/O RTN (J4-5). Once enabled, the status LED display should be an 8. The commanded motor speed will be the power up default, set to 0 during configuration. Because the parameters were saved in non-volatile memory, the controller can now be power cycled and, after power-up, be ready to run with the parameters established during this session.

WARNING

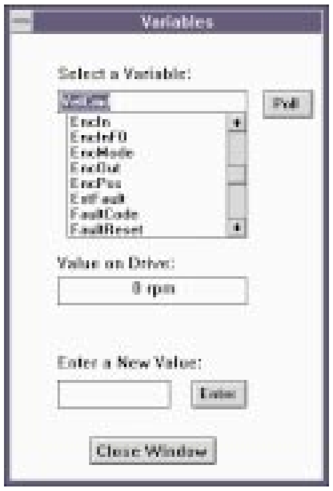
Before proceeding, the motor may need to be attached or temporarily clamped to the table or bench The inertial forces created during speed steps may make the motor hop around.

2.7 Using the Variables Window

Changing Motor Velocity

Move to the **Drive** menu with the mouse or by typing **<Alt+D>** and select the **Variables** option. The Variables window allows all parameters, variables, and commands to be examined, changed, or actuated as appropriate.

Type the proper key word name in the Select a Variable box or select one from the scrollable alphabetical list of all key words below that box. To change the shaft velocity, type **velcmd** in the Select a Variable box and press ↵. The current value of the motor velocity command in the drive will be displayed under Value on Drive.



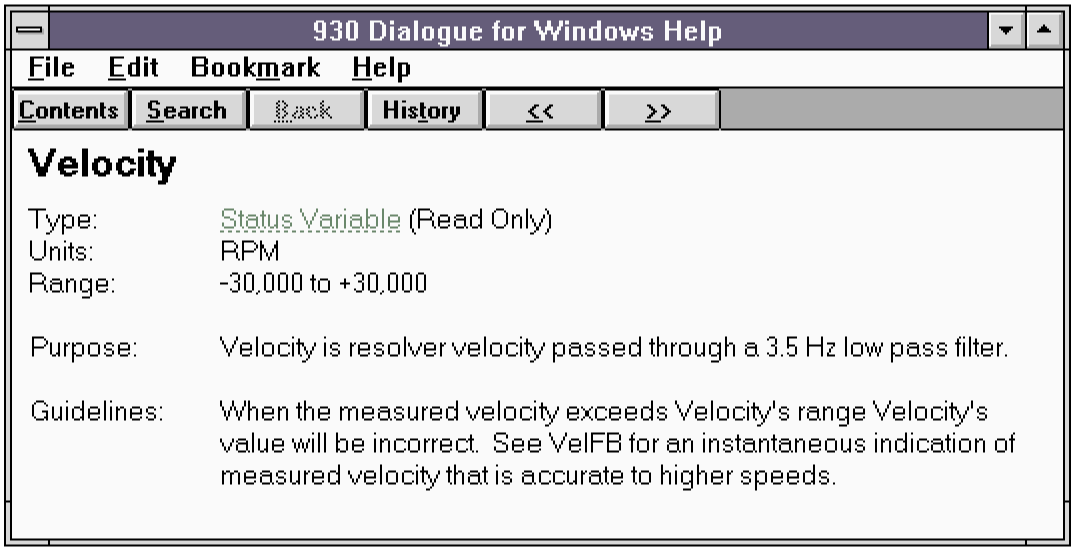
Variable name completion

Now press the <**Tab**> key or use the mouse, to get the cursor in the Enter a New Value box, enter 100, and press ↵. The motor shaft should now be spinning at 100 RPM in the clockwise direction when facing the motor shaft. To check the measured motor velocity type **veloc** in the Select a Variable box and press ↵. 930 Dialogue looked up "veloc", found it uniquely matched the key word Velocity, completed the spelling automatically, and updated the Value on Drive box.

To continuously read and update the measured velocity press the Poll button via the mouse or by using the <**Tab**> key to move the focus to that button. Pushing the button again will stop polling.

Getting Help

To get help information on a particular key word press the <**F1**> key while the cursor is located somewhere on that word in the Select a Variable box. With Velocity in that box <**F1**> should bring up the following help window.


930 Dialogue for Windows Help
File Edit Bookmark Help
Contents Search Back History << >>

Velocity

Type:	Status Variable (Read Only)
Units:	RPM
Range:	-30,000 to +30,000
Purpose:	Velocity is resolver velocity passed through a 3.5 Hz low pass filter.
Guidelines:	When the measured velocity exceeds Velocity's range Velocity's value will be incorrect. See VelFB for an instantaneous indication of measured velocity that is accurate to higher speeds.

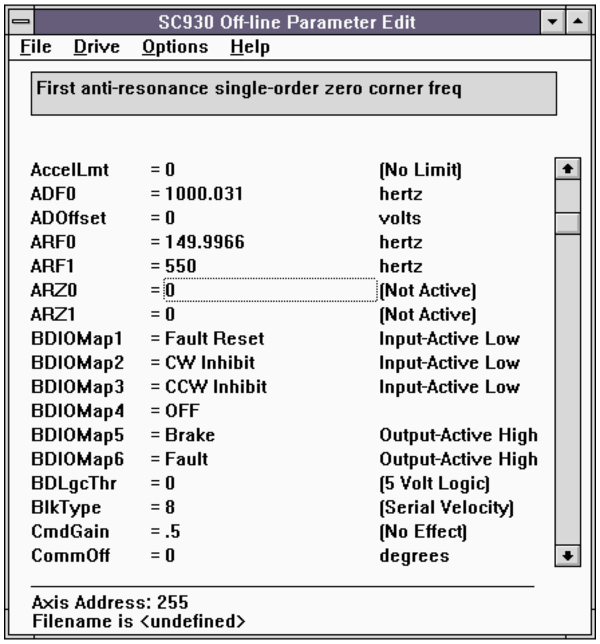
2.8 Reviewing and Editing Parameters

Close the Variables window by clicking on the **Close Window** Option or by pressing **<Alt+F4>**.

The Off-line Parameter Edit window will be displayed. The parameter values displayed are based upon selections made during Drive set up in Section 2.6. These values are stored in the PC's RAM. Changes made using the Variables Window do not change the PC's RAM copy.

Getting Help

A parameter in the PC's RAM can be edited by moving the highlighting bar to a parameter line and typing a new value followed by ↵**.** For example, move the highlighting bar to ARF1, type **550** and press ↵. The value of ARF1 changes to the new value.



First anti-resonance single-order zero corner freq

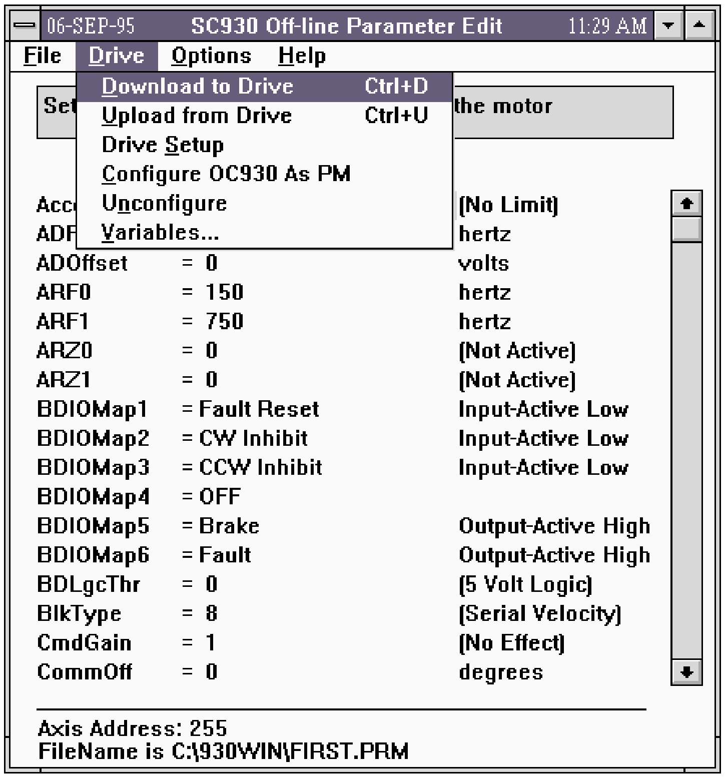
```
AccelLmt    = 0               [No Limit]
ADF0        = 1000.031        hertz
ADOffset    = 0               volts
ARF0        = 149.9966        hertz
ARF1        = 550             hertz
ARZ0        = 0               [Not Active]
ARZ1        = 0               [Not Active]
BDIOMap1    = Fault Reset     Input-Active Low
BDIOMap2    = CW Inhibit      Input-Active Low
BDIOMap3    = CCW Inhibit     Input-Active Low
BDIOMap4    = OFF
BDIOMap5    = Brake           Output-Active High
BDIOMap6    = Fault           Output-Active High
BDLgcThr    = 0               [5 Volt Logic]
BlkType     = 8               [Serial Velocity]
CmdGain     = .5              [No Effect]
CommOff     = 0               degrees
```

Axis Address: 255
Filename is <undefined>

Context sensitive help is also available in the Parameter Edit Window. Press **<F1>** to get help information about a highlighted variable. Information about all variables is available in this way.

2.9 Downloading Parameters to the 930

Changes to values on the Parameter Edit Window only affect the PC RAM copy. Changes made in the Parameter Edit Window must be downloaded to the drive in order to take affect.



Acc			[No Limit]
ADF			hertz
ADOffset	=	0	volts
ARF0	=	150	hertz
ARF1	=	750	hertz
ARZ0	=	0	[Not Active]
ARZ1	=	0	[Not Active]
BDIOMap1	=	Fault Reset	Input-Active Low
BDIOMap2	=	CW Inhibit	Input-Active Low
BDIOMap3	=	CCW Inhibit	Input-Active Low
BDIOMap4	=	OFF	
BDIOMap5	=	Brake	Output-Active High
BDIOMap6	=	Fault	Output-Active High
BDLgcThr	=	0	[5 Volt Logic]
BlkType	=	8	[Serial Velocity]
CmdGain	=	1	[No Effect]
CommOff	=	0	degrees

Axis Address: 255
FileName is C:\930WIN\FIRST.PRM

Select **Download to Drive** in the **Drive** Menu and press **<Enter>**. Select **Yes** to the question "Are You Sure?" and the parameters will be downloaded to the OC930. Also respond **Yes** to the question "Do an NVSAVE now?" This will save the parameters in the SC900's non-volatile memory.

Note: *930 Dialogue sets the drive variable Enable to 0 at the beginning of the download. To enable the drive you can use the Variables screen to set Enable = 1. If the downloaded parameters were NVSaved, cycling control AC power will also return Enable to its default value of 1.*

2.10 Uploading Parameters from the OC930

It is also possible to upload the current parameter values in the drive's RAM to the Parameter Edit Window on the PC by using the symmetric **Upload from Drive** function.

2.11 Saving Parameters on Disk

Select the **File** menu, select **Save As** and press ↵. Type the File Name **STARTING.PRM** and press ↵. The Parameter Edit window parameters will be saved on disk in a file named STARTING.PRM.

2.12 Opening a Disk File

Return to the **File** menu by pressing **<F10>** . Select the **Open** option and press ↵**.** Press the <**Tab**> key to move to the list of files. Use the arrow keys to select **STARTING.PRM** and press ↵. The Parameter-Edit screen for STARTING.PRM, that you just saved to disk, will be read from the disk and displayed. In this way, you can maintain a record of the drive's parameters. If it is ever required to make a clone, open the file in this way and use the **DownLoad to Drive** option of the **Drive** menu to download the parameters to the new drive.

2.13 Exiting 930 Dialogue

Return to the **File** menu by pressing **<F10>** and select **Exit**. Press ↵ and select **Yes** to the question "Are you SURE?" You will then exit 930 Dialogue to Program Manager.

You should now know how to start and exit 930 Dialogue, configure your drive, edit and save parameters values, and configure a driver using parameters stored in a disk file.

3 OC930 Interfaces and Connections

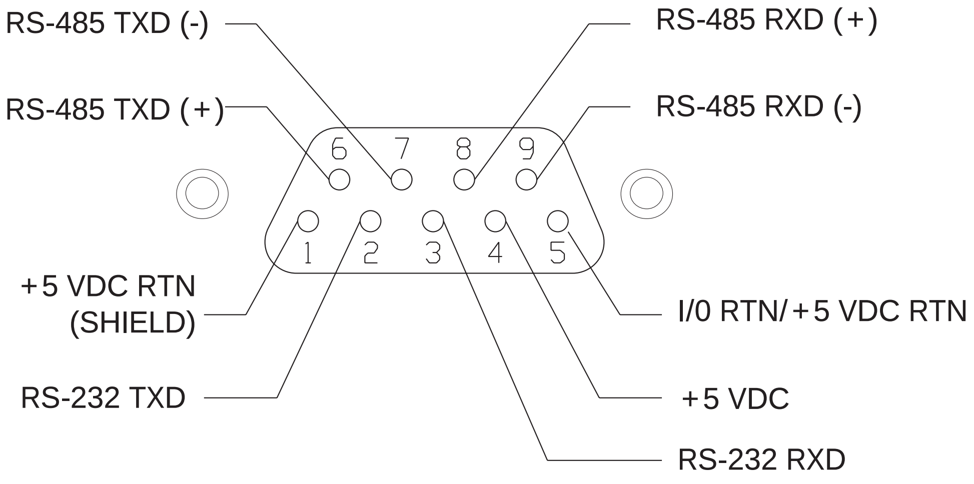
Introduction

This chapter describes the OC930's serial port (J31) and provides the information required to interface to it. This chapter also describes the serial port address DIP switch on the OC930.

3.1 Serial Port J31

The serial port (J31), utilizes the 9 contact female D subminiature style connector shown below. A brief description of each signal is included in the J31 I/O table on following page. For additional information, please refer to the OC930 Serial Communications Transceiver Schematic at the end of this chapter.

J31



I/O Table

Input/Output	Pin Number	Explanation
+5 VDC RTN/ Shield	J31-1	Common/shield -serial port interface
RS-232 TXD	J31-2	RS-232 transmitter output (from OC930)
RS-232 RXD	J31-3	RS-232 receiver input (to OC930)
+5 VDC	J31-4	+5 Vdc output (200 mA maximum between J31-4 & J4-25)
I/O RTN/+5 VDC RTN	J31-5	Common serial port interface
RS-485 TXD (+)	J31-6	RS-485 transmitter output (from OC930)
RS-485 TXD (-)	J31-7	
RS-485 RXD (+)	J31-8	RS-485 receiver input (to OC930)
RS-485 RXD (-)	J31-9	

The information provided in this section should be used to connect the SC930 to your computer for use with 930 Dialogue (due to the intelligent communications protocol utilized**, it is not possible to operate the OC930 with a dumb terminal**). Two communication links are available, RS-232 and RS-485. RS-485 allows a single computer to communicate with up to 32 SC930s in multi-axis configurations. A DIP switch on the OC930 selects the communications address for RS-485 communication. 930 Dialogue defaults to communicate with axis 255 upon start up.

3.2 Setting Up Serial Addresses using Switch S1

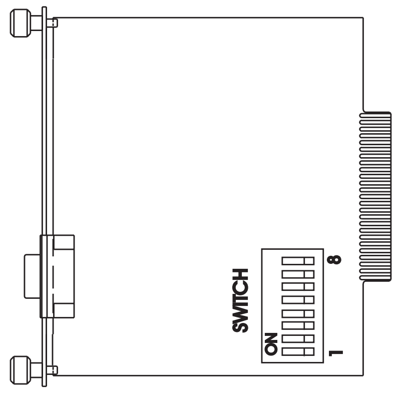
Definition

The S1 switch sets the communication address for each OC930. The ability to select different addresses is used with RS-485 for multi-drop communications.

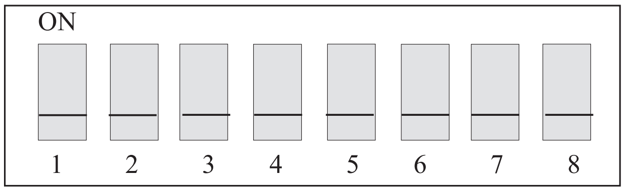
Procedure

Looking down at the top of the OC930, the following diagram shows the location of switch S1.

Note: *Each SC900 subsystem connected to a multi-drop master must have a unique serial address.*



The diagram below shows the S1 default switch settings. The OC930 factory default address is 255.



The switches are:

- On in the up position (away from number)
- Off in the down position (toward number)

Note: *When using RS-232 communications, it is recommended to leave the address set at 255.*

Procedure

To change the OC930 Address:

1. Remove power from the SC900 drive.
2. Remove the OC930 from the drive
3. Refer to the table below to set the SC900 to the appropriate address.

Address	1	2	3	4	5	6	7	8
0	On	On	On	On	On	On	On	On
1	Off	On	On	On	On	On	On	On
2	On	Off	On	On	On	On	On	On
3	Off	Off	On	On	On	On	On	On
4	On	On	Off	On	On	On	On	On
5	Off	On	Off	On	On	On	On	On
6	On	Off	Off	On	On	On	On	On
7	Off	Off	Off	On	On	On	On	On
8	On	On	On	Off	On	On	On	On
9	Off	On	On	Off	On	On	On	On
10	On	Off	On	Off	On	On	On	On
	⋮							
250	On	Off	On	Off	Off	Off	Off	Off
251	Off	Off	On	Off	Off	Off	Off	Off
252	On	On	Off	Off	Off	Off	Off	Off
253	Off	On	Off	Off	Off	Off	Off	Off
254	On	Off	Off	Off	Off	Off	Off	Off
255*	Off	Off	Off	Off	Off	Off	Off	Off

*(factory default) recommended for RS-232 operation

4. Re-connect power to the SC900 drive.
5. Repeat steps 1 through 4 for any other units on the bus. Make sure to give each unit a unique address.

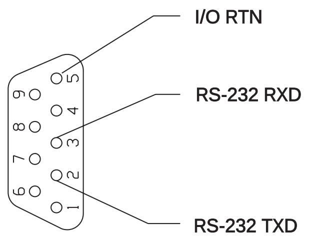
RS-232 Connections

RS-232 connections on J31 are shown below. Cable wiring required for connecting to either 9 or 25 pin serial ports of most computers are also shown.

Note: *Pinouts vary among computer manufacturers. Check the hardware reference manual for your machine before wiring.*

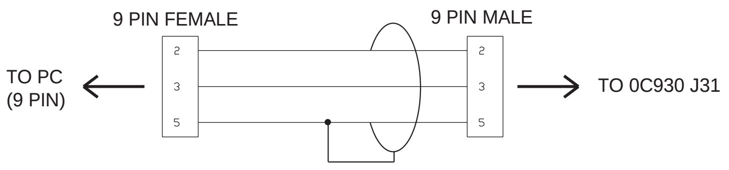
Cabling diagram

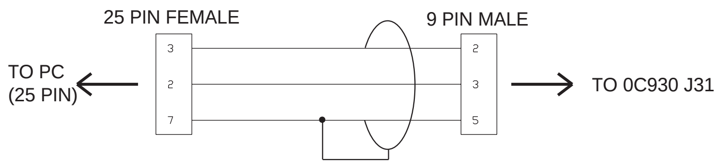
A 6 foot (1.8 m) RS-232 Cable with 9 pin connectors and a 9 pin to 25 pin adapter is available from Pacific Scientific. The Pacific Scientific order number is RS-232-750.



9 CONTACT FEMALE D SUBMINIATURE CONNECTOR

TYPICAL CABLES





Note: *Shielded wiring is recommended for the serial communications cable to minimize potential errors from electrical noise.*

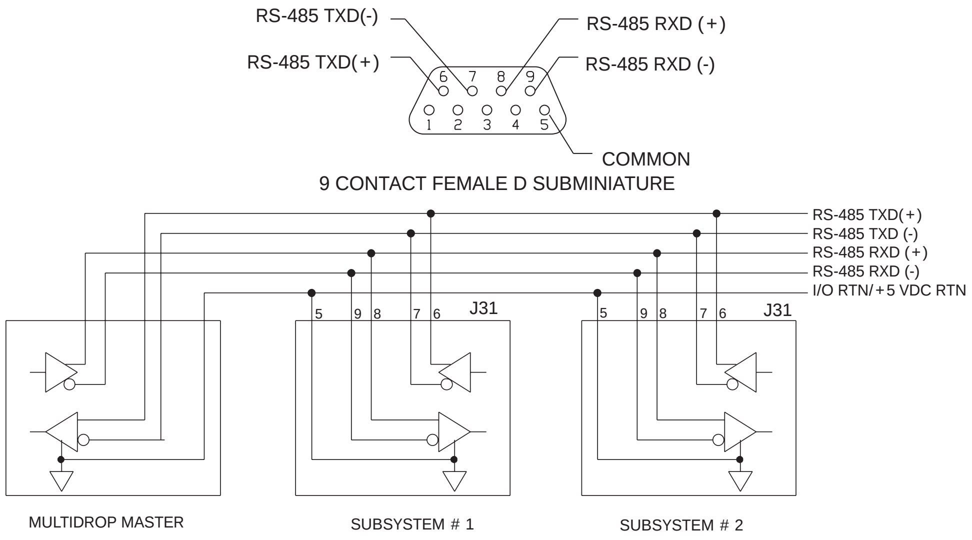
RS-485/RS-422 Connections

Up to 32 OC930s can be connected in parallel to a multidrop master. The OC930s must each have a unique address, set using switch S1 as described above. Once the address is set, the **Axis Selection** function in 930 Dialogue must be used to select the designated axis address. Then, either the RS-232 or the RS-485 link can be used to communicate with the selected axis.

For example, the RS-232 link can be used to completely setup and test an individual axis before connecting it into the multi-axis configuration.

RS-485/RS-422 connections to J31 are shown below. A multidrop interconnection diagram, showing multiple axes connected to a single host is also shown.

Connection diagram

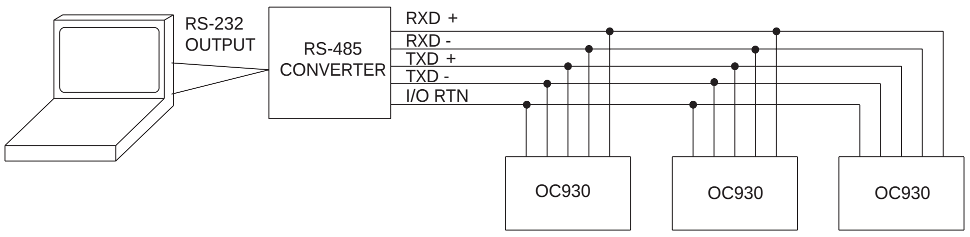


RS-232/RS-485 converter installation

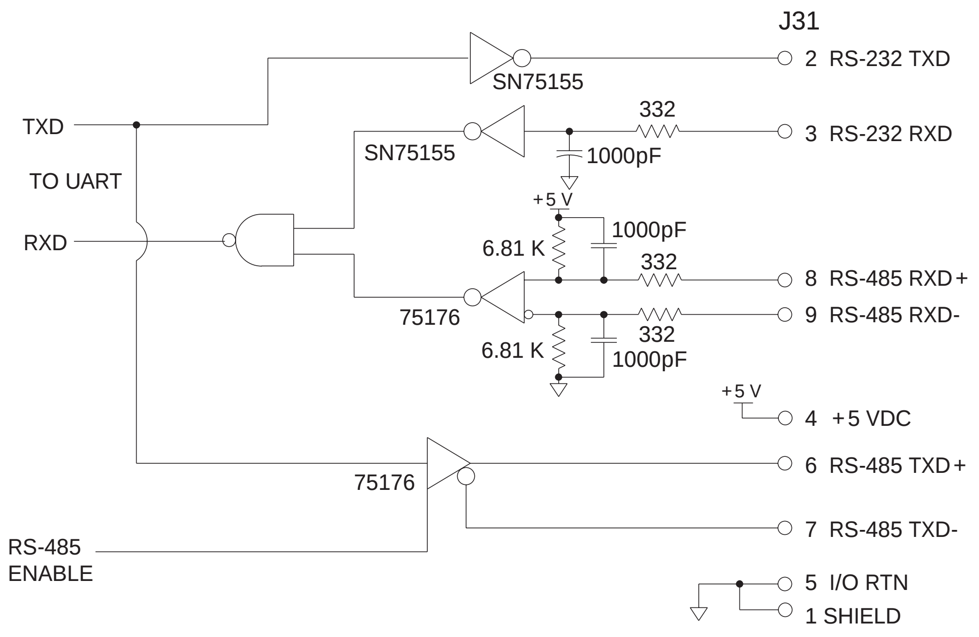
It is often convenient to use an RS-232 to RS-485/RS-422 converter so that an RS-232 port, available on all PCs, can be used to connect to multiple axes. The figure below shows a typical installation, using the B & B Model 422 RS-232 to RS-422 adapter. RS-232 to RS-485/RS-422 adapters are available from many sources.

Note: *An adapter can be powered from the serial port +5 Vdc output on J31-4 as long as the load current on J31-4 and J4-25 both total less than 200 mA.*

Installation diagram



OC930 Serial Communications Transceiver Schematic



4 Selecting Motor Control Functionality

Introduction

The SC900 family has three distinct modes of controlling the motor shaft and three distinct sources for the shaft command:

Modes

- Torque Control
- Velocity Control
- Position Control

Commands

- Analog Command
- Incremental Digital Pulse Command
- Serial Port Command with OC930

The SC930 implements seven of the nine possible combinations from the above list. The eighth possibility, Serial Port command torque block, is implemented indirectly and the ninth possibility, analog command position block is not implemented. The `BlkType` parameter sets most of the overall drive functionality. But, many other parameters need to be set to insure smooth and proper operation and this chapter goes over these requirements.

Within these eight combinations of mode and command, additional parameters allow further specialization. For example, `BlkType` = 1 (analog command velocity block) can be further enhanced to be an emulation of a superior performance clutch brake with a number of features. See section 4.2.3 for further details on this example.

Most of the drive's operating modes can be easily set up using **New Set Up** under the File menu or **Drive Set Up** under the Drive menu of 930 Dialogue. The others may be set up by using the **Variables** screen to change parameters or by changing appropriate entries in the parameter edit form and down loading the new configuration. The following sections give the details on these eight operating modes.

Refer to Appendix C for control block diagrams. Refer to Chapter 6 for additional information on velocity and position loops.

4.1 Torque Block Modes

4.1.1 Analog Command Torque Block (BlkType = 0)

This mode allows the differential analog voltage between terminals J4-1 and J4-2 to set the motor's terminal torque current amplitude. Since the actual motor current amplitude (`IFB`) times the motor's 0-peak line-line torque constant K_T times $\sqrt{3}/2$ is the shaft torque, then the analog input directly controls motor shaft torque. The easiest way to set up this mode is to select the **Torque Block - Analog Command** option when doing **New Set Up** under the File menu or **Drive Set Up** under the Drive menu of 930 Dialogue. The overall gain of this block, i.e. the output current amplitude in amps per input volt, is set by the `CmdGain` parameter directly in Amp/V and should be set by the user to the desired value.

Command processing

Figure 1 in Appendix C shows the analog torque block mode has the same signal processing as a velocity loop except that the velocity error signal (`VelErr`) is set to `VelCmdA` not to (`VelCmdA - VelFB`) and that the `VelCmd` clamp is bypassed. Thus, the analog input goes through a number of signal processing steps before becoming the motor torque current command `ICmd`.

1. Analog differential amplifier with 1200 Hz low pass filter.
2. High resolution A/D sampled at the velocity loop update rate and added to the `ADOffset` parameter.
3. `ADF0` adjustable low pass filter to become `AnalogIn`.
4. Bypass the `VelLmtHi`, `VelLmtLo` clamp.
5. d/dt slew limit clamped by the `AccelLmt`, `DecelLmt` parameters to become the `VelCmdA` variable.
6. Velocity error variable `VelErr` is set equal to `VelCmdA`.
7. The anti-resonance second order velocity loop compensation block controlled by the `ARF0, ARF1, ARZ0`, and `ARZ1` parameters to become the `FVelErr` variable.
8. The proportional and integral velocity loop compensation block controlled by the `KVP` and `KVI` parameters respectively.

9. And finally through the `IlmtPlus` and `IlmtMinus` current command clamp to become the `ICmd` motor torque current command variable.

Although this looks like a large amount of processing, the options are only there to allow tailoring the response to fit a particular application. Typically, most of the signal blocks are set to directly pass the signal so that `ICmd = CMDGain*(AnalogIn)` as directly as possible. The set of parameters below accomplish this result and are the values set by 930 Dialogue during Analog Torque block set up.

`AccelLmt`	= 0 (no Accel limiting)
`DecelLmt`	= 0 (no Decel limiting)
`ADF0`	= 100,000 Hz to bypass, 1000 Hz by auto set up
`ARF0`	= 100,000 Hz
`ARF1`	= 100,000 Hz
`ARZ0`	= 0 (not active)
`ARZ1`	= 0 (not active)
`KVP`	= 1 A/rad/sec
`KVI`	= 0 Hz

Important

The `KVP` parameter must be set to 1 A/rad/sec for the units on `CmdGain` to be correct. If `CmdGain` is set to 1 Amp/V and `KVP` to 2 A/rad/sec then an analog input of 1 volt will incorrectly give 2 amps of output torque current amplitude.

When changing the `BlkType` from something else to 0 to get an analog torque block you will generally need to additionally set `KVP` to 1, `KVI` to 0, and the other items in the above list to appropriate values to get the system working as desired.

4.1.2 Digital Frequency Command Torque Block (BlkType = 4)

This mode is the same as the analog command torque block mode except that the command input comes from the Incremental Position Command inputs on J4-21 through J4-24. The frequency on these input terminals is the variable `EncFreq` and is substituted for the `AnalogIn` input to the `CmdGain` scaling. The units on `CmdGain` for `BlkType` = 4 become Amp/kHz. For this mode to work the additional parameters `EncMode` and `EncInF0` must be set appropriately.

4.1.3 Serial Port Command Torque Block (BlkType = 0)

This mode requires the OC930 to work. It is not possible to directly command the current over the serial port. But, you can use the analog command torque block mode to virtually implement it. First, set `VelCmdSrc` = 1, which sets `VelCmd` = `VelCmd2`. Then, send the desired torque current command to `VelCmd2`. If `KVP` = $60/2\pi$ Amp/rad/sec then `VelCmd2` = 1 RPM would command 1 Amp.

4.2 Velocity Block Modes

4.2.1 Analog Command Velocity Block (BlkType = 1)

This mode allows the differential analog voltage between terminals J4-1 and J4-2 to set the motor's shaft velocity, also informally known as shaft speed. The easiest way to set up this mode is to select the **Velocity Block - Analog Command** option when doing **New Set Up** under the **File** menu or **Drive Set Up** under the Drive menu of 930 Dialogue. The overall gain of this block, i.e. the output shaft velocity per input volt, is set by the `CmdGain` parameter in kRPM/V and should be set by the user to the desired value.

Command processing

The analog input goes through a number of signal processing steps before becoming the actual motor velocity command `VelCmdA` as shown by Figure 1 in Appendix C.

1. Analog input differential amplifier with 1200 Hz low pass filter.
2. High resolution A/D sampled at the velocity loop update rate and added to the `ADOffset` parameter.
3. `ADF0` adjustable low pass filter to become the `AnalogIn` variable.
4. Range clamped by the `VelLmtHi, VelLmtLo` parameters.
5. d/dt slew limit clamped by the `AccelLmt, DecelLmt` parameters to become the `VelCmdA` variable.

Velocity loop compensation

The actual velocity command (`VelCmdA`) is then combined with the measured shaft velocity (`VelFB`) and processed by the velocity loop compensation to create the motor torque current command (`ICmd`). The detailed signal processing steps to create ICmd are listed below and shown in Figure 2 in Appendix C.

1. `VelErr` set equal to (`VelCmdA - VelFB`)
2. The anti-resonance second order velocity loop compensation block controlled by the `ARF0, ARF1, ARZ0`, and `ARZ1` parameters to become the `FVelErr` variable.
3. The proportional and integral velocity loop compensation block controlled by the `KVP` and `KVI` parameters respectively.
4. And finally through the `IlmtPlus` and `IlmtMinus` current command clamp to become the `ICmd` motor torque current command variable.

Although this looks like a lot of parameters to set, the automated set up capability of 930 Dialogue usually sets all of them properly. The only parameters that typically need user adjustment are the desired block gain (`CmdGain`) and the velocity loop tuning gain (`KVP`). For more information on tuning the velocity loop see Chapter 6 Servo Loop Parameters.

4.2.2 Digital Frequency Command Velocity Block (BlkType = 5)

This mode is the same as the analog command velocity block mode except that the command input comes from the Incremental Position Command inputs on J4-21through J4-24. The frequency on these input terminals is the variable `EncFreq` and is substituted for the `AnalogIn` input to the `CmdGain` scaling. The units on `CmdGain` for `BlkType` = 5 become kRPM/kHz. For this mode to work the additional parameters `EncMode` and `EncInF0` must be set appropriately.

4.2.3 Serial Port Command Velocity Block (BlkType = 8)

This mode requires the OC930 to operate. It is the same as the analog command velocity block mode except that the command input is the value of `VelCmd` set over the serial port. Note that `VelCmd` is a non-volatile parameter and when the SC900 base servo powers up in `BlkType` = 8 the initial value of the velocity command is this non-volatile value. Changing `VelCmd` over the serial port then sets a new volatile velocity command. Changing the non-volatile velocity command requires the additional step of issuing the `NVSave` command.

Clutch Brake Example

One useful variation of this mode is to implement a simple emulation of a mechanical clutch brake. The procedure below lists the steps required to set up this mode. Once set up, clutch brake emulation does not require the OC930 to operate.

1. Set the non-volatile value of `VelCmd` to the desired run speed, including direction, in RPM.
2. Set `VelCmd2` to 0.
3. Set the mappable input function `VelCmdSrc` to one of the `BDIO` discrete inputs.
4. Set `AccelLmt` and `DecelLmt` to the desired clutch activation acceleration and brake activation deceleration respectively.

The `VelCmdSrc` function selects between the normal source and the non-volatile parameter `VelCmd2` for `VelCmd`. So, using the clutch brake emulator is as simple as setting the `VelCmdSrc` `BDIO` to the active state to select `VelCmd2` for the brake state and `VelCmdSrc` to inactive for the clutch/run state.

Clutch brake example (cont'd)

This emulation allows much faster transition times between braking and running or visa versa than mechanical clutch brakes, has significanty longer life due to near zero mechanical wear, clutch/brake mode transitions are consistently the same, and the run speed regulation is often much better. The only disadvantage to this mode is that in the brake mode even though the velocity command is digitally 0 there can be a very small offset that could lead to drift. For example, using the drive's worst case digital offset drift of 0.00005 RPM, the shaft would only move 1 degree per hour, or only 0.02 mechanical degrees in one minute. When the drive needs to be out of the clutch activated mode for extended time periods, usually the drive would be disabled to insure no powered drift.

Note that this example could be used with the digital frequency or analog command modes to allow the run speed to be externally adjusted. Furthermore, the analog command mode combined with the `VelLmtHi` and `VelLmtLo` parameters could make a clutch brake emulator with two selectable run speeds such as forward and reverse. Just set `CmdGain` and `ADOffset` so that a digital signal connected to the differential analog input is guaranteed to exceed the clamps set with `VelLmtHi` and `VelLmtLo.`

It is possible to emulate a clutch brake with no brake drift and no holding torque by using RunStop instead of VelCmdSrc. RunStop disables the drive automatically once shaft speed decelerates to 0 or StopTime seconds after RunStop goes to stop the drive disables independent of speed.

See `BlkType` = 2 for a clutch brake emulation with zero drift in brake state with full position holding torque.

4.3 Position Block Modes

4.3.1 Digital Command Position Block (BlkType = 2)

This mode is just a velocity block mode with the VelCmd coming from the position loop. See Figure 3 in Appendix C. In particular,

`PosError = PosCommand` - Position Feedback

$$VelCmd = 2\pi * KPP * PosError + \frac{KVFF}{100} * \left[\frac{d}{dt}(PosCommand)\right]$$

where:

`PosCommand` is the position command in rad

`KPP` is the proportional position loop gain in Hz

`KVFF` is the velocity feed forward gain percentage

`VelCmd` is the net velocity command in rad/sec.

To finish setting this mode up `EncMode` and `EncF0` must be set to get the command working and `PulsesIn` and `PulsesOut` must be set to the desired block gain. The easiest way to set up this mode is to select either the **Position Block - Step and Direction** or **Position Block - Electronic Gearing** options when doing **New Set Up** under the File menu or **Drive Set Up** under the Drive menu of 930 Dialogue. See Chapter 5 “Setting Parameters for Electronic Gearing” for additional details on this mode.

When the SC900 is disabled and `BlkType = 2`, `PosCommand` is set to the position feedback value. This insures that when the drive is enabled, it picks up motion from its present position.

4.3.2 Serial Command Position Block (BlkType = 2)

This mode is identical to the digital command position block type except that the `EncMode` parameter is set to 3 to hold the input `EncPos` variable and the desired position command is sent over the serial port as the `PosCmdSet` variable.

CAUTION

Use this mode with extreme care. It is easy to change `PosCommand` by huge amounts via the `PosCmdSet` variable and this change will result in the motor flying off at maximum speed for extended periods of time which could be very dangerous. `PosCmdSet` should only be changed in small incremental amounts to perform a position move.

Zero Drift Clutch Brake

With `BlkType` = 2, `VelCmdSrc` switches differentiated PosCommand between the normal `EncIn` gearing command and `VelCmd2`. For detailed information, please see the Position Control Block Diagram in Appendix C.

To set up this form of clutch brake emulation, set `EncMode` = 3 and `VelCmd2` to the desired runspeed. With `VelCmdSrc` active, the motor runs at `VelCmd2` (clutch active state), and with `VelCmdSrc` inactive, the motor is in a position hold (brake state). The velocity trajectory between clutch and brake states is determined by the control loop dynamics and the accel/decel possible at maximum torque. For example, setting `KVFF` = 100 will always transition at maximum torque while setting KVFF = 0 the transition is controlled by `KPP`.

Note: *For Position Control modes (`e.g.` `BlkType` = 2), `AccelLmt` and `DecelLmt` no longer limit the commanded velocity slew rate. `AccelLmt` and `DecelLmt`* still limits *`VelCmdA`, but for position loops, `VelCmdA` should be determined by the position loop alone. Unless `AccelLmt` and `DecelLmt` are turned off with `BlkType` = 2, there will be unacceptable overshoots in the shaft motion.*

Page empty

5 Setting Parameters for Electronic Gearing

Introduction

This chapter provides procedures for setting up the SC900 for use as either as an electronic gearing slave with a master reference encoder or for use with a stepper indexer that generates step and direction signals.

Both electronic gearing and stepper emulation modes are based on configuring the SC900 with a position loop, setting the command scale factor via the `PulsesIn, PulsesOut, PulsesFOut` parameters, and selecting the proper command pulse stream encoding format. Figure 3 in Appendix C shows the command scaling and position loop block diagram.

The setup for either mode of operation is very straightforward using the **New Setup** selection from the File menu or the **Drive Setup** selection from the Drive menu.

5.1 Slaving the SC900 to a Master Encoder

The following procedure describes how to use 930 Dialogue to easily set up your SC900 for use as an Electronic Gearing Slave with a Master Reference encoder.

Procedure

1. Select **New Setup** from the Files menu.
2. Select **Automatic** for the Set Up Mode.
3. Enter your motor part number and press **<Enter>.**
4. Select your drive model number.
5. Select the type of line voltage, if applicable.
6. Select **Position Block-Electronic Gearing** on the Mode of Operation screen.

Procedure (cont'd)

7. Enter the number of motor resolver counts (1/65536 of a rev) that you want the motor to move for the specified number of input encoder quadrature counts. For example:

Example

If the input encoder line count is 2000 (8000 quadrature counts per encoder revolution) and the motor should make one revolution for every three encoder revolutions, then:

1 rev = 65,536 resolver counts per

3 revs = 24,000 encoder counts.

Because 65,536 is greater than the maximum value for PulsesOut, divide both numbers by four. This gives us 16,384 motor resolver counts for 6,000 input encoder counts.

Note: *If the calculated number of motor resolver counts is a non-integer value, use* `PulsesFOut` in addition to `PulsesOut`.

8. Hit **OK** and then make your bandwidth selection.

9. Enter the filename you would like to call your new parameter file.

10. Download the parameter set to the SC900 using the **Download to Drive** selection on the **Drive** drop-down menu.

At this point, when the drive is enabled, it will act as an electronic gearing slave and move relative to the enable time starting position.

The Position Block Electronic Gearing setup mode sets the following parameters to the values listed below:

`BlkType` = 2	Digital Position
`EncMode` = 0	Quadrature Decode
`Kvff` = 0	No velocity feed forward
`KPP` = BW selected	
`PulsesOut` = Resolver Counts entered	
`PulsesIn` = Encoder Counts entered	

Note: *In many electronic gearing applications, following error (position loop null error proportional to speed) with Kvff = 0 is a problem. The following error can be eliminated by setting Kvff to 100%. See Section 6.2 for more information of Kvff.*

5.2 Controlling the SC900 with a Stepper Indexer

The following procedure describes how to use 930 Dialogue to easily set up your SC900 for use with a stepper indexer.

Procedure

1. Select **New Setup** from the Files menu.
2. Select **Automatic** for the Set Up Mode.
3. Enter your motor part number and press **<Enter>.**
4. Select your drive model number.
5. Select the type of line voltage, if applicable.
6. Select **Position-Block-Step and Direction** on the Mode of Operation screen.
7. Specify the number of steps per motor revolution. This number must be evenly divisible by four. For industry standard step sizes, select from the table below.

1.8°/Full Step	**Steps/Rev**
Full	200
Half	400
1/5	1000
1/10	2000
1/25	5000
1/125	25000
1/250	50000

8. Hit **OK** and then make your bandwidth selection.
9. Enter the filename you would like to call your new parameter file.
10. Download the parameter set to the SC900 using the **Download to Drive** selection on the **Drive** drop-down menu.

At this point, when the drive is enabled, it can be controlled by a stepper indexer feeding it with step and direction signals.

The Position Block Step and Direction setup mode sets the following parameters to the values listed below:

`BlkType` = 2	Digital Position
`EncMode` = 1	Step & Direction
`Kvff` = 0	No velocity feed forward
`KPP` = BW selected	
`PulsesOut` = 16834 Resolver counts	
`PulsesIn` = (steps/motor rev)/4	

Step and direction applications are typically point to point positioning and `Kvff = 0` works fine. Crisper motion profile "corners" are possible when `Kvff` is properly adjusted in the rage (typically) of 70-80%. See Section 6.2 for additional information on `Kvff.`

5.3 Converting From Another Mode

It is not uncommon to start a system as a velocity block to either check out system mechanics or to optimize the velocity loop tuning. If this is the case, it is not required to go back and do a New Setup, as described in the previous two sections, to get the SC900 configured to do positioning. By using the Variables screen, the specific parameters listed at the end of each section can be set.

Procedure

To set the values of the parameters, use the following procedure while the drive is powered and disabled:

1. Set proper values for each of the six parameters listed.
2. NVSave the parameter set in the drive.
3. Upload the parameter set to the PC and save to a file for future reference.

5.4 Turning Gearing On and Off

The `VelCmdSrc` mappable BDIO input function allows gearing to be turned on or off via a digital input. If `VelCmd2` = 0 and `VelCmdSrc` is active, then the drive does a position hold. Setting `VelCmdSrc` to inactive returns the drive to normal gearing. See the Control Block Diagram in Appendix C.

When combined with an external computer or PLC, this function can also be used for a crude homing routine. Connect the home switch to the BDIO pin mapped to `VelCmdSrc` and set `VelCmd2` to the homing velocity.

6 Servo Loop Parameters

Introduction

This chapter describes setting parameters associated with the velocity and position loops. In many cases, satisfactory operation is achieved using **New Set Up** or **Drive Set Up** menu selections. However, in some cases the user must adjust control loop parameters due to large mismatches between motor and load inertia, mechanical resonances, backlash, etc. This chapter provides guidance for handling these situations.

Chapter 4 should be reviewed for a description of the control loop architecture. Refer to Appendix C for control loop block diagrams.

Note: *The two anti-resonant zeroes (`ARZ0` and `ARZ1`) are assumed to both be off (set to zero) for this discussion.*

6.1 Velocity Loop

The velocity loop block diagram is shown in Figure 2 of Appendix C. Velocity loop bandwidth is the key indicator of system performance. Systems with fast settling time must have high velocity loop bandwidth. Conversely, if the velocity loop bandwidth is low, attempting to achieve fast settling time by increasing the position loop bandwidth, `KPP`, leads to overshoot and ringing.

Velocity loop bandwidth

The velocity loop bandwidth (f_{vc}) is given by the equation:

$$f_{vc}(Hz) = \frac{KVP * K_T \sqrt{3}/2}{2\pi * J_{TOT}} \approx 0.138 * KVP * \frac{K_T}{J_{TOT}}$$

where:

`KVP` is the velocity loop proportional gain in amps/(rad/sec)
K_T is the 0-peak line-line motor torque constant in lb-in/amp
J_{TOT} is the total inertia (motor total + load total) in lb-in-sec^2.
(Any consistent set of units for K_T, J_{TOT}, such as MKS, that yields K_T/J_{TOT} in rad/sec^2/amp will work)

The motor torque constant is the value of K_T peak published in the Pacific Scientific Motion Control Solutions catalog.

Note: *f_{vc} is the unity gain open-loop crossover frequency of the idealized rigid single mass system. See hardware specifications for maximum f_{vc} value.*

Default bandwidths

The **New Setup** and **Drive Setup** utilities set KVP to achieve the velocity loop bandwidths shown below, assuming there is no load on the motor shaft and the motor has no mechanical brake or other secondary devices installed.

Note: *The bandwidth depends upon the user's selection for desired system response:*

	Gentle	**Medium**	**Stiff**
f_{vc} Velocity Loop Bandwidth (Hz)	25	75	200

Load inertia

From the formula for bandwidth, it is seen that bandwidth changes inversely with total inertia. If the load inertia equals the motor plus resolver inertia, the velocity loop bandwidth will be half the values shown. If the load inertia is ten times the motor plus resolver inertia, the bandwidths will be one eleventh these values. Clearly `KVP` must be increased to compensate for increased load inertia if bandwidth is to be maintained. Typically, load inertias up to 3(motor + resolver) give acceptable performance without further optimization.

The most common servo setup problem is adding large load inertia without a corresponding increase in KVP.

The value of `KVP` to achieve a desired bandwidth can easily be calculated as follows:

$$KVP = \frac{2\pi * f_{vc} * J_{TOT}}{K_T \sqrt{3}/2} \approx 7.26 * f_{vc} * \frac{J_{TOT}}{K_T}$$

Example calculation

For example, to achieve 75 Hz bandwidth with an R32G motor having 20 to 1 load inertia = 0.011 lb-in-sec^2:

J_{TOT}[1] = 0.00055 + 0.011 = 0.01155 lb-in-sec^2

K_T[2] = 4.4 lb-in/amp

$$KVP = 7.26 * 75 * \frac{0.01155}{4.4} = 1.43$$

1 Motor plus resolver inertia (0.00055 lb-in-sec^2) for the R32G motor can be found in the catalog or 930 Dialogue's motor data screen.

2 K_T can be found in the catalog as K_T peak (4.4 lb-in/amp) or by using the Back EMF Constant, K_E (52.0 Volts/kRPM) shown on 930 Dialogue's motor data screen in the following formula: $K_T = 0.084*K_E$ (volts/krpm).

930 Dialogue can also be used to make the calculation. Simply enter the total inertia in place of the motor plus resolver inertia when using the **New Setup** or **Drive Setup** utilities and 930 Dialogue will calculate the appropriate value for `KVP` to achieve 25, 75 or 180 Hz bandwidth depending upon the choice made for system response.

There is no specific answer to the general question “What should the bandwidth be?” In general, the higher the velocity loop bandwidth, the faster the settling time will be and the better the rejection of torque disturbances (increased stiffness). Typically, velocity loop bandwidths range from 30 to 100 Hz. However, too high a bandwidth can lower the damping of resonances in mechanical linkages, causing excessive ringing and/or wear in coupled mechanics. Remember, it is the resulting motion at the end of any mechanical linkages that typically matters, not the response at the motor shaft.

Problems with high load inertia

It would seem from the above that setting `KVP` is simply a matter of increasing its value to compensate for load inertia. Unfortunately, the following problems often interfere, particularly when the load inertia is large compared with the motor’s inertia:

1. Mechanical resonances between motor and load cause high frequency oscillation.
2. Backlash between motor and load effectively unload the motor over a small angle. Within this small angle the increased bandwidth results in oscillations.
3. Ripple in the velocity feedback signal results in large motor ripple current if `KVP` is large.

As a general rule, any system with `KVP` set higher than 5 times the medium bandwidth setting will require adjustment to the default `ARF0` and `ARF1` settings.

Resonances

Mechanical resonances are caused by springiness between motor inertia and load inertia. This may result from belts, flexible couplings, or the torsional stiffness of shafts. **In general, the stiffer the couplings, the higher the resonance frequency and the easier it is to tune the system for good performance.**

If the velocity loop breaks into an oscillation at a frequency well above the calculated velocity loop bandwidth, a resonance problem may well exist. A second symptom is that the frequency of oscillation is relatively constant in the presence of changes to `ARF0` and `ARF1`.

ARF0 & ARF1

Two digital anti-resonant low-pass filters `ARF0` and `ARF1` are included in the velocity loop. Their purpose is to lower the gain above f_{vc} and especially at any resonant frequency > f_{vc} so that oscillations do not occur. Default values, also a function of the selected system response, are shown below:

	Gentle	**Medium**	**Stiff**
ARF0 (Hz)	100	150	1500
ARF1 (Hz)	200	750	$1x10^5$

If the velocity loop bandwidth cannot be raised to an acceptable value without encountering a resonant oscillation, the procedure on the following page is recommended.

Procedure

1. Set both `ARF0` and `ARF1` to 400 Hz and set `KVP` low enough to prevent oscillation.
2. Increase `KVP` slowly until oscillation at the resonant frequency just begins. Then, reduce `KVP` slightly until the oscillation just stops. Compute the velocity loop bandwidth using the formula given at the beginning of this section. If the velocity loop bandwidth is less than .25 times the value of `ARF0` and `ARF1`, then proceed to Step 3. Otherwise, go to Step 4.
3. Decrease both `ARF0` and `ARF1` by 20% and go back to Step 2.
4. The velocity loop bandwidth should now be approximately one quarter the value of `ARF0` and `ARF1`. For margin, reduce `KVP`, `ARF0`, and `ARF1` by 20%.

Backlash

Some backlash may be unavoidable, especially when gear reduction is used. If backlash is present, the inertia match must be good (load inertia should be roughly equal to motor inertia) for good servo performance. Gearing reduces the inertia reflected to the motor by the square of the gear reduction from motor to load. Therefore, select a gear ratio to give the required match.

Current ripple

The velocity feedback signal in standard SC900 Drives operating with the standard 20 arcmin resolver can have up to 3% p-p ripple. The resulting motor torque current ripple, with no `ARF0/ARF1` filtering, can be calculated using the following formula:

$$\text{Current ripple (amps p-p)} = \frac{3}{100} * \text{Speed (RPM)} * \frac{2\pi}{60} * KVP$$

$$\approx 0.003 * \text{Speed (RPM)} * KVP$$

There can be cause for concern when this p-p number exceeds 40% of the drive's or motor's current rating. The motor current should be monitored using Dac Monitors on J4-3 to insure actual ripple current, with `ARF0/ARF1` filtering, is not excessive.

Motor current ripple can often be reduced by lowering the `ARF0`, `ARF1` low-pass filter break frequencies. This benefit is limited by velocity loop bandwidth and stability constraints. Velocity feedback ripple, and hence motor current ripple, can also be reduced by specifying a higher accuracy resolver.

KVI

The parameter `KVI` sets the so called "lag-break" frequency of the velocity loop. `KVI` is equal to the frequency in Hz where the velocity loop compensation transitions from predominantly integral characteristics to predominantly proportional characteristics. Drive rejection of torque disturbances increase as `KVI` increases. Default values for `KVI` are shown below:

	Gentle	**Medium**	**Stiff**
KVI (Velocity Loop Lag-Break Freq. (Hz))	1.7	5.0	13.3

If the Drive is to be used within a position loop (either with `BlkType` = 2 or when using an external position drive and `BlkType` = 1), `KVI` should be equal to or less than 0.1 times the velocity loop bandwidth. If no position loop is used, `KVI` can be set to 0.25 times the velocity loop bandwidth (or higher if some ringing can be tolerated). In general, the response to a velocity command step (or truncated ramp) will have velocity overshoot for non-zero values of `KVI`.

6.2 Position Loop

When `BlkType` is set equal to 2, a position loop is configured outside the velocity loop described in the previous section. Figure 3 in Appendix C illustrates the structure of the position loop. **The velocity loop must be set up and evaluated in terms of bandwidth before attempting to setup the position loop.**

KPP

The position loop proportional gain, `KPP`, determines the settling time of the position loop. `KPP` is the bandwidth of the position loop, in Hz, assuming an ideal velocity loop. Default values for `KPP` are shown below:

	Gentle	**Medium**	**Stiff**
KPP (Position Loop Bandwidth (Hz))	5	15	50

In general, the higher the value of `KPP`, the faster the settling time. However, **trying to set `KPP` to a high value with inadequate velocity loop bandwidth results in overshoot and ringing.** A good trade off is to set `KPP` to 0.2 times the velocity loop bandwidth. Slightly higher values can be used if overshoot can be tolerated.

KVFF

`KVFF` is the velocity feed forward gain. In the absence of velocity feed forward (`KVFF` = 0), the commanded velocity is proportional to the position (following) error. This means that the actual position will lag the commanded position by a value proportional to the speed. The error will be smaller for larger values of `KPP`.

The following table gives a feel for the following error magnitude.

Speed (rpm)	**KPP (Hz)**	**Following Error (revolutions)**
1000	10	0.27
2000	10	0.53
5000	10	1.33
1000	20	0.13
2000	20	0.27
5000	20	0.66

Note: *The following error can easily exceed one complete motor revolution. In many electronic gearing applications, such following errors are not acceptable (real gears don't have following errors!) Also, stepper systems don't have such errors.*

Feed forward takes advantage of the fact that the SC900 DSP knows the frequency of the encoder or step inputs and hence knows how fast the motor should be going at a given instant. All or part of this velocity can be added to the velocity command to reduce following error. If `KVFF` is set to 100 (%), then the steady state following error reduces to zero.

Overshoot

Setting `KVFF` equal to 100% can result in position overshoot. Somewhat lower values may be required if this is a problem. `KVFF` set to 70%-80% typically achieves the fastest step response with no overshoot. However, setting `KVFF` to less than 100% will give steady state following error when running at constant speed.

6.3 Advanced Velocity Loop Tuning

Continuous time transfer function approximation

The transfer function for the velocity loop compensation block is given below:

$$\frac{FVelErr}{VelErr}(s) = \frac{\left(\frac{s}{w_z}\right)^2 + \frac{1}{Q_z}\frac{s}{\omega_z} + 1}{\left(\frac{s}{\omega_f}\right)^2 + \frac{1}{Q_f}\frac{s}{\omega_f} + 1}$$

$$\frac{ICmd}{VelErr}(s) = \frac{\left(\frac{s}{\omega_z}\right)^2 + \frac{1}{Q_z}\frac{s}{\omega_z} + 1}{\left(\frac{s}{\omega_f}\right)^2 + \frac{1}{Q_f}\frac{s}{\omega_f} + 1}(KVP)\left(1 + \frac{2\pi(KVI)}{s}\right)$$

Definitions for the terms used in the equations above are shown on the following page.

For ARx0 > 0 both roots are real and:

$$\omega_x = 2\pi\sqrt{(ARx0)(ARx1)}$$

$$Q_x = \frac{\sqrt{(ARx0)(ARx1)}}{Arx0 + ARx1}$$

For ARx0 < 0 roots are a complex pair and:

$$\omega_x = -2\pi ARx0$$

$$Q_x = ARx1$$

Note: *When* `ARZ0` *and* `ARZ1` *are both zero, the numerator of* $\frac{FvelErr}{VelErr}(s)$ *reduces to 1. If* `ARZ0` *or* `ARZ1` *is individually 0 the numerator reduces to* $\frac{s}{2\pi ARZx}+1$

Discrete time transfer function

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$$s \rightarrow \frac{1}{T_s}\left(1 - z^{-1}\right)$$

where $T_s = 250\ \mu$sec**.**

Page empty

7 Parameter, Variable and Command Reference

In this chapter This chapter includes a quick reference guide to all 930 parameters, variables, and commands as well as detailed descriptions of each. The quick reference indicates the page number of the detailed description.

7.1 Quick Reference

The list defines the type of entry, default value, and page number in Section 7.2 where a detailed description can be found. "NV" in the type field indicates the parameter is stored in non-volatile memory. "MF" in the type field indicates the variable is a mappable function.

Note: *A default "value" of set up indicates that the value assigned by 930 Dialogue is a result of the controller set up function.*

Name	**Type**	**Default Value**	**Page #**
AccelLmt	NV parameter (float)	set up	7-8
ADF0	NV parameter (float)	set up	7-9
ADOffset	NV parameter (float)	set up	7-10
AlnNull	MF variable (integer)	0 if not mapped	7-11
AnalogIn	variable (float R/O)		7-12
AnalogOut1	variable (float)		7-13
AnalogOut2	variable (float)		7-14
ARF0	NV parameter (float)	set up	7-15
ARF1	NV parameter (float)	set up	7-16
ARZ0	NV parameter (float)	set up	7-17
ARZ1	NV parameter (float)	set up	7-18

Table (cont'd)

Name	Type	Default Value	Page #
AxisAddr	variable (integer R/O)		7-19
BDIn*x*	variable (integer R/O)		7-20
BDIOMap*x*	NV parameter (integer)	set up	7-21
BDLgcThr	NV parameter (integer)	set up	7-22
BDOut*x*	variable (integer)		7-23
BlkType	NV parameter (integer)	set up	7-24
Brake	variable (integer R/O)		7-25
CCDate	variable (integer R/O)		7-26
CCSNum	variable (integer R/O)		7-27
CcwInh	MF variable (integer)	0 if not mapped	7-28
Cfgd	variable (integer R/O)		7-29
CmdGain	NV parameter (float)	set up	7-30
CmdGain2	NV parameter (float)		7-31
CommEnbl	variable (integer)	1	7-32
CommOff	NV parameter (float)	set up	7-33
CommSrc	NV parameter (integer)	set up	7-34
Cwinh	MF variable (integer)	0 if not mapped	7-35
DecelLmt	NV parameter (float)	set up	7-36
DM1F0	NV parameter (integer)	set up	7-37
DM2F0	NV parameter (integer)	set up	7-38
DM1Gain	NV parameter (float)	set up	7-39

Table (cont'd)

Name	Type	Default Value	Page #
DM2Gain	NV parameter (float)	set up	7-40
DM1Map	NV parameter (float)	set up	7-41
DM2Map	NV parameter (float)	set up	7-42
DM1Out	variable (float R/O)		7-43
DM2Out	variable (float R/O)		7-44
ElecRev	MF variable (integer R/O)		7-45
Enable	variable (integer)	1	7-46
Enable2	MF variable (integer)	1 if not mapped	7-47
Enabled	MF variable (integer R/O)		7-48
EncFreq	variable (float R/O)		7-49
EncIn	NV parameter (integer)	set up	7-50
EncInF0	NV parameter (float)	highest speed	7-51
EncMode	NV parameter (integer)	set up	7-52
EncOut	NV parameter (integer)	set up	7-53
EncPos	variable (integer R/O)		7-54
ExtFault	variable (integer R/O)		7-55
Fault	MF variable (integer R/O)		7-56
FaultCode	variable (integer R/O)		7-57
FaultReset	MF variable (integer)	0 if not mapped	7-59
FVelErr	variable (float R/O)		7-60
FwV	variable (integer R/O)		7-61

Table (cont'd)

Name	Type	Default Value	Page #
HSTemp	variable (float R/O)		7-62
HwV	variable (integer R/O)		7-63
ICmd	variable (float R/O)		7-64
IFB	variable (float R/O)		7-65
ILmtMinus	NV parameter (integer)	set up	7-66
ILmtPlus	NV parameter (integer)	set up	7-67
Inputs	variable (integer R/O)		7-68
Ipeak	variable (float R/O)		7-69
IR	variable (float R/O)		7-70
IS	variable (float R/O)		7-71
IT	variable (float R/O)		7-72
ItF0	NV parameter (float)	set up	7-73
ItFilt	variable (float R/O)		7-74
ItThresh	NV parameter (integer)	set up	7-75
ItThreshA	variable (integer R/O)		7-76
Kii	NV parameter (float)	set up	7-77
Kip	NV parameter (float)	set up	7-78
Kpp	NV parameter (float)	set up	7-79
Kvff	NV parameter (float)	set up	7-80
Kvi	NV parameter (float)	set up	7-81

Table (cont'd)

Name	Type	Default Value	Page #
Kvp	NV parameter (float)	set up	7-82
MechRev	MF variable (integer R/O)		7-83
Model	NV parameter (integer R/O)		7-84
Motor	NV parameter (integer R/O)	set up	7-85
NVLoad	command		7-86
NVLoadOpt	command		7-87
NVSave	command		7-88
NVSaveOpt	command		7-89
OCDate	variable (integer R/O)		7-90
OCSNum	variable (integer R/O)		7-91
Outputs	variable (integer)		7-92
PoleCount	NV parameter (integer)	set up	7-93
PosCmd Set	variable (integer)		7-94
PosCommand	variable (integer R/O)		7-95
PosError	variable (integer R/O)		7-96
PosErrorMax	NV parameter (integer)	set up	7-97
Position	variable (integer R/O)		7-98
PulsesFOut	NV parameter (integer)		7-99
PulsesIn	NV parameter (integer)		7-100
PulsesOut	NV parameter (integer)		7-101
RemoteFB	NV parameter (integer)		7-102

Table (cont'd)

Name	Type	Default Value	Page #
ResPos	variable (integer R/O)		7-103
RunStop	MF variable (integer)	1 if not mapped	7-104
StopTime	NV parameter (float)	set up	7-105
UncfgDrv	command		7-106
UncfgOpt	command		7-107
VBus	variable (float R/O)		7-108
VBusThresh	NV parameter (float)		7-109
VdCmd	variable (float R/O)		7-110
VelCmd	NV parameter (float)		7-111
VelCmdA	variable (float R/O)		7-112
VelCmd2	NVparameter (float)	set up	7-113
VelCmdSrc	MF variable (integer)		7-114
VelErr	variable (float R/O)		7-115
VelFB	variable (float R/O)		7-116
VelLmtLo	NV parameter (float)	set up	7-117
VelLmtHi	NV parameter (float)	set up	7-118
Velocity	variable (float R/O)		7-119
ZeroSpeedThresh	NV parameter (float)		7-120

7.2 Keyword Reference

Introduction

This section is an alphabetical reference to SC930 keywords. These keywords give access to:

- parameters
- variables
- commands

The name, type of each keyword, and communications protocol code is listed at the top of the page. For additional information on the Serial Communications Protocol, please refer to Appendix A.

The keyword is then described based on the following categories:

Purpose

Units

Range or Value

Default

Related Parameters/Commands

Guidelines

Note: *"NV" indicates the parameter is stored in non-volatile memory. "MF" indicates the variable is a mappable function.*

AccelLmt

(NV Parameter, Float) f276

Units	RPM/sec
Range	0 to 1 x 10^9
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0 (no acceleration limiting).
Purpose	Slew rate limit on actual velocity command magnitude increases. See `VelCmdA` for the `VelCmd` value after slew limiting.
Guidelines	Setting `AccelLmt` to 0 turns off the `AccelLmt` slew limiting; `VelCmdA` can immediately increase to equal `VelCmd`. See `DecelLmt` for control of `VelCmdA` magnitude decreases. For position loops, setting either `AccelLmt` or `DecelLmt` to a value is not recommended as it may cause excessive overshoot.

ADF0

(NV Parameter, Float) f18

Units	Hertz
Range	0.01 to 4.17 x 10^7
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1000.
Purpose	`ADF0` is the first-order low-pass filter corner frequency for the analog input channel on J4-1 to J4-2.
Guidelines	`ADF0` is the corner frequency in Hz of the single-order low-pass filter. The purpose of the filter is to attenuate the high frequency components from the digitized input signal. Decreasing `ADF0` lowers the response time to input changes, but also increases the effective resolution of `AnalogIn` by removing more circuit noise.

ADF0	**AnalogIn**	
	Effective Bits	lsb Size
Max	14	1.6 mV
150	16	0.4 mV
10	18	0.1 mV

ADOffset

(NV Parameter, Float) f19

Units	Volts
Range	-15 to +15
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0.
Purpose	`ADOffset` adjusts the steady-state value of the analog command input.
Guidelines	`AnalogIn` is equal to the differential voltage between J4-1 and J4-2 plus the `ADOffset`.

AInNull
(Mappable Input Function, Variable, Integer) i283

Range	0 or 1
Default	0 at power up if not mapped to a BDIO point
Purpose	Function to null the dc in `AnalogIn` to 0.
Guidelines	When not mapped to a BDIO, setting `AInNull` to 1 starts the nulling function by temporarily setting `ADF0` to 1 Hz. When `AInNull` goes back to 0 for normal operation `ADF0` is restored and `ADOffset` is set to old `ADOffset` minus `AnalogIn` sampled at the 1 to 0 transition. This new `ADOffset` is then stored in NV memory.

AnalogIn

(Variable, Float, Read-Only) f0

Units	Volts
Range	-13.5 to +13.5
Default	none
Purpose	`AnalogIn` (Analog input) is the digitized value of the analog input channel, which is the differential voltage of J4-1 (+) relative to J4-2 (-) after `ADOffset` is added and passed through `ADF0` low-pass filter.
Guidelines	`AnalogIn` can be monitored to check the presence and voltage of signals at the analog input terminals.

AnalogOut1

(Variable, Float) f1

Units	Volts
Range	-5 to +4.961
Default	None
Purpose	`AnalogOut1` directly sets the voltage level of DAC Monitor 1 (J4-3) when `DM1Map` = 0.
Guidelines	When `DM1Map` is not equal to 0, `AnalogOut1` is not used.

AnalogOut2

(Variable, Float) f261

Units	Volts
Range	-5 to +4.961
Default	None
Purpose	`AnalogOut2` directly sets the voltage level of DAC Monitor 2 (J4-4) when `DM2Map` = 0.
Guidelines	When `DM2Map` is not equal to 0 `AnalogOut2` is not used.

ARF0

(NV Parameter, Float) f8

Units	Hertz
Range	0.01 to 10 x 10^6
	-10 x 10^6 to -0.01
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.
Purpose	`ARF0` is the first velocity loop compensation anti-resonance low-pass filter corner frequency.
Guidelines	`ARF0` is the corner frequency, in Hz, of one of two single-order low-pass anti-resonant filters or if < 0 it is the under damped pole pair frequency in Hz and `ARF1` would be the pole pair Q. The purpose of the anti-resonant filters is to attenuate the velocity loop gain at the mechanical resonant frequency. See `ARF1, ARZ0, ARZ1` and Chapter 6 for more information.

ARF1

(NV Parameter, Float) f9

Units	Hertz
Range	0.01 to 10,000,000 1 to 100 (Q)
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.
Purpose	`ARF1` is the second velocity loop compensation anti-resonance low-pass filter corner frequency.
Guidelines	`ARF1` is the corner frequency, in Hz, of one of two single-order low-pass anti-resonant filters or if `ARF0` is < 0, then `ARF1` is the Q of the under damped pole pair. The purpose of the anti-resonant filters is to attenuate the velocity gain at the mechanical resonant frequency. See `ARF0, ARZ0, ARZ1` and Chapter 6 for more information.

ARZ0

(NV Parameter, Float) f285

Units	Hertz
Range	20 to 1 x 10^5 -1 x 10^5 to -35
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0.
Purpose	`ARZ0` is the first velocity loop compensation zero.
Guidelines	`ARZ0` is generally not needed and should be set to 0, which eliminates it entirely. For very demanding compensation schemes it can be used to add lead compensation or with `ARZ1` to add a notch filter. `ARZ0` positive sets a zero frequency in Hz and if < 0 sets an under damped zero pair frequency in Hz while `ARZ1` is the zero pair Q. See `ARF0`, `ARF1`, `ARZ1` and Chapter 6 for more information.

ARZ1

(NV Parameter, Float) f286

Units	Hertz
Range	20 to 1 x 10^6
	-100 to 100 (Q)
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0.
Purpose	`ARZ1` is the second velocity loop compensation zero.
Guidelines	`ARZ1` is generally not needed and should be set to 0 which eliminates it entirely. For very demanding compensation schemes it can be used to add lead compensation or with `ARZ0` to add a notch filter. `ARZ1` sets a zero frequency in Hz or if `ARZ0` is set < 0 then `ARZ1` sets the under damped zero pair Q. See `ARF0, ARF1, ARZ0` and Chapter 6 for more information.

AxisAddr

(Variable, Integer, Read-Only) i78

Range	0 to 255
Default	Set by hardware DIP switch on OC930 Card
Purpose	`AxisAddr` indicates the address of the drive currently selected.
Guidelines	The axis address must be set in 930 Dialogue to correspond to the address setting of the dip switch of the SC930 option card. For most applications the default setting of address 255 is recommended.

930 Dialogue provides a find axis feature which can determine the axis setting of an SC930 option card, provided proper serial communications cabling and connections between the PC and the drive have been made.

BDIn*X*

(Variable, Integer, Read-Only) i17-i22

Range	0 or 1
Purpose	`BDIn1` reads the state of BDIO1, J4-7
	`BDIn2` reads the state of BDIO2, J4-8
	`BDIn3` reads the state of BDIO3, J4-9
	`BDIn4` reads the state of BDIO4, J4-10
	`BDIn5` reads the state of BDIO5, J4-11
	`BDIn6` reads the state of BDIO6, J4-12
Guidelines	`BDInX` indicates whether `BDIOX` input voltage is above or below the logic threshold selected by the variable `BDLgcThr`.
	`BDInX` = 0 indicates a logic low input
	`BDInX` = 1 indicates a logic high input

BDIOMapx

(NV Parameter, Integer) i270-i275

Range -2,147,483,648 to 2,147,483,648

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets their values as follows:

BDIOMap1	Fault Reset Input Active Low
BDIOMap2	CW Inhibit Input Active Low
BDIOMap3	CCW Inhibit Input Active Low
BDIOMap4	OFF
BDIOMap5	Brake Output Active High
BDIOMap6	Fault Output Active High

Purpose Sets the logical function of the BDIOs on J4-7 to J4-12.

Guidelines Although the value is a 32 bit integer, the value is easily set in the Variables Screen or the Parameter Form by menu pick. First select Off, Input, or Output, then pick a function, and finally select the logic polarity as active high or low.

Input Functions `FaultReset, RunStop, Enable2, VelCmdSrc, CwInh, CcwInh, AInNull, Position Block, Use CmdGain2`

Output Functions `Fault, Enabled, Brake, ElecRev, MechRev, Zero Speed, PosError Warning`

BDLgcThr

(NV Parameter, Integer) i256

Range	0 or 1
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0 (5 volt logic).
Purpose	`BDLgcThr` sets the switching threshold of all the discrete inputs and the pull up voltage for the BDIOs.
Guidelines	0 selects 5 volt logic compatibility 1 selects 24 volt logic compatibility

BDLgcThr	**Low (volts)**	**High (volts)**	**Pull up (volts)**
0	2.1	3.1	5.0
1	4.0	5.0	12.0

BDOut*X*

(Variable, Integer) i35-i40

Range	0 or 1
Default	none
Purpose	Allows setting the output logic state of BDIO outputs not mapped to an output function via `BDIOMap`.
	`BDOut1` sets the state of BDIO1, J4-7
	`BDOut2` sets the state of BDIO2, J4-8
	`BDOut3` sets the state of BDIO3, J4-9
	`BDOut4` sets the state of BDIO4, J4-10
	`BDOut5` sets the state of BDIO5, J4-11
	`BDOut6` sets the state of BDIO6, J4-12
Guidelines	0 turns on the pull down transistor
	1 turns off the pull down transistor

BlkType

(NV Parameter, Integer) i85

Range	0, 1, 2, 4, 5 or 8
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to that selected by the user.
Purpose	`BlkType` specifies configuration as a position, velocity, or torque block.
Guidelines	`BlkType` sets the overall control functionality of the drive. For block diagrams of the drive configurations, refer to the manual which describes the alternative `BlkType` settings. When used in any of the analog modes, the analog control is the differential voltage applied to the Analog Cmd+ and Analog Cmd- inputs (J4-1 and J4-2 respectively).

BlkType	Servo Configuration
0	Analog Torque Block
1	Analog Velocity Block
2	Digital Position Block
4	EncFreq Torque Block
5	EncFreq Velocity Block
8	Serial Port Command Velocity block

If `Position Block Input Select` is mapped to a BDIO input and the input is asserted, then the active Block Type will be Position Block (`BlkType` = 8). If the input is not asserted, then the active Block Type is controlled by `BlkType`.

Brake

(Mappable Output Function, Variable, Integer, Read-Only)

i277

Range	0 or 1
Purpose	Output function to indicate when the motor is not powered and a mechanical brake is needed to hold the motor.
Guidelines	0 = the motor is powered and the brake should be off 1 = the mechanical brake should engage To insure that a mechanical brake is engaged when a drive's control power is removed, the `Brake` function should be mapped active high to a BDIO pin.

CCDate
(Variable, Integer, Read-Only) i280

Range	0 to 2^31
Default	none
Purpose	`OCDate` gives the Control card date code.

CCSNum
(Variable, Integer, Read-Only) i279

Range	0 to 2^31
Default	none
Purpose	`CCSNum` gives the Control card serial number.

CcwInh

(Mappable Input Function, Variable, Integer) i164

Range	0 or 1 (0 is normal operation; 1 is function activated)
Default	0 at power up if not mapped
Purpose	`CcwInh` selects between normal operation and clamping `VelCmdA` to be only positive.
Guidelines	When `CcwInh` = 1, counter-clockwise rotation is inhibited. That is, `VelCmdA` is clamped to be only positive. For positioning `BlkType`, `PosError` must return to near 0 or be positive to exit this mode. When `CcwInh` and `CwInh` are both active `VelCmdA` is set to 0 and the first inhibit to go inactive stops that inhibit immediately independent of `BlkType`. While actively inhibiting CCW motion, the status LED alternates 8r. With Both CCW and CwInh active, the LED alternates 8n.

Cfgd
(Variable, Integer, Read-Only) i3

Range	-32,768 to 0
Purpose	Configuration state of the drive’s RAM. 0 is a fully configured drive, -1 is a completely unconfigured drive, and other minus numbers indicate partial configuration.

CmdGain

(NV Parameter, Float) f22

Units, Range	`BlkType` = 0 Amperes/Volt $\pm 10^{10} * I_{peak}$
	`BlkType` = 1 KRPM/Volt $\pm 10^{10}$
	`BlkType` = 2 Not Applicable (see `PulsesIn`, `PulsesOut`)
	`BlkType` = 4 Amperes/KHz $\pm 10^{8} * I_{peak}$
	`BlkType` = 5 KRPM/KHz $\pm 10^{7}$
	`BlkType` = 8 Not Applicable
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.
Purpose	`CmdGain` sets the scale factor of the analog input.
Guidelines	`CmdGain` is a floating point variable that sets the command gain of the analog input (voltage from J4-1 to J4-2) for `BlkType's` 0 (Analog torque block), and 1 (Analog velocity block) and the encoder input frequency for `BlkType's` 4 (EncFreq Torque) and 5 (EncFreq Velocity).
	If `Use CmdGain2` is mapped to a BDIO Input and the input is asserted, then the active value for Command Gain will be `CmdGain2`. If the input is not asserted, then the active value for Command Gain will be `CmdGain`.

CmdGain2

(NV Parameter, Float) f323

Units, Range	BlkType = 0	Amperes/Volt	$\pm 10^{10} * I_{peak}$
	BlkType = 1	KRPM/Volt	$\pm 10^{10}$
	BlkType = 2	Not Applicable	(see PulsesIn, PulsesOut)
	BlkType = 4	Amperes/KHz	$\pm 10^{8} * I_{peak}$
	BlkType = 5	KRPM/KHz	$\pm 10^{7}$
	BlkType = 8	Not Applicable	

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets this parameter to 0.0.

Purpose CmdGain2 sets the scale factor of the analog input when a BDIO Input that has been mapped to CmdGain2 function is asserted.

Guidelines CmdGain2 is a floating point variable that sets the command gain of the analog input (voltage from J4-1 to J4-2) for BlkType's 0 (Analog torque block), and 1 (Analog velocity block) and the encoder input frequency for BlkType's 4 (EncFreq Torque) and 5 (EncFreq Velocity).

CommEnbl

(Variable, Integer) i131

Range	0 or 1
Default	1
Purpose	`CommEnbl` allows/disallows normal commutation.
Guidelines	0 disables commutation; commutation angle set only by `CommOff`
	1 enables normal commutation

IMPORTANT NOTE

`CommEnbl` must always be 1 for normal operation. Leaving `CommEnbl` at 0 can overheat and possibly damage the motor.

CommOff

(NV Parameter, Float)f23

Units	Electrical Degrees
Range	0 to 360
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0 degrees.
Purpose	`CommOff` sets the origin for the electrical commutation angle.
Guidelines	Proper value for standard Pacific Scientific motors is 0.

Note: *For `CommSrc` = 1 (incremental encoder commutation) `CommOff` is set to 0 on every power up, independent of the value in the non-volatile memory. Drive RAM value is always read/write.*

CommSrc

(NV Parameter, Integer) i265

Range	0 or 1
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0 (commutate from motor resolver feedback).
Purpose	`CommSrc` selects between resolver or incremental encoder feedback for motor commutation.
Guidelines	0 selects resolver feedback commutation and `PoleCount` set to number of motor poles 1 selects incremental encoder feedback commutation `PoleCount` set to number of quadrature encoder counts per electrical cycle.

CwInh

(Mappable Input Function, Variable, Integer) i163

Range	0 or 1 (0 is normal operation; 1 is function activated)
Default	0 at power up if not mapped
Purpose	`CwInh` selects between normal operation and clamping `VelCmdA` to be only negative.
Guidelines	When `CwInh` = 1, clockwise rotation is inhibited. That is, `VelCmd` is clamped to be only negative. For positioning `BlkType`'s `PosError` must return to near 0 or be negative to exit this mode. When `CwInh` and `CcwInh` are both active `VelCmdA` is set to 0 and the first inhibit to go inactive stops that inhibit immediately independent of `BlkType`. While actively inhibiting CW motion, the status LED alternates 8¬. With Both CCW and CwInh active, the LED alternates 8n.

DecelLmt

(NV Parameter, Float) f277

Units	RPM/sec
Range	0 to 1 x 10^9
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets it value to 0 (no deceleration limiting).
Purpose	Slew rate limit on actual velocity command magnitude decreases. See `VelCmdA` for the `VelCmd` value after slew limiting.
Guidelines	Setting `DecelLmt` to 0 turns off `DecelLmt` slew limiting; `VelCmdA` can immediately decrease to equal `VelCmd`. See `AccelLmt` for control of `VelCmdA` magnitude increases. For position loops, setting either `AccelLmt` or `DecelLmt` to a value is not recommended as it may cause excessive overshoot.

DM1F0

(NV Parameter, Float) f17

Units	Hertz
Range	0.01 to 4.17 x 10^7
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1000.
Purpose	`DM1F0` sets the frequency in Hz of a single pole low-pass filter on the DAC Monitor 1 output (J4-3).
Guidelines	`DM1F0` can be used to attenuate high frequency components from the `DM1Map` selected signal. Setting `DM1F0` to 1 Hz and using `DM1Out` to examine the filtered value is an easy way to accurately measure the selected signal's dc value.

DM2F0

(NV Parameter, Float) f266

Units	Hertz
Range	0.01 to 4.17 x 10^7
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1000.
Purpose	`DM2F0` sets the frequency in Hz of a single pole low-pass filter on the DAC Monitor 2 output (J4-4).
Guidelines	`DM2F0` can be used to attenuate high frequency components from the `DM2Map` selected signal. Setting `DM2F0` to 1 Hz and using `DM2Out` to examine the filtered value is an easy way to accurately measure the selected signal's dc value.

DM1Gain

(NV Parameter, Float) f21

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.

Purpose Sets the multiplicative scale factor applied to the `DM1Map` selected signal before outputting on DAC Monitor 1 (J4-3).

Guidelines Changing `DM1Map` changes `DM1Gain`'s value unless `DM1Map` changes to a signal with identical units, such as `VelCmdA` to `VelFB` (`DM1Map` = 1 to 2). Set `DM1Gain` to keep the signal in the DAC Monitor ± 5 volt range. Below lists units when `DM1Gain` = 1.

Monitor#	Scale Factor	Monitor#	Scale Factor
0	No effect	15	1 v/Cycle
1	1 v/kRPM	16	1 v/Amp
2	1 v/kRPM	17	1 v/Amp
3	1 v/kRPM	18	1 v/Amp
4	1 v/kRPM	19	1 v/100%
5	1 v/Rev	20	1 v/100%
6	1 v/Rev	21	1 v/100%
7	1 v/Rev	22	1 v/v
8	1 v/Amp	23	1 v/Rev
9	1 v/Amp	24	1 v/Amp
10	1 v/v	25	1 v/Amp
11	1 v/Hz	26	1 v/100%
12	10 v/4096	27	1 v/100%
13	1 v/100%	28	1 v/kRPM
14	1 v/°C		

See also `DM1Map`, `DM1F0`, and `DM1Out`.

DM2Gain
(NV Parameter, Float) f263

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.

Purpose Sets the multiplicative scale factor applied to the `DM2Map` selected signal before outputting on DAC Monitor 2 (J4-4).

Guidelines Changing `DM2Map` changes `DM2Gain`'s value unless `DM2Map` changes to a signal with identical units, such as `VelCmdA` to `VelFB` (`DM2Map` = 1 to 2). Set `DM1Gain` to keep the signal in the DAC Monitor +- 5 volt range. Below lists units when `DM2Gain` = 1.

Monitor#	**Scale Factor**	**Monitor#**	**Scale Factor**
0	No effect	15	1 v/Cycle
1	1 v/kRPM	16	1 v/Amp
2	1 v/kRPM	17	1 v/Amp
3	1 v/kRPM	18	1 v/Amp
4	1 v/kRPM	19	1 v/100%
5	1 v/Rev	20	1 v/100%
6	1 v/Rev	21	1 v/100%
7	1 v/Rev	22	1 v/v
8	1 v/Amp	23	1 v/Rev
9	1 v/Amp	24	1 v/Amp
10	1 v/v	25	1 v/Amp
11	1 v/Hz	26	1 v/100%
12	10 v/4096	27	1 v/100%
13	1 v/100%	28	1 v/kRPM
14	1 v/°C		

See also `DM2Map`, `DM2F0`, and `DM2Out`.

DM1Map

(NV Parameter, Integer) i7

Range 0 to 65,537

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 9 (IFB).

Purpose DM1Map selects the signal sent to the DAC Monitor 1 output on J4-3.

Guidelines

Monitor #	Mnemonic	Monitor #	Mnemonic
0	AnalogOut1	16	IR
1	VelFB	17	IS
2	VelCmdA	18	IT
3	VelErr	19	VR
4	FVelErr	20	VS
5	Position*	21	VT
6	PosError*	22	VBus
7	PosCommand*	23	ResPos*
8	ICmd	24	Non-Trq Icmd
9	IFB	25	Non-Trq IFB
10	AnalogIn	26	Trq VCmd
11	EncFreq	27	Non-Trq Vcmd
12	EncPos*	28	VelCmd
13	ItFilt	65536	Clamp Off†
14	HSTemp	65537	Clamp On†
15	Comm Ang*		

* - Will wrap around when the signal exceeds the output voltage range.

† - The value of the selected signal does not change.

See also DM1Gain, DM1F0, and DM1Out

DM2Map

(NV Parameter, Integer) i258

Range 0 to 65,537

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1 (`VelFB`).

Purpose `DM2Map` selects the signal sent to the DAC Monitor 2 output on J4-4.

Guidelines

Monitor #	Mnemonic	Monitor #	Mnemonic
0	AnalogOut2	16	IR
1	VelFB	17	IS
2	VelCmdA	18	IT
3	VelErr	19	VR
4	FVelErr	20	VS
5	Position*	21	VT
6	PosError*	22	VBus
7	PosCommand*	23	ResPos*
8	ICmd	24	Non-Trq Icmd
9	IFB	25	Non-Trq IFB
10	AnalogIn	26	Trq VCmd
11	EncFreq	27	Non-Trq Vcmd
12	EncPos*	28	VelCmd
13	ItFilt	65536	Clamp Off†
14	HSTemp	65537	Clamp On†
15	Comm Ang*		

* Will wrap around when the signal exceeds the output voltage range.

† The value of the selected signal does not change.

See also `DM2Gain`, `DM2F0`, and `DM2Out`

DM1Out

(Variable, Float, Read-Only) f31

Range	Depends on `DM1Map` selected signal
Purpose	`DM1Out` indicates the value of the selected, filtered variable output to DAC Monitor 1 (J4-3). The value is reported in the units of the selected variable. For example `DM1Map` = 1 selects `VelCmdA` and the units would be RPM.
Guidelines	With `DM1F0` set low, such as 1 Hz, `DM1Out`'s value will accurately measure the `DM1Map` selected signal's dc component. `DM1Out` can also be used to examine variables that can not be directly queried such as motor phase voltage duty cycle, `DM1Map` = 19, 20, or 21.

DM2Out

(Variable, Float, Read-Only) f256

Range	Depends on `DM2Map` selected signal
Purpose	`DM2Out` indicates the value of the selected, filtered variable output to DAC Monitor 2 (J4-4). The value is reported in the units of the selected variable. For example `DM2Map` = 1 selects `VelCmdA` and the units would be RPM.
Guidelines	With `DM2F0` set low, such as 1 Hz, `DM2Out`'s value will accurately measure the `DM2Map` selected signal's dc component. `DM2Out` can also be used to examine variables that can not be directly queried such as motor phase voltage duty cycle, `DM2Map` = 19, 20, or 21.

ElecRev

(Mappable Output Function, Variable, Integer, Read-Only)

Not accessible over the serial port

Range	0 or 1
Purpose	Square wave whose frequency is equal to the motor electrical frequency.
Guidelines	There are PoleCount/2 motor electrical revolutions (cycles) per mechanical revolution.

$$\text{ElecRev (Hz)} = \frac{\text{(Shaft RPM)}}{60} * \frac{\text{Polecount}}{2}$$

Enable

(Variable, Integer) i10

Range	0 or 1
Default	Set to 1 at power up
Purpose	`Enable` = 0 prevents power from flowing out of the motor power terminals (J2). 0 (to disable the drive) 1 (to enable the drive)
Guidelines	Before power can flow to the motor, verify that the following conditions are all true: 1. Drive is not faulted. (Status display 0 or 8) 2. Enable/ input (J4-6) connected to I/O RTN. 3. `Enable2` function is = 1. 4. `Enable` variable is = 1. See also `Enabled`.

Enable2

(Mappable Input Function, Variable, Integer) i266

Range	0 or 1
Default	1 at power up if not mapped
Purpose	Second drive enable function that can be mapped to a BDIO pin with programmable polarity.

Guidelines

0 disables the drive

1 allows the drive to enable if other conditions permit

For incremental encoder based commutation `Enable2` should be mapped to a BDIO and be used to enable/disable motion since the dedicated hardware enable J4-6 is used to start commutation alignment.

Before applying power to the motor, verify that the following conditions are true:

1. Drive is not faulted. (Status display 0 or 8)
2. Enable/ input (J4-6) connected to I/O RTN.
3. `Enable2` function is = 1.
4. `Enable` variable is = 1.

Enabled

(Mappable Output Function, Variable, Integer, Read-Only)

i11

Range	0 or 1
Purpose	`Enabled` indicates whether power can flow to the motor. 0 (drive disabled) 1 (drive enabled)
Guidelines	Before power can flow to the motor, verify that the following conditions are all true: 1. Drive is not faulted. (Status display 0 or 8) 2. Enable/ input (J4-6) connected to I/O RTN. 3. `Enable2` function is = 1. 4. `Enable` variable is = 1.

EncFreq

(Variable, Float, Read-Only) f2

Units	(if `EncMode` = 0) Quadrature encoder counts per second
	(if `EncMode` = 1, 2) Steps per second
Range	-3,000,000 to +3,000,000
Purpose	`EncFreq` (Encoder Frequency) is the frequency in quadrature pulses per second of the external encoder, (or steps per second if step-and-direction format is used).
Calculation	`EncFreq` = Encoder Speed (RPM) * Encoder Line Count / 15
Guidelines	Calculated from delta `EncPos` at position loop update rate. Although the values returned do not have fractional parts this variable is communicated as a floating point quantity.
	See `EncInF0` for recommended maximum count frequencies.

EncIn

(NV Parameter, Integer) i12

Units	(if `EncMode` = 0) Encoder line count (if `EncMode` = 1) Steps per quarter-revolution
Range	1 to 65535
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1024.
Purpose	`EncIn` specifies the line count of the encoder being used, (or one-fourth the steps/revolution if step-and-direction input format is used).
Guidelines	`EncIn` is used to insure proper units in `KPP, KVP, VelFB` when using an encoder for Servo feedback. See also `RemoteFB`. When `RemoteFB = 0`, `EncIn` is not used.

EncInF0

(NV Parameter, Float) f287

Units Hertz

Range 4 values depending on `EncMode`

EncMode = 0 (Quadrature decode)

EncInF0 (Hz)	Max Hardware Quad Count limit (Hz)	Min Hardware Pulse Width (usec)
1,600,000	3,333,333	0.6
800,000	952,400	2.1
400,000	476,200	4.2
200,000	238,100	8.4

EncMode = 1 or 2 (Step, Dir or Up, Down)

EncInF0 (Hz)	Max Hardware Count limit (Hz)	Min Hardware Pulse Width (usec)
800,000	833,333	0.6
200,000	238,000	2.1
100,000	119,000	4.2
50,000	59,500	8.4

Default Parameter value set before last NVSave. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1,600,000 Hz.

Purpose `EncInF0` selects digital low pass filter frequency on the incremental encoder input connected to J4-21 through J4-24.

Guidelines `EncInF0` is the maximum recommended count frequency for reliable operation. If the maximum input frequency is < `EncInF0`, lowering it will give better noise rejection.

The maximum hardware count limits require ideal timing with exact 50% duty cycle, perfect quadrature symmetry, etc. The recommended `EncInF0` count takes real world signal tolerances into account. With the SC900's emulated encoder out wired to another SC900's encoder in, and `EncInF0` = 1,600,000 Hz, the count frequency can work reliably up to 2,000,000 Hz.

EncMode

(NV Parameter, Integer) i71

Range 0, 1, 2, or 3

Default Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0.

Purpose `EncMode` specifies the type of digital command expected at the incremental position command port.

Guidelines `BlkType` = 2 uses the incremental position command port (J4-21, J4-22, J4-23, J4-24) for its position command.

Value of EncMode	**Description**
0	Selects quadrature encoder pulses
1	Selects step-and-direction input signals
2	Selects up/down count input signals
3	Ignores input signal, `EncPos` value held

EncOut

(NV Parameter, Integer) i69

Units	Emulated encoder line count
Range	0, 128, 256, 512, 1024, 2048, 4096, 8192, 16384,
	125, 250, 500, 1000, 2000, 4000, 8000, 16000
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1024.
Purpose	`EncOut` selects the resolution of the incremental encoder shaft position output (J4-14, J4-15, J4-16, J4-17, and J4-19, J4-20) emulated from the resolver.
Guidelines	`EncOut` = 0 cross connects the incremental position command port input (J4-21, J4-22 and J4-23, J4-24) to the incremental shaft position output port to provide buffering. CH Z out (J4-19, J4-20) is held fixed for `EncOut` = 0.

EncPos

(Variable, Integer, Read-Only) i13

Units	`EncMode` = 0 Quadrature encoder counts
	`EncMode` = 1, 2 Steps
Range	-2,147,483,648 to +2,147,483,647
Purpose	`EncPos` (Encoder Position) indicates the position of the external encoder or the accumulation of step inputs if step-and-direction input format is used. For example, with a 1024 line encoder, each increment of `EncPos` is equal to 1/4096 of a revolution of the encoder shaft.
Guidelines	See `EncInF0` for maximum count frequencies. See `EncMode` for input decoding mode.

ExtFault

(Variable, Integer, Read-Only) i133

Range 0-16

Purpose `ExtFault` provides additional information on `FaultCode` Blinking 1 (1) or E (14) and Alternating F 3 (243), 0 otherwise.

Guidelines

In the variables window, poll the value of `ExtFault` for additional fault information. Values listed below:

LED Display	Value of ExtFault	Description
1	1	$\lvert VelFB \rvert > 21038$
	2	$\lvert VelFB \rvert > 1.5 * \max(\lvert VelLmtxx \rvert)$
E	0	No ExtFault information
E	1	Resolver Calibration data corrupted
E	2	Excessive dc offset in current feedback sensor
E	3	DSP incompletely reset by line power dip
E	6	Excessive dc offset in Analog Command A/D
E	7	Unable to determine option card type
E	8	DSP stack overflow
E	10	Firmware and control card ASIC incompatible
E	11	Actual Model does not match value in non-volatile memory
E	12	Unable to determine power stage
E	13	Control card non-volatile parameters corrupt
E	14	Option card non-volatile parameters corrupt
F 3	15	RAM failure
F 3	16	Calibration RAM failure

Fault

(Mappable Output Function) i286

Range	0 or 1
Purpose	`Fault` indicates whether the drive has faulted and is disabled.
Guidelines	0 is not faulted, normal operation
	1 is faulted, no power flow to motor
	See `FaultCode` and `ExtFault` for further information.

FaultCode

(Variable, Integer, Read-Only) i14

Range 0 to 255

Purpose `FaultCode` indicates a fault has occurred. When the status display is not a zero or an eight, a fault has occurred. Reset the drive by asserting the fault reset signal or cycling drive AC power.

Guidelines Under HELP menu, see Index topic FAULTCODES for remainder of fault codes. Value is 0/8 unless faulted.

Status LED	Value	No Fault Meaning
0 (Solid)	0	Not Faulted/Not Enabled
8 (Solid)	0	Not Faulted/Enabled
8¬ (Alternating)	0	Not Faulted/Enabled/CwInh active
8r (Alternating)	0	Not Faulted/Enabled/CCwInh active
8n (Alternating)	0	Not Faulted/Enabled/CwInh & CCwInh active

Faultcodes

Status LED	Value	Fault Meaning
(Blinking) 1	1	Velocity feedback (VelFB) over speed
(Blinking) 2	2	Motor Over-Temp
(Blinking) 3	3	Drive Over-Temp
(Blinking) 4	4	Drive I*t
(Blinking) 5	5	l-n Fault (9x3)
(Blinking) 6	6	Control ±12 V supply under voltage
(Blinking) 7	7	Output over current or bus over voltage
(Blinking) 9	9	Shunt Regulator Overload

Faultcodes (continued)

Table (cont'd)

Status LED	**Value**	**Fault Meaning**
(Blinking) A	10	Bus OV detected by DSP
(Blinking) b	11	Auxiliary +5V Low
(Blinking) C	12	Not assigned
(Blinking) D	13	Not assigned
(Solid) E*	14	Processor throughput fault
(Blinking) E*	14	Power Up Self Test Failure
(Alternating) E1	225	Bus UV, Bus Voltage VBUSTHRESH
(Alternating) E2	226	Ambient Temp Too Low
(Alternating) E3	227	Encoder commutation align failed (Only CommSrc=1)
(Alternating) E4	228	Drive software incompatible with NV memory version
(Alternating) E5*	229	Control Card hardware not compatible with drive software version
(Alternating) E6	230	Drive transition from unconfigured to configured while enabled
(Alternating) E7	231	Two AInNull events too close together
(Alternating) F1	241	Excessive Position Following Error Jam
(Alternating) F3	243	Parameter Checksum Error (Memory Error)

* `FaultReset` cannot reset these faults

See `ExtFault` for further information on Blinking E, Blinking 1, and Alternating F 3.

FaultReset
(Mappable Input Function, Variable, Integer) i263

Range	0 or 1
Default	0 at power up if not mapped
Purpose	`FaultReset` allows drive faults to be reset.
Guidelines	`FaultReset` active automatically disables the drive. When not mapped to a BDIO, setting `FaultReset` to 1 via the serial port will reset the latched fault condition. If the fault persists when `FaultReset` is active the drive remains faulted. If the fault condition does not persist, then setting `FaultReset` to 1 clears the latched fault and returning `FaultReset` to 0 resumes normal operation.

FVelErr

(Variable, Float, Read-Only) f30

Units	RPM
Range	-48000 to +48000
Purpose	`FVelErr` is commanded velocity - measured velocity (`VelCmdA` - `VelFB`) after being processed by the velocity loop compensation anti-resonant filter section. See also `ARF0, ARF1, ARZ0, ARZ1`.

FwV
(Variable, Integer, Read-Only) i84

Range	1000 to 65535
Purpose	`FwV` indicates the 930 firmware version number.
	For example: `FwV` = 1100 would be version 1.1

HSTemp

(Variable, Float, Read-Only) f269

Units	Degrees Centigrade
Range	-10 to +150
Purpose	`HSTemp` indicates the drive heat sink temperature.
Guidelines	The drive heat sink temperature is monitored to determine if the drive is within a safe operating region for the power electronics. This variable can be used to see how much thermal margin remains for a given application.

See also `ItThresh`.

HwV

(Variable, Integer, Read-Only) i130

Range	> 0
Purpose	`HwV` indicates the drive's control electronics hardware version number.
Guidelines	12 = first production control card version

ICmd

(Variable, Float, Read-Only) f28

Units	Amperes
Range	-Ipeak to +Ipeak
Purpose	`ICmd` indicates the commanded motor torque current.
	Note: *`ILmtMinus`* and *`ILmtPlus`* limit the range of this variable.

IFB

(Variable, Float, Read-Only) f29

Units	Amperes
Range	-Ipeak to Ipeak
Purpose	`IFB` indicates the measured motor torque current value.
Guidelines	`IFB` can be monitored to observe the actual torque current flowing in the motor. `IFB` should equal `ICmd`.

ILmtMinus

(NV Parameter, Integer) i4

Units	% (Percentage) of peak current rating of drive
Range	0 to 100
Default	The default value, for **New Set Up** and **Drive Set Up** menu picks, is based upon the selected motor and drive.
Purpose	`ILmtMinus` (Counter-Clockwise Current Limit) sets the maximum allowable torque current amplitude in the counter-clockwise direction. This is a percentage of the drive's peak current rating (`Ipeak`).
Guidelines	Only integer values may be entered (i.e. no fractional numbers).

Warning

If ILmtMinus*0.01*Ipeak > twice the motor's continuous current rating, the motor's over temperature sensor is not guaranteed to always respond fast enough to prevent motor winding damage.

ILmtPlus
(NV Parameter, Integer) i5

Units	% (Percentage) of peak current rating of drive
Range	0 to 100
Default	The default value, for **New Set Up** and **Drive Set Up** menu picks, is based upon the selected motor and drive.
Purpose	`ILmtPlus` (Clockwise Current Limit) sets the maximum allowable torque current amplitude in the clockwise direction. This is a percentage of the drive's peak current rating (`Ipeak`).
Guidelines	Only integer values may be entered (i.e. no fractional numbers).

Warning

If ILmtPlus*0.01*Ipeak > twice the motor's continuous current rating, the motor's over temperature sensor is not guaranteed to always respond fast enough to prevent motor winding damage.

Inputs

(Variable, Integer, Read-Only) i33

Range	0 to 63 (6 BDIOs)
Purpose	`Inputs` reads the state of BDIO inputs in parallel. This variable is determined by the voltage levels applied to the BDIO input pins J4-7 to J4-12.
Guidelines	`Inputs`' value is weighted so that BDIO 1's individual value is x1, BDIO 2's is x2, BDIO 3's is x4, etc. 0 corresponds to a low input while 1 corresponds to a high input. Inputs = 12 means that BDIO 1, 2, 5, 6 are low and BDIO 3, 4 are high. See `BDIn1-6` to query inputs individually.

Inputs =

1xBDIO1 + 2xBDIO2 + 4xBDIO3 + 8xBDIO4 + 15xBDIO5 + 32xBDIO6

Ipeak

(Variable, Float, Read-Only) f20

Units Amperes

Range single value (see Default below)

Default

Model Number	**I_{peak}**
932	7.5
933	15.0
934	30.0
935	60.0

Purpose `Ipeak` is the drive's maximum 0-peak current rating.

IR

(Variable, Float, Read-Only) f270

Units	Amps
Purpose	`IR` is the measured current flowing in Motor Phase R, J2-4.

IS

(Variable, Float, Read-Only) f271

Units	Amps
Purpose	`IS` is the measured current flowing in Motor Phase S, J2-3.

IT
(Variable, Float, Read-Only) f272

Units	Amps
Purpose	`IT` is the measured current flowing in Motor Phase T, J2-2.

ItF0

(NV Parameter, Float) f11

Units	Hertz
Range	Lower limit set by Model Upper limit > 10
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value based on the selected drive.
Purpose	`ItF0` specifies the corner frequency of the low-pass filter implementing the I*t drive thermal protection circuit.
Guidelines	`ItF0`, in conjunction with `ItThresh` specifies the thermal protection circuit for the drive. `ItF0` is the corner frequency of a low-pass filter which processes an estimate of the drive's power dissipation. Increasing `ItF0` makes the response more sensitive to over-current conditions. **Note:** *The minimum frequency for* `ItF0` *(slowest to fault) is limited to protect the drive's power electronics.*

ItFilt

(Variable, Float, Read-Only) f25

Units	% (Percentage) of drive peak current
Range	0 to 100
Purpose	`ItFilt` is the drive's output current amplitude low pass filtered by `ItF0` and normalized by `Ipeak` to a percentage. `ItFilt` is the input to the drive's I*t thermal protection fault.
Guidelines	`ItFilt` provides a means of evaluating the I*t protection circuit. When `ItFilt` exceeds the threshold specified by `ItThreshA`, the drive faults with `FaultCode` 4.

`ItFilt` = `ItF0` low pass filter of $\left(|I_r|+|I_s|+|I_t|\right)\dfrac{100}{2*Ipeak}$

ItThresh

(NV Parameter, Integer) i82

Units	% (Percentage) of drive peak current
Range	0 to 100 (Actual upper limit depends on Model)
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and& drive.
Purpose	`ItThresh` sets the desired maximum continuous output current, as a percentage of Ipeak, before the I*t thermal protection faults the drive.
Guidelines	`ItThresh`, in conjunction with `ItF0`, specifies the thermal protection fault for the drive. The actual I*t fault threshold may be lowered if the heat sink temperature (`HSTemp`) gets too high, see `ItThreshA`.

Note: *The maximum value for* `ItThresh` *is limited to protect the drive's power electronics.*

ItThreshA

(Variable, Float, Read-Only) f316

Units	percent
Range	0 to 100%
Default	none
Purpose	`ItThreshA` is the maximum continuous output current, as a percentage of Ipeak, trip level for the I*T thermal protection fault.
Guidelines	`ItThresh` sets the desired value for `ItThreshA` and the two are equal for lower heat sink temperatures, i.e. lower `HSTemps`. At higher `HSTemps`, `ItThreshA` may be clamped to protect the power stage. When `ItFilt` exceeds `ItThreshA` the drive will I*t fault. While doing a worst case motion profile examining `ItThreshA`, `ItFilt`, and `HSTemp` will indicate how much drive thermal margin remains.

Kii

(NV Parameter, Float) f264

Units	Hertz
Range	0 to 2546
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 50 Hertz.
Purpose	`Kii` sets the integral gain of the current loops.
Guidelines	`Kii` is the current loop's integral gain. It defines the frequency where the current loop compensation transitions from predominantly integral characteristics (gain decreasing with frequency) to predominantly proportional characteristics (constant gain with frequency). This value should typically be less than 10% of the current loop's bandwidth. See `Kip` for more information.

Kip

(NV Parameter, Float) f257

Units	Volts/Ampere
Range	0 to 161,712/Ipeak
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor.
Purpose	`Kip` sets the proportional gain of the current loop.
Guidelines	930 dialogue automatically sets `Kip` as long as the motor being used is in 930 Dialogue's motor database or the motor inductance is entered in Manual entry mode. Current loop bandwidth in Rad/sec is Kip/L, where L is the motor's line-to-line inductance (in henries). Recommended bandwidth is $2\pi*1000$ Rad/sec, maximum bandwidth is $2\pi*1500$ Rad/sec.

Kpp
(NV Parameter, Float) f14

Units	Hertz
Range	0 to 159.4
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected system response.
Purpose	`Kpp` sets the proportional gain of the position loop.
Guidelines	`Kpp` is defined by the following relationship:
	`VelCmd` (rad/sec) = $2\pi * KPP * PosError$ (radians)

Kvff

(NV Parameter, Float) f16

Units	% (Percentage)
Range	0 to 199.9
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 0.
Purpose	`Kvff` sets the proportion of velocity feed forward signal added to the velocity command from differentiated position command.
Guidelines	`Kvff` is functional only for positioning modes, `BlkType` = 2.

When `Kvff` = 0 the net velocity command in positioning mode results entirely from `PosError`. For this case, there will be a static non-zero `PosError` commanding a constant shaft speed. This error is known as the following error. Velocity feed forward adds a term to `VelCmd` proportional to delta `PosCommand` at the position loop update rate which can reduce following error.

Increasing `Kvff` reduces steady state following error and gives faster response time. However, if `Kvff` is too large, it will cause overshoot. Typically `Kvff` should not be set larger than 80% for smooth dynamics and acceptable overshoot, but should be set to 100% for minimum following error, which may be necessary in electronic gearing applications.

Kvi

(NV Parameter, Float) f10

Units	Hertz
Range	0 to 636.6
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected velocity loop bandwidth.
Purpose	`Kvi` sets the integral gain of the velocity loop.
Guidelines	`Kvi` is the velocity loop integral gain. It defines the frequency where the velocity loop compensation transitions from predominantly integral characteristics (gain decreasing with frequency) to predominantly proportional characteristics (constant gain with frequency). This value should typically be less than 10% of the velocity loop bandwidth. See `KVP`.

Kvp

(NV Parameter, Float) f15

Units	Amperes/(Radians/Second)
Range	0 to Ipeak * 12.6
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected velocity loop bandwidth.
Purpose	`Kvp` sets the proportional gain of the velocity loop.
Guidelines	`Kvp` is defined by the following relationship:

`Kvp` = Commanded motor torque current / Velocity Error;

where Commanded motor torque current has units of (amperes), and Velocity Error has units of (radians/second).

`Kvp` must be adjusted for total load inertia and motor torque constant as described in Chapter 6.

Note: *Idealized velocity loop bandwidth (in Rad/sec) equals Kvp * Kt / J, where J is the total shaft inertia and K_t/J units are $rad/sec^2/A$. Maximum recommended idealized bandwidth is 2π*400 Rad/sec.*

MechRev

(Mappable Output Function, Variable, Integer, Read-Only)

Not accessible over the serial port

Range	0 or 1
Purpose	Square wave whose frequency is equal to the resolver's electrical frequency which is typically equal to the mechanical Rev/sec.
Guidelines	Resolvers can have multiple electrical cycles per mechanical revolution and this is usually specified as the resolver speed.

$$\texttt{MechRev}\ (\text{Hz}) = \frac{\text{Shaft (RPM)}}{60} * \text{Resolver Speed}$$

The resolver speed is nearly always 1, so `MechRev` frequency is usually mechanical Rev/sec. See also `ElecRev`.

Model
(NV Parameter, Integer, Read-Only) i77

Range	932, 933, 934, 935
Purpose	`Model` indicates the drive model number (power level).

Motor
(NV Parameter, Integer, Read-Only) i134

Range	Up to any 4 ASCII characters
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor.
Purpose	`Motor` indicates the first 4 characters of the motor part number used to determine the Signature Series current waveshape used to eliminate torque constant ripple.
Guidelines	This is a read-only parameter that can only be set by 930 Dialogue's Drive Setup, New Setup, or Drive Download.

NVLoad

(command) 256

Purpose `NVLoad` loads all NV Parameters to working memory (RAM) from the control card non-volatile memory.

Guidelines Use `NVLoad` to update all NV parameters. This might be useful if you have changed one or more parameters using the serial link and now wish to restore the original values (assuming the `NVSave` command was **NOT** issued after changing parameters). See also `NVLoadOpt`.

NVLoadOpt

(command) 258

Purpose `NVLoadOpt` loads all NV Parameters to working memory (RAM) from the option card non-volatile memory.

Guidelines Use `NVLoadOpt` to update all NV parameters from the Option Card. This might be useful if you have changed one or more parameters using the serial link and now wish to restore the original values (assuming the `NVSaveOpt` command was **NOT** issued after changing parameters). See also `NVLoad`.

NVSave

(command) 257

Purpose `NVSave` stores all NV parameters from RAM (working memory) to the control card non-volatile memory.

Guidelines Use **`NVSave`** when you wish to save all parameters in non-volatile memory. The 930 will then utilize these values even after the power cycles. The NVRAM utilized is rated to allow 100,000 write cycles. This is far greater than should ever be needed. However, repeated use of `NVSave` from a host computer should be avoided. See also `NVSaveOpt`.

NVSaveOpt

(command) 259

Purpose `NVSaveOpt` stores all NV Parameters from RAM (working memory) to the option card non-volatile memory.

Guidelines Use `NVSaveOpt` when you wish to save all parameters in non-volatile memory on the option card. Saving the parameters to the option card non-volatile memory allows the drive control card to function with a removable Personality Module (the option card), when the drive control card non-volatile memory is not configured. The 930 will then utilize these values even after the power cycles. The NVRAM utilized is rated to allow 100,000 write cycles. This is far greater than should ever be needed. However, repeated use of `NVSaveOpt` from a host computer should be avoided. See also `NVSave`.

OCDate

(Variable, Integer, Read-Only) i282

Range	0 to 2^31
Default	none
Purpose	`OCDate` gives the Option card date code.

OCSNum
(Variable, Integer, Read-Only) i281

Range	0 to 2^31
Default	none
Purpose	`OCSNum` gives the Option card serial number.

Outputs

(Variable, Integer) i47

Range	0 to 63 (6 BDIOs)
Purpose	For BDIO outputs not mapped to an output function via BDIOMap, allows setting their output logic state in parallel.
Guidelines	`Outputs`' value is weighted so that BDIO 1's individual value is x1, BDIO 2's is x2, BDIO 3's is x4, etc. 0 will turn on the corresponding pull down transistor while 1 turns off the pull down transistor. `Outputs` = 12 would pull down BDIO 1, 2, 5, 6 and open circuit BDIO 3, 4.

Note: *BDIOs mapped to output functions via their* `BDIOMap` *are determined by that function and their value in Outputs will be ignored.*

See `BDOut1-6` to set outputs individually.

`Outputs` = 1xBDIO1 + 2xBDIO2 + 4xBDIO3 + 8xBDIO4 + 15xBDIO5 + 32xBDIO6

PoleCount

(NV Parameter, Integer) i72

Units	Motor poles
Range	2 to 65534 (even #'s only)
	1 to 65535 Encoder Counts per electrical cycle
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor.
Purpose	`PoleCount` matches the drive for the appropriate motor pole count or encoder quadrature counts per motor electrical cycle.
Guidelines	For `CommSrc` = 0 sets the number of motor poles
	For `CommSrc` = 1 sets the number of encoder quadrature counts/motor cycle

WARNING

When the `PoleCount` set does not match the actual pole count, the motor's operation will be erratic.

PosCmdSet

(Variable, Integer) i257

Units Counts (same units as position feedback)

Range -2,147,483,648 to +2,147,483,647

Purpose `PosCmdSet` (read/write commanded position) can be used to change the commanded position and therefore allows position control over the serial link.

Guidelines `PosCmdSet` can be used to change the commanded position, `PosCommand`, and therefore allows position control using the serial link.

When `RemoteFB = 0`, `PosCommand` is in resolver counts. When `RemoteFB = 1 or 2`, `PosCommand` is in `EncPos` units.

WARNING

Caution should be used when changing `PosCmdSet`. The new value becomes the input to the position loop (no profiling). Therefore, large changes to `PosCommand` via `PosCmdSet` will result in violent motion and, very likely, large overshoot.

If `PosCmdSet` is to be used for position control over the serial port, a sequence of closely spaced position commands should be issued over time to create a motion profile that can be followed by the drive, motor, and load.

Note: *This variable only makes sense for position control blocks, (i.e. when* `BlkType` *= 2).*

PosCommand

(Variable, Integer, Read-Only) i54

Units	Counts (same units as position feedback)
Range	-2,147,483,648 to +2,147,483,647
Purpose	`PosCommand` (position command) is the position loop command input.
Guidelines	`PosCommand` can be used to determine the position being commanded. It is a read-only variable; it cannot be used to change the commanded position. `PosCmdSet` allows `PosCommand` to be changed using the serial link.

When `RemoteFB = 0`, `PosCommand` is in resolver counts. When `RemoteFB = 1 or 2`, `PosCommand` is in `EncPos` units.

Note: *This variable only makes sense for position control blocks, (i.e. when* `BlkType` *= 2).*

PosError

(Variable, Integer, Read-Only) i55

Units	Counts (same units as position feedback)
Range	-134,217,728 to +134,217,727
Purpose	`PosError` (Actual Position Error) is equal to the difference between the position command (`PosCommand`) and the actual position (`Position`).

When `RemoteFB = 0`, `PosError` is in resolver counts. When `RemoteFB = 1 or 2`, `PosError` is in `EncPos` units.

Note: *This variable only makes sense for position control blocks, (i.e. when* `BlkType` *= 2).*

PosErrorMax

(NV Parameter, Integer) i285

Units	Counts (same units as position feedback)
Range	0 to 294,912,000 (4500 revs)
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 4096.
Purpose	`PosErrorMax` sets the maximum value in position feed back counts for the position loop following error fault.
Guidelines	The following error fault compares `PosError` with the `PosError` predicted from `EncFreq` and `Kvff` and if the magnitude of the difference is larger than `PosErrorMax` continuously for longer than 1 second or statistically larger over half the time the drive will following error fault ("F1"). If you have mapped one of the BDIO outputs as `PosErrorWarning`, then this output will be activated and the following error fault ("F1") will be disabled.

Position

(Variable, Integer, Read-Only) i57

Units	Resolver Counts
Range	-2,147,483,648 to +2,147,483,647
Purpose	`Position` indicates the measured resolver position, including integral resolver cycles since power up.
Guidelines	`Position` is always the measured resolver position.

PulsesFOut

(NV Parameter, Integer) i292

Units	2^{-16} resolver counts
Range	0 to 65535
Default	Parameter value set before last NVSave. Powering up an unconfigured drive sets this parameter to 0.
Purpose	`PulsesFOut` specifies the fractional number of resolver counts the motor will move for each `PulsesIn` number of `EncPos` command counts. Fractional part of numerator of the exact electronic gearing ratio.

Guidelines

```
PosCommand(New) = PosCommand(Old) +
Ratio*(EncPos(New) - EncPos(Old))
```

$$\text{Ratio} = \frac{PulsesOut + (2^{-16} * PulsesFOut)}{PulsesIn}$$

calculated once per position loop update period.

PulsesIn

(NV Parameter, Integer) i58

Units	(if EncMode = 1) Steps (if EncMode = 0) Quadrature counts of encoder input
Range	1 to 32,767
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1.
Purpose	PulsesIn specifies the number of steps, or quadrature encoder counts, used for selecting an exact gear ratio.
Guidelines	PosCommand(New) = PosCommand(Old) + Ratio*(EncPos (New) - EncPos (Old))

$$\text{Ratio} = \frac{PulsesOut + (2^{-16} * PulsesFOut)}{PulsesIn}$$

Calculated once per position loop update period.

PulsesIn specifies the number of EncPos command counts required to increase PosCommand by PulsesOut resolver counts.

PulsesOut

(NV Parameter, Integer) i59

Units	Resolver counts
Range	-32,768 to +32,767
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to 1.
Purpose	`PulsesOut` specifies the number of resolver counts the motor will move for each `PulsesIn` number of `EncPos` command counts. Integral part of numerator of the exact electronic gearing ratio.
Guidelines	`PosCommand`(New) = `PosCommand`(Old) + `Ratio*` (`EncPos` (`New`) - `EncPos` (`Old`))

$$\text{Ratio} = \frac{PulsesOut + (2^{-16} * PulsesFOut)}{PulsesIn}$$

calculated once per position loop update period.

RemoteFB

(NV Parameter, Integer) i267

Range	0, 1, or 2
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets it value to 0 (all loops closed around resolver).
Purpose	`RemoteFB` selects the source of the feedback signal for the loops.
Guidelines	0 Resolver velocity and resolver position feedback
	1 Resolver velocity and encoder position feedback
	2 Encoder velocity and encoder position feedback
	When `RemoteFB` is not equal 0, make sure `EncIn` is set to the proper value so that scaling of `KPP,` `KVP`, and `VelFB` will be in the default units.

ResPos

(Variable, Integer, Read-Only) i56

Units	Resolver Counts
Range	0 to 65535
Purpose	`ResPos` (Resolver Position) is the absolute mechanical orientation of the resolver relative to the motor housing.
Guidelines	`ResPos` varies from zero to maximum range and then back to zero as the motor rotates clockwise through one complete resolver electrical cycle. Standard resolvers have one electrical cycle per mechanical revolution.

RunStop

(Mappable Input Function, Variable, Integer) i264

Range	0 or 1
Default	1 at power up if not mapped
Purpose	`RunStop` selects between normal operation and setting the velocity command to zero and then disabling the drive once `VelFB` goes to 0.
Guidelines	This mappable function is a specialized form of mechanical clutch brake emulation where the shaft is left with no holding torque, e.g. brake and clutch off, once the speed drops to zero or once the `StopTime` is exceeded.

StopTime

(NV Parameter, Float) f262

Units	seconds
Range	0.002 to 65.534
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets it value to 0.
Purpose	Maximum time out to disable the drive for the `RunStop` mappable BDIO function.
Guidelines	When the `RunStop` function enters the stop state a timer starts and if the actual shaft velocity has not reached 0 by `StopTime` the drive is disabled regardless.

UncfgDrv

(command) 260

Purpose	`UncfgDrv` sets the control card non-volatile memory to the unconfigured state.

UncfgOpt

(command) 261

Purpose `UncfgOpt` sets the option card non-volatile memory to the unconfigured state.

VBus

(Variable, Float, Read-Only) f275

Units	Volts
Range	0 to 1000
Purpose	`VBus` is the voltage of the high voltage DC supply, rectified from the AC line, used to power the motor.
Guidelines	Monitoring this variable can be used to detect the presence of the AC line power for the motor DC supply.

For 115 VAC line power the Bus is nominally 160 VDC.

For 240 VAC line power the Bus is nominally 330 VDC.

For 480 VAC line power the Bus is nominally 670 VDC.

VBusThresh

(NV Parameter, Float) f274

Units	Volts
Range	-1 to +1000
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets its value to -1 (fault is disabled).
Purpose	`VBusThresh` is an adjustable parameter to allow the drive to fault if the AC line power for the motor DC supply is low.
Guidelines	When VBus `VBusThresh` the drive will fault and display a blinking "E 1". This functionality allows the drive to have an interlock so that it will not try to move the motor unless there is sufficient motor bus voltage. `VBusThresh` = 255 is a good value to detect a 230 Vac line more than 15% low. **Note:** *A value of -1 disables the Bus Under voltage Fault ("E 1").*

VdCmd

(Variable, Float, Read-Only) f320

Units	% (percentage)
Range	-300 to 300
Purpose	Motor terminal voltage PWM duty cycle amplitude command of the torque producing current control loop.
Guidelines	`VdCmd` is a fraction of the motor power dc bus voltage. See `VBus`.

VelCmd

(NV Parameter, Float) f26

Units	RPM
Range	VelLmtLo to VelLmtHi -21,000 to +21,000
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets it value to 0.
Purpose	`VelCmd` is the net desired velocity loop command input. See `VelLmtHi, VelLmtLo, AccelLmt, DecelLmt, and VelCmdA.`

VelCmdA
(Variable, Float, Read-Only) f268

Units	RPM
Range	VelLmtLo to VelLmtHi
Purpose	Actual velocity loop command.
Guidelines	`VelCmdA` is `VelCmd` (or `VelCmd2` if `VelCmdSrc` = 1) after being slew limited by `AccelRate`, `DecelRate` and potentially clamped by `CwInh` and/or `CcwInh`.

VelCmd2

(NV Parameter, Float) f267

Units	RPM
Range	VelLmtLo to VelLmtHi
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets it value to 0.
Purpose	Non-volatile second velocity command selected when `VelCmdSrc` = 1.
Guidelines	This parameter allows easy emulation of mechanical clutch brake functionality. If `VelCmd2` = 0 then `VelCmdSrc` = 1 corresponds to `VelCmd = 0` and the brake engaged while `VelCmdSrc` = 0 corresponds to the brake off and the clutch engaged.

VelCmdSrc

(Mappable Input Function, Variable, Integer) i276

Range	0 or 1
Default	0 at power up if not mapped
Purpose	`VelCmdSrc` controls whether `VelCmd` source is determined by `BlkType` or is set to `VelCmd2`
Guidelines	0 has `VelCmd` selected by `BlkType`
	1 sets `VelCmd` to `VelCmd2` for all `BlkType`s
	This parameter allows easy emulation of mechanical clutch brake functionality. If `VelCmd2` = 0 then `VelCmdSrc` = 1 corresponds to `VelCmd = 0` and the brake engaged while `VelCmdSrc` = 0 corresponds to the brake off and the clutch engaged.

VelErr
(Variable, Float, Read-Only) f27

Units	RPM
Range	-48000 to +48000
Purpose	`VelErr` is commanded velocity - measured velocity (`VelCmdA` - `VelFB`).

VelFB

(Variable, Float, Read-Only) f34

Units	RPM
Range	-48,000 to +48,000 for resolver
	-30,000 to +30,000 for encoder
Default	none
Purpose	Instantaneous value of the velocity feedback.
Guidelines	For normal operation, `RemoteFB` = 0 or 1, `VelFB` is the resolver velocity. For `RemoteFB` = 2, `VelFB` is based on delta `EncPos` at position loop update rate.

VelLmtHi

(NV Parameter, Float) f279

Units	RPM
Range	-21,039 to +21,039
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.
Purpose	Sets the highest `VelCmdA` value allowed and a `VelFB` overspeed fault threshold.
Guidelines	For `BlkTypes` that have a velocity loop, (BlkType = 1, 2, 5, 8) `VelCmd` and `VelCmd2` are clamped to be less than `VelLmtHi`. In torque control `BlkTypes` (0, 4) `VelLmtHi` has no clamping function. If `VelLmtHi` is reduced to below the current value of `VelCmd2` or `VelCmd`, then `VelCmd2` and/or `VelCmd` are reduced to `VelLmtHi`.

For all `BlkTypes`, a fault with `FaultCode = 1` will occur if $|VelFB| > 1.5 * \max of(|VelLmtLo|, |VelLmtHi|)$.

See also `VelLmtLo`.

VelLmtLo

(NV Parameter, Float) f280

Units	RPM
Range	-21,039 to +21,039
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** calculates its value based on the selected motor and drive.
Purpose	Sets the smallest `VelCmd`A value allowed and a `VelFB` overspeed fault threshold.
Guidelines	For `BlkTypes` that have a velocity loop, (BlkType = 1, 2, 5, 8) `VelCmd` and `VelCmd2` are clamped to be greater than `VelLmtLo`. In torque control `BlkTypes` (0, 4) `VelLmtLo` has no clamping function. If `VelLmtLo` is increased to above the current value of `VelCmd2` or `VelCmd`, then `VelCmd2` and/or `VelCmd` are increased to `VelLmtHi`.

For all `BlkTypes`, a fault with `FaultCode = 1` will occur if $|VelFB| > 1.5 * \min of (|VelLmtLo|, |VelLmtHi|)$.

See also `VelLmtHi`.

Velocity

(Variable, Float, Read-Only) f6

Units	RPM
Range	-30,000 to +30,000
Purpose	`Velocity` is `VelFB` passed through a 3.5 Hz low pass filter.
Guidelines	When the measured velocity exceeds `Velocity`'s range `Velocity`'s value will be incorrect. See `VelFB` for an instantaneous indication of measured velocity that is accurate to higher speeds.

ZeroSpeedThresh

(NV Parameter, Float) f324

Units	RPM
Range	0 to 16,000
Default	Parameter value set before last NVSAVE. 930 Dialogue **New Set Up** or **Drive Set Up** sets this parameter to 30.
Purpose	`ZeroSpeedThresh` sets the threshold speed for turning the ZeroSpeed output on or off. If the absolute value of the motor speed is below `ZeroSpeedThresh` then the ZeroSpeed output will be turned on. If the absolute value of the motor speed is greater than `ZeroSpeedThresh` then the ZeroSpeed output will be turned off.
Guidelines	To use this function, you must map a BDIO point to the ZeroSpeed function. This function may be used as a general speed indicator instead of as a zero speed indicator.

Appendix A OC930 Communications Protocol

Introduction

This section describes the communications protocol that governs communications between the OC930 and any host device, typically a PC. This is the same protocol that 930 Dialogue uses for communicating with the OC930.

Message Format

To the Drive:

Field Name	Width in Bytes
Header	2
Target Address	2
Command Code	2
Message Length	4
Message Data	Variable
Checksum	2

Response Format

From the Drive:

Field Name	Width in Bytes
Header	2
Drive Address	2
Command Code	2
Message Length	4
Message Data	Variable
Checksum	2

Header

The message header sequence delimits the beginning of every message and every response. It is always the two characters ^A^B (<01><02>).

Target Address / Drive Address

The address field contains the ASCII Hex number corresponding to the address of the drive that the message is being sent to. For response messages, the address field contains the ASCII Hex number corresponding to the address of the drive generating the response.

Command Code

The command code field of a message from the PC to a drive contains an ASCII Hex number specifying the type of message being sent. The types of messages are:

05	Read a Variable from the Drive
06	Write a New Value to the Drive
0B	Execute a Command

Response Code

The response code field of a response message from a drive to the PC contains an ASCII hex number indicating whether or not the corresponding message from the PC was received properly by the drive. The values for the response code are:

00	Message received properly
01	Message contained an error

Message Length

The message length field contains an ASCII Hex number corresponding to the number of characters in the Data Field of the message or response. The message length field is organized as most significant byte first.

Message Data

The message data field is message dependent. Read Messages and Write Messages are described below.

Checksum

The checksum field contains the ASCII Hex number corresponding to the modulo-256 sum of the ASCII values of all characters in the message. An example checksum calculation for a Read Message is shown below:

Header	^A^B	01 + 02
Address	FF	70 + 70
Command Code	05	48 + 53
Data Length	0005	48 + 48 + 48 + 53
Data	D000E	68 + 48 + 48 + 48 + 69

The sum of all the ASCII characters in the message is 722. The modulo-256 value of 722 is 210. This is equal to the hex number D2. Therefore, the checksum for this message will be D2. The complete message is:

^A^B FF 05 0005 D000E D2

Read Messages

Read Messages are used to read the value of variables in the drive. To read the value of a variable, you must specify its type (Integer or Floating Point) and its Identifier Number.

Message Data

The data field of a Read Message consists of five characters. The first character specifies whether the variables is Integer (D) or floating point (C). The next four characters are the ASCII hex value for the variables Identifier Number.

Response Data

The data field for a Read Message Response consists of eight characters. The meaning of these characters is different depending upon whether the variable is integer or floating point.

If the variable being read is an integer variable then the eight bytes are the ASCII hex representation of the 32 bit value for the variable. The ASCII hex number is least-significant-byte-first format.

If the variable being read is a floating point variable then the eight bytes are the ASCII hex representation of the 32 bit IEEE-754 Single Precision Floating Point value for the variable.

Write Messages

Write Messages are used to change the value of a variable on the drive.

Message Data

The data field of a Write Message consists of 13 characters. One character is used to specify whether the variable is a Integer variable (D) or a floating point variable (C). The next four characters are used to specify the particular variables Identifier Number in 16 bit ASCII hex format. The remaining 8 characters specify the value to be written to the variable as described below.

If the variable being written is an integer variable then the eight bytes are the ASCII hex representation of the new 32 bit value for the variable. The ASCII hex number is least-significant-byte-first format. All eight bytes must be sent even if most of the characters are zero.

If the variable being written is a floating point variable then the eight bytes are the ASCII hex representation of the new 32 bit IEEE-754 Single Precision Floating Point value for the variable.

Response Data

The data field of the response to a Write Message will be the two characters 00 if the message was received properly.

Examples

Read FaultCode from the Drive.

FaultCode is an Integer Variable (D). FaultCode's Identifier Number is 000E. For this example, let's assume the value of FaultCode is 3.

Message:	^A^B FF 05 0005 D 000E D2
Response:	^A^B FF 00 0008 03 00 00 00 3A

Read Velocity from the Drive.

Velocity is a Floating Point Variable (C). Velocity's Identifier Number is 0006. For this example, let's assume the value of Velocity is 1000.1.

Message:	^A^B FF 05 0005 C 0006 C2
Response:	^A^B FF 00 0008 66067A44 69

Write New Value for ILmtPlus to the Drive

IlmtPlus is an Integer Variable (D). IlmtPlus's Identifier Number is 0005. For this example, let's write a new value of 50 percent (decimal 50 = 32 hex).

Message:	^A^B FF 06 000D D 0005 32 00 00 00 57
Response:	^A^B FF 00 0002 00 11

Write New Value for AccelLmt

AccelLmt is a Floating Point Variable (C). AccelLmt's Identifier Number is 276. This corresponds to 0114 in hexadecimal. For this example, let's write a new value of 10,000.

Message:	^A^B FF 06 000D C 0114 00401C46 74
Response:	^A^B FF 00 0002 00 11

Examples (cont'd)

Restore Parameters from NV Memory

NVLoad's Identifier Number is 256, which corresponds to 0100 in hexadecimal.

Message:	^A^B FF 0B 004 0100 86
Response:	^A^B FF 00 0002 00 11

Appendix B Configuring an OC930 as a Personality Module

Introduction

You can use the OC930 Communications Option Card two ways:

- As a communication card only
- As a Personality Module

Description

In the communications option card configuration, the OC930 is used only to allow serial communications to the SC900 Servo Drive. All non-volatile parameters are stored on the control card of the SC900. The OC930 is not required in order to operate the drive.

In the Personality Module configuration, all non-volatile parameters are stored on the OC930, and the NV memory on the SC900 control card is unconfigured. On power-up, the SC900 reads the value of each non-volatile parameter from the OC930. In this configuration, the OC930 must be installed in order to operate the drive. The `AInNull` mappable function saves the new `ADOffset` to the NV memory on the OC930 when configured as a Personality Module.

Procedure

To configure an OC930 as a Personality Module, perform the following steps:

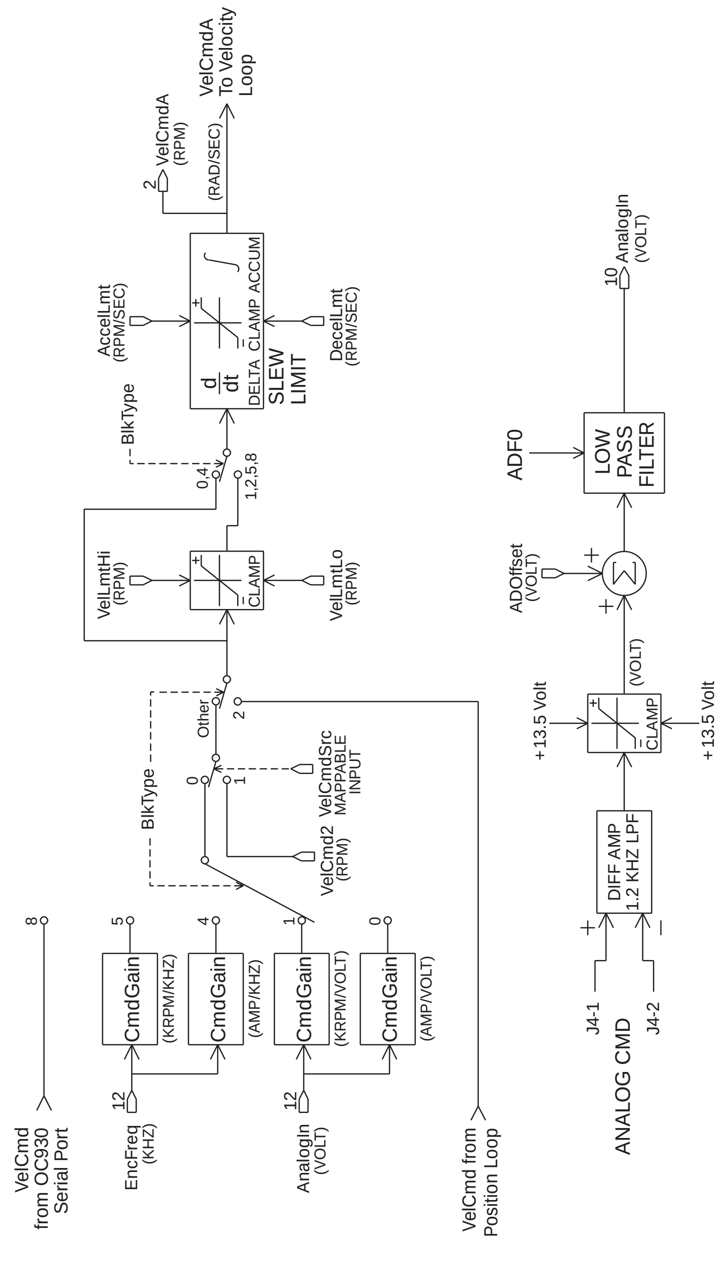
1. Select **Configure OC930 as PM** from the Drive drop-down menu.
2. A screen will appear to verify that you're sure you want to do this. Type **YES** and hit the **<Enter>** key.

Note: *Drive status LED will continue to flash* U - C until control ac power is cycled.

Page empty

Appendix C Control Block Diagrams

Figure 1



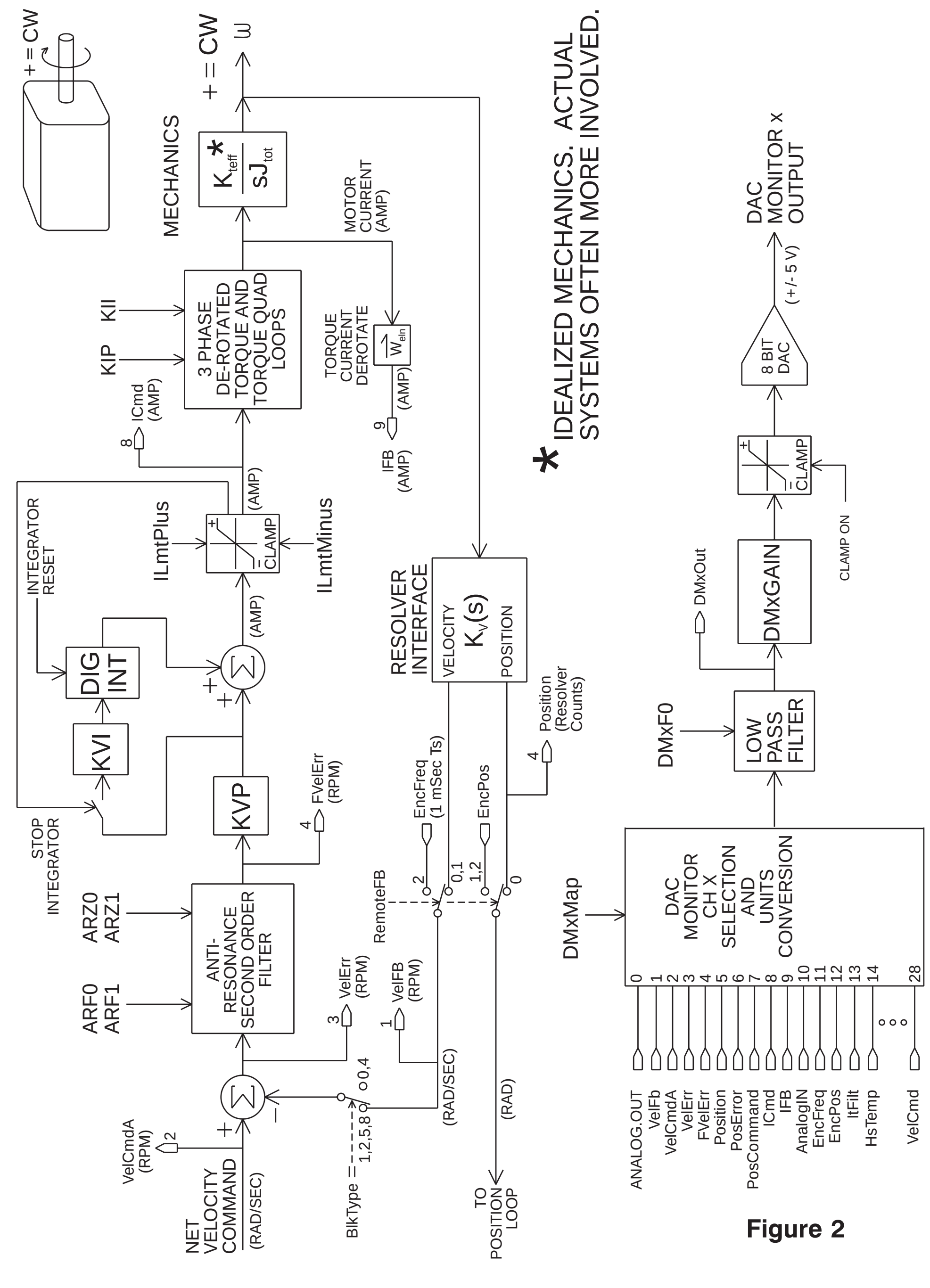
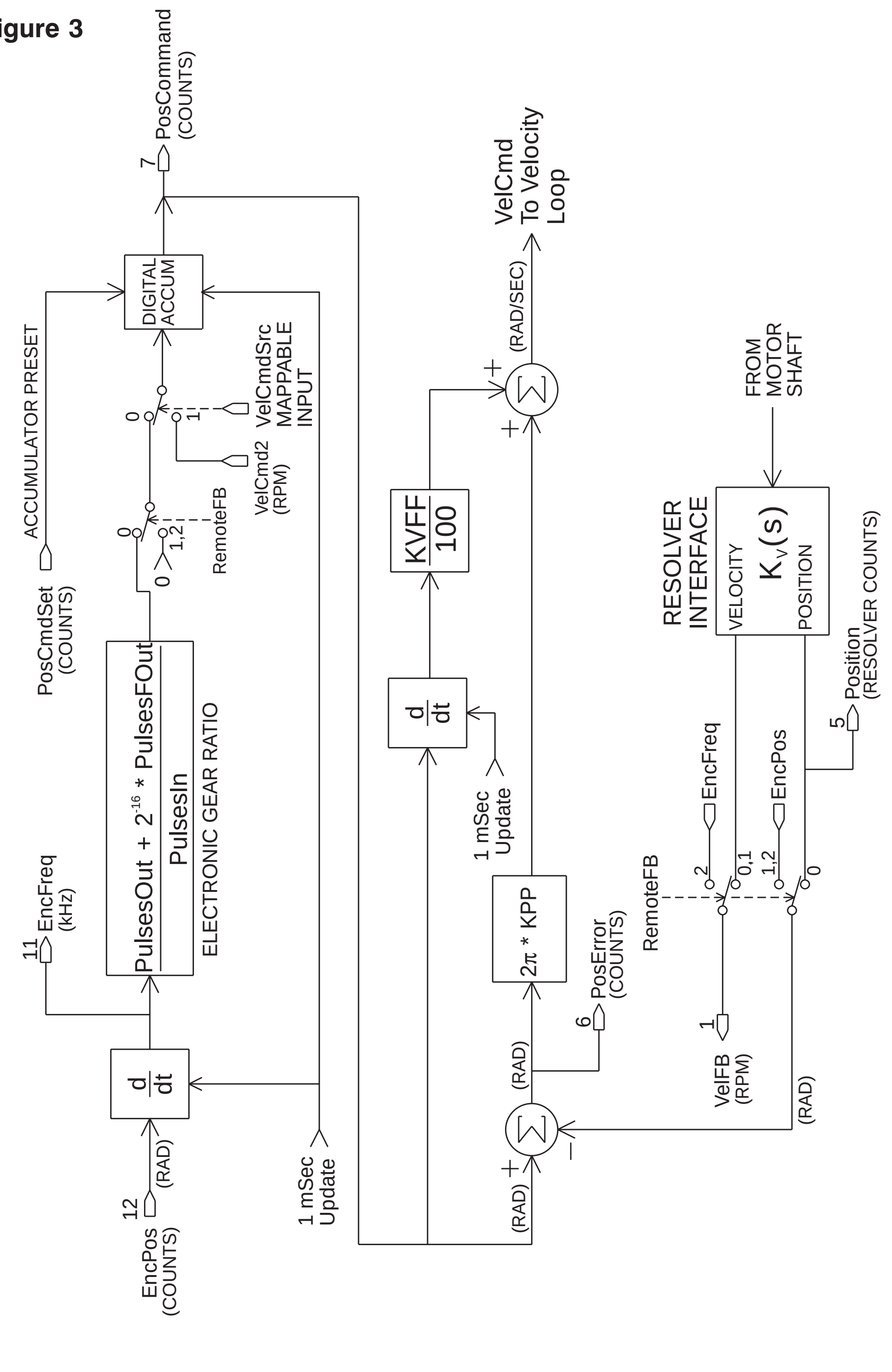

NET VELOCITY COMMAND
(RAD/SEC)
VelCmdA (RPM)
2
+
−
BlkType = 1,2,5,8
0,4
3 VelErr (RPM)
1 VelFB (RPM)
(RAD/SEC)
(RAD)
TO POSITION LOOP
ARF0
ARF1
ARZ0
ARZ1
ANTI-RESONANCE SECOND ORDER FILTER
STOP INTEGRATOR
KVP
4 FVelErr (RPM)
KVI
DIG INT
INTEGRATOR RESET
+
+
(AMP)
ILmtPlus
ILmtMinus
CLAMP
(AMP)
8 ICmd (AMP)
KIP
KII
3 PHASE DE-ROTATED TORQUE AND TORQUE QUAD LOOPS
TORQUE CURRENT DEROTATE
$\overrightarrow{W_{eln}}$
9 (AMP)
IFB (AMP)
MOTOR CURRENT (AMP)
MECHANICS
$\frac{K_{teff}}{sJ_{tot}}$ *
+ = CW
ω
+ = CW
RemoteFB
2 EncFreq (1 mSec Ts)
0,1
1,2 EncPos
0
RESOLVER INTERFACE
VELOCITY
$K_V(s)$
POSITION
4 Position (Resolver Counts)
* IDEALIZED MECHANICS. ACTUAL SYSTEMS OFTEN MORE INVOLVED.
DMxMap
DAC MONITOR CH X SELECTION AND UNITS CONVERSION
0 ANALOG.OUT
1 VelFb
2 VelCmdA
3 VelErr
4 FVelErr
5 Position
6 PosError
7 PosCommand
8 ICmd
9 IFB
10 AnalogIN
11 EncFreq
12 EncPos
13 ItFilt
14 HsTemp
28 VelCmd
DMxF0
LOW PASS FILTER
DMxOut
DMxGAIN
CLAMP
CLAMP ON
8 BIT DAC
(+/- 5 V)
DAC MONITOR x OUTPUT
Figure 2

Figure 3



PosCommand (COUNTS)
7
ACCUMULATOR PRESET
DIGITAL ACCUM
PosCmdSet (COUNTS)
VelCmdSrc MAPPABLE INPUT
VelCmd2 (RPM)
RemoteFB
0
1
1,2
PulsesOut + 2^-16 * PulsesFOut
PulsesIn
ELECTRONIC GEAR RATIO
11 EncFreq (kHz)
d/dt
12 EncPos (COUNTS)
(RAD)
1 mSec Update
VelCmd To Velocity Loop
(RAD/SEC)
KVFF/100
2π * KPP
6 PosError (COUNTS)
RESOLVER INTERFACE
VELOCITY
POSITION
Kv(s)
FROM MOTOR SHAFT
5 Position (RESOLVER COUNTS)
EncFreq
EncPos
2
0,1
1 VelFB (RPM)

Page empty

Appendix D Troubleshooting and Fault Diagnostic Guide

Introduction

The following table of problems, causes and appropriate actions complements the list of SC900 fault codes found in Section 7 (page 7-56).

Problem & Status Display	Possible Cause	Action
Velocity Feedback Over Speed Fault (Blinking 1)	Loose or open circuit wiring to the resolver feedback connector J3.	Check connections. Tighten TB screws on J3.
	VelFB > 1.5 * [Max Of (|VelLmtLo| or |VelLmtHi |)] or VelFB > 21,000 RPM	Limit `VelCmd` appropriately. **Note:** *For Encoder velocity feedback (`CommSrc` = 1 or 2) check that EncIn is set properly to correctly scale the `VelFB` units.*
Motor Over Temperature Fault (Blinking 2)	Loose or open circuit wiring to motor PTC thermal sensor (J3-8 and J3-9).	Check connections. Tighten TB screws on J3-8 and J3-9.
	High ambient temperature at motor.	Lower ambient temperature.
	Insufficient motor heat sinking from motor mounting.	Increase motor mounting heat sinking.
	Operating above the motor's continuous current rating.	Operate within continuous torque rating.
	Inoperative motor cooling fan.	Return to factory for fan replacement.

Problem & Status Display	**Possible Cause**	**Action**
Drive Over Temperature Fault (Blinking 3) **Note:** *See `HSTemp`, `ItFilt`, `ItThresh` , and `ItF0` for information on measuring thermal margin in an application.*	High drive ambient temperature.	Lower ambient temperature to below 50°C (60°C if I_{Out} is derated)
	Restriction of cooling air due to insufficient space around unit.	Provide sufficient cooling space.
	Operating above the drive's continuous current rating.	Operate within continuous current rating.
	Inoperative cooling fan.	Return to factory for fan replacement.
Drive I*t Fault (Blinking 4)	Mechanically jammed motor. Motion profile accelerations too high. Machine load on the motor increased, e.g. friction increased. Problem with wiring between drive and motor yielding improper motion. Drive and/or motor under sized for application.	Ensure motor shaft is not jammed. Change profile or load. **Note:** *See `HSTemp`, `ItFilt`, `ItThresh` , and `ItF0` for information on determining the continuous current margin in an application.*
Line-Neutral Fault (Blinking 5)	Short circuit in the motor and/or drive-motor cable.	Check cable.
	Motor power cable is longer than the data sheet specification by enough to cause excessive motor line to earth ground/neutral capacitance.	Shorten power cable.

Problem & Status Display	Possible Cause	Action
Control ± 12V Supply Under Voltage Fault (Blinking 6)	Insufficient control ac voltage on voltage on J1-5 to J1-6.	Check voltage with meter.
	Internal drive failure.	Contact distributor.
Output Over Current or Bus Over Voltage Fault (Blinking 7)	Motor power wiring (J2-2, 3, or 4) short circuit.	Check for short.
	Line-to-line or line-to-ground/neutral internal motor winding short circuit.	Check for short.
	Insufficient motor inductance for output over current faults.	Check motor inductance against drive minimum specification.
	Motor ac power input voltage too high.	Reduce ac input voltage to within specification.
	Disconnected regeneration resistor on J5, or external regeneration resistor ohmage too small for Bus Over Voltage fault.	Check the connections on J5.
Shunt Regulator Overload (Blinking 9)	Excessive regen in application.	
	Improper external regen wiring or components on J5.	Check connections on J5.

Problem & Status Display	Possible Cause	Action
Bus Over Voltage Detected By DSP (Blinking A)	Actual bus over voltages are usually, but not always, detected and displayed as a blinking 7 fault, see that entry for more information.	
	Drive improperly set up in the factory.	Contact distributor. Return to factory.
Auxiliary +5V Supply Fault (Blinking b)	Short circuited wiring on the output.	Check for short.
	Load exceeds the current rating of this supply.	Reduce load.
Processor Throughput Fault (Solid E)	Drive hardware failure.	Cycle control power.
	Drive software bug.	Cycle control power.
Power Up Self Test Failure (Blinking E)	Internal drive software error.	See `ExtFault` for further information about the exact failure. The drive control power must be cycled to clear this fault.
Bus Voltage Fault (Alternating E, 1)	Motor power bus voltage dropped below `VBusThresh`.	Check the measured bus voltage `VBus` and the fault threshold `VBusThresh`.
Ambient Temperature Too Low Fault (Alternating E, 2)	Ambient temperature is below drive specification.	Raise ambient temperature above 0°C.
	Drive's internal temperature sensor has a wiring problem.	Contact distributor. Return to factory.

Problem & Status Display	**Possible Cause**	**Action**
Encoder Commutation Alignment Fault (Alternating E, 3)	Problems with encoder feedback wiring to J4.	Check wiring.
	Load inertia more than 100 times the motor inertia leading to settling times long compared to the 2.0 second alignment.	Artificially extend the alignment time by pulsing the hardware enable (J4-6).
Firmware Version Incompatible with NV Memory Version (Alternating E, 4)	OC930-001-01 (drive software upgrade card) was used to set up an old drive and then removed.	Re-install OC930-001-01 (drive software upgrade card).
Firmware Version Incompatible with Hardware (Alternating E, 5)	Non-volatile parameter memory was written with a newer drive software than the drive just powered up with.	Check the drive software version via the `FwV` status variable. Contact factory for upgrade details.
Attempted to Configure with Drive Enabled (Alternating E, 6)	Unconfigured drive (Status LED alternates U, C) was fully configured with the drive motor power enable active.	This fault can be reset or the control ac power cycled to get the drive-motor operating.
Two `AInNull` Events Too Close Together (Alternating E, 7)	`AInNull` function was re-activated too soon after going inactive.	Ensure at least 0.5 second pause between `AInNull` activations. Check for switch bounce.

Problem & Status Display	**Possible Cause**	**Action**
Excessive Position Following Error Fault (Alternating F, 1)	Motor is either stalled or partially jammed.	Ensure motor is not jammed.
	`PosErrorMax` is set too sensitive.	Increase the value of `PosErrorMax`.
Parameter Checksum Error Fault (Alternating F, 3)	Glitch while last saving the NV parameters.	Download parameters again and `NVSave`.
	Option card has corrupted NV memory contents.	See `ExtFault` status variable to determine whether NV memory corruption is inside the drive or on the option card.
	Hardware problem with the NV memory.	Cycle power.

Index