

ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

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

Visit our website - *Click HERE*

Compumotor

ZX/ZXF Indexer Drive
User Guide

Compumotor Division
Parker Hannifin Corporation
p/n 88-011631-01 G **May 1997**

Parker
Motion & Control

IMPORTANT

User Information

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

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

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

© ***Compumotor Division of Parker Hannifin Corporation, 1997***
—All Rights Reserved—

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

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

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

North America and Asia:
Parker Hannifin
Compumotor Division
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070
Fax: (707) 584-3793
FaxBack System: (800) 936-6939
BBS: (707) 584-4059
E-Mail: tech_help@cmotor.com

Europe (non-German speaking):
Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
441-202-690-911
441-202-600-820

Germany, Austria, Switzerland:
Hauser Elektronik GmbH
Robert-Bosch-Str. 22
D-77656 Offenburg
Germany
49-781-509-300
49-781-509-176

Parker
Motion & Control

USER GUIDE CHANGE SUMMARY

The following is a summary of the primary changes to this user guide since the last version was released. This user guide, version 88-011631-01**G**, supersedes version 88-011631-01**F**.

The entire user guide has been changed according to the new Compumotor user guide styles, format, and illustration standards. Also, the chapters have been renumbered and reorganized. Technical changes to each chapter are summarized below.

Chapter ① Introduction is unchanged.

Chapter ② Getting Started is unchanged.

Chapter ③ Installation

The following information was updated or changed:

- ❑ Programmable Input I/O drawing was changed

Chapter ④ Application Design

Chapter ⑤ ZXF Following is unchanged.

Chapter ⑥ Z Series Shunt Regulator

Chapter ⑦ Hardware Reference

The following information was updated or changed:

- ❑ Typical Input & Output Circuit drawing was corrected
- ❑ Motor vendor information was deleted
- ❑ Corrected Data (Voltage Constant on Line 2 of Motor Specifications

Chapter ⑧ Maintenance & Troubleshooting

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

How To Use This User Guide iv
Assumptions iv
Installation Process Overview iv
Installation Procedures iv
Contents of This User Guide iv
Conventions v
Commands v
Related Publications v

① INTRODUCTION 1
Product Description 1
Product Features 1
Following Option (ZXF) 2
Theory of Operation 2

② GETTING STARTED 5
What You Should Have 5
Drive/Motor Configuration 5
ZX/ZXF Front Panels 6
Check-Out Procedure 6
ZXF Option 11

③ INSTALLATION 13
Environmental Considerations 13
Complete System Configuration 13
Mounting
Drive 13
Panel Layout 15
Motor Mounting 16
System Wiring 26
Connections 26
Resolver Connections 28
I/O[1] Connections 29
I/O[2] Connections 33
ZXF Encoder Connection 34
Line Power Connections 34
Installation Verification 36

④ APPLICATION DESIGN 41
Basic Servo Tuning Theory 41
Servo Tuning 43
Alphanumeric Display and Pushbuttons 45
Motion Profile Application Considerations 48
Positional Accuracy vs. Repeatability 48
Move Times—Calculated vs Actual 48
Registration 51
Jogging the Motor 52
Backlash Compensation 53
Defining a Home Location 53
Homing to a Switch 54
Creating Motion Programs and Sequences 55
Sequence Commands 55
Creating and Executing Sequences 57
Subroutines 57
Asynchronous Events—FAULT and WHEN 58
Power-Up Sequence Execution 60

Sequence Debugging Tools 61
High-Level Programming Tools 65
Variables 65
Math Operations 67
Complex Branching and Looping 68
Conditional Branching 70
Interfacing to the ZX 76
Programmable Inputs and Outputs 76
Switches 77
Thumbwheel Interface 80
Selecting Sequences with the Thumbwheel Module 83
PLC Operation 85
Interfacing with a PLC 85
Front Panel Operation 87
Remote Panel Operation 87
Host Computer Operation 87
ZX Application Examples 88
Application Development Approach 88
Application Example #1 89
Application Example #2 92
Application Example #3 93

❁ ZXF FOLLOWER 95
What is Following? 95
Types of Following 96
Velocity Following 96
Setting Up Velocity Following 97
Preset Moves In Velocity Following 98
Position and Velocity Following 99
Primary Axis at Rest 99
Primary Axis in Motion 100
Following Acceleration 101
How Following Acceleration Works 103
Decelerating 104
Position and Velocity Following 104
Following Acceleration Application Bottle Filling 105
Recede and Advance While Following 109
Advance Following Example 109
FP Delay Examples 114
Recede vs. Advance 118
Cam Following 119
Synchronization 123
Other Following Features 126
Registration in Following Mode 126
Jogging in Following Mode 127
Changing Following Ratio 128
Following a Step and/or Direction Signal 129
Following Equation and Command Summary 129
Following Command Summary 129
Following Equation Summary 130

⑥ Z SERIES SHUNT REGULATOR 135
What Is A Shunt Regulator? 135
Shunt Regulator Installation 135
Selecting a Z Series Shunt Regulator 136

⑦ HARDWARE REFERENCE 141
ZX600 Electrical Specifications 141
Input Power 141
Output Power 141
ZX600 Motor/Drive Configuration 141
Technical Data ZX600 Series 142
Positional Repeatability 142
Positional Accuracy 142
Motor/Drive Compatibility 142
Motor Brakes 143
Motor Data 143
Speed/Torque Curves 151
ZX800 Electrical Specifications 153
Input Power 153
Output Power 153
Motor/Drive Configuration 153
Technical Data ZX800 Series 153

Positional Repeatability....154
Positional Accuracy....154
Motor/Drive Compatibility....154
Motor Brakes....154
Motor Data....154
Speed/Torque Curves....161
ZX900 Electrical Specifications....163
Input Power....163
Output Power....163
ZX900 Motor/Drive Configuration....163
Technical Data ZX900 Series....163
Positional Repeatability....164
Positional Accuracy....164
Motor Drive Compatibility....164
Motor Brakes....164
Motor Data....164
Speed/Torque Curves....169
I/O Data & Descriptions....170
Factory Default Settings....172
RS-232C Communications....172
ZX Device Address....172
Hardware Interfaces....172
Limits....172
Motion Parameters....172
MAINTENANCE & TROUBLESHOOTING....173
Motor Maintenance....173
ZX Maintenance....173
Spare Parts....173
Battery....173
Troubleshooting....174
Diagnostic LEDs....174
ZX Warning Codes....175
ZX Fault Codes....175
RS-232C....176
Software Debugging Tips....176
Reducing Electrical Noise....178
Common Mistakes....178
Returning The System....178
APPENDIX....181
Command Listing....181
INDEX....183

How To Use This User Guide

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

Assumptions

This user guide assumes that the user has a fundamental understanding of the following information.

- ❑ Basic electronic concepts (voltages, switches, current, etc.)
- ❑ Basic motion control concepts (torque, velocity, distance, force, etc.)

Installation Process Overview

To ensure trouble-free operation, you should pay special attention to the environment in which the Z Drive equipment will operate. Environmental conditions include the layout, mounting, and wiring and grounding practices used. These recommendations are intended to help you easily and safely integrate the Z Drive into your facility.

Installation Procedures

Before you attempt to install this product, you should complete the following steps:

① Review this user guide. Become familiar with the user guide's contents so that you can quickly find the information you need.

② Develop a basic understanding of all system components, their functions, and interrelationships.

③ After you have read Chapter ③ and clearly understand what must be done to properly install the system, begin the installation process. Do not deviate from the sequence or installation methods provided.

④ Before you customize your system, check all of the system functions and features to ensure that you have completed the installation process correctly.

The successful completion of these steps will prevent subsequent performance problems and allow you to isolate and resolve any potential system difficulties before they affect your system.

Contents of This User Guide

This user guide contains the following information.

Chapter ① Introduction — This chapter provides a description of the product and a brief account of its features.

Chapter ② Getting Started — This chapter contains a detailed list of items you should have received with your ZX shipment. It will help you become familiar with the system and ensure that each component functions properly.

Chapter ③ Installation — This chapter provided instructions for you to properly mount the system and make all electrical and non-electrical connections. Upon completion of this chapter, your system should be completely installed and ready to perform basic operations.

Chapter ④ Application Design — This chapter provides additional information that will help you customize the system to meet your application's needs. Important application considerations are discussed. Sample applications are provided.

Chapter ⑤ ZXF Follower — This chapter discusses the following capabilities of the ZXF Drive.

Chapter ⑥ Z Series Shunt Regulator — This chapter contains information on the shunt regulator option for ZX/ZXF Drive/Indexers. Installation instructions, dimensions, and selection criteria (400W version vs. 800W version) are included.

Chapter ⑦ Hardware Reference — This chapter contains information on system specifications (dimensions and performance). This chapter may be used as a quick-reference tool for proper I/O connections.

Chapter 8
Maintenance & Troubleshooting

This chapter describes Compumotor's recommended system maintenance procedures. It also provides methods for isolating and resolving hardware and software problems. Diagnostic error codes are listed.

Conventions

To help you use this user guide effectively, conventions used throughout this user guide are explained here.

Commands

All commands that you are instructed to enter are shown in capital letters. The symbol **>**, is the ZX Indexer/Drive command prompt. The command is shown in boldface. A delimiter (space or carriage return) is required after each command. A description is provided next to each command example.

Command	Description
> CCA2Ø.ØØ	Sets average current limit to 20.00A

The system ignores command syntax that is not within the valid range for a specific command. A **?** prompt will be returned by the drive when the last command entered was not understood, or a parameter limit was exceeded.

Related Publications

The following publications may be helpful resources.

- Seyer, Martin. RS-232C Made Easy: Connecting Computers, Printers, Terminals and Modems. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1984
- Current Parker Compumotor Motion Control Catalog
- Manual for the IBM or IBM-compatible computer that you may use with the Z Drive
- Schram, Peter (editor). *The National Electric Code Handbook (Third Edition).* Quincy, MA: National Fire Protection Association

C H A P T E R ①

Introduction

The information in this chapter will enable you to:

- ❏ Understand the product's basic functions, features

Product Description

The ZX Series servo system incorporates all the functions of a high-level indexer with the power of a Z servo system. The ZX is a brushless servo positioning system that includes a brushless servo motor, brushless resolver feedback, a digital signal processor (DSP) based closed-loop drive amplifier, and a powerful X language indexing system.

The ZX has Compumotor's latest X language enhancements. It offers registration and complex segmented-move capabilities. The ZX provides PLC communication and thumbwheel input functionality with its seven inputs and four outputs. The ZXF also provides position and velocity following capabilities.

☞ ***Helpful Hint:***
ZX's are configured at the factory as packaged systems that compensate for your typical load and performance requirements.

The ZX's servo drive uses a DSP and a sophisticated servo control algorithm for superior closed-loop performance. The control loop is closed around the motor shaft. The motor's resolver provides position feedback. All servo performance parameters are stored in battery-backed memory. You do not have to make analog tuning potentiometer adjustments that conventional systems require.

The ZX's power amplifier section uses a rugged, bipolar 7 kHz pulse width modulation (PWM) sinusoidal current control scheme. This construction improves reliability, power regulation, and low-speed smoothness.

Product Features

The ZX provides the following flexible programming features.

- ❏ One registration input that is given the highest priority
- ❏ Calculation delay times for motion as low as 500 µs
- ❏ A complex motion profiling system that allows you to:
  - Change velocity based on distance without stopping
  - Change distance, or turn on outputs on-the-fly
- ❏ High-level programming commands such as:
  - `IF/THEN/ELSE/WHILE`
  - `REPEAT/UNTIL`
  - `GOTO AND GOSUB`
- ❏ Complex evaluations such as checking input levels, error conditions, boolean evaluations, and variable comparisons for basic programming branching decisions can be made
- ❏ An output can be configured to provide pulse and direction to second axis to control velocity and distance
- ❏ PLC functionality and interfacing capability using the 7 inputs and 4 outputs

The ZX provides the following hardware and performance features.

- ❏ Brushless servomotor
- ❏ Brushless resolver feedback
- ❏ Torques to 9,000 oz-in continuous (18,000 oz-in peak)
- ❏ User programmable resolutions from 200 - 65535 steps/rev
- ❏ Multi-processor control: no drift, no analog pots to adjust
- ❏ Fan-cooled compact drive enclosure
- ❏ 7 kHz PWM switching frequency
- ❏ Analog (±10V) output monitor for either velocity or torque
- ❏ Various parameters factory-set and stored in battery-backed RAM (random access memory)
- ❏ High-noise immunity due to optical isolation and brushless resolver technology
- ❏ Simple pushbutton adjustment of servo gains
- ❏ Alphanumeric display for fault and servo data

Following Option (ZXF)

The ZXF option can perform velocity following and distance following moves at a *following ratio*. The ZXF can follow from an incremental encoder.

You can program the ZXF for following applications with its command language and report back/verification feature. You can enter following ratios via thumbwheels and change them on-the-fly.

You can perform preset moves at a specified velocity ratio. You can perform registration moves while in the Following mode. Registration moves can either follow at a ratio of the master velocity or be executed in the standard motion modes. The ZXF can jog the motor in Following mode to help set up a system. You can change the following ratio incrementally with the **UP** and **DOWN** pushbuttons on the front panel.

You can use the ZXF's special Synchronization mode to compensate for system errors (e.g., stretching in a web processing system).

Following Option Features

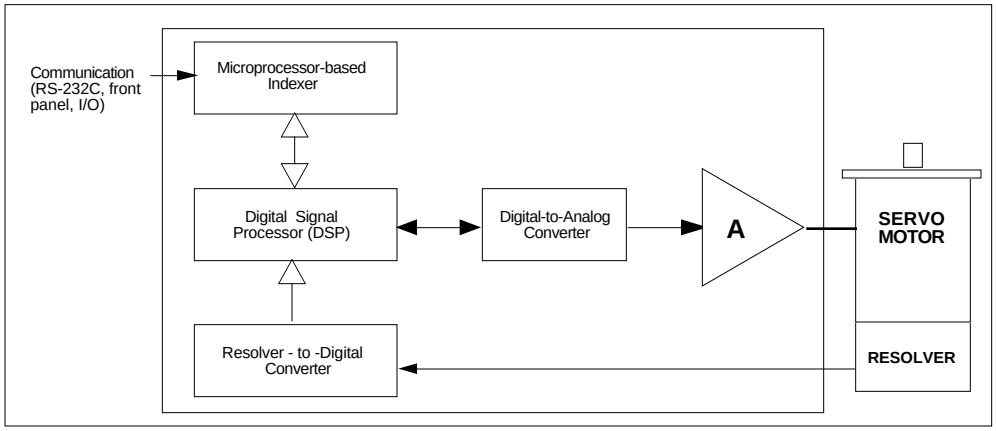
The Model ZXF provides the following additional features:

- ❏ Controls a speed based on a ratio of a primary axis speed
- ❏ Makes preset moves at a velocity ratio of a primary axis
- ❏ Synchronizes speed to a primary axis based on registration marks on material
- ❏ Changes following ratio and other functions based on the encoder position of a primary axis (*Cam Following*)

Theory of Operation

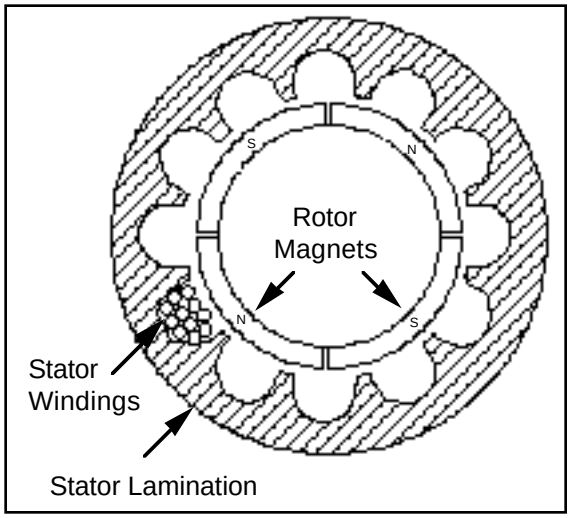
The ZX 's microprocessor-based indexer generates position set point commands that are sent to the ZX's DSP to control the drive. The ZX's DSP reads its actual position from the motor's resolver and compares it with the commanded set point. Any difference between the actual and commanded position creates a positional error. The drive converts this error into a commanded torque to correct the positional error.

The figure below illustrates three major components of the ZX Indexer/ Drive digital servo system: the servo motor, the drive, and the resolver.



ZX Indexer/Drive Digital Servo System

The ZX motor family consists of brushless, 3-phase, AC motors. The figure below illustrates the basic construction of the ZX servo motor. The permanent magnets are securely held in place by metal bands and composite fiber materials to allow high-speed performance. The rotors are precision-balanced, which provides low- and high-speed smoothness. The windings are located in the outer portion of the motor (stator). This *inside-out* construction allows better heat dissipation than conventional brush-type motors. As a result, higher continuous torque and horsepower ratings are achieved for a given motor size.



4-Pole Brushless Motor

B*rushless DC* motors produce a trapezoidal EMF (Electro-Motive Force) waveform. ZX Drive motors are *brushless AC* motors that produce sinusoidal back EMF waveforms. These motors provide smoother low-speed operation than trapezoidal motors.

The ZX Drive is a 3-phase inverter that controls the amplitude, frequency, and phase of each of the three motor stator currents. Controlling the current amplitude relates directly to the magnitude of the torque being generated. Since the motor is *synchronous*, controlling the frequency of the stator currents controls the mechanical rotation of the shaft. The following equation describes this relationship.

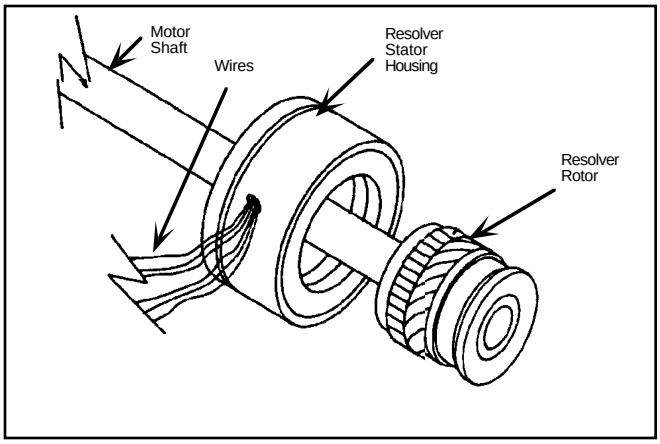
$\omega_e = \omega_r * (P/2)$

ω_e = Electrical frequency of the stator currents

ω_r = Mechanical frequency of the shaft

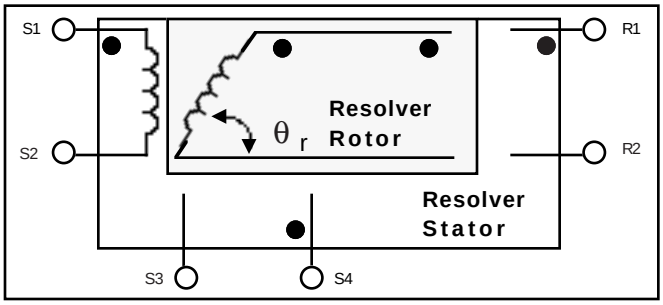
(P/2) = Number of motor pole pairs

Controlling the phase of each of the motor's stator currents ensures balanced 3-phase operation and minimizes torque ripple. Actual motor currents are sensed, and the current is adjusted using pulse width modulation. A single-speed, brushless, pancake-type resolver provides feedback. The figure below shows the resolver mounted directly to the motor shaft. This eliminates the need for internal coupling. The resolver stator windings are mounted to the motor housing.



Resolver Mechanical Drawing

The figure below shows the resolver's internal windings. S1/S2 and S3/S4 are stator windings. R1 and R2 are rotor windings. When a voltage is excited in the rotor winding, a voltage is induced in the stator windings. The stator (S1/S2 and S3/S4) phase voltages are compared to obtain a digital value (θ**r**) representing the motor shaft's physical position.



Resolver Electrical Drawing

C H A P T E R ②

Getting Started

The information in this chapter will enable you to:

- ❑ Verify that each component of your system has been delivered safely
- ❑ Properly configure and *check out* the system

What You Should Have

Inspect your ZX shipment upon receipt for obvious damage to its shipping container. Report any damage to the shipping company immediately. Compumotor cannot be held responsible for damage incurred in shipment.

Part/Quantity	Available Types
ZX or ZXF Drive	ZX600 (120VAC or 240VAC) or ZX900 (240VAC)
Motor	ZX605, ZX606, ZX610, ZX620, ZX630, ZX640, ZX805, ZX806, ZX810, ZX820, ZX830, ZX840, ZX910, ZX920, ZX930, or ZX940
Motor Cable	25 ft., 50 ft., or 100 ft.
Resolver Cable	25 ft., 50 ft., or 100 ft.
ZX Drive User Guide	88-011631-01 D
ZX Software Reference Guide	88-011798-01 E

Ship Kit (ZX System)

Part/Quantity	Available Types
ZX Drive or ZXF (1)	ZX600 (120VAC or 240VAC) or ZX900 (240VAC)
ZX Drive User Guide	88-011631-01 D
ZX Software Reference Guide	88-011798-01 E

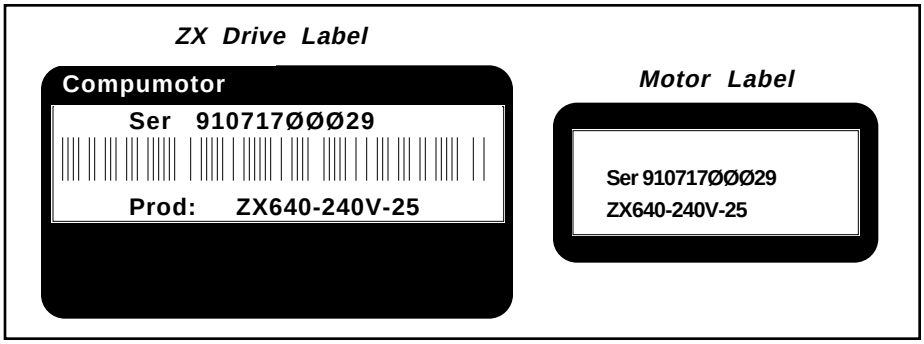
Ship Kit (ZX Indexer/Drive Only)

Drive/Motor Configuration

Compumotor configures ZXs according to the motor that you order. *If you purchased the ZX to run 600 Series motors (ZX605, ZX606, ZX610, ZX620, ZX630, and ZX640), it will not run 800 Series (ZX805, ZX806, ZX810, ZX820, ZX830, and ZX840) or ZX900 series motors (ZX910, ZX920, ZX930, and ZX940). All ZX Series drives* ***must*** *be run with the same ZX Series motors.* You can determine which motor you have by checking the model numbers on the drive and motor labels.

The ZX600 and ZX800 are available in a ***120VAC version*** and a ***240VAC version***. The ZX900 Drive is available in a **240VAC** version only.

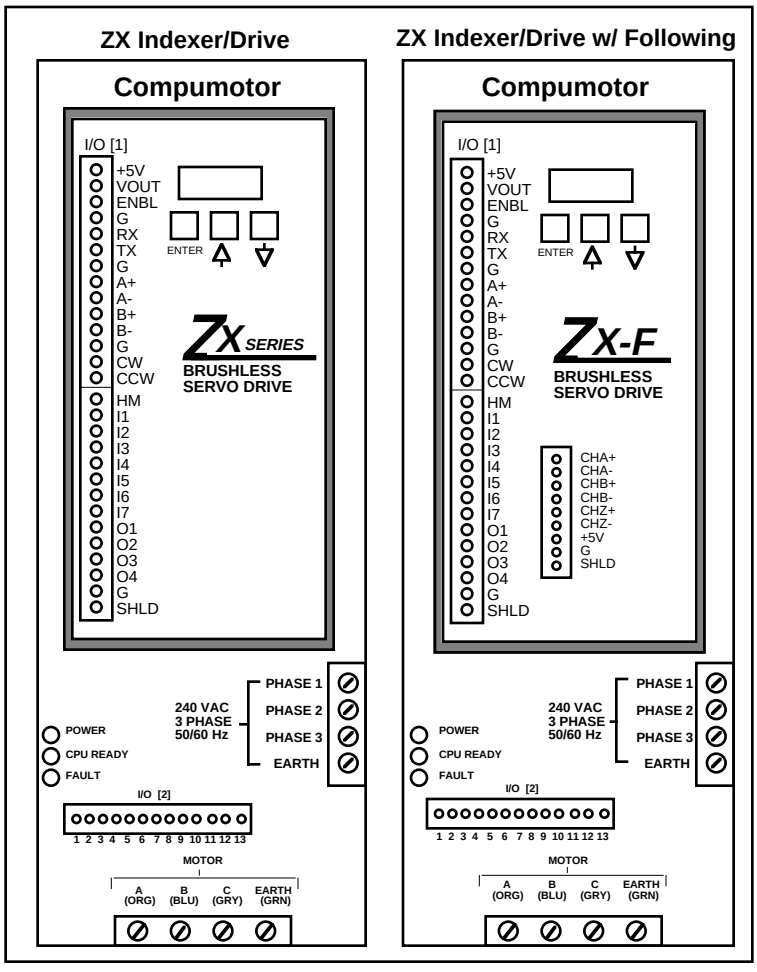
☛ Be sure you have the drive version you ordered before you connect power to the drive. The model numbers on the drive and motor labels identify the 3-phase input power level (120V or 240V). ***The model numbers must match***.



Product Labels

ZX/ZXF Front Panels

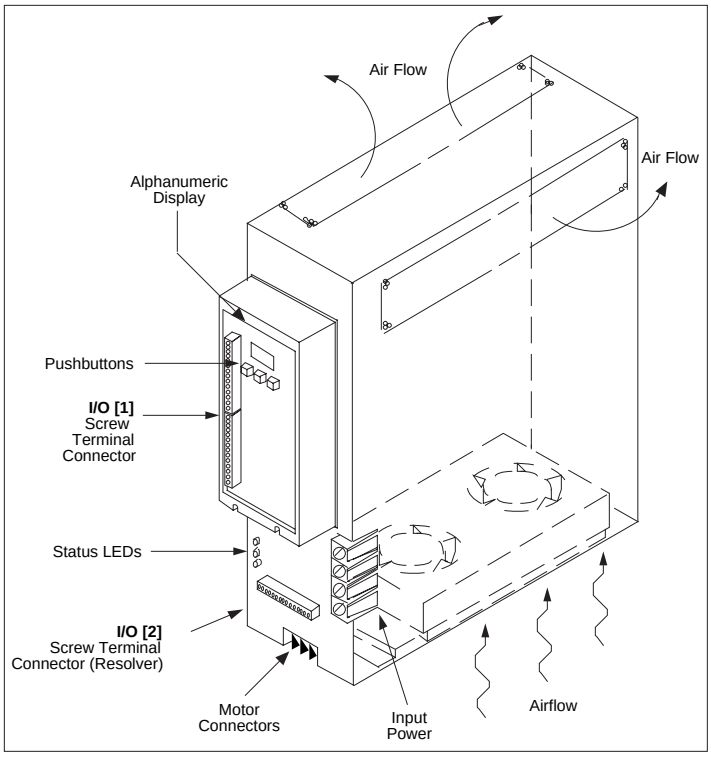
The front panels for the ZX and ZXF are different. ***The ZX's front panel is used throughout this user guide to demonstrate wiring instructions.*** *The ZXF's front panel will be shown when following functions are described.*



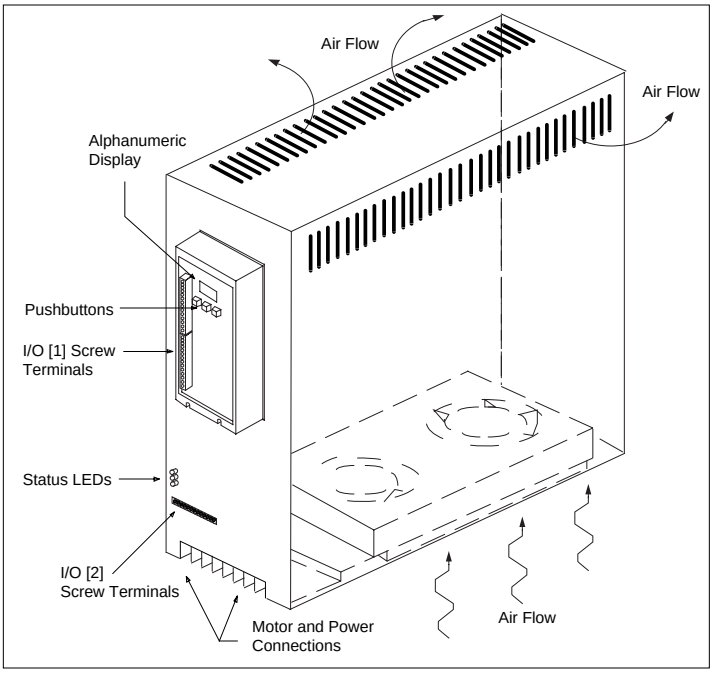
Front Panels—ZX and ZXF

Check-Out Procedure

This section provides a basic ***bench test*** of the ZX. **You must complete the steps in this chapter before you permanently install the ZX and connect the motor to its intended load.**



ZX600/ZX800 Bench Test



ZX900 Bench Test

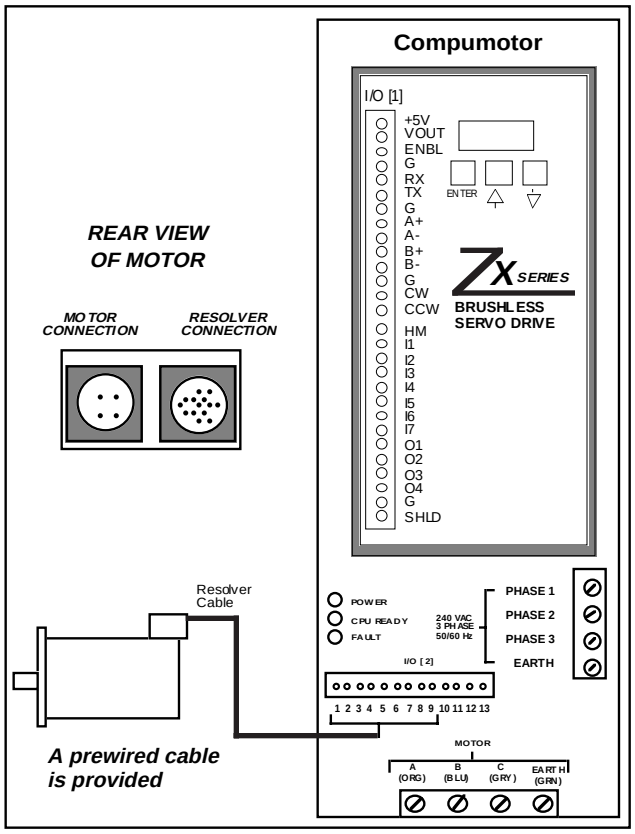
Tools

To complete the Check-Out Procedure, you must have the following tools:

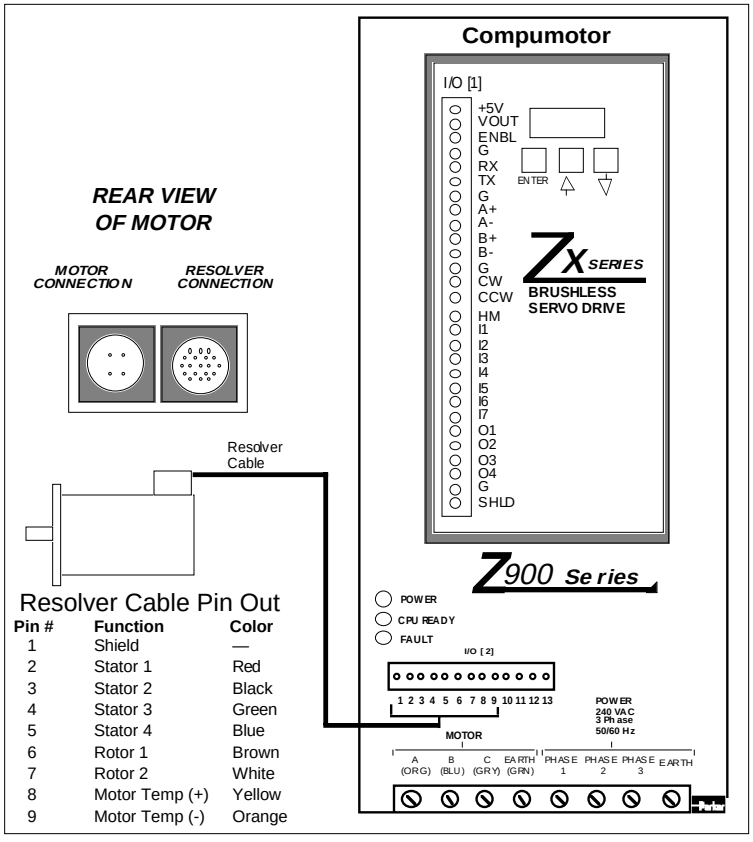
- ❏ Phillips-head screwdriver
- ❏ Large flat blade screwdriver
- ❏ Small flat blade screwdriver
- ❏ Power cable

Step ①

Connect the resolver cable between the ZX **I/O[2]** connector and the servo motor as shown in the following figures (for ZX600/ZX800 or ZX900).



ZX600/800 Resolver Cable Connection



Resolver Cable Pin Out

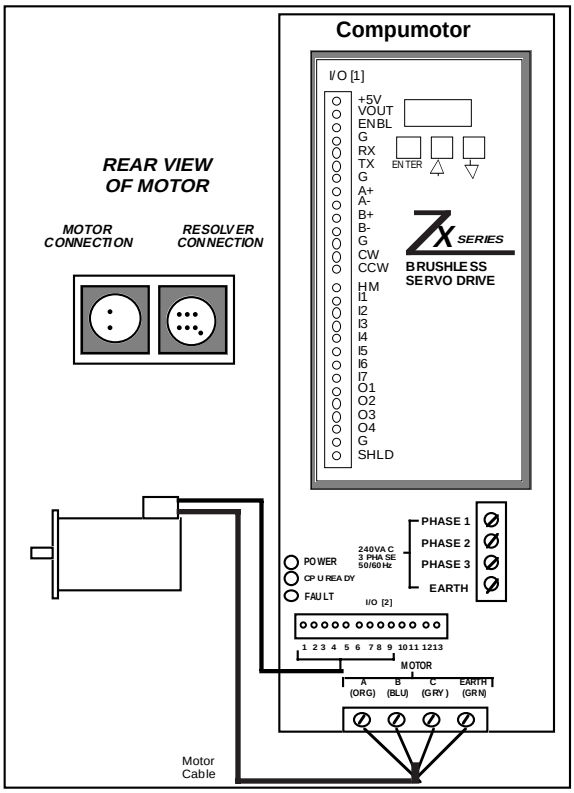
Pin #	Function	Color
1	Shield	—
2	Stator 1	Red
3	Stator 2	Black
4	Stator 3	Green
5	Stator 4	Blue
6	Rotor 1	Brown
7	Rotor 2	White
8	Motor Temp (+)	Yellow
9	Motor Temp (-)	Orange

ZX900 Resolver Cable Connection

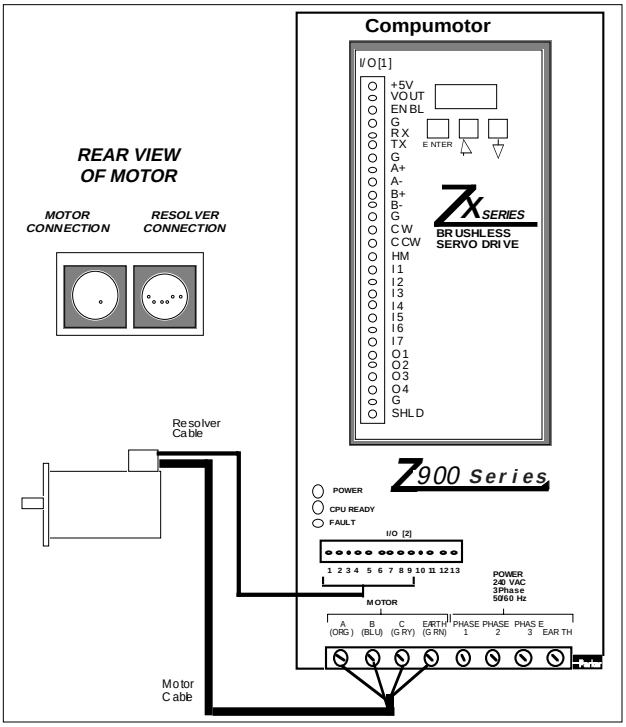
Step ②

Connect the cable between the ZX motor connector and the servo motor. ***Match the color of the wire with the proper terminal.***

> ***WARNING***
>
> Never disconnect the motor cable with the power on. The motor produces lethal voltages. Be sure the motor is properly grounded to reduce the chance of electrical shock.



ZX600/800 Motor Cable Connection

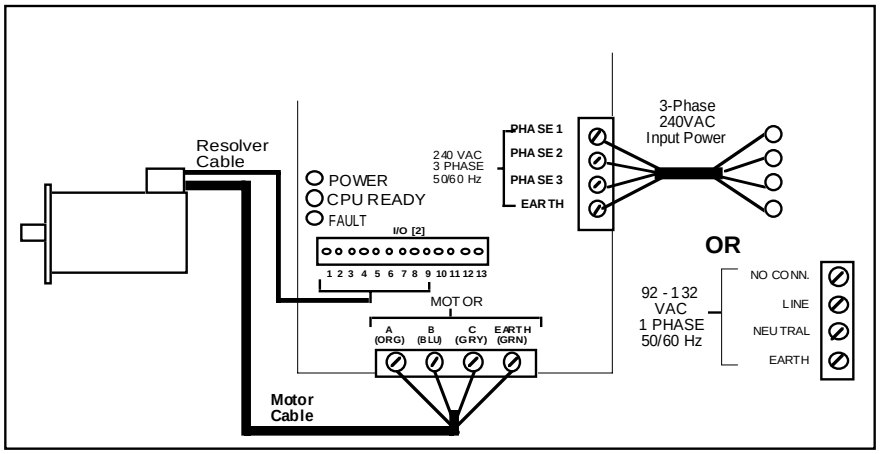


ZX900 Motor Cable Connection

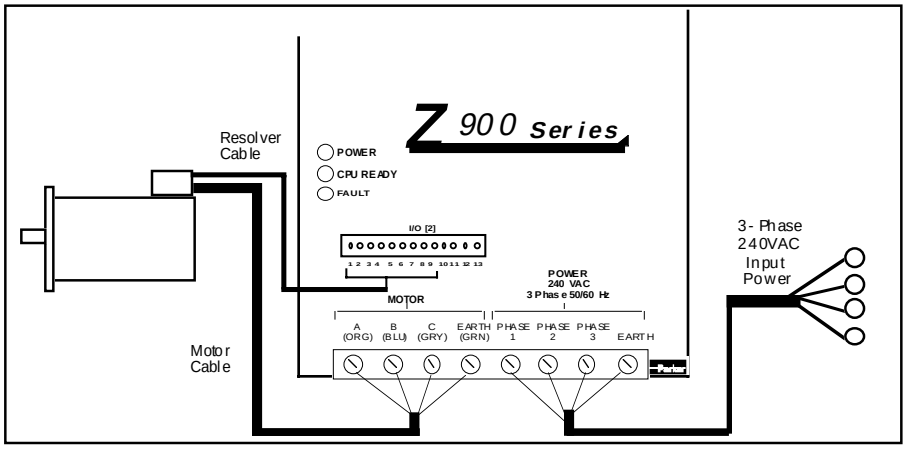
Step ③

Measure and verify your incoming 3-phase 240VAC (*or 120VAC*) line with a voltmeter. Connect, ***but do not energize***. The ZX is designed to operate best on a balanced 3-phase input. Additional wiring options for this connection are provided in *Chapter ③ Installation*.

> ***WARNING***
> Be sure the drive is properly grounded to reduce the chance of electrical shock.



ZX600/800 Power Cable Connection



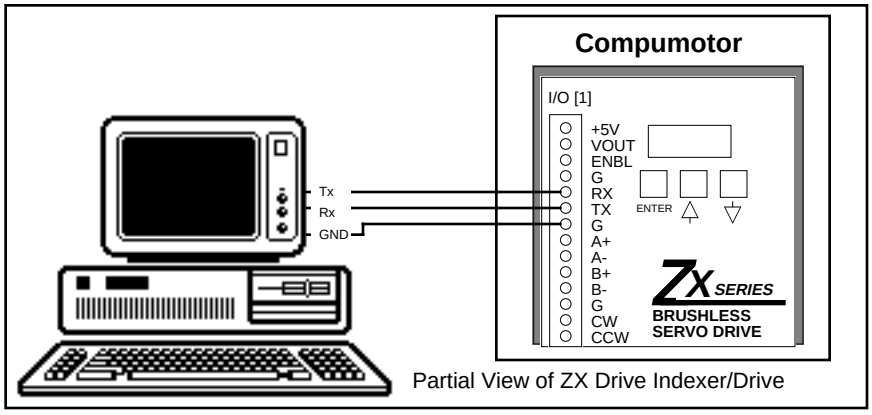
ZX900 Power Cable Connection

Step ④

Connect a terminal to the RS-232C connector (**I/O[1]**) using the **Rx** (receive), **Tx** (transmit), and **G** (ground) connections. The basic communications parameters are listed below:

- ❏ Baud Rate: 9600
- ❏ Data Bits: 8
- ❏ Stop Bits: 1
- ❏ Parity: None
- ❏ Full Duplex mode

Refer to the operations manual that accompanied your terminal for instructions on setting the communication parameters listed above. ***Chapter ③ Installation*** *provides detailed RS-232C information for the ZX.*



RS-232C Connection

Be sure that the ZX's ***ENBL*** *input (****I/O [1]****) is jumpered to ground. The ZX is shipped with the* ***ENBL*** *input grounded.*

If you use a host computer to send characters to the ZX via RS-232C at a high rate, you may need to insert time delays (***pacing characters***) to ensure that the ZX has enough time to process each character. The ability to set a variable time delay is available in most terminal emulator packages.

Step ⑤ **Apply power to all system components now**. The following response should appear on the terminal:

```
*READY
>
```

The following response should appear on the ZX's 4-digit display:

```
OK
```

Step ⑥ You must now disable the hardware limits. Type the following command to disable the limits: **`> LD3`**

To check the state of your hardware limits command, enter the following command: **`> 1LD`**

You should receive the following response: **`*3_No_Limits_Enabled`**

Step ⑦ Enter the following commands to move the motor 25,000 steps.

Command	Description
`> A1Ø`	Sets acceleration to 10 rps^2
`> AD1Ø`	Sets deceleration to 10 rps^2
`> V1`	Sets velocity to 1 rps
`> D25ØØØ`	Sets distance to 25,000 steps
`> G`	Initiates motion

Since the default drive resolution is 5,000 steps/rev, the motor should have moved 25000 steps or 5 revolutions.

This completes the basic configuration tests. The successful completion of this test means that you wired the resolver, motor, terminal (optional), and power connections correctly. The components of your system are functioning properly. If the motor does not move as commanded, check your connections and try the test again. In Chapter ③ you will install and test the rest of your system's components.

ZXF Option

The ZXF can be used as an indexer. However, the ZX cannot perform following functions. The ZXF is configured as an indexer when you receive it from the factory. You can use software commands to enter or exit the Following mode. In *Chapter ③ Installation*, encoder installation and use of the Following mode are explained. *Chapter ④ Application Design* provides some practical examples of Following mode operation.

C H A P T E R ③

Installation

The information in this chapter will enable you to:

- ❑ Mount all system components properly
- ❑ Connect all electrical system inputs and outputs properly
- ❑ Ensure that the complete system is installed properly

You must complete the Check-Out Procedure in *Chapter ② Getting Started* before proceeding with the permanent installation process.

Environmental Considerations

You must consider the environment in which your system will be operating. Proper mounting, wiring, and grounding will ensure trouble-free operation. The ZX is designed to operate in an industrial environment; however, severe atmospheric contamination, electrical noise, or temperature extremes can affect performance of the system. ***Operate the drive and motor within its designed specifications.***

Compumotor recommends that you operate and store the ZX in the following conditions.

Storage Temperature: -40°F to +185°F (-40°C to +85°C)

Operating Temperature: 32°F to 122°F (0°C to 50°C) With adequate air flow (10 cubic feet per minute)

Complete System Configuration

Safety is the primary concern when installing any motion control system. This chapter provides installation guidelines that are designed to preserve the safety of the operator and the equipment. You should install all Compumotor hardware in conformity with local and national electrical and safety codes.

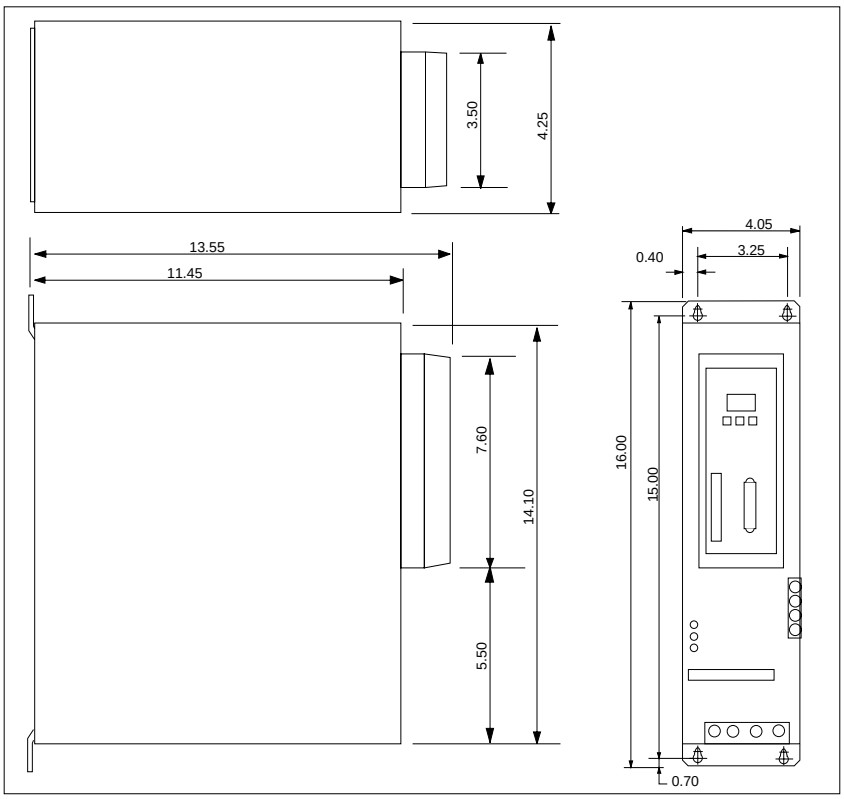
This chapter provides detailed instructions on all aspects of the ZX's installation and configuration. Once the system has been properly installed and initial adjustments are made, there should be little or no adjustment required to maintain normal operation.

Drive Mounting

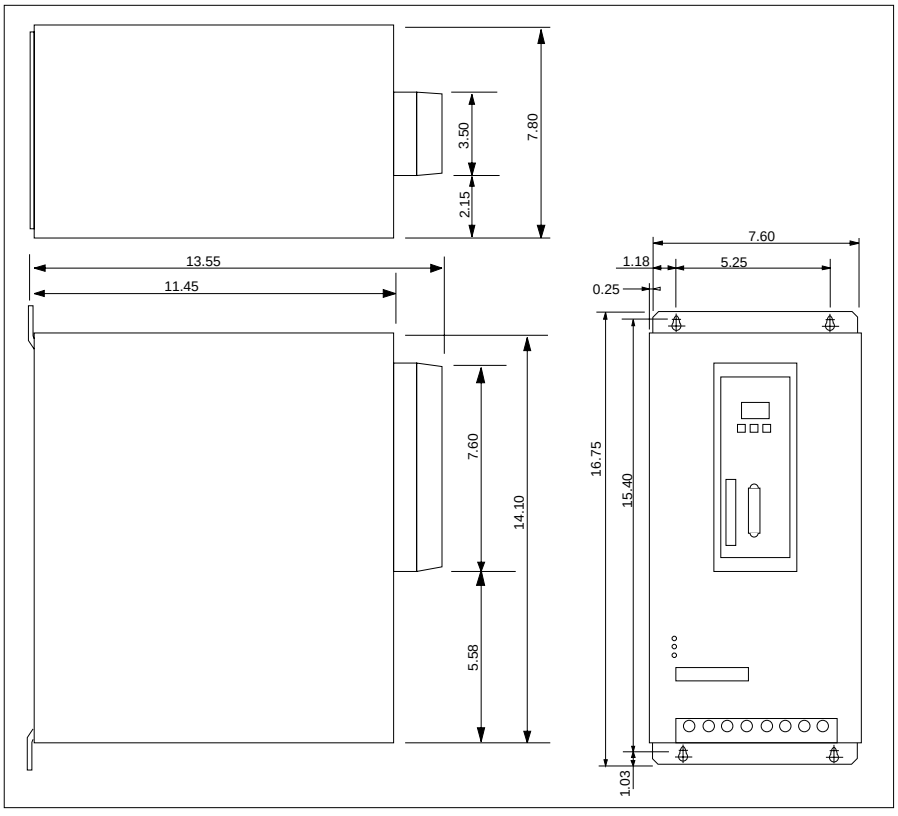
☛ The National Electrical Manufacturers Association (NEMA) has established standards that define the degree of protection that electrical enclosures provide.

The ZX should be installed in an enclosure that will protect it from atmospheric contaminants such as oil, metal flakes, moisture, and dirt. Industrial application environments may contain airborne contaminants, so the enclosure used should conform to a *NEMA TYPE 12 standard*.

The ZX mounting bracket is notched with keyhole type slots to accept screws on either end for flat panel surface mounting. Refer to the figures below for drive dimensions and keyhole slot locations. Use 10-32 or 1/4 x 20 screws into captured nuts to mount the ZX in a panel mount configuration. Use locking type fasteners to prevent the ZX from coming loose due to vibration.



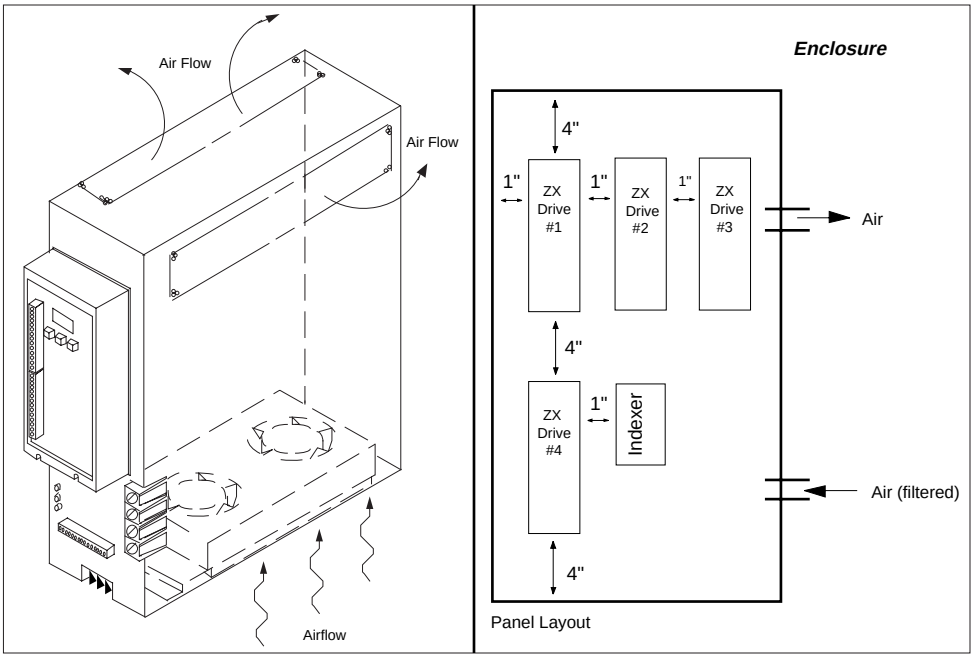
ZX600/ZX800 Indexer/Drive Dimensions



ZX900 Indexer/Drive Dimensions

Panel Layout

Proper panel layout is essential to ensure sufficient cooling of equipment within the enclosure. The ZX is fan cooled. It is designed to operate at its rated speed and torque specifications in a 32°F to 122°F (0°C to 50°C) ambient environment. When you design your panel layout, allow sufficient space for unrestricted airflow through each drive. Since the fan draws air upward over the heatsink, the air directly beneath the unit must not exceed 122°F (50°C). The ZX has two temperature sensors. One sensor is mounted to the heatsink—the other is mounted to the control board circuitry. If the drive overheats, the front-panel display will provide an error message.



ZX Air Flow and Panel Layout Guidelines

The ZX produces heat, which must be dissipated. The ZX600/ZX800 may generate as much as 500 watts (the ZX900 can generate 750 watts) when it is operated in continuous duty at maximum current. The actual dissipation will vary depending on the application duty cycle, motor size, and load inertia.

Observe the following rules when mounting ZX systems in an enclosure:

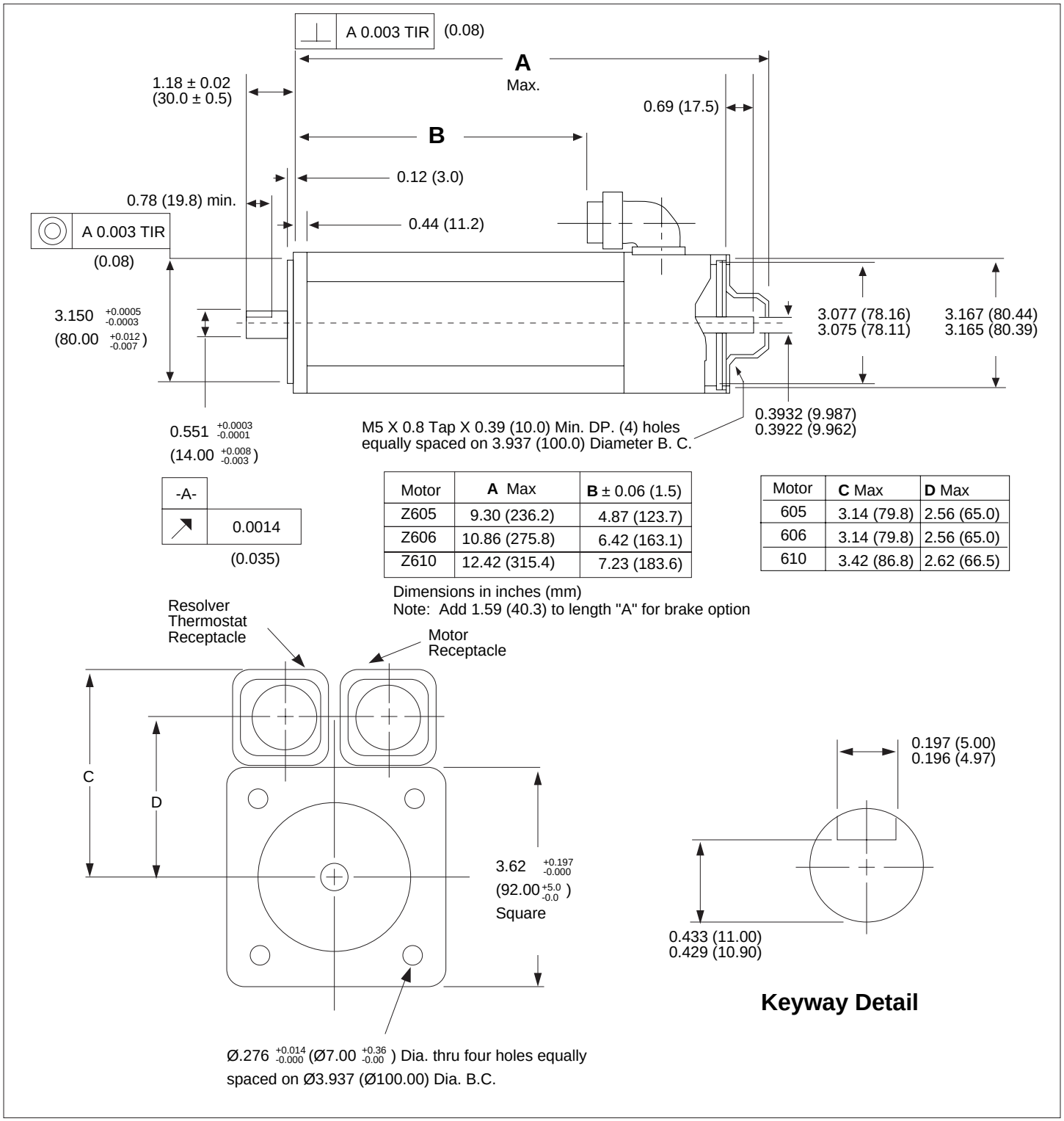
① The vertical distance between the ZX and other equipment, or the top and bottom of the enclosure, should be no less than 4 inches.

② The horizontal distance between the ZX and other equipment, or the side walls of the enclosure, should be no less than 1 inch.

③ Large heat-producing equipment (such as transformers) should not be mounted directly beneath the ZX.

Motor Mounting

Mount the motors with the four flange bolts that are on the front face plate. Center the motor with the pilot on the front face. The subsequent figures contain dimensional information for ZX motors (Z600, Z800, and Z900 Series).

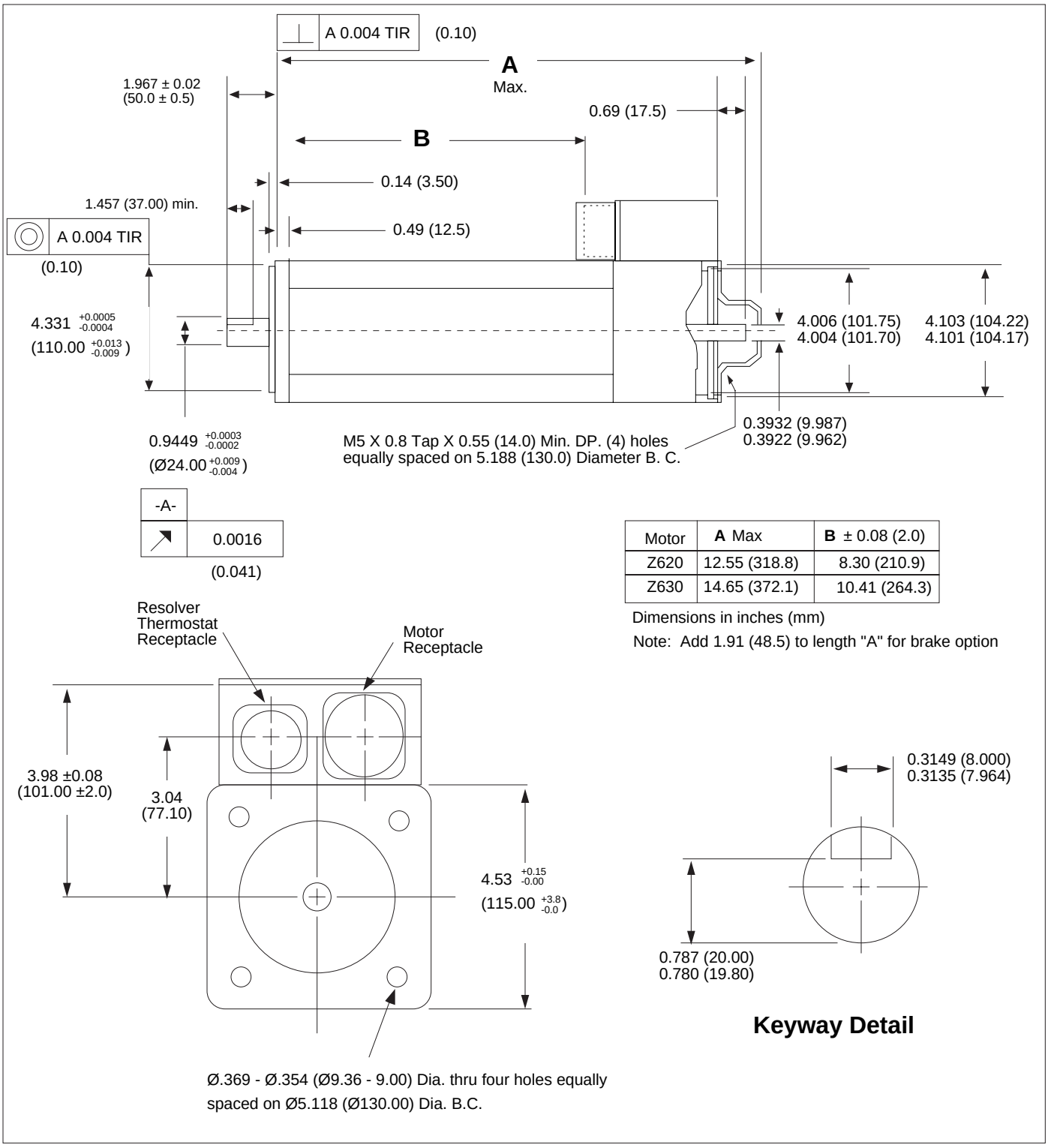


Motor	**A** Max	**B** ± 0.06 (1.5)
Z605	9.30 (236.2)	4.87 (123.7)
Z606	10.86 (275.8)	6.42 (163.1)
Z610	12.42 (315.4)	7.23 (183.6)

Motor	**C** Max	**D** Max
605	3.14 (79.8)	2.56 (65.0)
606	3.14 (79.8)	2.56 (65.0)
610	3.42 (86.8)	2.62 (66.5)

Dimensions in inches (mm)
Note: Add 1.59 (40.3) to length "A" for brake option

ZX Motor Dimensions (Z605 - Z610)

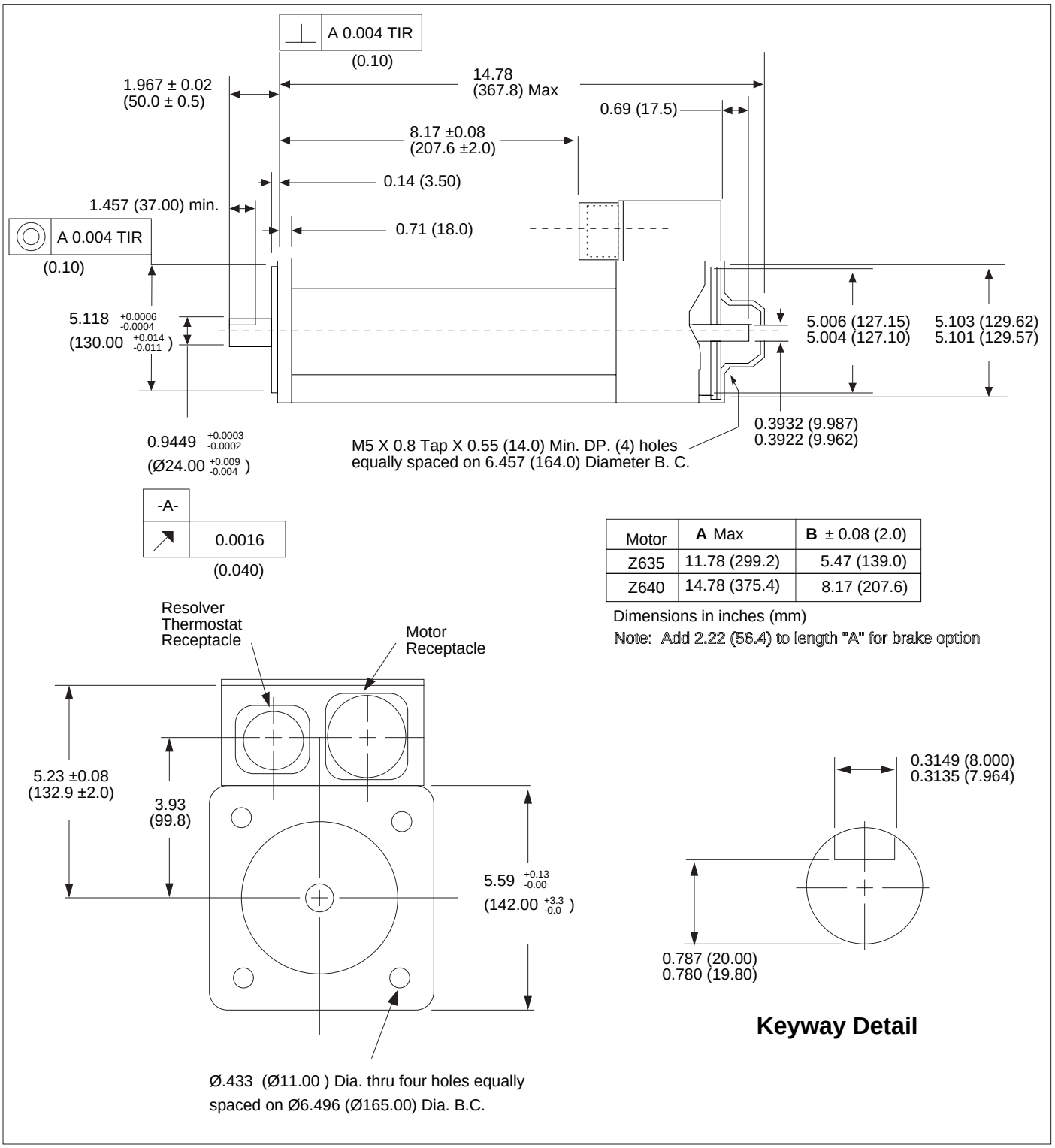


Motor	**A** Max	**B** ± 0.08 (2.0)
Z620	12.55 (318.8)	8.30 (210.9)
Z630	14.65 (372.1)	10.41 (264.3)

Dimensions in inches (mm)

Note: Add 1.91 (48.5) to length "A" for brake option

ZX Motor Dimensions (Z620 & Z630)

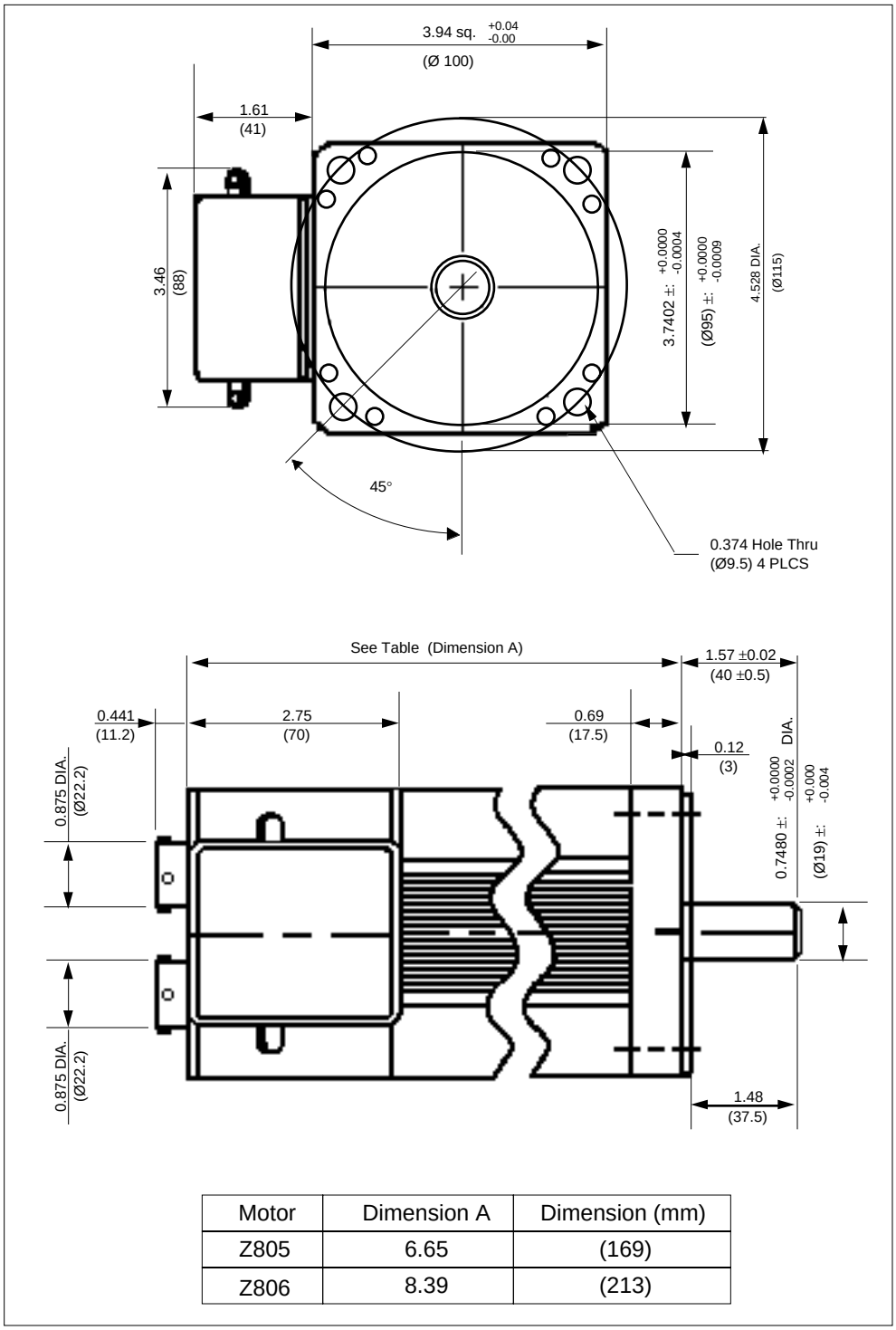


Motor	**A** Max	**B** ± 0.08 (2.0)
Z635	11.78 (299.2)	5.47 (139.0)
Z640	14.78 (375.4)	8.17 (207.6)

Dimensions in inches (mm)

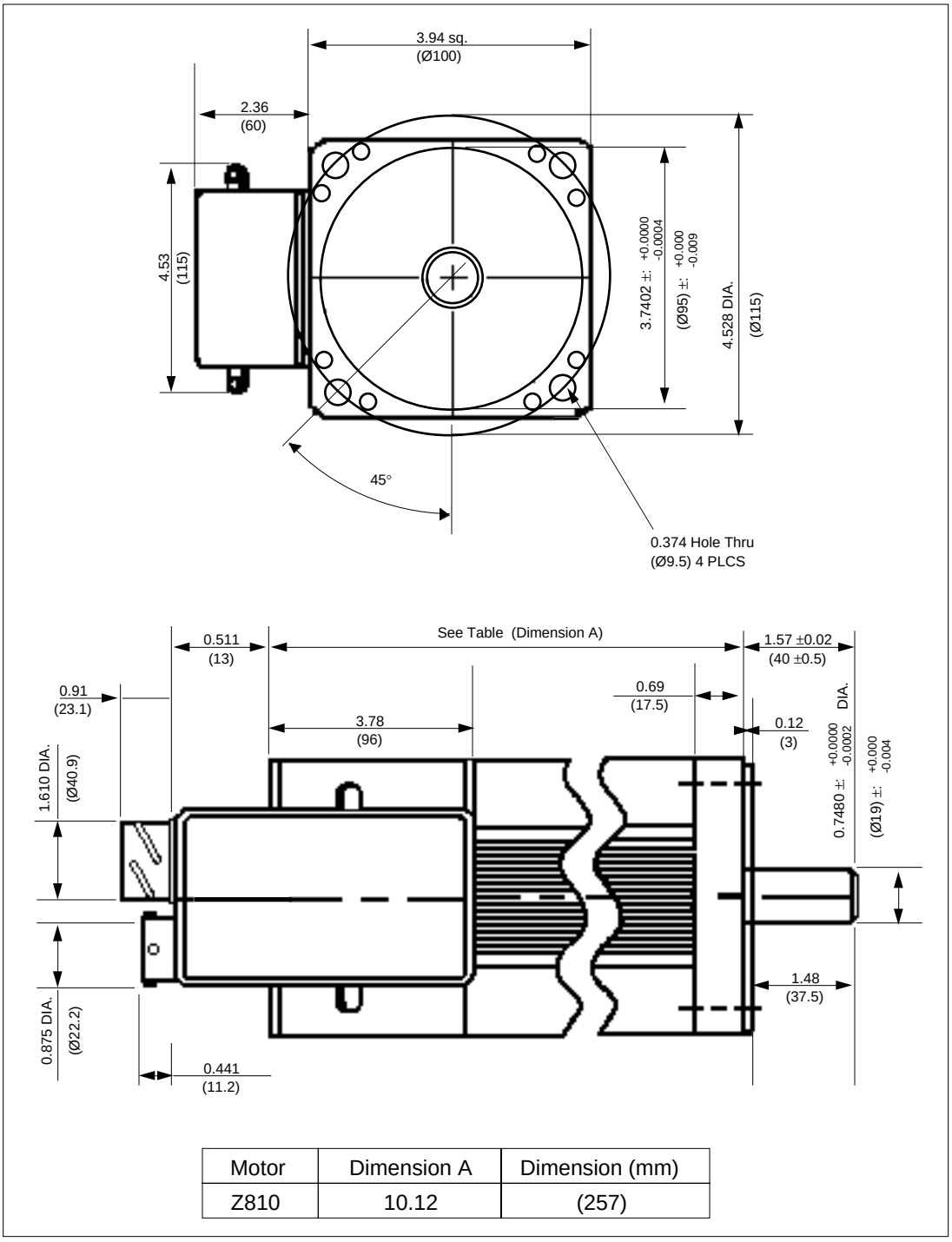
Note: Add 2.22 (56.4) to length "A" for brake option

ZX Motor Dimensions (Z635 & Z640)



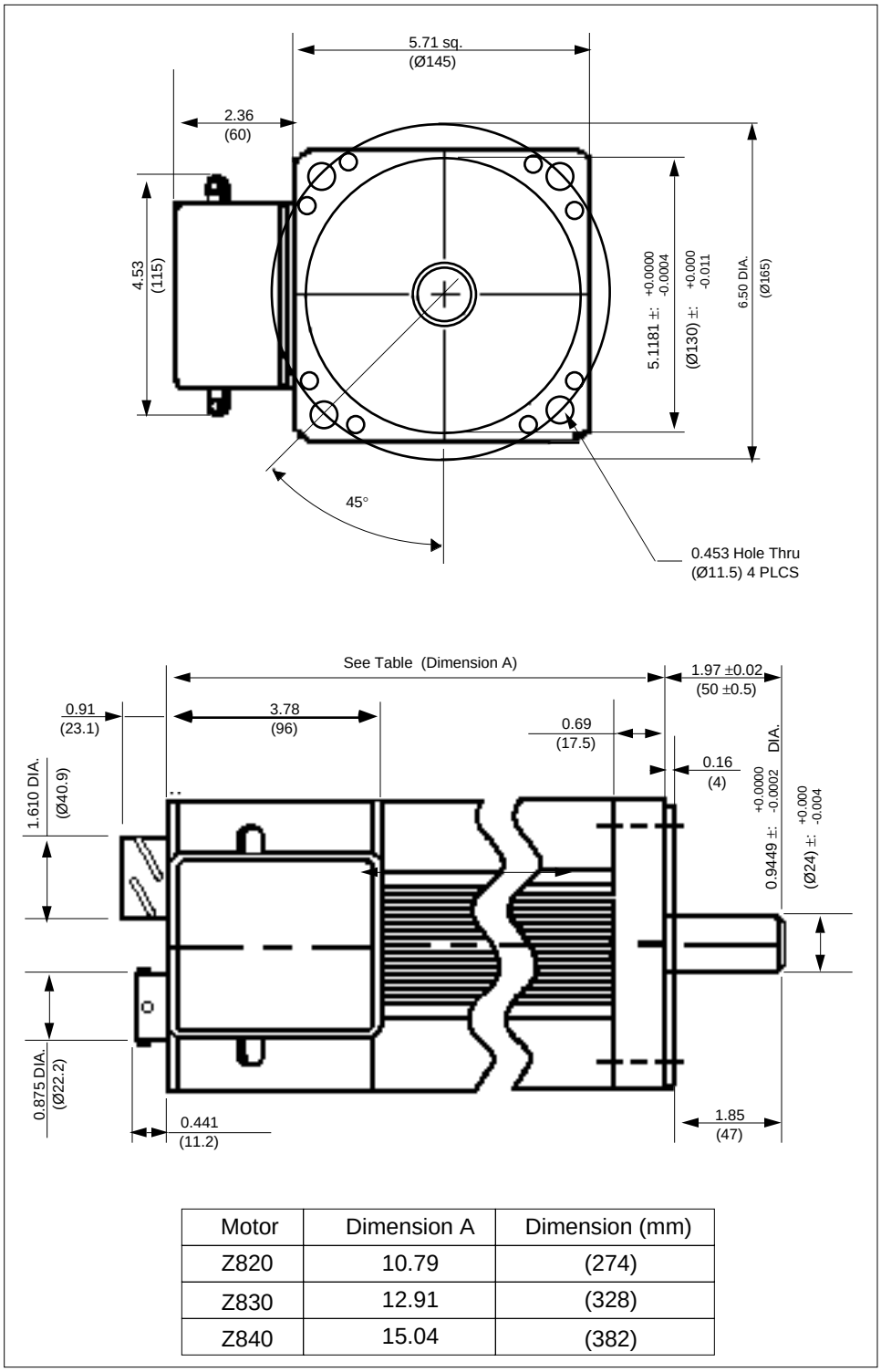
Motor	Dimension A	Dimension (mm)
Z805	6.65	(169)
Z806	8.39	(213)

ZX Motor Dimensions (Z805 & 806)



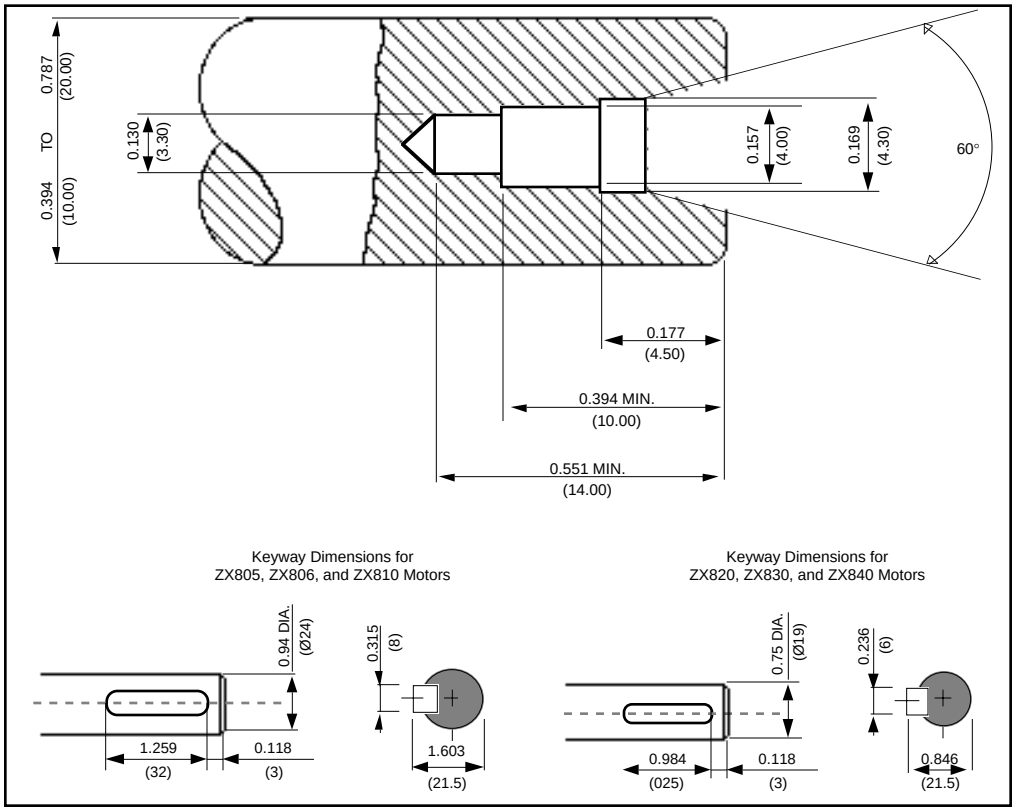
Motor	Dimension A	Dimension (mm)
Z810	10.12	(257)

ZX Motor Dimensions (Z810)

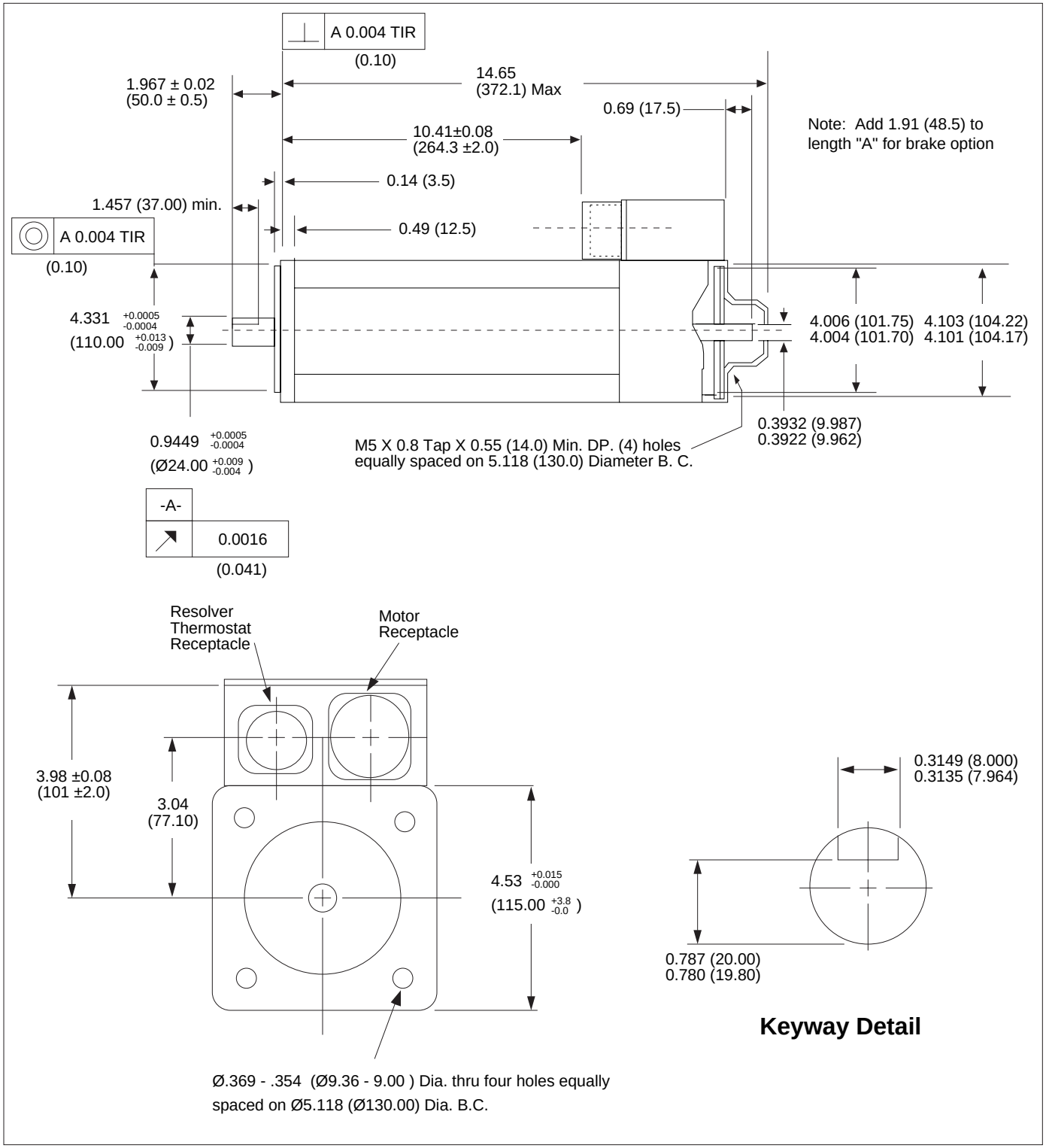


Motor	Dimension A	Dimension (mm)
Z820	10.79	(274)
Z830	12.91	(328)
Z840	15.04	(382)

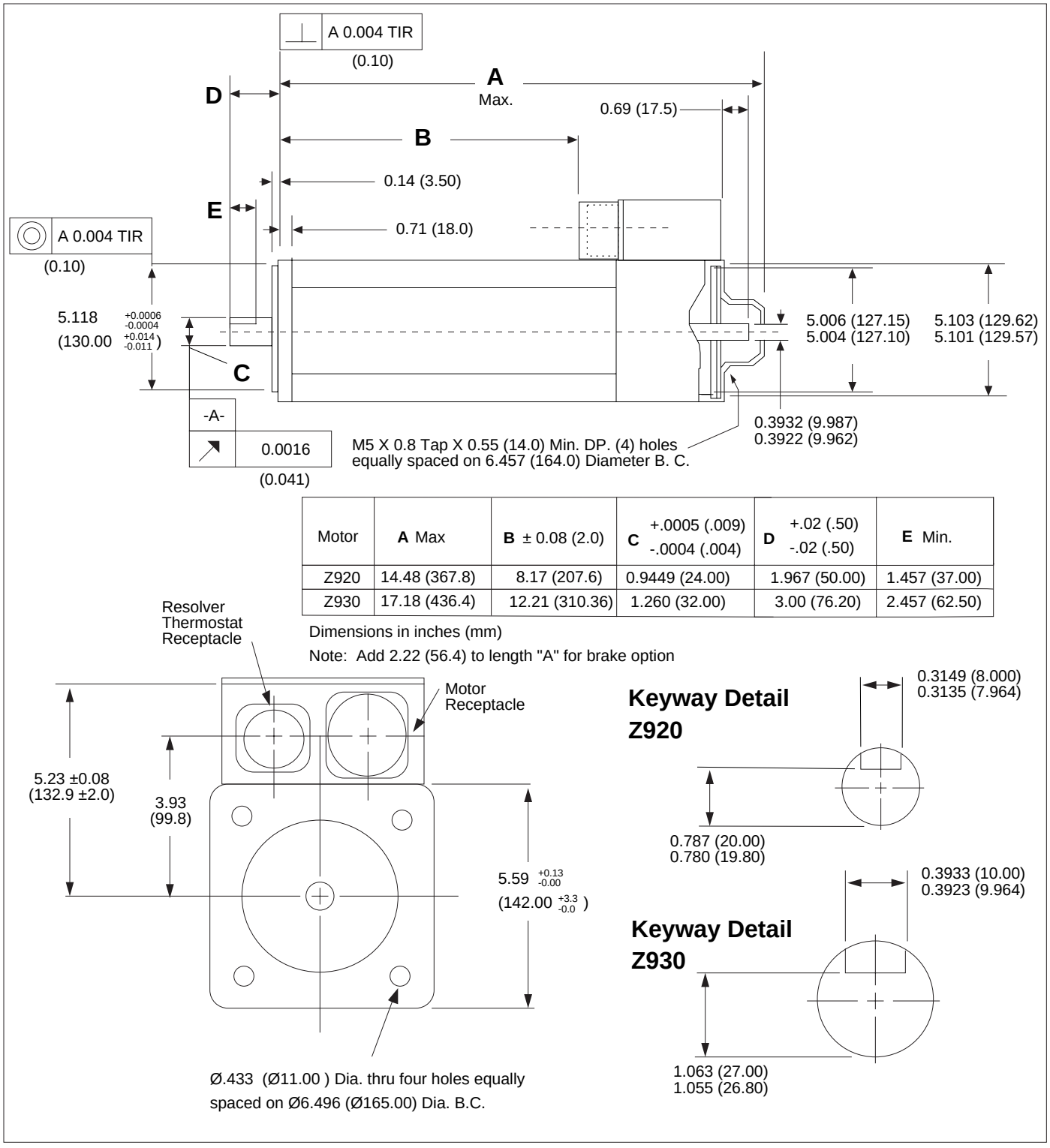
ZX Motor Dimensions (Z820, 830, & 840)



ZX800 Series Motor Shaft Dimensions



ZX Motor Dimensions (Z910)

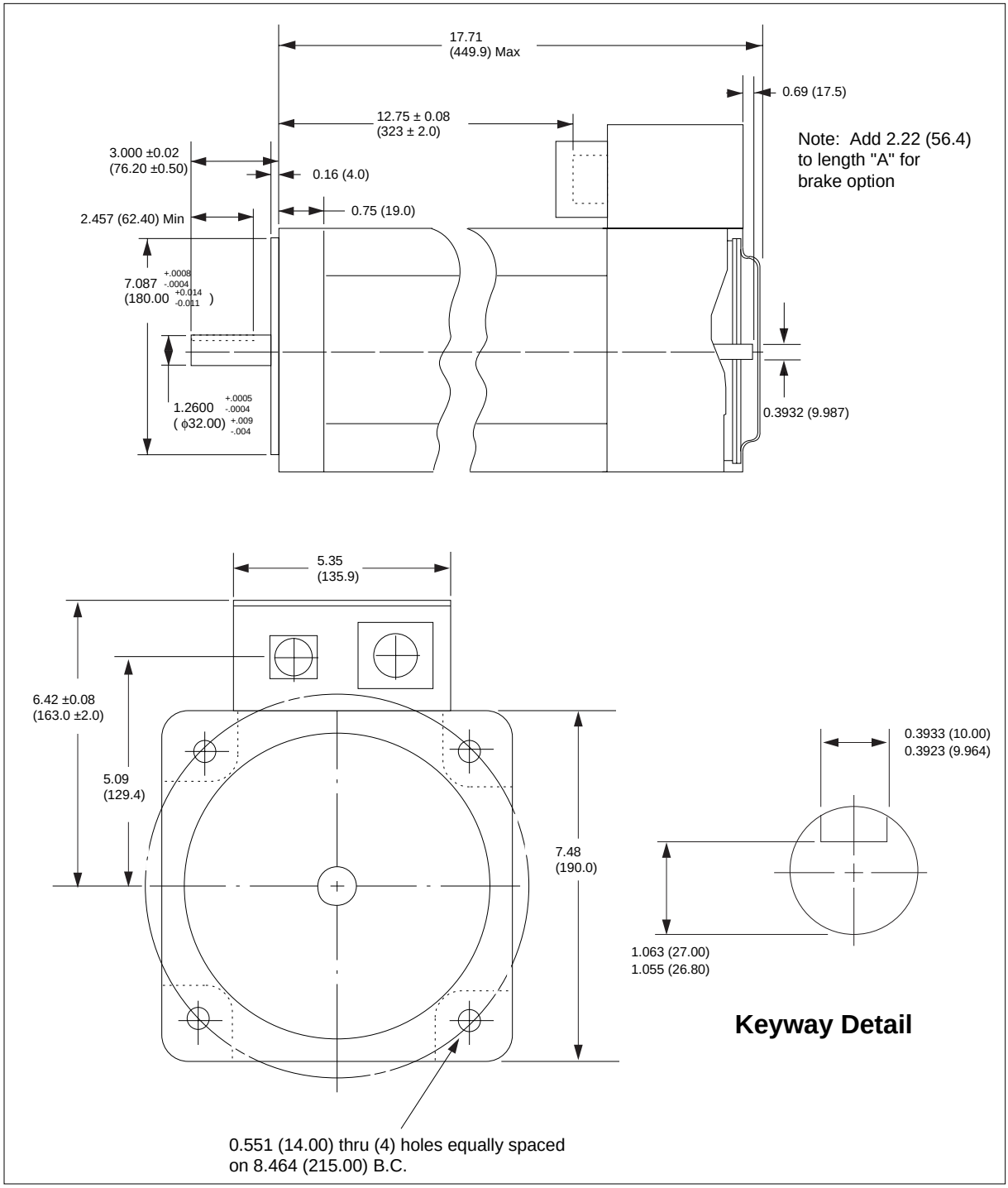


Motor	**A** Max	**B** ± 0.08 (2.0)	**C** +.0005 (.009) -.0004 (.004)	**D** +.02 (.50) -.02 (.50)	**E** Min.
Z920	14.48 (367.8)	8.17 (207.6)	0.9449 (24.00)	1.967 (50.00)	1.457 (37.00)
Z930	17.18 (436.4)	12.21 (310.36)	1.260 (32.00)	3.00 (76.20)	2.457 (62.50)

Dimensions in inches (mm)

Note: Add 2.22 (56.4) to length "A" for brake option

ZX Motor Dimensions (Z920 & 930)



ZX Motor Dimensions (Z940)

ZX motors can produce substantial steady-state torques and rapid accelerations. A combination of these forces can shear shafts and mounting hardware if the mounting design is faulty. Rapid acceleration can produce shock and vibration loads that require heavier hardware than normally required. The motor itself, under certain load profiles, can generate low-frequency vibrations within the mounting structure as well. These vibrations can cause fasteners to loosen if they are not locked. Compumotor recommends that you use elastic nut type fasteners.

> ***CAUTION***
>
> Vibrations produced by a cycling motor can induce metal fatigue within structural members. Have all plans reviewed by a mechanical engineer.

Couplers

Shaft misalignments are unavoidable due to fabrication tolerances. Common shaft misalignment problems include parallel, angular, and end float misalignment. ***Parallel misalignment*** is the offset of two shaft center lines where the center lines remain parallel to each other. ***Angular misalignment*** occurs when the shaft center lines are not parallel, but intersecting at some angle not equal to 0°. ***End float misalignment*** is a variance in the relative distance between the ends of the two shafts.

Couplings can help you accommodate these misalignment problems while maintaining desired torque. Several coupling types are available.

- ❏ Single-flex
- ❏ Double-flex
- ❏ Rigid

A single-flex coupler will correct only for angular misalignment while a double-flex handles both angular and parallel problems. Depending upon their design both types may or may not accommodate end play problems.

When joining two shafts that are fixed in the radial and angular direction, you should use a double-flex coupling. Single-flex couplings are not recommended for this type application because uncompensated parallel misalignment will eventually bend shafts and place bearings under unacceptable loads.

If only one of the shafts is free to move radially, Compumotor recommends that you use single-flex couplings. A double-flex coupling may allow too much freedom of movement leading to destructive vibration generation.

Applications that require rigid couplings are extremely limited. Rigid couplings do not compensate for any misalignments and are therefore not recommended for most applications. You should use them only if the motor is on floating mounts that allow for alignment compensation.

Ensure that any coupling you select for your application is within the range of the couplings performance specifications.

System Wiring

This section provides instructions for all ZX system wiring.

- ❏ Motor Connections
- ❏ Resolver Connections
- ❏ I/O[1] & I/O[2] Connections
  - Limit Inputs
  - Home Inputs
- ❏ Power Connections
- ❏ Encoder Connections (*for ZXF only*)

> ***WARNING***
>
> All AC power must be disconnected prior to installation wiring. Failure to observe safety precautions may expose you to dangerous voltages.

Connections

The ZX is shipped with a pre-assembled motor cable. The motor end of the cable has an MS-type connector.

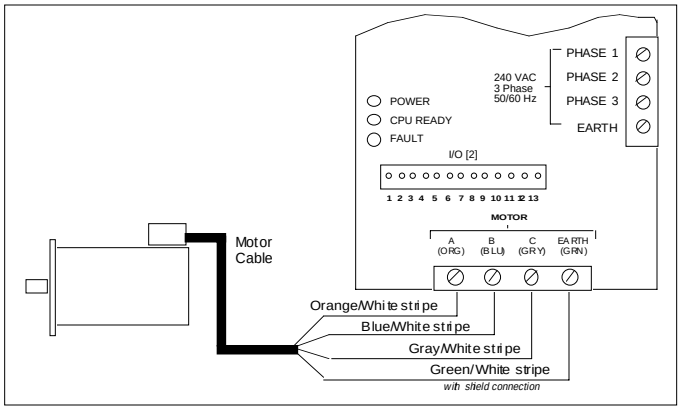
Motor Type	Motor Cable Connector	Resolver Cable Connector
Z606 - Z606	851-Ø6EC14-5S5Ø	851-Ø6EC14-19S5Ø
Z610 - Z630	CIRØ65-18-1ØS	851-Ø6EC14-19S5Ø
Z640	CIRØ6F-22-22S	851-Ø6EC14-19S5Ø
Z806 - Z806	PT06A-14-5S	PTØ6E-14-12S
Z810 - Z840	97B-3106-24-22S MS-3057-6A 3420-12	PTØ6E-14-12S
Z910	CIRØ65-18-1ØS	851-Ø6EC14-19S5Ø
Z920 - Z940	CIRØ6F-22-225	851-Ø6EC14-19S5Ø

ZX Series Resolver and Motor Connectors

The drive end has four leads with spade terminals that should be attached prior to applying power. *The ground wire is connected at the spade terminal to an uninsulated, stranded wire that touches the motor cable shield.* Motor screw terminal connections to the ZX are marked

Motor A, B, C, and Earth. The motor connectors are the same for all Zxs.

WARNING
Never disconnect the motor cable with the power on. The motor produces lethal voltages. Be sure the motor is properly grounded to reduce the chance of electrical shock.



Motor Wiring Pin Out and Color Codes

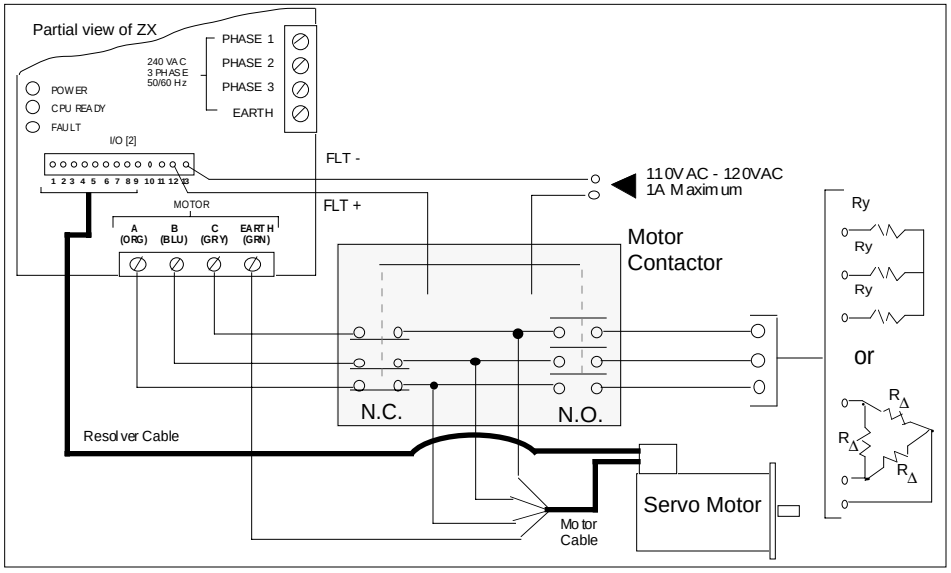
The standard ZX600 Drive motor cable is a 4-conductor, 12 AWG, 25-foot cable. The standard ZX900 cable is a 4-conductor, 8 AWG, 25-foot cable. If the motor and drive must be mounted further than 25 feet apart, Compumotor can provide cables with the MS connector installed for cable lengths of up to 100 feet. Standard cables are 25, 50, and 100 feet. *To order special cables, call Compumotor's Customer Service Department (800-722-2282). These cables must be ordered separately*. Cable lengths in excess of 100 feet are not recommended. *Compumotor recommends installing the motor and resolver cables in separate conduits to minimize noise coupling as well as for safety.*

Motor Braking

If the ZX faults, for any reason, the amplifier will be disabled and the motor will freewheel. Refer to *Chapter ⑧ Maintenance & Troubleshooting*, for all fault conditions. If a freewheeling load is unacceptable, you must install auxiliary motor braking resistors.

The figure below illustrates a typical resistor braking scheme. In this scenario, the motor contactor is normally energized to provide a direct connection between the motor and drive. The motor contactor (*N.O.—normally open/N.C. —normally closed*) is controlled by the fault relay+ (**FLT+**)/fault relay- (**FLT-**) output on the ZX's **I/O[2]** connector. If the amplifier faults or if the line voltage is disconnected, the contactor connects the motor braking resistors across the motor. The braking resistors can be sized by analyzing specific applications.

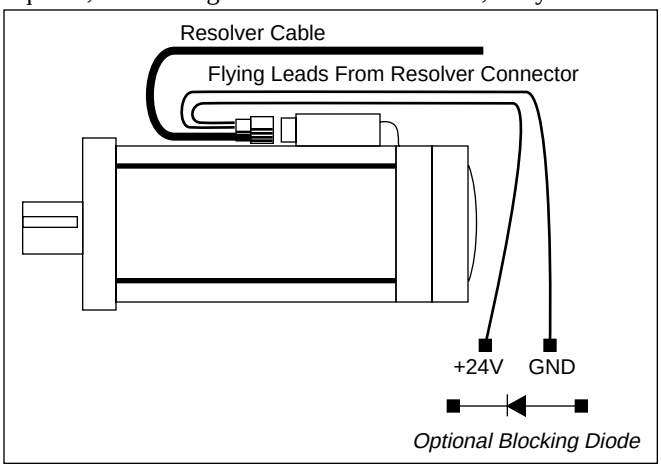
☞ ***Helpful Hint:***
Braking resistors provide very little braking at zero velocity. If standstill braking or emergency stopping are required, you can order the spring-type motor brake option from Compumotor (800) 722-2282. The brake is activated from a separate 24VDC supply and operates in a fail-safe mode.



Motor Braking (Dynamic)

If the total load inertia is less than five times the rotor inertia, non-inductive 200 watt power resistors can be used as the braking resistors. For a *wye* configuration, use 5Ω or more (R_Y = **5**Ω). For a *delta* configuration, use 15Ω or more (R_Δ = **15**Ω). If quicker stopping action is required, the braking resistors can be lowered, but you must increase their power ratings.

Wiring the Motor Brake Option



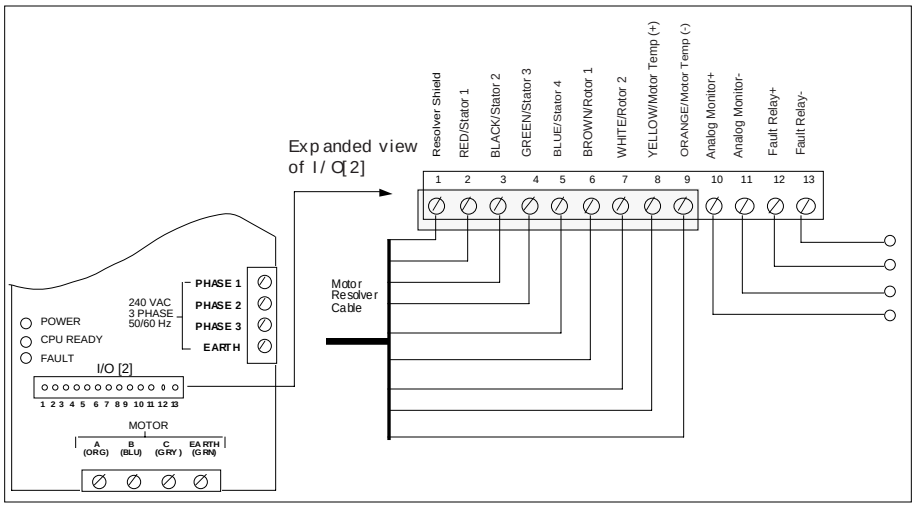
Wiring Z Series Motor Braking Option (-B)

Apply +24V to one of the flying leads from the resolver connector, and GND to the other lead (this energizes a coil that releases the brake). It may be necessary to place a blocking or flyback diode across the brake coil to prevent noise spike from falsely triggering other devices when the brake is engaged or disengaged (a 1N4936 should be sufficient). ***Install the diode between the wires as shown in the figure above.***

Resolver Connections

The ZX is equipped with a pre-assembled 25-foot resolver cable. The motor end of the cable includes an MS-type connector, Amphenol (part #**PT06E-14-12S**). The drive end of the cable is a screw terminal connector (**I/O[2]**).

☞ ***Helpful Hint:***
All ZXs have the same resolver connector.

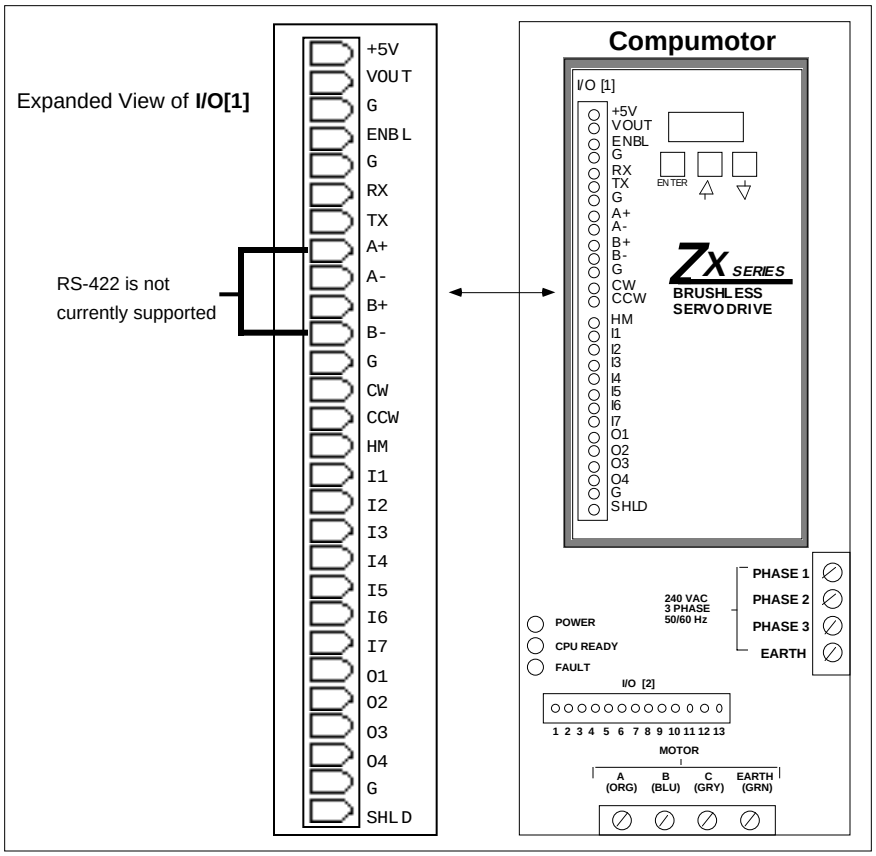


Resolver Pin-Out I/O[2]

I/O[1] Connections

The ZX's **I/O[1]** connectors provides the following communication, input, and output connections.

- ❏ Communication
  - RS-232C
  - RS-422C (*not currently supported*)
- ❏ Inputs
  - VOUT
  - Seven programmable inputs
  - ENBL
  - Limits
  - Home
- ❏ Outputs
  - Four programmable outputs

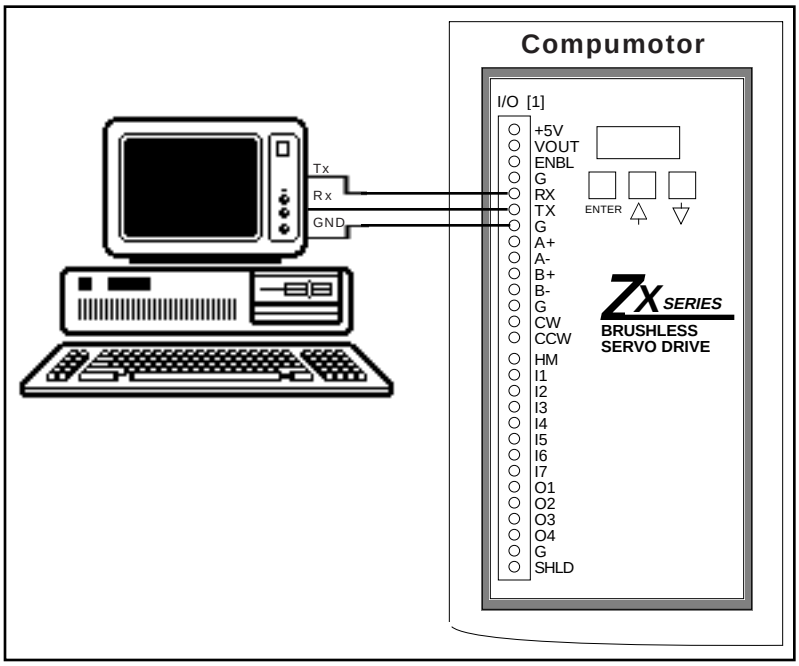


Screw Terminal I/O [1]

RS-232C Connections (RX, TX, GND) I/O[1]

The ZX can communicate to any terminal or host computer that can be configured for RS-232C. The ZX has a set of commands that can be used to set up the drive, program the drive, and report back drive information.

The ZX has a three-wire, optically isolated RS-232C interface that is compatible with RS-232C specifications. Receive Data (**Rx**), Transmit Data (**Tx**), and ground (**GND**) signals are connected on the screw terminal **I/O[1]**. Proper shielding of the RS-232C signal wires is required. The shield should be connected to an earth ground point on the terminal.



RS-232C Connections

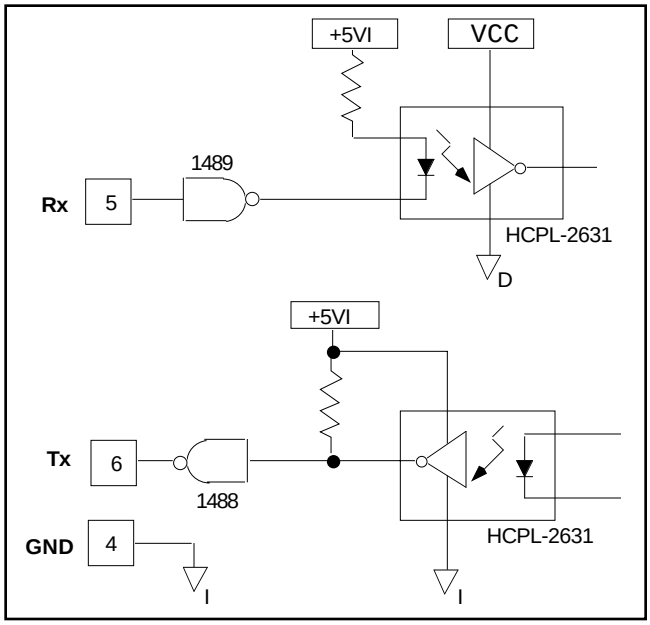
☞ ***Helpful Hint:***
The default communication parameters are:

- ❏ Baud Rate: 9600
- ❏ Data Bits: 8
- ❏ Stop Bit: 1
- ❏ Parity: None

Handshaking is not supported. The terminal should be set for Full Duplex mode.

You can change the baud rate with the front-panel pushbuttons (see *Chapter ④ Application Design* for more information). Baud rates of 300, 600, 1200, 2400, 4800, and 9600 are available. The RS-232C communication interface is optically isolated.

☞ ***Helpful Hint:***
Schematic of the RS-232C communication interface



RS-232C Input

ZX Daisy Chain Wiring

You may daisy chain up to 99 ZXs. Individual drive addresses are set with the ZX's push buttons. When daisy chained, the units may be addressed individually or simultaneously. You should establish a unique device address for each ZX. If you daisy chain more than 10 units, the baud rate should not exceed 2400.

Commands prefixed with a device address command only the unit specified. Commands

without a device address command all units on the daisy chain. The general rule is: ***Any command that causes the drive to transmit information from the RS-232C port (such as a status or report command), must be prefixed with a device address***. This prevents daisy chained units from all transmitting at the same time.

Attach device identifiers to the front of the command. The Go (**G**) command instructs all units on the daisy chain to go, while **1G** tells only unit one to go.

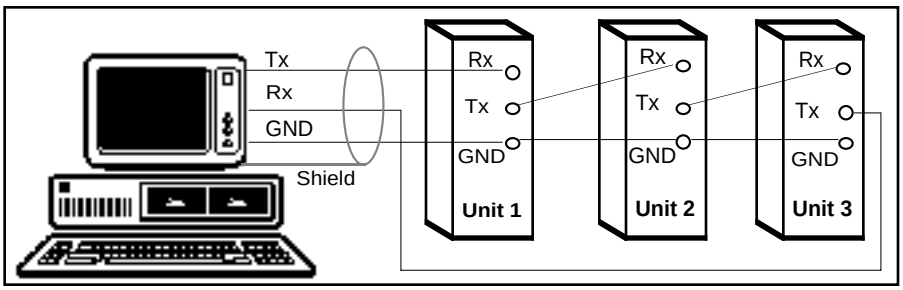
When you use a single communications port to control more than one ZX, all units in a daisy chain receive and echo the same commands. Each device executes these commands, unless this command is preceded with an address that differs from the units on the daisy chain. This becomes critical if you instruct any indexer to transmit information. To prevent all of the units on the line from responding to a command, you must precede the command with the device address of the designated unit.

No ZX executes a device-specific command unless the unit number specified with the command matches the ZX's unit number. Device-specific commands include both buffered and immediate commands.

You must use status-request commands in an orderly fashion. Commands should only be issued when the host is ready to read the response. You should not send new commands until you receive a response from the previous status-request command. In particular, you should not issue a immediate-status command until the host receives a buffered command status response. If this is not followed, the command responses will be intertwined, rendering the data useless.

If you enable the Interactive mode (**SSI1**), only the ZX that is set to Address 1 will respond with a prompt (>). This prevents all the ZXs from sending out > in a daisy chain. Typically, you should disable the Interactive mode when you use a host computer with the ZX.

☞ ***Helpful Hint:*** Multiple-drive configuration (daisy-chain) of RS-232C ports from one controlling terminal or computer



RS-232C Daisy Chain Configuration

Sample Applications and Commands

Three ZXs are on an RS-232C daisy chain. Send the following commands:

Command	Description
> MN	Sets unit to Preset mode
> A5	Sets acceleration to 5 rps^2 for all three controllers
> V1Ø	Sets velocity to 5 rps for all three controllers
> LD3	Disables limits (if they are not connected)
> 1D25ØØØ	Sets Axis 1 distance to 25,000 steps
> 2D5ØØØØ	Sets Axis 2 distance to 50,000 steps
> 3D1ØØØØØ	Sets Axis 3 distance to 100,000 steps
> G	Moves all axes

Unit 1 moves 25,000 steps, unit 2 moves 50,000 steps, and unit 3 moves 100,000 steps. All three units use the same acceleration and velocity rates. Units 1, 2, and 3 move at approximately the same time.

VOUT Input I/O[1]

This input may be used to set the output voltage level of the programmable outputs (O1 - O4). The VOUT input, referenced to ground, should be no more than 24V. Refer to *Chapter ⑦ Hardware Reference* for the VOUT input's hardware specifications.

ENABLE Input I/O[1]

The ZX 's ENBL input can enable or disable the ZX servo amplifier.

- ❏ To disable the amplifier, open the connection from ground.
- ❏ To enable the amplifier, pull the input to ground, and reset the drive (or type **ON**). *This input must be pulled down to ground for the drive to be active.*

See *Chapter ⑦ Hardware Reference* for the ENBL input's hardware specifications.

Limits I/O[1]

The ZX has two dedicated hardware end-of-travel limits (CCW and CW on the front panel). When you power up the ZX, these inputs are enabled. If you want to test the ZX without connecting the CCW and CW switches, disable the limit inputs with the **LD3** command. If you disable a move without disabling the inputs, the ZX motor will not turn. You can use the **RA** (Limit Switch Status Report), **IS** (Input Status), and **IN** (Set Input Functions) commands to test the limits' status. Refer to *Chapter ⑦ Hardware Reference* for the limit inputs' hardware specifications.

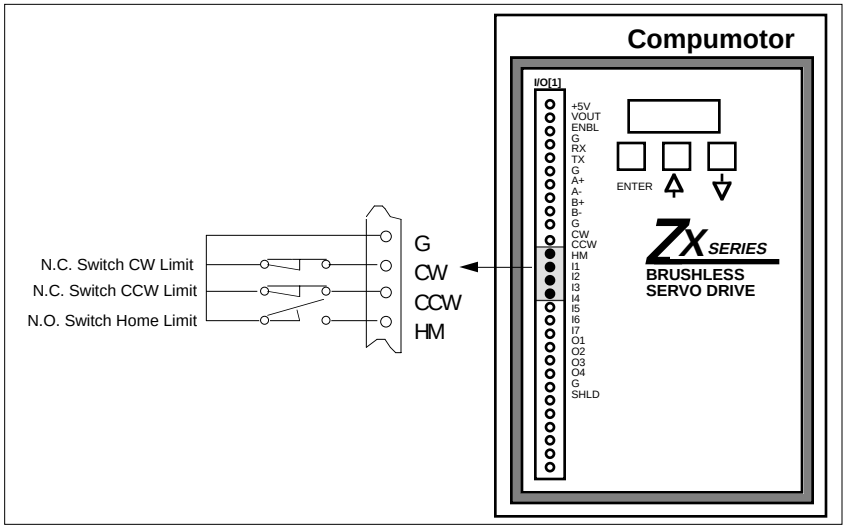
The ZX also has software limit capabilities. The software limits are disabled when you power up the system. If you need software limit capabilities, you can enable and define these software limits. Refer to the ***ZX Software Reference Guide***—Software Limits (**SL**) command.

Home Limit Input I/O[1]

The ZX's dedicated home HM input provides a reference for your application's motion. This input can command a machine to start an operation from a repeatable position. You may also use this input in conjunction with the Go Home (**GH**) command or the Go Home input with the Set Input Functions (**IN**) command. When the ZX executes a **GH** command, it scans the home limit input until the switch activates the home limit input.

Refer to *Chapter ⑦ Hardware Reference* for the home limit input's hardware specifications. *The homing function is discussed later in this chapter*.

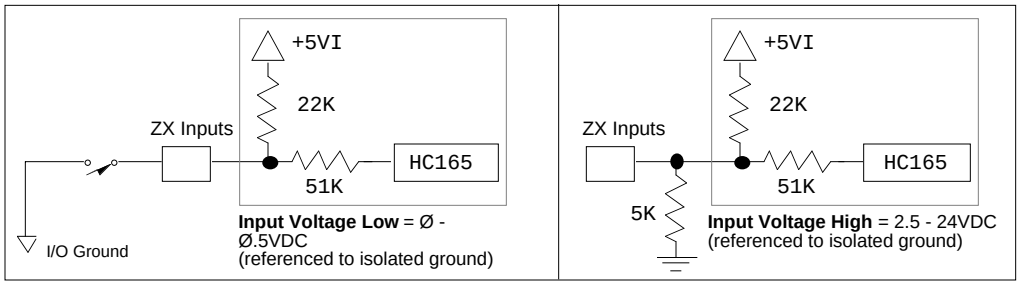
☛ You can activate the home limit input with the Define Active State of Home Switch (**OSC**) command.



Home and Limit Input Wiring

Programmable Inputs I/O[1]

The ZX has seven programmable inputs. These inputs can be connected directly to 24VDC. Each input can be programmed to perform 24 different functions. The inputs can be used with PLCs and configured with the outputs to interface with thumbwheel switches. Refer to *Chapter ⑦ Hardware Reference* for the programmable inputs' hardware specifications.

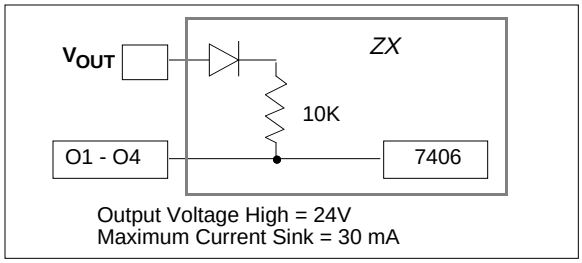


Enable, Home, Limits, and Programmable Inputs

Programmable Outputs I/O[1]

The ZX has four general programmable outputs. These outputs can sink up to 30 mA. You can program these outputs to perform 16 functions. They are open collector outputs, but they are pulled up internally with a 10K resistor. **V_{OUT}** is jumpered at the factory to 5V. To use a higher pull-up voltage, disconnect the jumper and apply an external power supply to **V_{OUT}**. The pull-up voltage will be equivalent to **V_{OUT}** - 0.3V. ***Remember that 5V < V_{OUT} < 24V***.

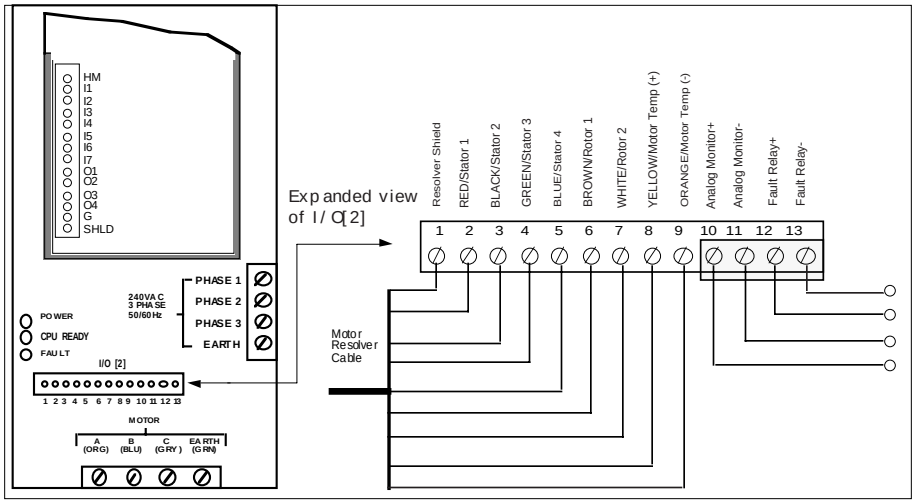
Refer to *Chapter ⑦ Hardware Reference* for the programmable output's hardware specifications.



V_{OUT} Input and Programmable Outputs 1 - 4

I/O[2] Connections

In addition to resolver signals and motor thermal signals, the **I/O[2]** screw terminal connector also has two outputs. Resolver cables on all ZX Indexers are the same.



Screw Terminal I/O (2) Outputs

Analog Monitor Output I/O[2]

This output provides an analog voltage (±10V) proportional to the velocity of the motor shaft.

- ❏ **MONITOR+** (pin #10) labeled **MON+**
- ❏ **MONITOR-** (pin #11) labeled **MON-**

You can connect these pins to an oscilloscope to facilitate tuning.

Drive Fault Relay Output I/O[2]

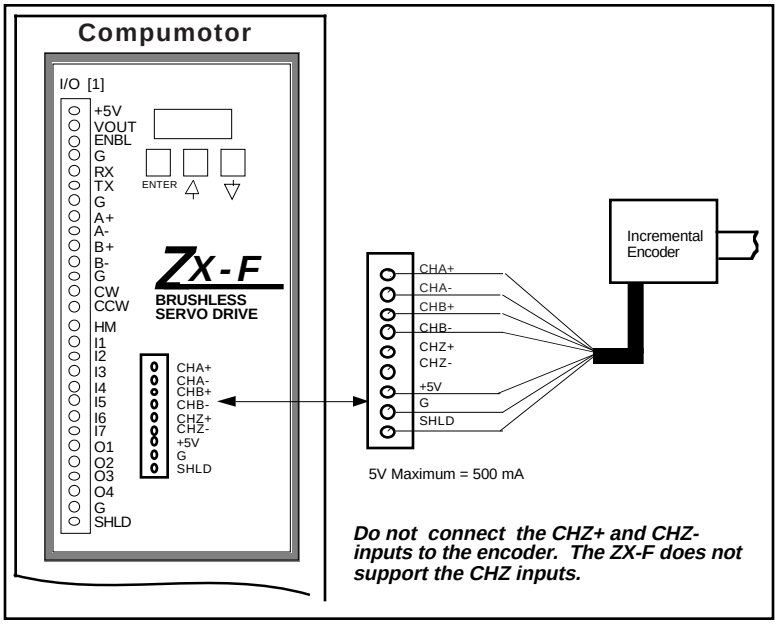
This isolated output is active during normal operation.

- ❏ **FAULT RELAY+** (pin #12) labeled **FTL+**
- ❏ **FAULT RELAY-** (pin #13) labeled **FTL-**

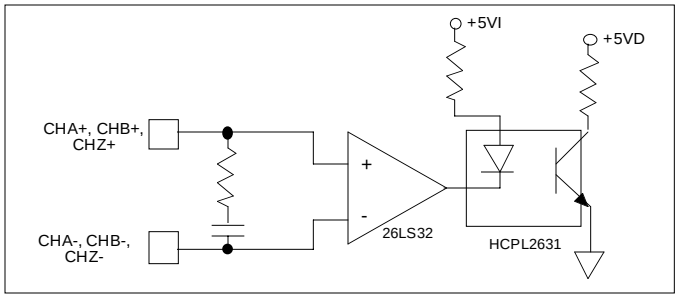
The relay will open during a fault condition or loss of power. This output is capable of 1A at 24VDC or 120VAC, resistive. The minimum current of 100mA at 12VDC is required to ensure contacts will not become contaminated.

ZXF Encoder Connection

The encoder connection is only applicable to the ZXF. ***If you are using a ZX, you may skip this procedure***.



Encoder Connection



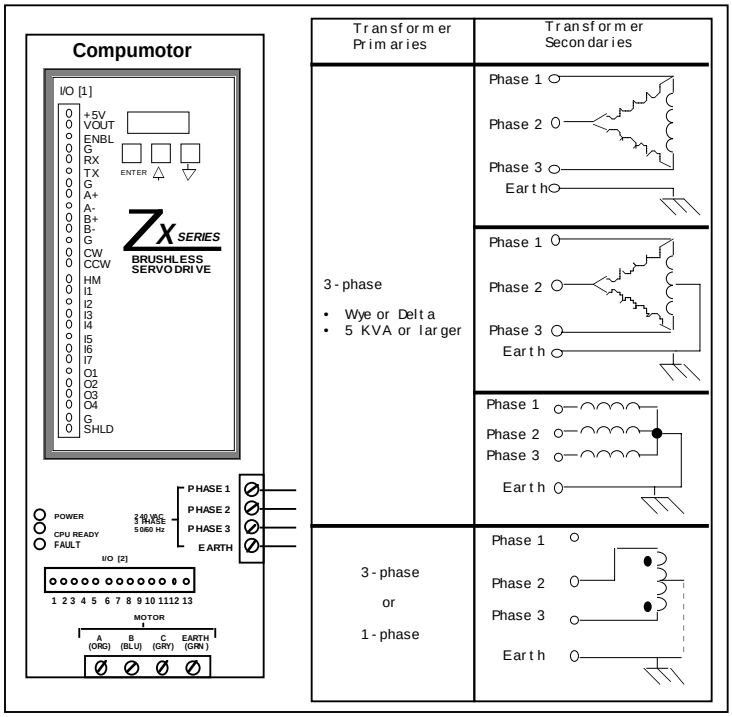
Encoder Inputs

Line Power Connections

The ZX600 and ZX800 are designed to be operated from a single or 3-phase 240VAC or single-phase 120VAC power source. The ZX900 is designed for single or 3-phase 240VAC operation. ***Only qualified personnel should install and service the equipment. Serious injury or death could result from miswiring***. The figures below illustrate the various ways to provide the main input power. The three input phase terminals feed a 3-phase rectifier that has current in-rush protection (each leg of the 3-phase input is fused separately). A 5KVA isolation transformer may be used for the ZX600 and ZX800 series, and a 10KVA isolation transformer may be used with the Z900 series, but transformers are generally not necessary if proper grounding procedures are followed and the proper line voltage is available.

> ***WARNING***
>
> Never disconnect the motor cable with the power on. The motor cable produces lethal voltages. This may cause a fatal injury. Be sure the drive is grounded properly to reduce the chance of electrical shock.

☞ ***Helpful Hint:***
Fuses: The ZX600 and ZX800 are fused for 240VAC 3-phase power (20A). If you suspect that a fuse has blown, call Compumotor (800-358-9070). ***Do not remove the outer chassis to access these fuses. Removing the chassis will void your warranty.***

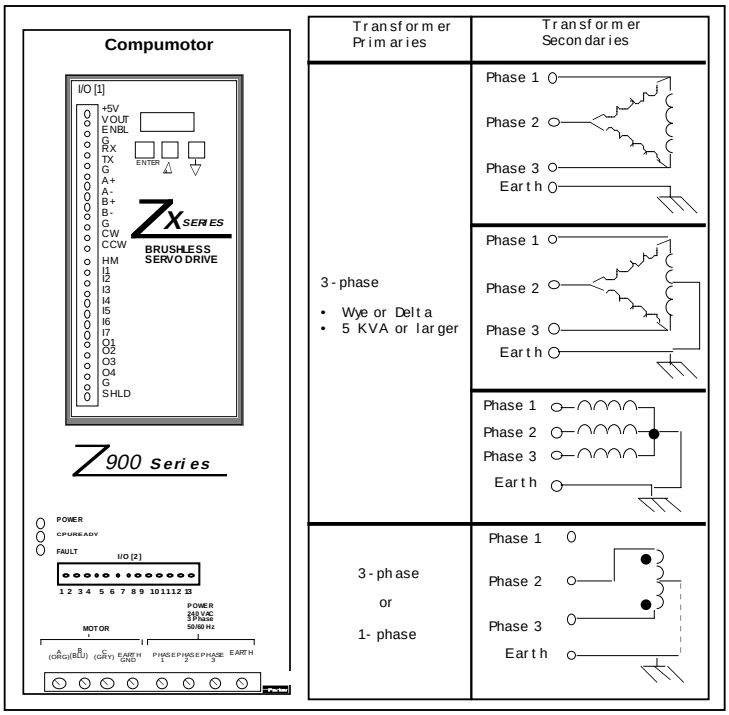


ZX600/800 Power Connections

Safety Interlock Switch

Put a 3-phase safety interlock switch between the drive and the power source to conveniently de-energize the drive in an emergency and/or service situation. A re-settable circuit breaker can also be used.

☞ ***Helpful Hint:***
Fuses: The ZX900 drives are fused for 240VAC 3-phase power (45A). The fuse block is located on the bottom of the drive beside the fan. These fuses may be replaced with Buss type SC45 fuses.



ZX900 Power Connections

Installation Verification

After you have completed all of the wiring instructions, you should complete the steps in this section to ensure that you have wired the motor, resolver, inputs, outputs, encoder (ZXF only), and power properly.

Programmable Input Functions

The input and output modes of the ZX offer flexibility and many function options for the input and output pins. You can program each input pin or designate them to correspond to a particular function. There are 24 functions that you can designate. To designate each input pin to a particular function, use the Set Input Functions (**IN**) command. (For a complete description of the **IN** command see the ***ZX Software Reference Guide***.) The input pins are numbered **I1** - **I7** on the front panel. The three remaining inputs are dedicated to **CW**, **CCW,** and **HM LIMIT** programmable inputs. Each function has been assigned a letter:

A: Trigger
B: Sequence Select
C: Kill
D: Stop
E: Drive Shutdown
F: Pause/Continue
G: Go
H: Direction
I: Synchronization
J: Jog+ (CW)
K: Jog- (CCW)
L: Jog Speed Select
M: Terminate Loop
N: Data
O: Reserved
P: Memory Lock
Q: Registration input (*can only be used with input #7)*
R: Reset
S: Go Home
T: Position Zero
U: User Fault
V: Data Valid
W: Data Sign
X: Increase Following Ratio
Y: Decrease Following Ratio
Z: Reserved

To program a particular input line to perform one of the designated functions, enter the **IN** command, followed by the pin number (1 - 7), followed by the letter of the function that you are designating the pin to perform. You may define the active input level with the Set Input Active Level (**INL**) command. The factory setting is **INLØ** (Active low).

Command	**Description**
> IN1A	Assigns input one to a trigger input

To display the present function assigned to each input type the following. To display one input at a time, type the number of the input following **1IN**.

Command	**Description**
> 1IN	Displays the input function assignments

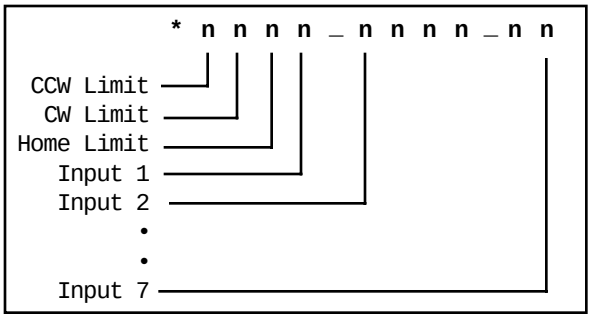
Verifying Proper Input Wiring

To verify that you have wired your inputs properly, enter the **IS** command. This command reports the active state of your inputs regardless of what function is assigned to them. Perform the following steps to verify proper wiring of your inputs.

Step ① Place the inputs in the active low state and display their current state.

Command	Description
> INLØ	Active level low
> 1IS	Response is ***ØØØØ_ØØØØ_ØØ**

Step ② Activate input #1 by shorting the input to ground and entering the status request **IS** again.



Command	**Response**
> 1IS	***ØØØ1_ØØØØ_ØØ**

This indicates that the ZX has recognized that the input number 1 has changed state. In this manner, you can verify that each input is changing state correctly.

Programmable Input Example

The following steps show how to configure input #1 (**IN1**) as a Remote Go input, and input #2 (**IN2**) as a stop input for control from your inputs.

Step ①

Command	Description
> MC	Sets to the continuous mode
> INLØ	Sets input level to be active low (off)
> LD3	Disables CW and CCW Limits
> IN1G	Sets up first input (**IN1**) as Go input.
> IN2D	Sets up second input (**IN2**) as a stop input
> A1ØØ	Sets acceleration to 100 rps^2
> AD1ØØ	Sets deceleration to 100 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps

Step ②

Command	Response
> 1IN1	*Ø1_G_GO_INPUT_(STATUS_OFF)
> 1IN2	*Ø2_D_STOP_INPUT_(STATUS_OFF)

These commands verify that inputs **IN1** and **IN2** have been configured properly. If not, go back and try step 1 again.

Step ③ To start motion, make sure that power is applied to the ZX and activate input #1 (**IN1**).

Step ④ To stop motion, activate input #2 (**IN2**).

This example shows how to use the inputs programmability Other **IN** features are discussed in *Chapter ④ Application Design.*

Programmable Output Functions

You can program the ZX outputs in the same way that you program the inputs. There are 16 possible functions that the outputs can perform (using the **OUT** command). Refer to the **OUT** command in the ***ZX Software Reference Guide*** for a detailed explanation. The possible output configurations are listed below:

A: Programmable Output
B: Moving/Not Moving
C: Sequence in Progress
D: At Limits
E: At Position Zero
F: Fault
H: Shutdown command indicator
J: Strobe (Outputs 1 - 3 only)
K: Command Error
L: Position Error Fault
N: CW Software Limit Reached
P: CCW Software Limit Reached
R: CW Hardware Limit Reached
S: CCW Hardware Limit Reached
T: Output Based on Position
U: Pulse output (Half axis)
V: Output When Within Deadband Range
W: Output When Within Deadband Range and Not Moving

The outputs are labeled on the front panel of the ZX as **O1** - **O4**. They are assigned in the same manner as the input pins. You can set the active level of the output with the Set Active Output Level (**OUTL**) command. The factory setting is **OUTLØ** (current flows when turned on). All of the outputs (**O1** - **O4**) are set up as programmable outputs as the factory default setting. The pull-up voltage is jumpered at the factory to 5V (through internal 10K resistors). If a higher pull-up voltage is desired, remove the jumper and apply the external power supply to **VOUT** (5V < **VOUT** < 24V). To increase the amount of current that the outputs can sink (30 mA maximum), you can add a resistor or apply voltage to the outputs (up to 24VDC).

Verify Proper Output Wiring

You can directly control each output by using the Output (**O**) command. Use the **O** command to turn on outputs 1 and 2 and turn off output 3. Use a voltmeter to verify that the output has changed state.

Command	Description
> O11Ø	Directly controls the outputs

Programmable Output Example

The following steps demonstrate how you can set up **O1** as a moving/not moving output and **O2** as a programmable output.

Step ① Make sure the outputs are properly wired.

Step ②

Command	Description
> MN	Sets unit to Normal mode
> OUT1B	Sets up **O1** as a moving/Not moving output
> OUT2A	Sets up **O2** as a programmable output
> A1ØØ	Sets acceleration to 100 rps^2
> AD1ØØ	Sets deceleration to 100 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps
> G	Executes the move
> T3	Waits 3 seconds
> O1	Turns on output 2 (**O2**)
> T3	Waits three seconds
> OØ	Turns off output 2 (**O2**)

While the motor is making the first move, **O1** should stay on. When the first move is finished, **O1** should go off. After 3 seconds, **O2** should come on and stay on for four seconds before turning off. Use the Output (**O**) command to display the current state of the outputs (type **1O**).

The successful completion of this test verifies that you can configure the ZX outputs to work as a programmable outputs and Moving/Not Moving outputs. In *Chapter ④ Application Design*, the use of other functions that you can program with the Configure Output (**OUT**) command are explained.

Limit Switches

Descriptions of the limits (**CW** & **CCW**) and a limit switch wiring example are provided earlier in this chapter. Before you test the limit switches, check the following:

- ❑ Ensure that the limit switches are wired properly.
- ❑ Ensure that the load is not attached to the motor.
- ❑ Ensure that you can manually open and close the limit switches.

To test the limit switches, complete the following steps:

Step ① Turn off (open) the CW and CCW limit switches.

Step ② Type **1IS**. Assuming that you have not turned on any other inputs, you should receive the following response:

***11ØØ_ØØØØ_ØØ**

This means that both CW and CCW limits are on (open).

Step ③ Turn on (close) the CW and CCW limit switches.

Step ④ To test the CW limit, enter the following commands:

Command	Description
> LDØ	Enables the CW and CCW limits
> INLØ	Normally closed limit inputs
> MC	Sets controller to Continuous mode

> A1ØØ	Sets acceleration to 100 rps^2
> AD1ØØ	Sets deceleration to 100 rps^2
> V3	Sets velocity to 3 rps
> H+	Changes the direction of the motor (CW)
> G	Executes the move (Go)

Step ⑤ Turn off (open) the CW limit switch. The motor should stop moving. Error **41** will scroll across the ZX's display. This verifies that the CW limit switch is working properly.

Step ⑥ Turn on (close) the CW limit switch. Error **41** will scroll across the ZX's display until a Go (**G**) command is executed in the opposite direction.

Step ⑦ To test the CCW limit, enter the following commands:

☞ ***Helpful Hint:*** The motor should move CCW

Command	Description
> H-	Changes the motor's direction (CCW)
> G	Executes the move (Go)

Step ⑧ Turn on (close) the CCW limit switch. The motor should stop moving and error **42** will scroll across the ZX's display. This verifies that the CCW limit switch is working properly.

Step ⑨ Turn on (close) the CCW limit switch. The motor should begin to move in the CCW direction. Error **42** will scroll across the ZX's display until a Go (**G**) command is executed.

If you are not able to stop the motor with the limit switches, reverse the CW and CCW limit input wiring. Perform this test again. If you are still unsuccessful, refer to *Chapter ⑧ Maintenance & Troubleshooting*.

Homing The Motor

Descriptions of the home limit (**HM**) and a home limit switch wiring example are provided earlier in this chapter. In this section, you will test your home limit switch and home the motor. You can initiate the Go Home function by entering the Go Home (**GH**) command over the RS-232C interface or by enabling the Go Home input using the Configure Input (**IN**) command. You must also define the Go Home Velocity (**GHV**), Go Home Acceleration (**GHA**), Go Home Deceleration (**GHAD**), and Final Go Home Velocity (**GHF**) to properly initiate the Go Home function. You can define the initial direction of the homing function with the Go Home Velocity (**GHV**) command. The **OS** commands set up the final go home approach and the edge of the switch to stop on.

When you command the motor to go home, it begins to move in the direction and at the velocity you specified. It performs this move at the acceleration rate specified with the **GHA** command, and looks for the **HM** input to go active. If the motor encounters an end-of-travel limit while it searches for home, it will reverse direction and look for the **HM** input to go active in the opposite direction. If the motor encounters the other limit before it detects the home signal, the Go Home move will be aborted and the motor will stop. To test the functionality of the **HM** limit switch, complete the following steps:

Step ① Manually open the **HM** limit switch and enter **1IS.** Assuming your end-of-limits and all other inputs are open (on), the system should respond with:

***11ØØ_ØØØØ_ØØ**

Step ② Close the **HM** limit switch and type **1IS.** The system should respond with ***111Ø_ØØØØ_ØØ**. This verifies that the home limit switch is functioning properly.

Step ③ Open the **HM** limit switch.

Step ④ To test the ZX's homing function, enter the following commands:

Command	Description
> IN1S	Sets up **IN1** (Input 1) as a remote Go Home input
> 1GHA1ØØ	Sets go home acceleration to 100 rps^2
> 1GHAD1ØØ	Sets go home deceleration to 100 rps^2
> 1GHF.5	Sets final go home speed to 0.5 rps.
> GHV+1	Sets the go home velocity to 1 rps (CW)
> 1OSB1	Enables the back up to home switch function
> 1OSGØ	The final approach direction is CW
> 1OSHØ	The CW edge is the edge the move stops on
> GH	Executes the Go Home function

Step ⑤ Ensure that the home limit input switch is open (on).

Step ⑥ Close **I1** (Input 1) and then open it. The motor will start to search for the **HM** limit switch again.

Step ⑦ Ground (turn off) the **HM** limit input. The motor decelerates to a stop. It then reverses

direction and moves off the switch at the final approach speed so that it can make its final approach to the home switch in the CW direction.

Step ⑧ Open the switch to simulate moving back off the home switch. Once it is off the switch, it begins its final approach in the CW direction.

Step ⑨ Activate the home switch again. The ZX will continue moving because it has reached only the CCW edge of the limit switch. Open the home limit switch again. This will be the CW edge and the motor will stop. This location will become the zero position. *Chapter ④ Application Design* explains the back-up to home limit function in detail. You will use the **OS** commands to set up the exact homing sequence that your application needs.

Verify Encoder Installation

This test is for ZXF users only. ***If you are using the ZX, you may skip this step***. To verify that your encoder is connected properly, enter the following series of commands.

Command	Description
> LD3	Disables the limits
> FSI1	Sets ZXF to Following mode
> 1PF	Reports the encoder count

Turn the encoder one CW revolution (move it in the positive direction). Check the encoder count again.

Command	Description
> 1PF	Reports the encoder count—it should have ***increased*** by the amount of your manual CW revolution

Turn the encoder one CCW revolution (move it in the negative direction). Check the encoder count again.

☞ ***Helpful Hint:***
If you pass this test, exit Following mode with the `FSIØ` *command.*

Command	**Description**
> 1PF	Reports the encoder count—it should have ***decreased*** by the amount of your manual CCW revolution

Command	**Description**
> S	Stops motion
> FSIØ	Exits Following mode, sets the ZXF to Indexer mode

You have now installed and tested your ZX system. *Chapter ④ Application Design* provides application information pertaining to motion, modes of operation, programming features, and other aspects of completing your application. Proceed to *Chapter ④ Application Design*.

C H A P T E R ④

Application Design

The information in this chapter will enable you to:

- ❏ Recognize and understand important considerations that must be addressed before you implement your application
- ❏ Understand the capabilities of the system
- ❏ Use examples to help you develop your application

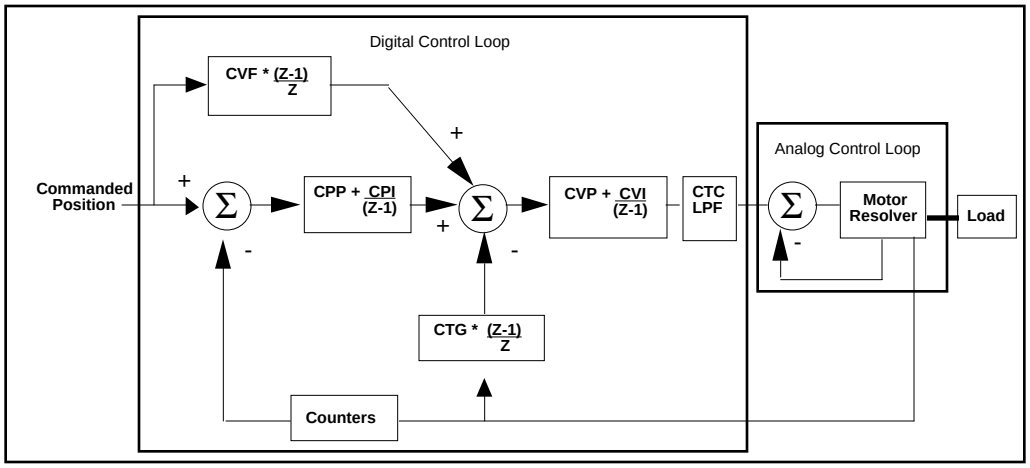
Basic Servo Tuning Theory

The ZX employs two basic control loops.

- ❏ Digital Control Loop
- ❏ Analog Control Loop

The *digital control loop* uses information from the resolver and user inputs to determine what the commanded motor currents should be.

The *analog control loop* takes current commands from the digital control loop and pulse width modulates the bus voltage to achieve these currents in the motor.



Digital and Analog Control Loops

The digital control loop has 15 parameters that you can adjust to obtain optimal shaft performance. You cannot adjust the analog control loop. It is configured to run all sixteen ZX motors (605, 606, 610, 620, 630, 640, 805, 806, 810, 820, 830, 840, 910, 920, 930, and 940) at optimum performance without modification. To ensure that the system operates properly, you must select the correct motor size with the Configure Motor (**CMTR**) command (refer to the ***ZX Indexer/Drive Software Reference Guide***).

Tuning parameters can vary significantly in each operating mode (Position mode, Velocity mode, or Torque mode). To simplify the task of tuning, default tuning parameters are stored for each motor size in each mode. If the default parameters do not provide adequate performance, you can manually tune the drive with the front panel interface or the RS-232C interface. All tuning parameters are accessible via the RS-232C interface; however, only some are accessible via the front panel.

Command	RS-232C	Front Panel	Tuning Commands
CPD	yes	yes	Configure Position Derivative
CPDM	yes	no	Configure Position Derivative Maximum
CPI	yes	yes	Configure Position Integral
CPIM	yes	no	Configure Position Integral Maximum
CPP	yes	yes	Configure Position Proportional
CPPM	yes	no	Configure Position Proportional Maximum
CTC	yes	no	Configure Time Constant
CTG	yes	yes	Configure Tach Gain
CTGM	yes	no	Configure Tach Gain Maximum
CVF	yes	yes	Configure Velocity Feed-Forward
CVFM	yes	no	Configure Velocity Feed-Forward Maximum
CVI	yes	yes	Configure Velocity Integral
CVIM	yes	no	Configure Velocity Integral Maximum
CVP	yes	yes	Configure Velocity Proportional
CVPM	yes	no	Configure Velocity Proportional Maximum

Tuning Parameter Commands

Tuning Procedure

If you are using the ZX for the first time, Compumotor recommends that you use the RS-232C interface. This interface provides access to all of the tuning parameters and gives you real-time access to some of the control variables. Two basic commands—**DDI** (Display Drive Information) and **DSP** (Display Servo Picture)—are designed to help you tune the drive. The **DDI** command lists all the tuning parameters as well as motor's resolution, drive configuration, etc. You can use this command to verify the drive's current operating mode.

☞ ***Helpful Hint:***
Sample **DDI** command response

```
          PP    PI    PD    VP    VI    VF    TG
*PERCENT  50    05    00    10    00    60    60
*MAXIMUM  10000 00400 32000 10000 00000 32000 32000

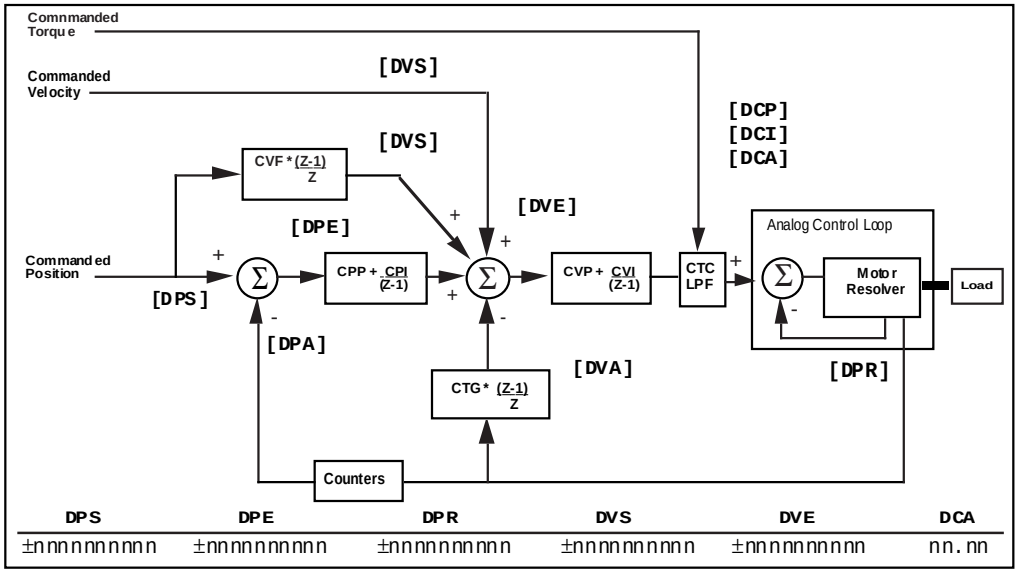
*TIME_CONSTANT=00005_(*100_MICROSECONDS)
*AVE_CURRENT_LIMIT=20.00_AMPS
*PEAK_CURRENT_LIMIT=40.00_AMPS
*MOTOR_RESOLUTION=05090
*RESOLVER_RESOLUTION=AUTO
*MOTOR_TYPE=2620
```

*All the gain commands have a maximum limit. This provides a wide dynamic range. For example, the gain term **PP** (Position Proportional) can vary from 0-99% using the **CPP** command. The maximum **CPPM** value can vary from 0 - 32,767. The equation below illustrates the number that is actually used in the control loop assuming that **CPP** = 50 and **CPPM** = 10000.*

```
Position Proportional gain = (CPP/100) * CPPM
                           = 50% * 10000
                           = 5000
```

The **DSP** command gives near real-time servo parameters. Use **DSP** to get an approximate real-time preview of what the control loop is doing and how changing the parameters will affect the system.

☞ ***Helpful Hint:***
The figure illustrates the response to a **DSP** command and the different parameters that you can display while the drive is operating. The software commands in brackets **[ ]** are the actual data in the loop. These variables are continuously updated.



Display Servo Picture Command (DSP) Response

The following display commands will help you tune the ZX.

DCA: Display Current Average — This command displays the ZX's average current.

DCP: Display Current Peak — This command displays the ZX's peak current.

DPS: Display Position Setpoint — This command displays the actual number of steps received from an indexer or pulse generator. This display is inactive in velocity and torque mode operation.

DPA: Display Position Actual — This command displays the motor shaft's actual position.

DPE: Display Position Error — This command displays the difference between the commanded and actual position in user-defined resolution.

DPR: Display Position Resolver — This command displays the position of the resolver. It *rolls over* numerically every mechanical revolution.

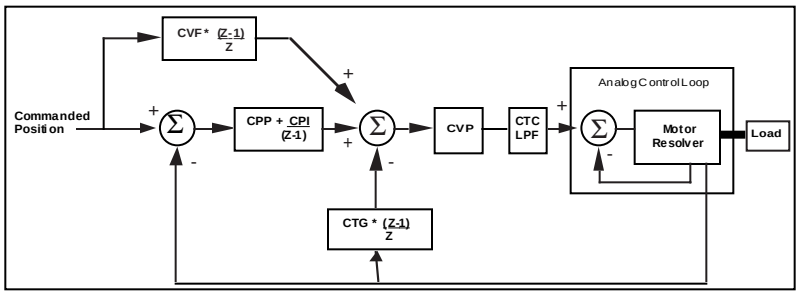
DVS: Display Velocity Setpoint — This command displays the desired velocity. In Position mode, this would correspond to the rate of change in steps.

DVA: Display Velocity Actual — This command displays the actual motor shaft velocity in rpm.

DVE: Display Velocity Error — This command displays the difference in the commanded velocity and the actual velocity in rpm.

Servo Tuning

The ZX's microprocessor-based indexer internally generates position setpoint commands while the ZX's **DSP** closes the position loop.



ZX Tuning

The following table contains the ZX's tuning commands.

Command	Tuning Commands
CPD	Configure Position Derivative
CPDM	Configure Position Derivative Maximum
CPI	Configure Position Integral
CPIM	Configure Position Integral Maximum
CPP	Configure Position Proportional
CPPM	Configure Position Proportional Maximum
CTC	Configure Time Constant
CTG	Configure Tach Gain
CTGM	Configure Tach Gain Maximum
CVF	Configure Velocity Feed-Forward
CVFM	Configure Velocity Feed-Forward Maximum
CVP	Configure Velocity Proportional
CVPM	Configure Velocity Proportional Maximum

Tuning Commands

CPP: Configure Position loop Proportional Gain

This command directly reflects the *stiffness* of the system. Generally, you want this gain as high as possible without causing the system to oscillate.

CPI: Configure Position Loop Integral Gain

This command directly influences the final position accuracy. In the default mode, it is turned on, but only slightly. It is error-limited to prevent integral windup.

CPD: Configure Position Loop Derivative Gain

This command sets both the digital tach gain and the velocity feed-forward gain to the same value. It has the effect of *damping* the system response. This gain is increased if the motor oscillates at zero commanded position.

CVP: Configure Velocity Loop Proportional Gain

This command directly reflects the *stiffness* of the system similar to the **CPP** command. Generally, you want this gain as high as possible without causing the system to oscillate. The only difference with this command relative to **CPP** is that it takes into account the velocity tach gain.

CVF: Configure Velocity Feed-forward Gain

This term reduces the position loop following error only when the shaft is turning. It does not affect the system's dynamics.

CTG: Configure Tach Gain

This term allows additional damping. If you increase this term, the system will become sluggish, but you will be able to stabilize large inertias.

CTC: Configure Torque Time Constant

This command filters the digital controller's output response. The motor is *commutated* every 100 µs and the servo loop is updated every 500 µs. In between each servo update, the commutation can use an average torque commanded value. The default is set to 500 µs (**CTC5**). This effectively low-pass filters the torque command signal with a -3dB frequency of 2000 Hz. You can change this value to decrease the low-pass filter frequency. This will lower the drive's bandwidth.

Position Mode Tuning Procedure

Use the following steps to tune the ZX.

Step ①

Set motor resolution to the proper number of steps/rev you desire (refer to the **CMR** command in the ***ZX Indexer/Drive Software Reference Guide***). The default is 5000 steps/rev.

Step ②

Check to make sure the **CMTR** command reports back the actual motor you are using. If it is wrong, change it with the **CMTR** command (refer to the ***ZX Indexer/Drive Software Reference Guide***).

Step ③

Attach the load and make your desired move with the default settings. Pay careful attention to the response time, end-of-position overshoot, following error, etc.

Step ④

Vary parameters to improve your performance if needed. Some common performance problems and suggested tuning procedures on how to improve performance are listed below.

Shaft Seems Spongy

Solution Procedure:

1. Increase **CPP**
2. Increase **CVP**
3. Decrease **CTG**
4. Increase **CPI**

Shaft Oscillates

Solution Procedure:

1. Increase **CTG**
2. Decrease **CVP**
3. Decrease **CPP**
4. Decrease **CPI**

Shaft Overshoots at End of Move

Solution Procedure:

1. Decrease **CPI**
2. Increase **CTG**

Shaft Has Too Much Following Error During Move

Solution Procedure:

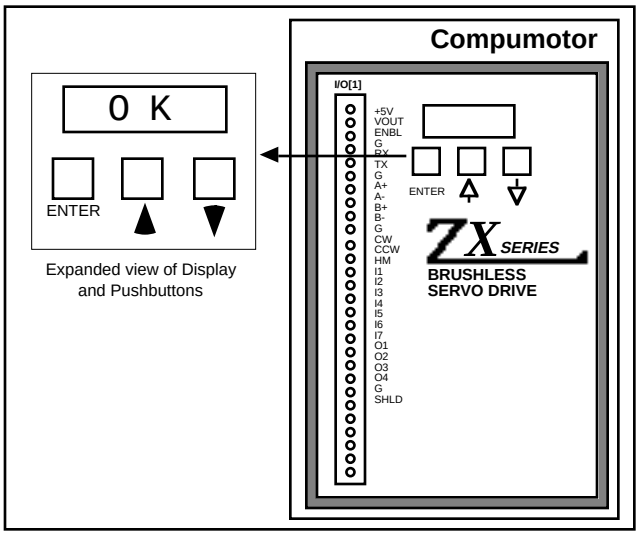
1. Increase **CVF**
2. Decrease **CTG**
3. Increase allowable **CPE**

Step ⑤

When performance is acceptable, you can save your gain parameters with a Save (**SV**) command over RS-232C or with the front panel display (press the **ENTER** pushbutton in the SAVE display).

Alphanumeric Display and Pushbuttons

The ZX has a four-character, dot-matrix, alphanumeric display. All error messages are scrolled across the display when a fault occurs. You can modify many drive parameters with the three pushbuttons.



ZX Drive Display and Pushbuttons

Fault Messages

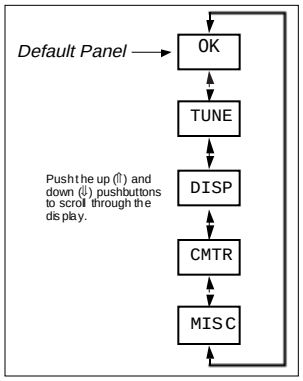
When a fault occurs, the appropriate ZX fault messages will be displayed. A fault code followed by a description of the fault scroll across the display.

Example: **ERROR #04 > OVER_VOLTAGE**

Fault messages are displayed continually until the fault is removed and the ZX is turned on again. Refer to *Chapter ⑧ Maintenance & Troubleshooting* for a complete list of error messages and troubleshooting methods.

Pushbutton Operation

You can use the ZX 's pushbuttons to modify drive parameters and to display several drive variables. The figure to the left is an overview of the Main menu panels and sub-panels. Although only one panel is shown on the display at a time, the ZX's display operates in a menu-driven format.



Main Menu Panel (Overview)

*The default value for the ZX's Configure Pushbuttons (**CPB**) command is **CPB1**. This fully enables the ZX's front panel. **CPBØ** provides you with access to all front panel displays. **CPBØ**, however, will not let you activate any of the menus.*

OK is the default message. It indicates that you are in the main menu. Use the up and down pushbuttons to view the menu items in the following order:

- ❑ **OK** Default user message, the home panel
- ❑ **TUNE** Tune Menu
- ❑ **DISP** Display Menu
- ❑ **CMTR** Configure Motor Menu
- ❑ **MISC** Miscellaneous Menu

To choose a menu, press the up and down pushbuttons to display the menu you want. Press **ENTER** to access the sub-panel menu. The table below shows the main menus and sub-menus.

O K	T U N E	D I S P	C M T R	M I S C
Home Panel	P P n n	D V E L	6 0 5	S A V E
	P I n n	D E R R	6 0 6	R F S
	P D n n	D C A	6 1 0	B R m m
	T G n n		6 2 0	A D p p
	V P n n		6 3 0	FOLL/NTFL
	V I n n		6 4 0	SEQU
	V F n n		F M C A	REV#
				JOG

Main Menu Panel for a ZX600 System (Overview)

Pressing the up and down pushbuttons at the same time will return the display to the *Home* panel, regardless of the sub-menu that you are currently using. If you do not press any pushbuttons for several seconds, the display will also return to the home panel.

If an error message is scrolling when the front panel is accessed, the scrolling will be interrupted. When no pushbutton is pressed for several seconds, the scrolling message will return.

If you hold a pushbutton, the selected feature will repeat automatically. If you hold a pushbutton for several seconds, the selected feature will repeat automatically at an accelerated pace. To reset the ZX, press the **UP**, **DOWN**, and **ENTER** pushbuttons together (works like the Reset [**Z**] command).

TUNE Menu

You can select the **TUNE** menu to adjust the system gains for optimum performance. ***The ZX is factory-configured for typical user loads. Hence, many applications do not require tuning***. The following gains are available.

- ❑ **PPnn** Position Proportional Gain
- ❑ **PInn** Position Integral Gain
- ❑ **PDnn** Position Derivative Gain
- ❑ **TGnn** Tachometer Gain
- ❑ **VPnn** Velocity Proportional Gain
- ❑ **VInn** Velocity Integral Gain
- ❑ **VFnn** Velocity Feed-forward Gain

The variable **nn** represents a percentage ranging from **00** to **99**. Use the **UP** and **DOWN** pushbuttons to locate the desired gain parameter on the display panel. To change the gain value, press and hold the **ENTER** pushbutton while using the **UP** or **DOWN** pushbuttons to increase or decrease the gain. When the desired value is reached, release the **UP** or **DOWN** pushbutton and the **ENTER** pushbutton. After you modify the gain, you can now change another gain or press the **UP** and **DOWN** pushbuttons together to return to the main menu. *To change the maximum gain values, you must use a terminal and communicate via RS-232C*.

DISP Menu

Select the **DISP** menu to display ZX parameters on the front panel. To review the respective numerical values, press the **ENTER** pushbutton. The following parameters are may be displayed:

- ❑ **DVEL** Display Actual Shaft Velocity in rpm
- ❑ **DERR** Display Position error in steps (-999 to +999)
- ❑ **DCA** Display Average Current X 100 (0234=2.34 amps)

☞ ***Helpful Hint:***
To return to the Main menu, press the **UP** and **DOWN** pushbuttons simultaneously.

CMTR Menu

Select the **CMTR** menu to configure the motor type (**CMTR** command). The following choices are available:

- ❑ **605 805 910**
- ❑ **606 806 920**
- ❑ **610 810 930**
- ❑ **620 820 940**
- ❑ **630 830**
- ❑ **635 840**
- ❑ **640**
- ❑ **FMCA Find Motor Commutation Angle**

To select a motor size, locating the desired motor size with the up and down pushbuttons and press the **ENTER** pushbutton. The preset motor size is designated by an asterisk. Changing motor sizes also changes some of the tuning parameters. The ZX has been configured at the factory for the motor type that you ordered. *If you change motor sizes, be sure to enter the proper **CMTR** value.* To change from one series to another (i.e., 600, 800, or 900) you must use the **CMTR** command via the RS-232C interface.

> ***WARNING***
> Disconnect the load prior to re-commutating the motor. System damage and/or personal injury can occur during re-commutation if the load is attached.

☞ ***Helpful Hint:***
All of Compumotor's resolvers are pre-aligned to the rotors at the factory, so this procedure is not usually necessary.

CMTR recalculates the mechanical offset between the rotor poles and the stator poles. The offset is factory-set to zero, but you can recalibrate the offset if you select the **FMCA** panel and press the **ENTER** pushbutton to select the **FMCA** command. This command locates the rotor magnets relative to the stator windings and allows you to properly commutate the motor.

MISC Menu

Selecting the **MISC** menu allows you to perform a variety of functions. The following section explains the submenu choices and their functions.

SAVE

Saves the servo tuning parameters to battery-backed RAM. To use, press the **ENTER** pushbutton. ***SV*** will be displayed when this function is executed.

RFS

This option returns all servo parameters to factory settings. To use this command, press the **ENTER** pushbutton. **FSET** will be displayed after the command is executed.

BRnn

This option allows you to change the baud rate (**mm** = 03, 06, 12, 24, 48, and 96—these values represent baud rates 300, 600, 1200, 2400, 4800, and 9600 respectively). To change the baud rate, press the **ENTER** and **UP** or **DOWN** pushbuttons simultaneously (as appropriate).

ADpp

This option allows you to change the device address (**pp** represents a device address from 01 to 99). To change the device address, press the **ENTER** and **UP** or **DOWN** pushbuttons simultaneously (as appropriate).

FOLL or NTFL

If you have a ZX, **NTFL** will be displayed in this menu. No further access is granted.

If you have a ZXF, **FOLL** will be displayed. To change the following ratio on the fly, press the **ENTER** pushbutton. The four least significant digits of the **FOL** command will be displayed. The least significant digit represents the following ratio in tenths (it should be blinking). This indicates that any changes to the following ratio will be in 1/10 increments. To select a higher magnitude of ratio change, press either the **UP** or **DOWN** pushbuttons. This will respectively move the blinking cursor to the left or right. Any changes to the following ratio, will now be at this new digit's magnitude.

100s	10s	1s	0.1s	FOL weighting
a	b	c	d	Blinking digit location

To change the ratio at the blinking digits magnitude, simultaneously press the enter key with the appropriate arrow key. *You must press the **UP** and **DOWN** pushbuttons simultaneously to return to the Main menu.*

SEQU

This option allows you to select and run any of 99 sequences. Press the **ENTER** pushbutton. **XSnn** will be displayed (the variable **nn** represents the current sequence selected). To select a new sequence, press the **ENTER** and **UP** or **DOWN** pushbuttons simultaneously (as appropriate). To run the selected sequence, press only the **ENTER** pushbutton. The display should show **XRnn** and run sequence **nn** repetitively. Pressing only the **ENTER** pushbutton again will return you to the **XSnn** display, where you can select a new sequence. *You must press the **UP** and **DOWN** pushbuttons simultaneously to return to the Main menu.*

REV#

When you press the **ENTER** pushbutton, this menu displays the current microprocessor and

DSP software revision levels (respectively).

JOG

Use the following steps to execute jogging from the front panel:

① Enable the Jog function with the **OSE1** command via RS-232C.
② Enable the pushbuttons with the **CPB1** command via RS-232C.
③ Press ENTER—**HI** will appear. This indicates that the axis will jog at the high jog velocity (**JVH**). Press ENTER again to change **HI** to **LO** (**JVL**).
④ Press ENTER and the up or down pushbuttons (***simultaneously***) to begin jogging. **UP** selects CW motion. **DOWN** selects CCW motion. If no motion occurs, check the status of your limits.
⑤ To return to the Main menu, press the **UP** and **DOWN** pushbuttons simultaneously.

Motion Profile Application Considerations

This section contains information that you should consider and evaluate when designing and developing your system.

Positional Accuracy vs. Repeatability

Some applications require high absolute accuracy. Others require repeatability. You should clearly define and distinguish these two concepts when you address the issue of system performance.

If the positioning system is taken to a fixed place and the coordinates of that point are recorded. The only concern is how well the system repeats when you command it to go back to the same point. For many systems, what is meant by accuracy is really repeatability. Repeatability measures how accurately you can repeat moves to the same position.

Accuracy, on the other hand, is the error in finding a random position. For example, suppose the job is to measure the size of an object. The size of the object is determined by moving the positioning system to a point on the object and using the move distance required to get there as the measurement value. In this situation, basic system accuracy is important. The system accuracy must be better than the tolerance on the measurement that is desired. Consult the technical data section of the Compumotor Catalog for more information on accuracy and repeatability.

Move Times—Calculated vs Actual

You can calculate the time it takes to complete a move by using the acceleration, velocity, and distance values that you define. However, you should not assume that this value is the actual move time. There is calculation delay and motor settling time that makes your move longer. You should also expect some time for the motor to settle into position. The ZX has minimal calculation-delay time associated with a Go (**G**) command. This delay can be as low as 500 µs. The ZX has an internal timer that allows you to monitor the elapsed time of your move. The response of the **TM** command shows you the previous move's execution time.

Predefined Gos

For the fastest possible calculation move times, predefined gos can be used. The **GDEF** command is used to execute predefined gos. The predefined gos are faster because any calculations are performed ahead of time and stored in memory. Refer to the **GDEF** command to execute a predefined go. Perform the following commands to make a predefined go. You can have up to 16 predefined moves.

Command	Description
> GDEF1,A1ØØ,AD1ØØ,V4,D5ØØØØ	Defines predefined move #1
> GD1	Execute predefined move #1

Preset Mode Moves

A preset move is a move distance that you specify in motor steps. You can select preset moves by putting the ZX into Normal mode (**MN** command). Preset moves allow you to position the motor in relation to the motor's previous stopped position (incremental moves) or in relation to a defined zero reference position (absolute moves). You can select incremental moves with the Mode Position Incremental (**MPI**) command. You can select absolute moves with the Mode Position Absolute (**MPA**) command. At any time, you can change the mode you are in and request the state in which the ZX is configured by issuing the **DR** command.

Continuous Mode Moves

The Continuous mode (**MC**) command accelerates the motor to the velocity that was last specified with the Velocity (**V**) command. The motor continues to move at the specified velocity until Stop (**S**) or Kill (**K**) is issued (or a velocity change is specified). To change velocity while the motor is moving, use the instantaneous velocity command (**IV**). Another way to change velocity while moving is to enter Motion Profiling mode (**MPP**).

In Motion Profiling mode, all buffered commands are executed immediately—therefore you only have to enter the **V** command to change the velocity. No **G** is needed following the **V**.

Continuous mode is useful for the following applications:

- ❑ Applications that require constant movement of the load and motion is not based on distance, but on internal variables or external inputs
- ❑ When the motor must be synchronized to external events such as trigger input signals.

☞ ***Helpful Hint:***
In this example, velocity is changed based on external inputs

Command	Description
> PS	Pauses motion until a **C** command is reached
> MPP	Places the ZX in **MPP** mode
> IN1A	Sets up **I1** (Input 1) as trigger bit 1
> IN2A	Sets up **I2** (Input 2) as trigger bit 2
> LD3	Disables the CW and CCW limits ***(This command is not necessary if the limits are installed)***
> MC	Sets unit to the Continuous mode
> A25	Sets acceleration to 25 rps^2
> AD25	Sets acceleration to 25 rps^2
> V1	Sets velocity to 1 rps
> G	Executes the move (Go)
> T1	Waits 1 sec after the motor reaches constant velocity
> V5	Sets velocity to 5 rps
> TR1Ø	Waits for trigger bit 1 to go on and bit 2 to go off
> STOP	Stops the motor
> NG	Ends Profiling mode
> C	Continues execution of commands

These commands cause the ZX to run in Continuous mode. The motor reaches 1 rps, waits for 1 second, changes velocity to 5 rps, waits for you to turn **I1** (Input 1) on and turn **I2** (Input 2) off, and then stops. *The **VØ** and **STOP** commands stop the motor (the **S** command is not a buffered command and cannot be used in this situation, unless you wish to halt the operation in the middle of the program).* The **DIN** command (an immediate command) simulates the state you want the inputs to be in. In the example above, you could simulate the activation of the trigger state without physically toggling the inputs, by using the **DIN** command as follows.

```
> DINEEE1ØEEEEE
```

E means *do not affect the input*. A **1** makes the input one, a **Ø** makes the input zero. Each **E**, **1**, or **Ø** represents an input bit. There are 10 inputs. The **1** and **Ø** in this example correspond to **I1** and **I2** on the front panel. The first 3 **E**'s correspond to the CCW, CW, and Home limits.

Incremental Mode Preset Moves

When you are in Incremental mode (**MPI**), a preset move moves the motor the specified distance from its starting position. You can specify the direction with the optional sign (**D+8ØØØ** or **D-8ØØØ**), or you can define it separately with the Change Direction (**H+** or **H-**) command.

Command	Description
> LD3	Disables CW & CCW limits (*not needed if limits are installed*)
> MPI	Sets unit to Incremental Position Mode
> MN	Places the ZX in the preset mode
> PZ	Zeroes the position counter
> A25	Sets acceleration to 25 rps^2
> AD25	Sets deceleration to 25 rps^2
> V5	Sets velocity to 5 rps
> D8ØØØ	Sets distance to 8,000 steps
> G	Executes the move (Go)
> D12ØØØ	Sets distance to 12,000 steps
> G	Initiates motion
> 1PR	Reports the setpoint (commanded) position Response: **2ØØØØ**

Absolute Mode Preset Moves

A preset move in the Absolute mode (**MPA**) moves the motor to the distance in an absolute coordinate system that you specify relative to an absolute zero position. You can set the absolute position to zero with the Position Zero (**PZ**) command or by cycling the power to the indexer. The absolute zero position is initially the power-up position.

The direction of an absolute preset move depends upon the motor position at the beginning of the move and the position you command it to move to. If the motor is at absolute position +12,800, and you instruct the motor to move to position +5,000, the motor will move in the negative direction a distance of 7,800 steps to reach the absolute position of +5,000.

The ZX powers up in Incremental mode. When you issue the Mode Position Absolute (**MPA**) command, it sets the mode to absolute. When you issue the Mode Position incremental (**MPI**) command the unit switches to Incremental mode. The ZX retains the absolute position, even while the unit is in the Incremental mode. You can use the Position Report (**PR**) command to read the absolute position.

In the following example, the motor performs the same commands as the incremental position example. In this case, the **PR** command will report a different position because it is working in an absolute coordinate system.

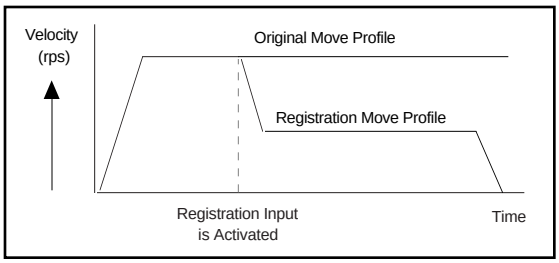
☞ ***Helpful Hint:***
The motor will move to absolute position 8,000. The second move is 4,000 more steps to the absolute position of 12,000 steps. The **PR** command reports a setpoint (commanded position) of 12,000 steps.

Command	Description
> LD3	Disables the CW & CCW limits (*not needed if limits are installed*)
> MPA	Sets unit to Incremental Position Mode
> PZ	Zeroes the position counter
> A25	Sets acceleration to 25 rps^2
> AD25	Sets deceleration to 25 rps^2
> V5	Sets velocity to 5 rps
> D8ØØØ	Sets distance to 8,000 steps
> G	Executes the move (Go)
> D12ØØØ	Sets distance to 12,000 steps
> G	Initiates motion

> 1PR Reports the setpoint(commanded) position
Response: **12ØØØ**

Registration

Registration with the ZX provides the ability to change the move profile which is being executed to an unrelated move profile defined as a registration move. **This unrelated registration move is executed when an input to the ZX transitions from a high to a low level.** You can only define input #7 (**I7**) as a registration input (refer to the **IN** command [**IN7Q**] in the ***ZX Indexer/Drive Software Reference Guide***). The registration input has the highest priority of any input. Upon receiving the registration input the motor's current position is captured within 50 microseconds. The registration move profile is executed during the next update period. The registration move profile uses the actual position captured (within 50 microseconds) as its starting or zero reference point.

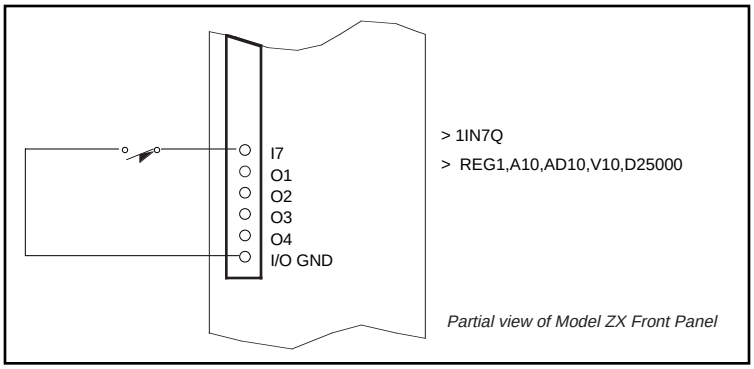


Registration Move

The interrupt is *edge-sensitive* to the voltage high-to-low transition (if you have a bouncy switch for the registration input, use the debounce [**TDR**] command to ensure that false registration interrupts do not occur). With the ZX, a registration interrupt can interrupt another registration move.

The ZX has 7 programmable inputs labeled **I1** through **I7**. Only **I7** can be defined as registration input. **REG1** defines the move when input **I7** is configured as a registration input.

☞ ***Helpful Hint:***
When **I7** is defined as a registration input, it **cannot** be toggled at a frequency higher than 1 kHz, or once every millisecond.



ZX Registration

The syntax for defining a registration move using input **I7** is:

> REG1,A1Ø,AD1Ø,V1Ø,D25ØØØ

The registration move **REG1** will be performed when **I7** is activated. The acceleration (**A1Ø**) will be 10 rps^2, the deceleration will be 10 rps^2, the velocity will be 10 rps, and the distance traveled will be 25,000 steps.

Bouncy Registration Inputs

The switch for the registration may be bouncy or noisy and may take a few milliseconds to settle (since the registration inputs to the ZX are interrupts that are edge sensitive). Hence, a bouncy switch each edge appears like a registration interrupt and the registration move is made from the distance position at which the latest edge was detected. The ZX allows you to debounce the inputs (with a software command). You can ignore any bouncing transitions from your switch after the initial registration interrupt occurred. The time in which the interrupts or false edges are ignored is determined by the number you enter for the **TDR** command. **TDR** is the debounce time in milliseconds that you specify so the inputs cannot cause another registration interrupt until the switch settles.

In the following example, an application needs a registration move. The motor has a resolution of 5,000 steps per revolution. The move must turn the motor 1 revolution at 10 rps. If an input does not occur, the move will be a 500,000-step move. The ZX is

configured as follows:

☞ ***Helpful Hint:***
If **I7** is toggled (voltage high to low), the corresponding registration move is run. The **DIN** command cannot be used to activate the registration input (the registration inputs are hardware oriented).

Command	Description
> 1IN7Q	Defines **I7** as a registration input
> REG1, A1Ø, AD1Ø, V1Ø, D5ØØØ	Defines the registration move
> D5ØØØØØ	Sets distance to 500,000 steps
> V5	Sets velocity to 5 rps
> G	The preset move is initiated

Jogging the Motor

In some applications, you may want to move the motor manually. You can configure the ZX to allow you to move the motor manually with the Configure Input (**IN**) command. Define the jogging velocity with the Jog Velocity High (**JVH**) and Jog Velocity Low (**JVL**) commands. You can define three different inputs for jogging: CW Jog input (**IN#J**), CCW Jog input (**IN#K**), and Jog Speed Select High/Low (**IN#L**). You must also enable the jogging feature with **OSE1**. Once you set up these parameters, you can attach a switch to the jog inputs that you defined and perform jogging (**#** represents digits 1 - 7, which you enter). The following example shows how to define power-up sequence 100 to set up jogging.

Step ①

Define a power-up sequence.

Command	Description
> XE1ØØ	Erase sequence #100
> XD1ØØ	Define sequence #100
> LD3	Disables the limits (*not needed if you have limit switches installed*)
> JA25	Set jog acceleration to 25 rps^2
> JAD25	Set jog deceleration to 25 rps^2
> OSE1	Enables Jog function
> JVL.5	Sets low-speed jog velocity to 0.5 rps
> JVH5	Sets high-speed jog velocity to 5 rps
> IN1J	Sets **I1** as a CW jog input
> IN2K	Sets **I2** as a CCW jog input
> IN3L	Sets **I3** as a speed-select input
> XT	Ends sequence definition

Step ②

Reset the ZX.

Command	Description
> **Z**	Resets the ZX

Step ③ Turn on **I1** to move the motor CW at 0.5 rps (until you turn off **I1**).

Step ④ Turn on **I2** to move the motor CCW at 0.5 rps (until you turn off **I2**).

Step ⑤ Turn on **I3** to switch to high-speed jogging.

Step ⑥ Repeat steps ③ and ④ to perform high-speed jogging.

Backlash Compensation

The ZX has the capability to compensate for backlash in the gearing of your system. You can specify different compensations depending on the direction the motor is moving. You will use the **BL** command for backlash compensation. Refer to the ***ZX Indexer/Drive Software Reference Guide*** for a detailed description of the backlash command. The syntax of the command is as follows:

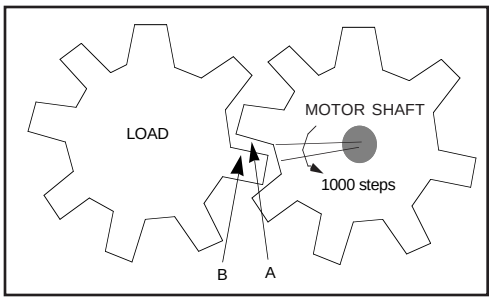
Command	Description
> BLn,m	Variable **n** is the amount of CCW steps that should be compensated. Variable **m** is the number of CW steps that will be compensated.

If a CW move is made, an extra **m** steps will be made to account for the backlash. The concept is that the load will not begin to move until the motor moves enough to make contact with the gearing. The same is true for the opposite direction. The backlash compensation may be different in either direction so you can program them independently. The load will not move until point A contacts point B. This is the backlash associated with changing direction. Assuming this distance to be 1,000 motor steps the Backlash command will be typed as follows:

Command	Description

> BL1ØØØ,2ØØØ **1ØØØ** is the amount of CCW steps that should be compensated. **2ØØØ** is the number of CW steps that will be compensated.

For a 25,000-step CCW move, the motor will actually move 26,000 steps to remove the backlash but the position counter will only change by 25,000 steps. This is done because the load is expected to move whenever the motor moves. The load should move a certain distance when the motor moves a certain distance. If there is backlash in the system, the load will not move the correct distance when the motor is commanded to move in the opposite direction from its previous move. In this example, the motor was moving in the CW direction. The gearing was flush and the teeth were touching. When the direction changes, the teeth must move 1,000 steps before they are again in contact with the load gear. Thus for the load to move 25,000 steps the motor will have to move 1,000 steps until the teeth are in contact then move 25,000 steps so that the load will actually move 25,000 steps. This mode allows you to compensate for the system error between the motor and the actual load position. *The compensation only occurs when you change direction*.



Backlash Compensation

Defining a Home Location

The ZX's go home function brings the motor to a home reference position. The homing function allows backs the motor to a home switch and stops it on a specific edge of the switch. You can program the active level of the switch. The homing function also allows you to home to the Z channel of the ZX motor's resolver. This occurs at the rollover between resolver position 0 and 65535. Use the Display Position Resolver (**DPR**) command to locate this position.

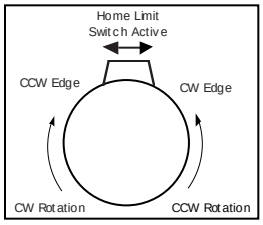
Homing to a Switch

The ZX home position is located where the edge selected with the **OS** command of the Home Limit input occurs (i.e., the ZX recognizes the home position as the position where the home limit signal makes a transition from on to off, or from off to on, depending on the selected edge and the initial direction of the go home move).

Once it recognizes the selected edge , the motor decelerates to a stop. After stopping, the ZX positions the servo motor away from the selected edge of home limit signal in the opposite direction of the final approach direction of the go home move. After coming to a stop a second time, the ZX creeps the motor towards the selected edge at Go Home Final Velocity (**GHF**) until the home limit input becomes active again or the home limit and the ZX's resolver Z Channel pulse are active.

You must ensure that the final approach starts from the opposite side of the signal from the selected edge. If the final approach direction is positive , the final move must start from the negative side of the selected edge. If there is significant backlash and friction in the system, and the indexer is instructed to go home in the CW direction, the motor may end up on the wrong side of the signal and execute its final approach in the wrong direction.

This problem can also occur if the motor's go home speed is high and the Home limit signal is delayed (by a relay or PLC for example). In such a situation, initiate homing operations from the opposite side of the selected edge of home. When the homing operation is complete, the indexer resets its internal position counter.



Go Home Description

You can use the home limit input in conjunction with the resolver's internal Z Channel input to select a final home position. To enable **Z** Channel homing, you must activate the **OSD1** command. In this situation, a load-activated switch connected to the Home Limit input locates the general home position area, and the ZX's internal resolver roll-over position is used for final Home positioning. ***The internal Z Channel pulse and the Home Enable input must both be active to mark the home position***.

Go Home

Under interface control, the Go Home (**GH**) command has the form **GH**<direction><velocity>. This indicates which direction to move, and at what velocity. For example, the command **GH-2** sends the motor in the negative direction at 2 rps in search of the home signal. The go home velocity can also be set by the **GHV** command. The last specified value will be used for the go home.

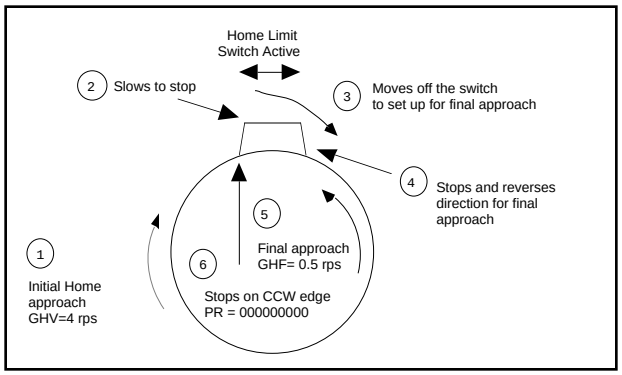
If you use an input to implement a go home move, the velocity provided by the **GHV** command is used to execute the homing function. The acceleration and deceleration parameters for this move are the home value for acceleration (**GHA**) and deceleration (**GHAD**). If an end-of-travel limit is activated before home is found, the indexer reverses direction and attempts to find the home position again. If the other limit is activated before the indexer finds home, the indexer will stop trying to go Home. The indexer can indicate whether or not the homing process was successful by responding to the Request Indexer Status (**R**) and Go Home Status (**RG**) commands. After a successful home move, the position counter is reset to zero.

Go Home Example

Command	Description
> LD3	Disables positive and negative limit inputs
> OSB1	Back up to home limit
> OSG1	The final approach direction is CCW
> OSH1	The reference edge is the CCW edge
> GHA1	Sets go home acceleration to 1 rps^2
> GHAD1	Sets go home deceleration to 1 rps^2
> GH.5	Sets go home in the positive direction at 0.5 rps

The motor starts to move (CW) toward the home position. Upon reaching the home switch, the motor stops. It continues to move CW at the final go home speed until it is off of the home switch. It will then be in position to reverse direction and make its final go home approach (CCW). It will encounter the home switch on the CW edge first and will continue to move until the switch is inactive which will be the CCW edge.

☞ ***Helpful Hint:***
This figure illustrates the homing procedure for these set up commands.



Go Home Example

Creating Motion Programs and Sequences

You must program the ZX to perform motion functions. A motion program consists of initialization (or ZX set-up), move profiles, and an I/O or RS-232C interface to execute motion instructions.

Sequence Commands

Sequences are the building blocks of motion programs for the ZX. A sequence can be one command or up to 8 K bytes of commands. The sequences are stored in battery backed RAM. A sequence is a set of commands that is executed by issuing that sequence number. The ZX has programming capabilities that send program control from one sequence to another (i.e., a **GOTO** statement). The ZX also allows you to transfer program control to another sequence and return to the point of transfer (i.e., a subroutine [**GOSUB**] call). The following

commands define, erase, and run sequences as well as other specialized sequence functions. Refer to the ***ZX Indexer/Drive Software Reference Guide*** for detailed descriptions and syntax of the following commands.

Sequence Status Commands

Command	Description
> XBS	Reports the number of bytes available for sequence programming
> XC	Sequence checksum report
> XDIR	Reports defined sequences and the bytes of memory they occupy
> XSD	Sequence status definition report
> XSR	Sequence status run report
> XSS	Sequence status report

Sequence Programming Commands

Command	Description
> XD	Starts sequence definition
> XT	Ends sequence definition
> XE	Erases a sequence
> XEALL	Deletes all sequences from battery-backed RAM

Sequence Execution Commands

Command	Description
> XQ	Sets/resets Interrupted Run mode
> XRP	Runs a sequence with a pause
> XR	Runs a sequence
> SSJ1	Runs a sequence defined by binary weighted sequence inputs

Sequence Branching Commands

Command	Description
> XG	Exits current sequence and moves to execute another sequence
> GOTO	Exits current sequence and moves to execute another sequence
> XR	When used within a sequence it jumps to execute another sequence then returns to the sequence from which it was called
> GOSUB	Jumps to execute another sequence then returns to the sequence from which it was called

Sequence Debugging Commands

Command	Description
> XTR	Sequence Trace mode
> XST	Sequence Single Step mode
> XS	Sequence Execution status
> #	Step sequence command
> DIN	Simulate input state command
> DOUT	Simulate output state command

Special Sequence Commands

Command	Description
> WHEN	Special condition command
> XWHEN	Special condition sequence
> XFK	Fault sequence

A sequence is a series of commands. These commands are executed in the order in which they are programmed when the sequence is run. Immediate commands cannot be stored in a sequence, just as they cannot be stored in the command buffer. Only buffered commands may be used in a sequence.
The ZX has 8,000 bytes of nonvolatile memory to store up to 100 sequences. You can use the **XBS** command to determine how many bytes are available in the sequence buffer and the **XDIR** command to determine what sequences have been programmed. The sequence buffers may have variable lengths, so you may have one long sequence or several short ones, as long as the total length does not exceed the 8,000 bytes of allocated space.

The commands that you enter to define a sequence are presented vertically in the examples below. This was done to help you read and understand the commands. When you are actually typing these commands into your terminal, they will be displayed horizontally.

To begin sequence definition, enter the Define Sequence (**XD**) command immediately followed by a sequence number (1 to 100) and a delimiter. The Terminate Sequence (**XT**) command ends sequence definition. All commands that you enter after **XD** and before **XT** will be executed when the sequence is run (see example below). Type **DR** to see the state of the ZX.

Command	Description
> 1DR	Displays the present state of the ZX

Perform the following commands.

Command	Description
> MPI	Places the ZX in the incremental mode
> MN	Places the ZX in the preset mode
> FSIØ	Places the ZX in the indexer mode
> LD3	Disables the ZX's limits

Command	Description
> XE1	Erases Sequence #1
> XD1	Begins definition of Sequence #1
> A25	Sets acceleration to 25 rps^2
> AD25	Sets deceleration to 25 rps^2
> V1Ø	Sets velocity to 10 rps
> D5ØØØ	Sets distance to 5,000 steps
> G	Executes the move (Go)
> H	Reverses direction
> G	Executes the move (Go)
> XT	Ends definition of sequence
> XR1	Runs Sequence #1

You can run a sequence by entering the **XR** command immediately followed by a sequence identifier number (1 to 100) and a delimiter.

Once you define a sequence, it cannot be redefined until you delete it. You can delete a sequence with the **XE** command immediately followed by a sequence number (1 to 100) and a delimiter, then redefine the sequence. *You can use **XEALL** to delete all defined sequences from battery-backed RAM. **Use XEALL with extreme caution—erased sequences cannot be retrieved**.*

Sequence #100 is a power-up sequence (if you have defined it). It is always run when you power up the system or when you reset the indexer with the Reset (**Z**) command. For convenience, you may find it advantageous to place all of your set-up commands in Sequence #100.

Sequences that you define are automatically saved into the ZX's nonvolatile memory. The only way to erase these sequences is by using the Erase Sequence (**XE**) or Erase All Sequences (**XEALL**) commands.

Creating and Executing Sequences

You can create sequences via RS-232C. Before you create sequences, you must understand the types of motion and the required user interfaces. To determine the proper user interface, you should be familiar with the methods of selecting sequences within your application.

Selecting Sequences

After you define the sequences from the RS-232C interface, you can execute the sequences by using one of the following modes of operation:

- ❏ *Stand Alone*: Use thumbwheel switches to select and run the sequence. See *Stand Alone Operation* in this chapter for more on this feature.
- ❏ *Computer Interface*: Use the Execute Sequence (**XR**) command to run sequences. See *Stand Alone Operation* in this chapter for more on this feature.
- ❏ *PLC (Programmable Logic Controller):* Use the sequence select inputs to run a sequence. See *Stand Alone Operation* in this chapter for more on this feature.
- ❏ *Front Panel Pushbutton Execution*: Refer to the *Pushbutton Operation* section earlier in this chapter.
- ❏ *Remote Panel*: Use a remote panel to select programs. See *Standalone Operation* in this chapter for more on this feature.

Subroutines

When you use the **GOTO** Sequence (**XG**) and execute a sequence (**XR**) command, you can execute different sequences from within a sequence. These commands can be substituted for the **GOTO** and **GOSUB** commands respectively. If you use **XG** or **GOTO**, the program will move to the sequence that you specified in **XG** or **GOTO**. After executing the specified sequence, the system will not return to the original sequence. It will remain in the current sequence, unless it receives another execution command (**XG/GOTO** or **XR/GOSUB**). However, if you use the **XR** or **GOSUB** command, the program will return control to the original sequence that contained **XR** or **GOSUB**. Program control will return to the original sequence when a Terminate Sequence (**XT**) command is reached. This prompts the program to return to the sequence that initiated the move to another sequence.

☞ ***Helpful Hint:***
The **XG** command has no limit since the program will not return control to the original sequence.

You can nest up to 16 different levels of sequences within one program. For example, when you exit Sequence #1 to execute Sequence #2 with the **XR2** command, you can execute Sequence #3 from Sequence #2. This nesting procedure can be repeated 16 times.

☞ ***Helpful Hint:***

Command	Description

In the previous example, when you execute Sequence #1, the program moves to Sequence #2. After executing Sequence #2, the program returns to Sequence #1. The program then moves to execute Sequence #3. Trace mode was enabled to help you see how the sequence was executed.

> XE2	Erases Sequence #2
> XD2	Defines Sequence #2
> A1ØØ	Sets acceleration to 100 rps^2
> AD1ØØ	Sets deceleration to 100 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25000 steps
> G	Executes the move (Go)
> XT	Ends Sequence 2 definition
> XE3	Erases Sequence #3
> XD3	Defines Sequence #3
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D-25ØØØ	Sets distance to 25000 steps
> G	Executes the move (Go)
> XT	Ends Sequence #3 definition
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
> XR2	Executes Sequence #2
> GOSUB3	Executes Sequence #3 (same as an **XR** command)
> XT	Ends Sequence #1 definition
> 1XTR1	Enables Trace mode
> XR1	Executes Sequence #1

Asynchronous Events—FAULT and WHEN

The ZX has special sequences that can run when a certain condition occurs. One such sequence is the *power-up sequence* (sequence 100). The **FAULT** and **WHEN** sequences operate similarly.

FAULT Sequence

The fault sequence is automatically executed when a fault condition or a Kill (**K**) occurs. Any condition that faults the ZX (error flashes) activates the fault sequence—if one is defined. If a kill command is issued, the sequence will also run. You can use the fault sequence to place the ZX in a safe state and turn off outputs that may be harmful to the rest of the system. You can use an **IF** command to determine what condition caused the fault so the fault can be remedied. The following steps illustrate the use of a fault sequence. The **IF** statement section explains more about the **ERnnnnn** flag, which indicates what fault condition occurred.

Step ① Define an input as a user fault input. You can use this input to indicate that a fault has occurred somewhere external to the system. This input will cause a fault condition.

Command	Description
> IN1U	Defines input #1 as a user fault input

Step ② Designate Sequence #10 as the fault sequence.

Command	Description
> XFK1Ø	Designates sequence 10 as the fault sequence
> XE1Ø	Erases sequence #10
> XD1Ø	Defines sequence #10
> 1"External_Fault	Quote command
> XT	End definition of sequence #10

The quote command sends a message over the RS-232C link to the terminal to tell the operator that a fault has occurred. You can use the Quote command to write statements to the terminal. Sequence #10 will now be executed whenever a fault occurs. You may now program. In the following example, an alternating loop will be performed.

Step ③ The normal *state* of this example application will be an alternating loop.

☞ ***Helpful Hint:*** The motor will alternate back and forth continuously.

Command	Description
> A5Ø	Acceleration is 50 rps^2
> AD4Ø	Deceleration is 40 rps^2
> D5ØØØØ	Distance is 50000 steps
> V7	Velocity is 7 rps
> L	Loop command
> G	Initiates motion
> H	Change direction
> N	End the loop

Step ④ You will now cause a system user fault error. Input states can be simulated with the **DIN**

command.

☞ ***Helpful Hint:***
Sequence #10 should automatically execute when the fault occurs.

Command	Description
> DINEEE1	Input #1 is activated

Error #66 should scroll across the ZX's display. To clear the fault, enter the following command.

Command	Description
> ON	Clears fault message

An **IF** statement could have been used to determine what fault condition occurred, then branched to a sequence that handled that fault condition appropriately. For example, the fault sequence will run for several faults. These faults are *limits reached*, *general servo fault*, and *user faults*. Each bit in the error flag (**ERnnnnnnn**) corresponds to one of these faults. Use the **IF** statement to determine which one occurred. Retype sequence #10 (the fault sequence) as follows:

☞ ***Helpful Hint:***
Depending on which error caused the program to branch to the fault sequence, the appropriate message will be displayed. Disable the fault sequence with the **XFKØ** command.

Command	Description
> 1XE1Ø	Erases sequence 10
> 1XD1Ø	Defines sequence 10
> IF(ER1)	IF the CCW limit is hit then
> 1"CCW_LIMIT_WAS_HIT	Display a message
> NIF	End the IF (ER1) statement
> IF(ERX1)	IF CW limit is hit then
> 1"CW_LIMIT_WAS_HIT	display a message
> NIF	End the IF (ERX1) statement
> IF(ERXXXXXX1)	IF the user fault occurred, display a message
> 1"USER_FAULT	
> NIF	Ends the IF (ERXXXXXX1) statement
> XT	Ends fault sequence statement

WHEN Sequence

The **WHEN** sequence runs when a specific condition is true. This could be a variable having a certain value, the inputs being in a specific state or the user flag having a set state. Whatever sequence or program is currently running will be interrupted when the condition is true and the sequence designated by the **XWHEN** command will be executed. In the following example, when *Variable 3* > 50, or *Input #1* is on, the ZX will execute the **XWHEN** sequence.

Command	Description
> WHEN(VAR3>5Ø_OR_INXXX1)	The when statement that must evaluate true in order for the **XWHEN** sequence to be run is defined.

Step ① The **WHEN** sequence is now defined. Sequence #8 is the **WHEN** sequence.

Command	Description
> XWHEN8	Sequence 8 is designated as a **WHEN** sequence
> XE8	Erases sequence #8
> XD8	Begin definition of Sequence #8
> A5Ø	Acceleration is 50 rps^2
> AD4Ø	Deceleration is 40 rps^2
> D3ØØØØ	Distance is 30000 steps
> V5	Velocity is 5 rps
> G	Initiates motion
> XT	Ends the sequence definition

Step ② The normal program to be executed is defined and executed here.

Command	Description
> VAR3=Ø	Variable 3 is initialized to 0
> A5Ø	Acceleration is 50 rps^2
> AD4Ø	Deceleration is 40 rps^2
> D5ØØØØ	Distance is 50000 steps
> V7	Velocity is 7 rps
> L	Loop command
> G	Initiates motion
> VAR3=VAR3+1	Variable 3 is increased
> N	End the loop

Step ③ You can cause the **WHEN** sequence to occur using either the **DIN** command to activate the input that will satisfy the **WHEN** condition or the current program will run 50 times, then the **WHEN** sequence will be executed.

Step ④ Enter **XWHENØ**. You can use the **WHEN** statement to change the mode of operation. This command can also preview the effect of multiple **WHEN** statements. Using an input to execute the **XWHEN** sequence can change or interrupt the program. Within the **XWHEN** sequence, you can use an **IF** statement to check the state of one or more inputs. Based on the state of the inputs, different sequences can be executed. Disable the **XWHEN** sequence before continuing with this procedure by entering **XWHENØ.**

Power-Up Sequence Execution

The ZX can be programmed to execute a sequence of commands on power up (sequences can be used as subroutines). Refer to the ***ZX Indexer/Drive Software Reference Guide*** for detailed descriptions and syntax of the following commands.

☞ ***Helpful Hint:***
Sequence #100 always runs on power up. To run another sequence on power up, put an **XR**<num> (or **XG**<num>) at the end of sequence #100. If sequence #100 is empty, nothing happens on power up.

Command	Description
> XE1ØØ	Erases sequence #100
> XD1ØØ	Begins definition of sequence #100
> LD3	Disables limits
> A2Ø	Sets acceleration to 20 rps^2
> AD2Ø	Sets deceleration to 20 rps^2
> V5	Sets velocity to 5 rps
> D125ØØ	Sets distance to 12,500 steps
> MN	Sets to Normal mode
> MPI	Sets into the incremental mode
> FSIØ	Sets unit to the indexer mode
> XG1	Go to sequence #1
> XT	Ends sequence definition
> XE1	Erases sequence #1
> XD1	Defines sequence #1
> G	Executes a go command
> XT	Ends definition of the sequence
> Z	Resets the indexer and runs sequence #100

A power-up sequence typically stores set-up or initialization parameters that your application requires. *Having motion in your power up sequence is not recommended.* Examples of set-up commands are listed below.

☞ ***Helpful Hint:***
You can put any buffered commands into sequence #100 (if you want to execute them during power up).

Command	Description
> SSJ1	Continuous Sequence Scan Mode
> SN	Scan time
> JA	Jog acceleration
> JVL	Jog velocity low
> JVH	Jog velocity high

Sequence Debugging Tools

After sequences are created, you may need to debug them to ensure that they perform the functions properly. The ZX provides several debugging tools.

- ❏ In Trace mode, you can trace a sequence as it is executing.
- ❏ You can set the state of the ZX 's inputs & outputs via software commands.
- ❏ You can enable error messages to explain why the ZX has stopped execution due to a programming error.

Trace Mode

You can use the Trace mode to debug a sequence or a program of sequences. Trace mode tracks, command-by-command, the entire sequence as it runs. It displays to your terminal, over the RS-232C serial link, all commands as they are executed. The following example demonstrates Trace mode.

Step ① Create the following sequence:

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
> A1Ø	Acceleration is 10 rps^2
> AD1Ø	Deceleration is 10 rps^2
> V5	Velocity is 5 rps
> L5	Loop 5 times
> GOSUB3	Jump to sequence #3
> N	Ends the loop
> XT	Ends the definition of sequence #1

Step ② Define sequence #3.

Command	Description
> XE3	Erases a sequence
> XD3	Defines sequence #3
> D5ØØØØ	Sets the distance to 50000 steps
> G	Initiates motion
> XT	Ends the definition of sequence #3

Step ③ Enter the following command to enable Trace mode.

Command	Description
> 1XTR1	Enables Trace mode

Step ④ Execute the sequence. The commands will be displayed on the terminal as each command in the sequence is executed. Enter the following command.

Command	Description
> XR1	Run sequence #1—response:

```
*SEQUENCE_ØØ1____COMMAND_A1Ø
*SEQUENCE_ØØ1____COMMAND_AD1Ø
*SEQUENCE_ØØ1____COMMAND_V5
*SEQUENCE_ØØ1____COMMAND_L5
*SEQUENCE_ØØ1____COMMAND_GOSUB3____LOOP_COUNT_1
*SEQUENCE_ØØ3____COMMAND_D5ØØØØ____LOOP_COUNT_1
*SEQUENCE_ØØ3____COMMAND_G____LOOP_COUNT_1
*SEQUENCE_ØØ3____COMMAND_XT____LOOP_COUNT_1
*SEQUENCE_ØØ1____COMMAND_N____LOOP_COUNT_1
*SEQUENCE_ØØ1____COMMAND_GOSUB3____LOOP_COUNT_2
*SEQUENCE_ØØ3____COMMAND_D5ØØØØ____LOOP_COUNT_2
*SEQUENCE_ØØ3____COMMAND_G____LOOP_COUNT_2
*SEQUENCE_ØØ3____COMMAND_XT____LOOP_COUNT_2
*SEQUENCE_ØØ1____COMMAND_N____LOOP_COUNT_2
*SEQUENCE_ØØ1____COMMAND_GOSUB3____LOOP_COUNT_3
*SEQUENCE_ØØ3____COMMAND_D5ØØØØ____LOOP_COUNT_3
.
.
.
*SEQUENCE_ØØ3____COMMAND_G____LOOP_COUNT_5
*SEQUENCE_ØØ3____COMMAND_XT____LOOP_COUNT_5
*SEQUENCE_ØØ1____COMMAND_N____LOOP_COUNT_5
*SEQUENCE_ØØ1____COMMAND_XT
```

The format for Trace mode display is: **Sequence Number_Command_Loop Count**

Step ⑤ To exit Trace mode, enter the following command:

Command	Description
> XTRØ	Exits Trace mode

Single-Step Mode

Single-Step mode is another level of debugging. This mode allows you to execute one command at a time as desired. Use **XST** to enable Single-Step mode. To execute a command, use the **#** sign. By entering **#** followed by a delimiter, you will execute the next command in the sequence. If you follow the **#** with a number (*n*) and a delimiter, you will execute the next *n* commands. The following steps demonstrate Single-Step mode.

Step ① Enable Single-Step mode.

```
> XST1
```

Step ② Begin execution of sequence #1

```
> XR1
```

Step ③ You will not execute any commands until you use the **#** command.

Command	Description
> #	Executes one command

The response will be:

```
*SEQUENCE_ØØ1____COMMAND_A1Ø
```

Step ④ To execute more than one command at a time, follow the **#** with the number of commands you want to execute.

Command	Description
> #3	Executes 3 commands, then pauses sequence execution

To complete the sequence, use the **#** sign until all the commands are completed. To exit Sequence-Step mode, type:

```
> XSTØ
```

Simulating I/O Activation

If your application has inputs and outputs that integrate the ZX with other components in your system, you can simulate the activation of these inputs and outputs so that you can run your sequences without activating the rest of your system. Thus, you can debug your program independently of the rest of your system. This is the same way in which a PLC program can be debugged by simulation of input and output states to run various portions of the program. The ZX uses two commands that allow you to simulate the input and output states desired. The **DIN** command controls the inputs and the **DOUT** command controls the outputs.

You will generally use the **DIN** command to cause a specific input pattern to occur so that a sequence can be run. Use the **DOUT** command to simulate the output patterns that are needed to prevent an external portion of your system from operating. You can set the outputs in a state that will be the inactive state of your external system. When you execute your program, a part in the program that will activate the outputs will not actually turn the outputs on to their active state because the **DOUT** command overrides this output and holds the external portion of the machine in an inactive state. When the program is running smoothly without problems you can activate the outputs and the ZX will affect the external system.

Outputs

The following steps describe the use and function of the **DOUT** command.

Step ① Display the state of the outputs with the **OUT** and **O** commands.

Command	Description
> 1OUT	Displays the state of the outputs

The response will be:

```
*1_A_PROGRAMMABLE_OUTPUT_____(STATUS_OFF)
*2_A_PROGRAMMABLE_OUTPUT_____(STATUS_OFF)
*3_A_PROGRAMMABLE_OUTPUT_____(STATUS_OFF)
*4_A_PROGRAMMABLE_OUTPUT_____(STATUS_OFF)
```

Command	Response
> 1O	*ØØØØ
> DOUT11EE	

Step ② Change the output state using the **O** command.

Command	Description
> O111Ø	

Step ③ Display the state of the outputs with the **OUT** and **O** commands.

Command	Description
> 1OUT	Displays the state of the outputs

The response will be:

```
*1_A_PROGRAMMABLE_OUTPUT_____(DISABLED_ON)
*2_A_PROGRAMMABLE_OUTPUT_____(DISABLED_ON)
*3_A_PROGRAMMABLE_OUTPUT_____(STATUS_ON)
*4_A_PROGRAMMABLE_OUTPUT_____(STATUS_OFF)
```

Command	Response
> 1O	*111Ø

Step ④ You can now disable the outputs into the inactive state using the **DOUT** command. An **E** does not affect the output.

> DOUT00EE

Step ⑤ Display the state of the outputs with the **OUT** and **O** commands.

Command	Description
> 1OUT	Displays the state of the outputs

The response will be:

```
*1_A_PROGRAMMABLE_OUTPUT_____(DISABLED_OFF)
*2_A_PROGRAMMABLE_OUTPUT_____(DISABLED_OFF)
*3_A_PROGRAMMABLE_OUTPUT_____(STATUS_ON)
*4_A_PROGRAMMABLE_OUTPUT_____(STATUS_OFF)
```

Command	Response
> 1O	*ØØ1Ø

Inputs

The following steps describe the use and function of the **DIN** command. You can use it to cause an input state to occur. The inputs will not actually be in this state but the ZX treats them as if they are in the given state and will use this state to execute its program.

Step ① This sequence will wait for a trigger state to occur and will then begin moving in Continuous mode. An input that is configured as a stop (**S**) input will stop motion.

Command	Description
> 1IN1A	Input #1 is a trigger input
> 1IN2A	Input #2 is a trigger input
> 1IN3D	Input #3 is a stop input
> 1INLØ	The active input level is low
> 1XE1	
> 1XD1	Define sequence #1
> TR11	Waits for the input trigger state to be 11
> A1ØØ	Acceleration is set to 100 rps^2
> AD1ØØ	Deceleration is 100 rps^2
> V5	Velocity is 5 rps
> MC	The continuous mode is activated
> TR1Ø	Waits for a trigger input state of 10
> G	Begins motion
> XT	Ends sequence definition

Step ② Turn on the Trace mode so that you can view the sequence as it is executed.

Command	Description
> 1XTR1	Turns on Trace mode

Step ③ Execute the sequence.

Command	Description
> XR1	Runs sequence #1

Step ④ When the **TR11** command is encountered, program execution will pause until the trigger condition is satisfied. Simulate the input state with the **DIN** command. *Inputs with an **E** value are not affected.*

> 1DINEEE11EEEEE

Step ⑤ When the **TR1Ø** trigger is encountered, program execution will pause for the new input pattern. Use the **DIN** command to simulate the desired input state.

> 1DINEEE1ØEEEEE

Step ⑥ The motor will move continuously until a the stop input is activated. Activate the stop input with the **DIN** command.

> 1DINEEE1Ø1EEEE

☞ ***Helpful Hint:*** To deactivate the I/O simulation commands:

> 1DINEEEEEEEEE

> 1DOUTEEEE

Error Messages

The ZX has an Error Message mode that displays an error message when an invalid command is attempted. This error message can be useful in debugging a sequence. The **SSN** command enables Error Message mode. The following commands demonstrate this mode.

Command	Description
> 1SSN1	Enters the error message mode
> 1D1ØØØØØØØØØ	

The ZX will respond with an error message: ***INVALID_DATA_FIELD**

Command	Description
> 1DK1ØØØØ	

The ZX will respond with an error message: ***INVALID_COMMAND**

The error message mode can be exited by typing:

Command	Description
> 1SSNØ	Exits the error message mode

Data

Inputs can be defined as data inputs to allow for external entry of motion data, loop counts, sequence select, time delays, and variable values. The following commands will read and enter data from the inputs.

☞ ***Helpful Hint:*** The recommended method of controlling the input lines to read data is through thumbwheels. A PLC may also be used to enter data.

- ❑ **VARD** Variable Read
- ❑ **DRD** Distance
- ❑ **VRD** Velocity
- ❑ **LRD** Loop Count
- ❑ **TRD** Time Delay
- ❑ **XRD** Sequence Number
- ❑ **FRD** Following Ratio

The input lines when configured as data inputs are weighted differently than sequence select inputs. The weighting for data inputs is *binary coded decimal* or BCD. This weighting allows you to enter data via thumbwheels. To use the data inputs to enter data, the outputs must also be used. Outputs 1 - 3 must be configured as data strobe outputs. They are used to select or strobe the appropriate while reading data. Up to 16 digits of data and one sign bit may be entered.

Delays

You can use the Time (**T**) command to halt the operation of the indexer function for a preset time. In *Continuous mode*, you may use the Time (**T**) command to run the motor at continuous velocity for a set time, then change to a different velocity. In *Preset mode*, the motor finishes the move before the indexer executes the time delay.

Command	Description
> PS	Waits for the indexer to receive a **C** command before executing the next command
> D25ØØØ	Sets distance to 25000 steps
> G	Moves motor 25,000 steps
> T5	Waits 5 seconds after the move ends
> H	Changes motor direction
> G	Moves motor 25,000 steps in the opposite direction
> C	Continues execution

High-Level Programming Tools

The Model ZX's X-language includes some commands that are common in most high-level programming languages (Pascal, Fortran or BASIC). In addition to these commands, 30 variables **VAR1-VAR3Ø** are provided for performing mathematical functions and Boolean comparisons. You can access some system variables—**POS** (Commanded Position or Setpoint) and **FEP** (Following Encoder Position).

Branching commands evaluate condition statements (see below) to make branching decisions. If the condition is true, one set of commands is processed—if the condition is false, another set of commands will be executed.

☞ ***Helpful Hint:***
The condition statements that are evaluated can be very complex. The condition statements support all of the following decisions.

IF (condition true)—*execute these commands*
ELSE—*execute these commands*
NIF

WHILE (condition true)—*execute these commands*
NWHILE

REPEAT—*Execute these commands*
UNTIL (condition true)

Variables

The ZX has up to 30 variables that can perform multiplication, division, addition, and subtraction. You can assign these variables to various motion parameters. These parameters and the syntax of assigning a variable to them are listed here.

☞ ***Helpful Hint:***
Variable assignments can be made in sequences or in Immediate Terminal mode. If a variable with a fractional portion is assigned to a parameter that requires a whole number (such as distance) the variable is rounded to the nearest whole number and assigned to the parameter.

Command	Description
D(VAR1)	Loads the distance with the value of variable 1
V(VAR4)	Loads the velocity with the value of variable 4
A(VAR5)	Loads the acceleration with the value of variable 5
AD(VAR7)	Loads the deceleration with the value of variable 7
FP(VAR9)	Loads the following point with the value of variable 9
DP(VAR1)	Loads the distance point with the value of variable 1
L(VAR3Ø)	Loads the loop count with the value of variable 3Ø
XR(VAR21)	Executes the sequence number held in variable 21
T(VAR3)	Loads the time delay with the value of variable 3
FOL(VAR29)	Loads the following ratio with the value of variable 29

Assigning Variables to Constants

You can set parameters as variables in a sequence (**D**(**VAR**)). You can define these variables by assigning a constant value. When the sequence is run, this value is assigned to the corresponding parameter in the sequence.

Step ①

The sequence below executes a move and travels the distance provided in variable #1 and the velocity provided in variable #2.

Command	Description
> LD3	Disables limits (*Not needed if limits are installed*)
> MPI	Places the ZX in incremental mode
> MN	Enters the normal mode
> FSIØ	Places a ZXF in the indexer mode. *Required only with Following option.*
> XE1	Erases sequence #1
> XD1	Defines sequence #1
> A1Ø	Sets the Acceleration
> AD1Ø	Sets the Deceleration
> V(VAR2)	Assigns variable #2 to the velocity term
> D(VAR1)	Assigns variable #1 to the distance term
> G	Initiates motion
> XT	Ends the sequence definition

Step ②

Assign numbers to the variables **VAR1** and **VAR2**

Command	Description
> VAR1=25ØØØ	The distance parameter is assigned 25000
> VAR2=5	The velocity parameter is assigned 5

Step ③

Execute sequence #1 (**XR1**). The motor will move 25000 steps at 5 rps. Verify that the distance (**D**) and velocity(**V**) have been assigned these values.

Command	Description
> 1V	*V05.00000
> 1D	*D+0000250ØØ

Step ④

Now change the values of the variables as follows.

Command	Description
> VAR1=75ØØØ	The distance parameter is 75,000
> VAR2=1Ø	The velocity parameter is assigned 10

Step ⑤

Repeat steps ③ and ④.

Entering Variables Via RS-232C

When using variables in sequences, the **RSIN** command can interactively prompt the user to enter a variable number.

Step ①

Define the following sequence:

☞ ***Helpful Hint:***
To enter the variable, you must precede the number with an exclamation point (!).

Command	Description
> 1XE1	
> 1XD1	Defines sequence #1
> 1"ENTER_THE_NUMBER_OF_PARTS	Prompts you for the # of parts to make
> 1CR	Inserts a carriage return
> 1LF	Inserts a line feed
> VAR5=RSIN	Sequence stops and waits for a # to be entered into variable 5
> 1CR	Inserts a carriage return
> 1LF	Inserts a line feed
> L(VAR5)	The loop count is loaded with the number of parts to make
> D25ØØØ	Distance is 25000 steps
> G	Initiates motion
> N	Ends the loop

```
> 1"FINISHED_WITH_PARTS_RUN    Indicates that the run is finished
> 1CR                          Inserts a carriage return
> 1LF                          Inserts a line feed
> XT                           Ends sequence definition
```

The **RSIN** command prompts you to enter a variable via RS-232C that will be used in a sequence. When a variable is assigned to **RSIN**, the execution of the sequence is stopped until you enter the variable.

Step ② Turn on Trace mode and run the sequence.

```
> 1XTR1
> XR1
```

Step ③ The program will stop at the **RSIN** command and wait for you to be enter a variable number. Enter the variable number.

```
> !1Ø
```

The sequence executes a 25,000-step move 10 times making 10 parts.

Math Operations

In addition to assigning constants to variables, two system parameters can be assigned to a variable.

- **POS**—Commanded Position
- **FEP**—Following Encoder Position

These parameters are assigned as if they were constants. For example:

Command	Response
> VAR1=POS	Assigns the current value of the command position to variable #1

Performing Math Operations with Variables

The ZX has the ability to perform simple math functions with its variables (add, subtract, multiply, and divide). When performing these math functions, the ZX *truncates* the decimal after five places to the right. The following sequence of steps illustrates the math capabilities of the ZX.

Step ① **Addition:**

Command	Response
> VAR1=5	
> VAR23=1ØØØ.565	
> VAR11=VAR1+VAR23	
> VAR23=1Ø+VAR23	
> VAR1=VAR1+5	
> 1VAR1	*+ØØØØØØØØØØØØØ1Ø.ØØØØØ
> 1VAR11	*+ØØØØØØØØØØØ1ØØ5.565ØØ
> 1VAR23	*+ØØØØØØØØØØØ1Ø1Ø.565ØØ

Step ② **Subtraction:**

Command	Response
> VAR3=1Ø	
> VAR2Ø=15.5	
> VAR3=VAR3-VAR20	
> VAR19=VAR2Ø-VAR3	
> 1VAR3	*-ØØØØØØØØØØØØØ5.5ØØØØ
> 1VAR2Ø	*+ØØØØØØØØØØØØØ15.5ØØØØ
> 1VAR19	*+ØØØØØØØØØØØØØ21.ØØØØØ

Step ③ **Multiplication:**

Command	Response
> VAR3=1Ø	
> VAR2Ø=15.5	
> VAR3=VAR3*VAR2Ø	
> VAR19=99	
> VAR19=VAR2Ø*VAR19	
> 1VAR3	*+ØØØØØØØØØØØØ155.ØØØØØ
> 1VAR2Ø	*+ØØØØØØØØØØØØØ15.5ØØØØ
> 1VAR19	*+ØØØØØØØØØØØ1534.5ØØØØ

Step ④ **Division**:

Command	Response

```
> VAR3=1Ø
> VAR2Ø=15.5
> VAR3=VAR3/VAR2Ø
> VAR3Ø=75
> VAR19=VAR3Ø/VAR3
> 1VAR3                    *+ØØØØØØØØØØØØØØØ.64516
> 1VAR2Ø                   *+ØØØØØØØØØØØØØ15.5ØØØØ
> 1VAR3Ø                   *+ØØØØØØØØØØØØØ75.ØØØØØ
> 1VAR19                   *+ØØØØØØØØØØØØ116.25Ø23
```

Complex Branching and Looping

The ZX supports the high-level language structures for branching and looping. Each conditional branch or loop evaluates a condition statement. Depending on whether this condition statement evaluates true or not determines where the ZX will branch to. The unconditional branching and looping statements have been introduced already. These are the **GOTO** (Branch), **GOSUB** (Branch & Return) and **L** (Loop) command. The **L** command is explained further here.

Unconditional Looping

The loop command is an unconditional looping command You may use the Loop (**L**) command to repeat a series of commands. You can nest Loop commands up to 16 levels deep.

☞ ***Helpful Hint:***
The motor moves a total of 10,000 steps

Command	Description
> PS	Pauses execution until the indexer receives a Continue (**C**) command
> MPI	Sets unit to Incremental mode
> A50	Sets acceleration to 50 rps^2
> V5	Sets velocity to 5 rps
> L5	Loops 5 times
> D2ØØØ	Sets distance to 2,000 steps
> G	Executes the move (Go)
> T2	Delays 2 seconds after the move
> N	Ends loop
> C	Initiates command execution to resume

☞ ***Helpful Hint:***
The example below shows how you can nest a loop inside a loop. In this example, the motor makes two moves and returns a line feed. The unit repeats these procedures until you instruct the it to stop [Stop (**S**) or Kill (**K**)].

```
Description                 Command
Pauses command execution    > PS
Loops indefinitely          > L ──────────────────┐
Sends a line feed           > 1LF                 │
Loops twice                 > L2 ──────┐          │
Executes 2,000-step move    > G        │ Nested   │ Loop
Waits Ø.5 seconds           > T.5      │ Loop     │
Ends loop                   > N ───────┘          │
Ends loop                   > N ──────────────────┘
Continues command           > C
execution
```

Unconditional Branching

The unconditional branching commands **GOTO** and **GOSUB** were explained earlier in the sequence section.

Conditional Statements

You can use the following types of conditional statements with the ZX.

- ❏ Error Flags
- ❏ User Flags
- ❏ Input State
- ❏ Boolean Comparisons

Error Flags

The error flag (**ERXXXXXXX**) is useful if you want to trap different error conditions and create different sequences to respond to them. The ***ZX Indexer/Drive Software Reference Guide*** explains the errors that can be trapped in the evaluation command description. An example of the statement's use is provided below.

```
> IF(ER110XXXX)
> GOSUB2
> NIF
```

User Flags

You can set the user flag (**FLØØØ111XØ**) and modify it within sequences to mark where the program has gone or to indicate any special state so a conditional statement can be made.

```
> WHILE(FLØØ11XXXX)
> D1ØØ
> G
```

```
> NWHILE
```

Input State

An example of this statement's use (**IN1111ØØØ1Ø**) is provided below.

```
> REPEAT
> GOSUB4
> UNTIL(IN0Ø1XX11)
```

Variable Comparisons

- VARn>VARm
- VARn<VARm
- VARn=VARm

```
IF(VAR1>VAR2)
WHILE(VAR3=1Ø)
GOSUB2
NWHILE
NIF
```

Boolean Comparisons

An example of the statement's use is provided below.

```
WHILE(VAR3>1Ø_AND_IN11ØØ11)
GOSUB8
NWHILE
```

Conditional Looping

The ZX supports two conditional looping structures—**REPEAT/UNTIL** and **WHILE**.

REPEAT/ UNTIL

With the **REPEAT** command, all commands are repeated between **REPEAT** and **UNTIL**. The ZX stops executing the commands when the **UNTIL** condition is true.

Step ①

Command	Description
`> VAR5=Ø`	Initializes variable 5 to 0
`> 1XE1Ø`	Erases sequence #10
`> 1XD1Ø`	Defines sequence #10
`> REPEAT`	Begins the **REPEAT** loop
`> A5Ø`	Acceleration is 50 rps^2
`> AD5Ø`	Deceleration is 50 rps^2
`> V5`	Sets velocity to 5 rps
`> D25ØØØ`	Distance is 25,000 steps
`> G`	Executes the move (Go)
`> VAR5=VAR5+1`	Variable 5 counts up from 0
`> UNTIL(INXXX111Ø_OR_VAR5>1Ø)`	When 111Ø input condition occurs or VAR5 is > 10, the loop will stop
`> 1"DONE_LOOPING`	Quote command indicates that the loop is finished
`> 1CR`	Inserts a carriage return
`> 1LF`	Inserts a line feed
`> XT`	Ends sequence definition

Step ②

Use the Trace mode to display the commands as they are run.

```
> 1XTR1
> XR1Ø
```

The loop can be exited either by using the **DIN** command to satisfy the input state or letting the **VAR5** counter count to 10. When using a **REPEAT/UNTIL** loop, all of the commands between **REPEAT** and **UNTIL** are executed at least once before the loop reaches the evaluation statement in the **UNTIL** line. This is not true with the **WHILE** loop.

WHILE

With the **WHILE** command, all commands between **WHILE** and **NWHILE** are repeated **WHILE** condition evaluates true. The evaluation of the condition is done at the end of any current move in progress.

Step ①

Command	Description
`> VAR5=Ø`	Initializes variable 5 to 0
`> 1XE1Ø`	Erases sequence #10
`> 1XD1Ø`	Defines sequence #10
`> WHILE(INXXX111Ø_OR_VAR5<1Ø)`	While the input pattern is equal to XXX111Ø or variable 5 is less than 10, repeat the loop
`> A5Ø`	Acceleration is 50 rps^2
`> AD5Ø`	Deceleration is 50 rps^2
`> V5`	Velocity is 5 rps
`> D25ØØØ`	Distance is 25000 steps
`> G`	Executes the move (Go)
`> VAR5=VAR5+1`	Variable 5 counts up from 0
`> NWHILE`	Last command in the WHILE loop
`> 1"DONE_LOOPING`	Indicates that the loop is done

> 1CR	Inserts a carriage return
> 1LF	Inserts a line feed
> XT	Ends sequence definition

Step ② Use the Trace mode to display commands as they are run.

```
> 1XTR1
> XR1Ø
```

You can exit the loop with the **DIN** command (cause the input state to not match the **IN** command). You can also exit the loop by setting the **VAR5** counter to 10. If the input pattern is not **XXX111Ø**, the loop will not be run.

Conditional Branching

You can use the **IF** statement for conditional branching. All commands between **IF** and **ELSE** are executed if the condition is true. If the condition is false, the commands between **ELSE** and **NIF** are executed. If the **ELSE** is not needed, it may be omitted. The commands between **IF** and **NIF** are executed if the condition is true. Examples of these statements are provided.

- ❏ Error Flag
- ❏ User Flag
- ❏ Input State

Error Flag

The **IF** command can check for error conditions. If an error exists, a conditional command may be executed. This command is useful if you wish to trap different error conditions (Drive Disabled, User Fault Input Activated, Excessive Position Error, etc). Refer to the ***ZX Indexer/Drive Software Reference Guide*** for more information on the **ER** evaluation with the **IF** command.

Command	Description
> XE1Ø	Erases sequence #10
> XD1Ø	Defines sequence #10—when a fault occurs, sequence #10 will run—the sequence is defined with **XFK1Ø** (below)
IF(ER1)	If hardware CCW limit switch is reached, perform following commands:
1"CCW_LIMIT_HIT	Display the error message
NIF	Ends IF statement
IF(ERX1)	If hardware CW limit switch is reached, perform following commands:
1"CW_LIMIT_HIT	Display the error message
NIF	Ends IF statement
XT	Ends Sequence loop
> XFK1Ø	Sets Sequence #10 as the Fault sequence

User Flag

This command uses the pattern set by the User Flag (**SFL**) command to run conditional commands. This command is useful if you want to make a decision based on previous sequence executions that will set or clear the user flag bits. For example, if an application has several sequences, you can assign different bit patterns with the **SFL** command at the end of each sequence. If you select these sequences from the host computer, you may wish to make different moves depending on the sequence you ran.

☞ ***Helpful Hint:*** Refer to the ***ZX Indexer/Drive Software Reference Guide*** for a description of **SFL**.

Command	Description
> PS	Waits for the indexer to receive a Continue (**C**) command before executing the next command
> SFL1Ø1Ø	Sets user flag bits 7 & 5 and clears bits 6 & 4, remaining bits not altered
•	
•	
•	
> IF(FL1Ø1Ø)	If user flag bits 5 & 7 are set, and bits 6 & 4 are clear, perform following commands
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps
> G	Executes the move (Go)
> NIF	Ends IF statement
> C	Continues execution

If the **FL** pattern matches the **SFL** setting, the motor moves 25,000 steps. You can change the **SFL** pattern at different points in sequences to map a path for sequence execution.

Input State

This command compares the input pattern (**CW**, **CCW**, **HM** and **I1** - **I7**) to execute the conditional commands. This command is useful for branching and performing conditional moves using the programmable inputs. For a detailed description of this command, refer to the ***ZX Indexer/Drive Software Reference Guide***.

Command	Description
> 1XE5	Erases sequence #5
> 1XD5	Defines sequence #5
IF(INXXX1Ø)	If **I1** is active and **I2** is not active, issue the following commands:
A1Ø	Sets acceleration to 10 rps^2
V5	Sets velocity to 5 rps
D25ØØØ	Sets distance to 25,000 steps
G	Executes the move (Go)
NIF	Ends IF statement
IF(INXXXØ1)	If **I1** is not active (open) and **I2** is active (closed), issue the following commands:
A1Ø	Sets acceleration to 10 rps^2
V5	Sets velocity to 5 rps

D-5ØØØ	Sets distance to 5,000 steps in the opposite direction
G	Executes the move (Go)
NIF	Ends IF statement
IF(INXXX1)	If **I1** is active, do the following command.
1"DONE	Ends message saying done
NIF	Ends IF statement
> 1XT	Ends sequence definition

Use **DIN** or the inputs to execute the different trigger input states. You can use the Trace mode to see what commands are executed.

Branching Using Variables and Boolean Logic

You can use the **IF** statement to branch based on variable values. Multiple comparisons can be made in one condition statement using the Boolean **OR** and **AND** functions. The limitation of how many comparisons can be made is limited to a command line length of 80 characters.

Command	Description
> XE8	Erases sequence 8
> XD8	Defines sequence 8
VAR5=15	Variable 5 = 15
IF(VAR5>1Ø_AND_VAR4=2Ø)	If Variable 5 is greater than 10 and variable 4 = 20 then perform the commands until the ELSE
A1ØØ	Sets acceleration to 100 rps^2
AD1ØØ	Sets deceleration to 100 rps^2
V5	Sets velocity to 5 rps
D25ØØØ	Sets distance to 25,000 steps
G	Executes the move (Go)
VAR5=VAR5-1	Variable 5 decrements one
ELSE	Ends IF statement
A1ØØ	Sets acceleration to 100 rps^2
AD1ØØ	Sets deceleration to 100 rps^2
V5	Sets velocity to 5 rps
D-5ØØØ	Sets distance to 5,000 steps in the opposite direction
G	Executes the move (Go)
NIF	Ends the IF
> XT	Ends sequence definition

Motion Profiling Mode—On-the-Fly Changes

The Motion Profiling mode allows you to execute buffered commands while a move is being made (on-the-fly). When you enter this mode, the ZX will process all the commands that you enter immediately. You can enter and exit this mode from within a sequence. This mode allows you to change velocity on-the-fly based on distance, turn on outputs based on distance, perform math and other commands while in motion. The following commands are used with Motion Profiling mode.

- **MPP**—Enter Motion Profiling Mode
- **NG**—Exits Motion Profiling Mode
- **DP**—Sets Distance Points within Motion Profiling Mode
- **FP**—Sets Distance Points from a following encoder with Motion Profiling mode (ZXF only, refer to *Chapter ⑤ Following*)

While the ZX is in Motion Profiling mode, you can execute any command while a move is being made. When the ZX encounters an **MPP** command in a sequence, all subsequent commands will be executed until the **NG** command is encountered. An example of the **MPP** command is provided below.

```
XD1 D5ØØØØ V1 MPP G O1 TR1X1 V4 NG XT
```

In this example, a 50,000-step move is made. The initial velocity is 1 rps. Motion begins with the **G** command. Output 1 is turned on with the **O1** command. The ZX then waits at the trigger command (**TR**) until the condition is true, at which point it changes the velocity. When the ZX encounters the **NG** command, it will not receive any additional commands until the 50,000-step move is completed. If the Trigger condition is not met during the 50,000 step move, the move is completed and the program waits on the trigger condition before finishing the sequence.

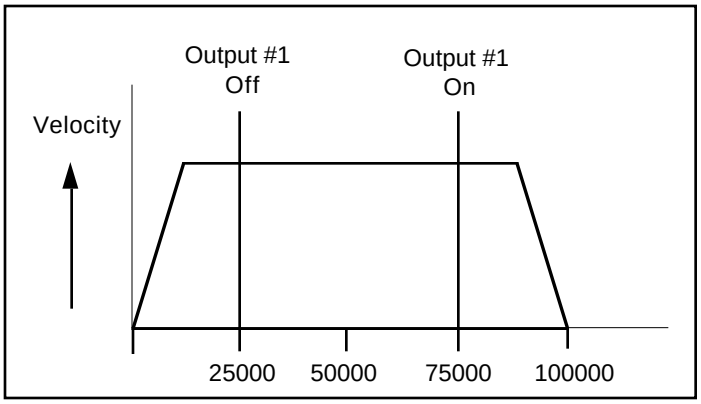
Changes Based on Distance

Changes are made based on distance using the distance point (**DP**) command. This command causes a delay in the processing of the commands until the motor reaches the specified distance point. Processing will continue once the distance point is reached. In this way, velocity changes and the activation of outputs can be done based on distance.

The distance point is interpreted differently for Absolute mode versus Incremental mode. To change velocity on-the-fly based on distance, the Motion Profiling Mode (**MPP**) command must be used with the Distance Point (**DP**) command.

☞ ***Helpful Hint:***
The following sequence example executes the profile shown in the figure below.

```
1XD1 PZ D1ØØØØØ V2 MPP G DP25ØØØ O1 DP5ØØØØ OØ NG XT
```



In Incremental mode, commands are processed until the DP command is reached. The ZX pauses at the **DP** command until the motor moves 25,000 steps. The ZX then turns on output #1. The 25,000 steps are counted from the point at which the command is encountered. When **DP5ØØØØ** is reached, the ZX pauses until the motor moves 50,000 steps. The ZX then turns output #1 off. In this example, if the ZX in Incremental mode, the output will be turned on at 25,000 steps and turned off after the motor has moved a total of 75,000 steps from the beginning of the first move.

In absolute mode, the value specified with **DP** is interpreted as an absolute position relative to the zero point location. In this example, the output would be turned on at 25,000 steps and turned off after the motor had passed the 50,000 step point.

Stopping Motion with a Stop Input

A common use of the Motion Profiling mode is to perform a continuous move and stop the move from the inputs. This can be done in two ways. You can define an input as a stop input. Motion can be stopped by activating the stop input.

The other method is to use the **STOP** command and place it within a sequence. The following step-by-step example illustrates these two methods.

Step ① Define the input as a stop input.

Command	Description
> IN1D	Defines input #1 as a stop input

Step ②

Create the sequence with a continuous move.

Command	Description
> XD1	Begins definition of sequence #1
V5	Sets velocity to 5 rps
A1ØØ	Sets acceleration to 100 rps^2
MC	Sets the ZX to Continuous mode
MPP	Sets the ZX to Motion Mode Profiling
G	Executes the move (Go)
NG	Exits Motion Profiling mode
> XT	Ends sequence #1 definition

Step ③ Execute sequence #1 with **XR1**. The motor will move at 5 rps and will not stop until you activate the stop input (input #1). Activate the stop input.

The motor will stop at a controlled deceleration. In this case, the buffer will be dumped and the sequences will not be finished. The **STOP** command caused program control to exit the sequence. The **G** command was the last command that was run.

Step ④ Issue the Display Parameters (**DR**) command. The ZX is still in Motion Profiling mode. Enter the **NG** command to exit the mode.

Stopping Motion (STOP Command)

The following step-by-step example illustrates the method of stopping a continuous move with the **STOP** command.

Step ① Configure input #1 as a trigger input.

Command	Description
> IN1A	Configures input #1 as a trigger input

Step ② Create a sequence with a **STOP** command after the trigger.

Command	Description
> XD1	Begins the definition of sequence #1
V5	Sets velocity to 5 rps
A1ØØ	Sets acceleration to 100 rps^2
MC	Sets the ZX to Continuous mode
MPP	Sets the ZX to Motion Mode Profiling
G	Executes the move (Go)
TR1	Activates trigger #1
STOP	Stops motion when the trigger condition is met
NG	Exits Motion Profiling mode
> XT	Ends sequence #1 definition

Step ③ Issue the Display Parameters (**DR**) command. The ZX is still in Motion Profiling mode.

In both of the examples, an **NG** command was required after motion was stopped. When a **STOP** command is issued, the command buffer is emptied. Therefore, the commands that have not been executed in the sequence at the time the stop occurs will not be executed. In both examples, the **NG** command is not executed because motion was stopped. In fact, **NG** can never be executed under these conditions. ***The important point is that if the STOP command is used to stop continuous motion, the NG must be issued either at the beginning of the next sequence or directly via the RS-232C interface***.

To prevent the ZX from stopping without finishing the sequence that it is currently executing, a software switch has been provided that will cause the ZX to continue executing the sequence it was running when the **STOP** was issued. By entering the Clear/Save the Command Buffer on Stop (**SSH1**) command, the ZX will stop motion when it encounters a **STOP** and continue processing the commands in the sequence. *Enter **SSH1** and repeat the previous example. Notice how the Motion Profiling mode is exited.*

Sequence Scan Mode and the Stop Command

In applications that use the Sequence Scan mode (**SSJ1**) and the **STOP** command, you must enable **OSI1** to prevent the Sequence Scan mode from being disabled after a stop. Under normal conditions, Sequence Scan mode is aborted when the ZX encounters a **STOP** command. If the **STOP** command is issued from a stop input or from within a sequence, Sequence Scan mode will be aborted. Usually, you will not want to abort the mode. The **SSH1** command will allow the ZX to complete the execution of the current sequence from the point at which the **STOP** was issued. The following example illustrates such a sequence.

Step ① Configure input #1 as a trigger input and sequence #2 as a sequence-select input.

Command	Description
> IN1A	Configures input #1 as a trigger input
> IN2B	Configures input #2 as a sequence select input
> OUT1A	Enables output #1 as a programmable output

Step ② Enable the Sequence Scan mode with the **SSJ1** command.

Enable the Save Buffer on Stop with the **SSH1** command.

Enable the Preserve Sequence Scan mode with the **OSI1** command.

Step ③ Create a sequence with a **STOP** command after the trigger.

Command	Description
> XD1	Begins the definition of sequence #1
V5	Sets velocity to 5 rps
A1ØØ	Sets acceleration to 100 rps^2
MC	Sets the ZX to Continuous mode
MPP	Sets the ZX to Motion Mode Profiling
G	Executes the move (Go)
TR1	Activates trigger #1
STOP	Stops motion when the trigger condition is met
NG	Exits Motion Profiling mode
O1	Turn on output #1
> XT	Ends sequence #1 definition

Step ④ Execute the sequence by activating input #2. Stop the move at any time by activating input

#1. Output #1 will be activated after the stop.

Step ⑤ The ZX is still in Sequence Scan mode and the move can be repeated by activating input #2 again (and stopped with input #1).

Other Uses of the Motion Profiling Mode

Motion Profiling mode provides flexibility in the complexity of the ZX control during motion. You can change gains on-the-fly based on distance or following error. You can turn on outputs, change velocity, and perform math functions. The primary application concern to consider during sequence execution is the time required to perform the commands. In some cases, the execution of commands may depend on the motion. The following examples show additional uses of the Motion Profiling mode.

Changing Gains On-the-Fly

Command	Description
> 1XD1	Begins the definition of sequence #1
1D400000	Sets distance to 400,000 steps
1V5	Sets velocity to 5 rps
1MPP	Sets ZX to Motion Profiling mode
1G	Executes the move (Go)
1DP150000	Sets distance point to 150,000
1V2	Changes velocity to 2 rps
1BCPP20	Changes gain to 20% of maximum
1DP200000	Sets distance point to 200,000
V5	Changes velocity to 2 rps
BCPP45	Changes gain to 45% of maximum
NG	Exits ZX from Motion Profiling mode
> XT	Ends the definition of sequence #1

Turning on Inputs, Using Time Delays, and Math

Command	Description
> 1XD1	Begins the definition of sequence #1
1PZ	Sets position counter to Ø
1MC	Sets the ZX to Continuous mode
1MPP	Sets ZX to Motion Profiling mode
1G	Executes the move (Go)
1T.5	Sets a 0.5 second delay
1O11Ø	Turns outputs #1 and #2 on, #3 off
1T.5	Sets a 0.5 second delay
1OØØ1	Turns outputs #1 and #2 off, #3 on
REPEAT	Starts repeat loop
VAR1=VAR1+1	Increases variable #1 by 1
T.1	Sets a 0.1 second delay
UNTIL(POS>400000)	Continues looping until the ZX's position is > 400,000
STOP	Halts command processing
NG	Exits ZX from Motion Profiling mode
> XT	Ends the definition of sequence #1

Triggers, input states (**INXX111**) time delays, and the distance points are the commands that will allow you to control when and where procedures occur during motion in your program. The Motion Profiling mode offers you the flexibility to accomplish a variety of different application needs.

Interfacing to the ZX

This section discusses the various interfaces that you may use with the ZX.

- ❏ Operation from Programmable Inputs and Outputs
  - Switches
  - Thumbwheels
- ❏ PLC Operation
- ❏ Front Panel Operation
- ❏ Remote Panel Operation
- ❏ Host Computer Operation

Programmable Inputs and Outputs

The ZX has a flexible scheme for defining I/O in a way that is suitable for most applications. There are 7 programmable inputs (**I1** - **I7** on the front panel). The other three inputs are dedicated for limits. There are also 4 programmable outputs. Using the inputs in combination with the outputs you can use up to 16 digits of thumbwheels with the ZX. This section explains some of the functions that the inputs and outputs can perform and how to use thumbwheels for an interface with the ZX.

Output Functions

You can turn the programmable outputs (**O1** - **O4**) on and off with the Output (**O**) and Immediate Output (**IO**) commands. Outputs **O1** - **O4** are factory set as programmable outputs. However, you can configure all of the outputs to perform different functions (Moving/Not Moving, Amp Off, Strobe, etc.) with the Configure Output (**OUT**) command. Refer to the **OUT** command in the ***ZX Indexer/Drive Software Reference Guide*** for descriptions of the available functions.

☞ ***Helpful Hint:***
You can use these outputs to turn on and off other devices (i.e., lights, switches, etc.). The output functions have unique letter assignments.

A:	Programmable Output	**K:**	Invalid Command Error
B:	Moving/Not Moving	**L:**	Position Error Fault
C:	Sequence in Progress	**M:**	Not Used
D:	At Soft of Hard Limits	**N:**	CW Software Limit Reached
E:	At Position Zero	**P:**	CCW Software Limit Reached
F:	Fault Indicator	**R:**	CW Hardware Limit Reached
G:	Not Used	**S:**	CCW Hardware Limit Reached
H:	Amp Off	**T:**	Output Based on Position
J:	Strobe Out	**U:**	Programmable Pulse output

☞ ***Helpful Hint:***
To see how the outputs are currently defined, type **1OUT**.

Command	Description
> PS	Pauses command execution until the indexer receives a Continue (**C**) command
> MN	Sets unit to Normal mode
> A1Ø	Sets acceleration to 10 rps^2
> V5	Sets velocity to 5 rps
> D25ØØØ	Sets distance to 25,000 steps
> OUT1A	Sets **O1** as a programmable output
> OUT2A	Sets **O2** as a programmable output
> OUT3B	Sets **O3** as a Moving/Not Moving output
> O1Ø	Turns **O1** on and **O2** off
> G	Executes the move
> OØ1	Turn **O1** off and **O1** on
> C	Initiates command execution to resume

O1 and **O2** are programmable outputs and **O3** as a Moving/Not Moving output. Before the motor moves 25,000 steps, **O1** turns on and **O2** turns off. These outputs remain in this state until the move is completed, then **O1** turns off and **O2** turns on. While the motor is moving, **O3** remains on.

Pulse Output Half Axis

The ZX can programmed a second half axis. The output half axis can control distance and constant speed for another drive/motor system. A second output can control the distance. The following commands set up a second axis and make it move.

Command	Description
> OUT1U	Configures the output to be a pulse output
> OUT2A	The output will be used to control direction
> PU1ØØØ,2	The output will send 1000 pulses at a rate of 2 ms
> PUL	This command begins the pulse train out of output 1

Input Functions

The inputs can individually be programmed to perform any of the following functions. Each function has an assigned letter:

A:	Trigger	**N:**	Data
B:	Sequence Select	**0:**	No Function assigned
C:	Kill	**P:**	Memory Lock
D:	Stop	**Q:**	Registration input (I7 Only)
E:	Command Enable	**R:**	Reset
F:	Pause/Continue	**S:**	Go Home
G:	Go	**T:**	Position Zero
H:	Direction	**U:**	User Fault
I:	Synchronization	**V:**	Data Valid
J:	Jog+ (CW)	**W:**	Data Sign
K:	Jog- (CCW)	**X:**	Increase Following ratio
L:	Jog Speed Select	**Y:**	Decrease Following ratio
M:	Not Used	**Z:**	No Function assigned

To designate each input pin to a particular function, use the Set Input Functions (**IN**) command. To current input definitions, type **1IN**. To see the inputs' states, use **1IS**. Enter the following commands.

> 1IN: Change input 1 to be a stop input by entering:

> IN1D: Check that it was assigned properly by again entering:

> 1IN1: With this method, you can assign all the inputs to any of the functions listed above. (*Except the registration input function. Only **I7** can be assigned as a registration input.*)

Switches

This section covers ZX triggers and sequence scanning with inputs.

Triggers

You can use the Trigger (**TR**) command to pause a sequence of buffered commands until one or more inputs come to a preferred state. **I1** - **I7** are set at the factory (default setting) to function as trigger inputs.

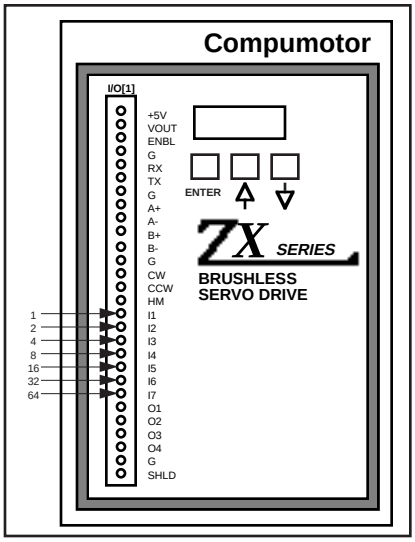
Command	Description
> IN1D	Sets **I1** as Stop(S) input
> IN2A	Sets **I2** as trigger input 1
> IN3A	Sets **I3** as trigger input 2
> IN4A	Sets **I4** as trigger input 3
> MC	Sets the unit to Continuous mode
> MPP	Enters the Motion Profiling mode
> A15	Sets acceleration to 15 rps^2
> AD15	Sets acceleration to 15 rps^2
> TR1	Waits for trigger input 1 (**I2**)to be on
> G	Executes a go (Go) command
> L	Loops infinitely
> V5	Sets velocity to 5 rps
> TRXØ1	Waits for trigger input 2 (**I3**) to be off and trigger input 3 (**I4**) to be on
> V1	Sets velocity to 1 rps
> TRX1Ø	Waits for trigger input 2 (**I3**) to be on and trigger input 3 (**I4**) to be off
> N	Ends the loop

This program configures **I1** as a stop input and **I2** as a trigger (start) input. **I3** and **I4** are programmable trigger inputs. If you activate the start input (**I2**), the motor will move. Since the ZX is in Motion Profiling mode, it will execute the loop and subsequent commands. As it reaches each trigger statement, it waits for that state to become true and changes the velocity to the velocity following the command. The loop is infinite, so it toggles continuously between 1 - 5 rps as the trigger statements come true and will not stop until the stop input is activated. If you activate **I1** during the operation of the ZX, the indexer immediately stops the operation and clears the command buffer (assuming that **SSH** is set to Ø).

Sequence Select & Sequence Scan

Inputs can be defined as sequence select inputs. This allows you to execute sequences defined via RS-232C by activating the sequence select inputs. *Binary weighting is fixed on the ZX. Input 3 (I3) will always select Sequence #4.*

☞ ***Helpful Hint:***
This figure shows the binary weights of the ZX's inputs when all 7 inputs are configured as sequence select inputs.



Dedicated Binary Weight of ZX Inputs

One possible input configuration and binary weighting is given below.

Input	Function	Binary Weighting
I1	Sequence Select	1
I2	Sequence Select	2
I3	Sequence Select	4
I4	Sequence Select	8
I5	Sequence Select	16
I6	Trigger	—
I7	Stop	—

Input Configuration/Binary Weighting Example

If the inputs are configured as in Table 4-4, sequence #6 will be executed by activating inputs #2 and #3. Sequence #19 will be executed by activating inputs #1, #2, and #5. If input `I1` is set-up as a trigger input, then only even numbered sequences can be used.

To execute sequences, the ZX must be placed in Sequence Scan mode. In this mode, the ZX will continuously scan the input lines and execute the sequence selected by the active sequence select lines. The **SSJ** command is used to enable/disable the sequence scan mode. When **SSJ1** is entered, the Sequence Scan mode is enabled. To disable the mode, enter **SSJØ**.

Once enabled (**SSJ1**), the ZX will run the sequence number that the active sequence select inputs and their respective binary weightings represent. After executing and completing the selected sequence, the ZX will scan the inputs again and run the selected sequences. If a sequence is selected that has not been defined via RS-232C, no sequence will be executed.

If it is not desirable for the ZX to immediately executed another sequence after running the currently selected sequences the Sequence Interrupted Run Mode (**XQ1**) can be enabled. In this mode, after executing a sequence, all sequence select lines must be placed in an inactive state before a new sequence can be selected. The active state of the inputs is determined by the **INL** command.

The Scan (**SN**) command determines how long the sequence select input must be maintained before the indexer executes the program. This delay is referred to as ***debounce time***. The following examples demonstrate how to use the Scan mode.

Step ①

Define a power-up sequence.

☞ ***Helpful Hint:***
The ZX executes these commands on power up—the ZX can read up to 99 sequences from the sequence-select inputs.

Command	Definition
> XE1ØØ	Erases sequence #100
> XD1ØØ	Defines sequence #100
SSJ1	Executes sequences via PLC input
SN2Ø	Sets scan time to 20 msec
XQ1	Sets ZX to interrupted run mode
A1Ø	Sets acceleration to 10 rps^2
AD1Ø	Sets deceleration to 10 rps^2
V2	Sets velocity to 2 rps
IN1B	Sets Input 1 as a sequence-select input
IN2B	Sets Input 2 as a sequence-select input
IN3B	Sets Input 3 as a sequence-select input
IN4B	Sets Input 4 as a sequence-select input
IN5B	Sets Input 5 as a sequence-select input
IN6B	Sets Input 6 as a sequence-select input

IN7B	Sets Input 7 as a sequence-select input
OUT1C	Sets Output 1 as sequence-in-progress output
LD3	Disables the limits (**not necessary if limits are connected**)
> XT	Ends the sequence definition

Step ② Define any sequences that your application may require.

Command	Description
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
D2000	Sets distance to 2,000 steps
G	Executes the move (Go)
> XT	Ends Sequence #1 definition

Command	Description
> XE2	Erases Sequence #2
> XD2	Defines Sequence #2
D4000	Sets distance to 4,000 steps
G	Executes the move (Go)
> XT	Ends Sequence #2 definition

Command	Description
> XE3	Erases Sequence #3
> XD3	Defines Sequence #3
D8000	Sets distance to 8,000 steps
G	Executes the move (Go)
> XT	Ends Sequence #3 definition

Command	Description
> XE99	Erases Sequence #99
> XD99	Defines Sequence #99
D-14000	Sets distance to -14,000 steps
G	Executes the move (Go)
> XT	Ends Sequence #99 definition

Step ③ Verify that your programs were stored properly by uploading each entered sequence (**XU**). If you receive responses that differ from what you programmed, re-enter those sequences.

Step ④ Run each program from the RS-232C interface with the Run Sequence (**XR**) command. Make sure that the motor moves the distance that you specify.

Step ⑤ Apply switches to the inputs that are normally open.

Switch #	Input
Switch #1	I1
Switch #2	I2
Switch #4	I3
Switch #8	I4
Switch #16	I5
Switch #32	I6
Switch #64	I7
GND	

Applying Switches to Inputs

Step ⑥ To execute sequences, cycle power to the ZX. The system will execute sequence #100.

Step ⑦ You can now execute sequences by closing the corresponding switches.

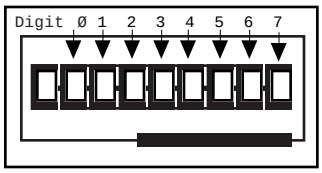
- ❑ Close switch 1 to execute sequence #1
- ❑ Close switch 2 to execute sequence #2
- ❑ Close switches 1 & 2 to execute sequence #3
- ❑ Close switches 1, 2, 6, and 7 to execute sequence #99

Thumbwheel Interface

With the ZX, you can use up to 16 digits of thumbwheels. The ZX uses a multiplexed BCD input scheme to read thumbwheel data. Therefore, a decode circuit must be used for thumbwheels. Compumotor recommends that you purchase Compumotor's TM8 Module if you desire to use a thumbwheel interface. ***The following section assumes that you are using Compumotor's TM8 module with the ZX***. The ZX commands and format that allow for thumbwheel data entry are:

Command	Description
DRDxyz	Read distance via thumbwheels
VRDxyz	Read velocity via thumbwheels
LRDxyz	Read loop count via thumbwheels
TRDxyz	Read time delay via thumbwheels
VARDn,xyz	Read variables via thumbwheels
XRDxy	Read sequence count via thumbwheels
FRDxyz	Read following ratio via thumbwheels

Variables **x** and **y** may range from Ø to 7. They represent the thumbwheel digits to be requested. The left-most digit of the TM8 Module is x=Ø and the right-most is y=7. The other digits are sequentially represented between Ø and 7.



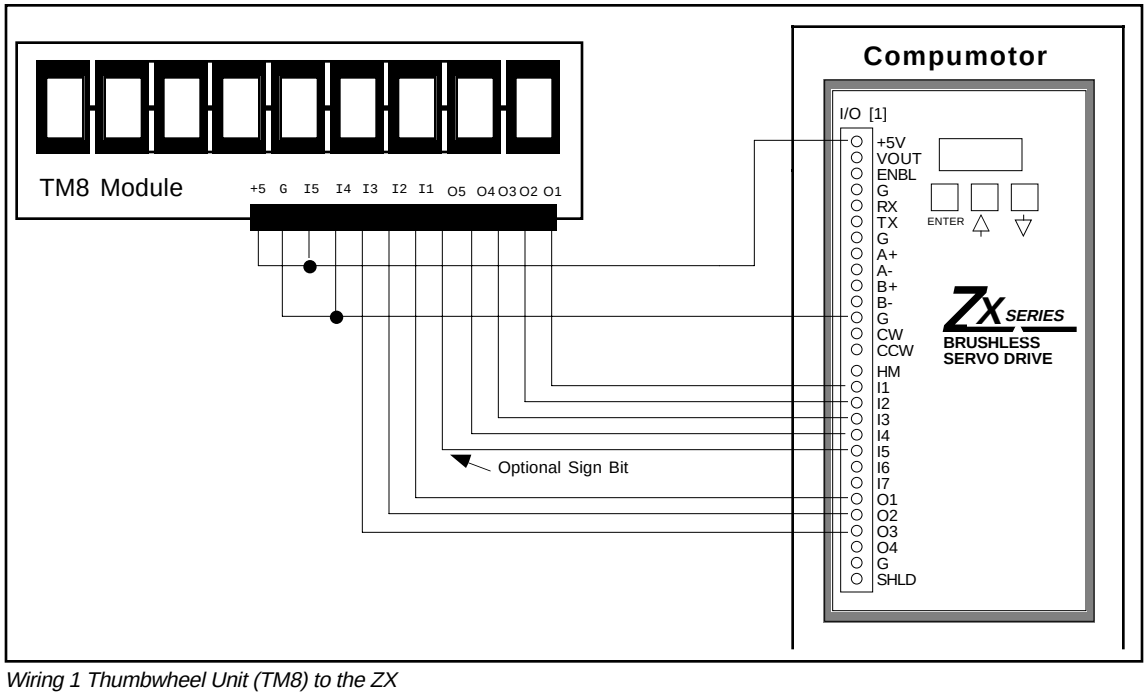
To Request One Digit **x** = **y** = the desired digit number

To Request All Digits **x** = Ø and **y** = 7 (*or do not use the x, y, and z fields*)

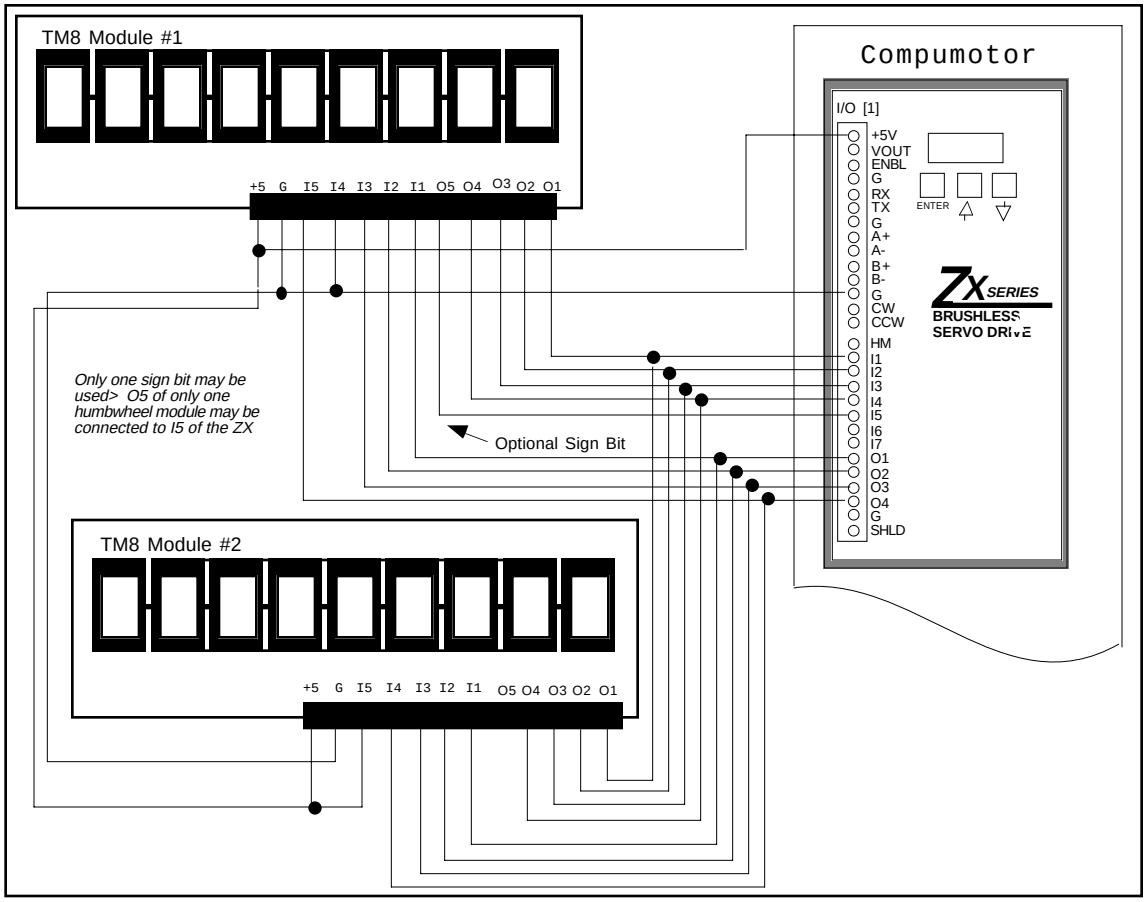
To Request Several Digits Ø < **x** < **y** < 7

The **z** field scales the thumbwheel read value by 10^{z}. When reading digits from the thumbwheel, the least significant digits will be filled first. The z field allows you to position the decimal point in these commands where needed. ***All of the fields (x,y, and z) must be used or none of the fields must be used.*** Refer to the ***ZX Indexer/Drive Software Reference Guide*** for more information. To use Compumotor's TM8 Module, follow the procedures below.

Step ① If you use one TM8 Module, wire your module to the ZX as shown in the figure below. If you use two TM8 Modules (the maximum allowed per **each** ZX).



Wiring 1 Thumbwheel Unit (TM8) to the ZX



Wiring 2 Thumbwheel Units (TM8) to the ZX

Step ②

Configure your ZX as follows:

Command	Description
> OUT1J	**O1** configured as a strobe
> OUT2J	**O2** configured as a strobe
> OUT3J	**O3** configured as a strobe
> IN1N	**I1** configured as a data input
> IN2N	**I2** configured as a data input
> IN3N	**I3** configured as a data input
> IN4N	**I4** configured as a data input
> IN5W	**I5** configured as a sign input (optional)
> INLØ	Inputs configured active low
> STR5Ø	Data strobe time of 50 ms per digit read. If one TM8 Module is used, you should now be ready to read in thumbwheel data. If two TM8 Modules are used, enter the additional set-up commands. *The minimum strobe time recommended for the TM8 module is 10 ms.*
> OUT4A	**O4** configured as a programmable output
> OUTL1	Outputs set active high
> O1	Set output 4 high—this enables TM8 Module #1

Step ③

Set the thumbwheel digits on your TM8 Module to **+12345678**. If two TM8 Modules are used, set the second module to **-87654321**. To verify that you have wired your TM8 Module(s) correctly and configured your ZX I/O properly, enter the following commands:

Command	Description
> DRD	Request distance data from all 8 thumbwheel digits
> 1D	Displays the distance read—**D+12345678.** If the response shown is not received , retry step 1.

If you are using two TM8 Modules, enter the following commands:

> OØ	O4 becomes 0V—this disables Module #1 and enables Module #2.
> DRD	Request distance data from all 8 thumbwheel digits
> 1D	Displays the distance read**—D+87654321** *The sign is positive.* Only one sign digit may be used when two TM8 Modules are used. If the response shown is not received , retry step 1.
> O1	Reenables the first TM8 Module.

Thumbwheel Examples

The following thumbwheel examples are to clarify the TM8 Module usage. If using two thumbwheel modules, enter the following commands:

☞ ***Helpful Hint:***
Note how the 10^2 field can position the decimal point where needed.

Command	Description
> DRD05Ø	Request distance data from the first six thumbwheels
> 1D	**D+123456**
> DRD052	Same as above, but scaled by 10^2
> 1D	**D+12345600**
> VRD67Ø	Request the velocity data from the last two TM8 Module thumbwheels
> 1V	**V.0078**
> VRD673	As above, but now scaled by 10^3
> 1V	Displays the velocity read—**V7.8ØØØ**

☞ ***Helpful Hint:***
The following commands show how you can use the TM8 module to enter variables.

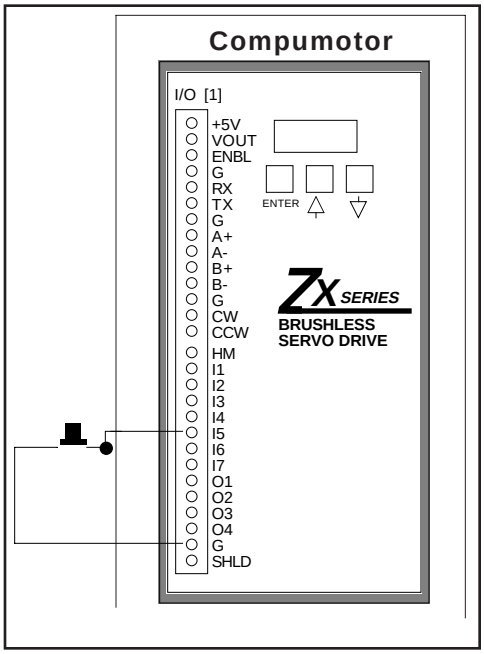
Command	Description
> 1VARD12,45Ø	This requests data from the TM8 Module's 5th and 6th digits
> 1VAR12	VAR12=*0000000000.00056
> 1VARD12,459	As above, but scaled by 10^9
> 1VAR12	VAR12=*0000056000.00000

Selecting Sequences with the Thumbwheel Module

The following example shows how the ZX is typically used with thumbwheels. In the example, only three sequences are entered. As many as 100 sequences may be defined and up to 99 may be executed with the Compumotor Thumbwheel Module. Sequence #100 is automatically executed during power-up or reset. Refer to the Reset (**Z**) command in the ***Z X Indexer/Drive Software Reference Guide***.

Ensure that the thumbwheel module is properly installed. Wire input 5 with a normally open switch as shown in the figure below. The switch shown in this configuration is a sequence-start switch.

☞ ***Helpful Hint:***
Position Mode Tuning Procedure



Sequence Start Configuration

Step ①

Define a power-up sequence. Set inputs **I1** - **I4** as data inputs and **I5** as a data valid input. Enter the following program.

Command	Description
> XE1ØØ	Erases Sequence #100
> XD1ØØ	Begins definition of Sequence #100
IN1N	Sets **I1** as a data input
IN2N	Sets **I2** as a data input
IN3N	Sets **I3** as a data input
IN4N	Sets **I4** as a data input
IN5V	Sets **I5** as a data valid input
INLØ	Sets ØV as active level
STR1Ø	Sets strobe time of 10 ms per digit
OUT1J	Sets O1 as a strobe output
OUT2J	Sets O2 as a strobe output
OUT3J	Sets O3 as a strobe output
L	Start a continuous loop
XRDØ1	Run the sequence displayed on thumbwheel digits Ø & 1
N	Ends loop
> XT	Ends definition of Sequence #100

Step ②

Define any sequences that your application may need.

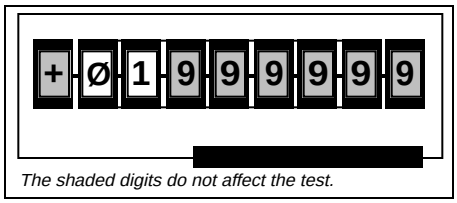
Command	Description
> XE1	Erases Sequence #1
> XD1	Begins definition of Sequence #1
MN	Sets mode to Normal
A25	Sets acceleration to 25 rps^2
AD25	Sets deceleration to 25 rps^2
V5	Sets velocity to 5 rps
D25ØØØ	Sets distance to 25000 steps
G	Executes the move (Go)
> XT	Ends definition of Sequence #1
> XE2	Erases Sequence #2
> XD2	Begins definition of Sequence #2
MN	Sets mode to Normal

	A25	Sets acceleration to 25 rps^2
	AD25	Sets deceleration to 25 rps^2
	V5	Sets velocity to 5 rps
	D10000	Sets distance to 10000 steps
	G	Executes the move (Go)
>	**XT**	Ends definition of Sequence #2
>	**XE99**	Erases Sequence #99
>	**XD99**	Begins definition of Sequence #99
	MN	Sets mode to Normal
	A25	Sets acceleration to 25 rps^2
	AD25	Sets deceleration to 25 rps^2
	V5	Sets velocity to 5 rps
	D-35000	Sets distance to -35000 steps
	G	Executes the move (Go)
>	**XT**	Ends definition of Sequence #99

Step ③ Reset the ZX. This will prepare the power-up sequence (sequence #100) to be executed when you apply power to the ZX.

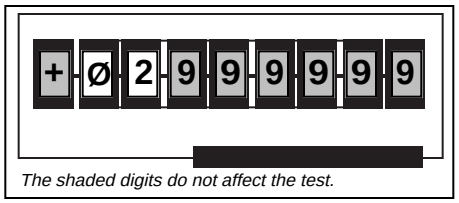
Command	Description
> Z	Resets the ZX

Step ④ Set thumbwheel digits Ø and 1 to **Ø1** and push start to move the servo motor 35,000 steps CW.



The shaded digits do not affect the test.

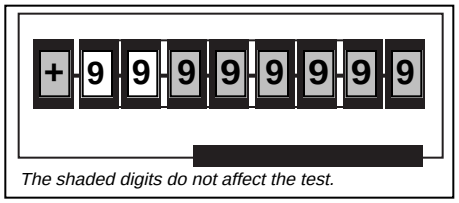
Step ⑤ Set thumbwheel digits Ø and 1 to **Ø2** and push start to move the servo motor 10,000 steps CW.



The shaded digits do not affect the test.

Step ⑥ Set thumbwheel digits Ø and 1 to **99** and push start to move the servo motor 10,000 steps CCW.

☞ ***Helpful Hint:***
If you select an invalid or unprogrammed sequence with the thumbwheels, no motion will occur (nothing will happen).



The shaded digits do not affect the test.

This section explains and provides examples of how to use a PLC with the ZX Indexer/Drive.

Interfacing with a PLC

In many applications it is desirable to interface to a PLC. The ZX performs the motion segment of a more involved process controlled by a PLC. In these applications, the PLC will execute sequences, load data, manipulate inputs and perform other specific input functions to control ZX and the motion segment of a process.

Data Read with a PLC

As in the thumbwheel case, the PLC can be used to enter data for sequence select (**XRD**), distance (**DRD**), velocity (**VRD**), loop count (**LRD**), time delay (**TRD**), and variable data (**VARD**). Detailed explanations of these commands are given here.

To read data from the PLC, the ZX's inputs #1 - #4 must be configured as data inputs with an active low level (**INLØ**).

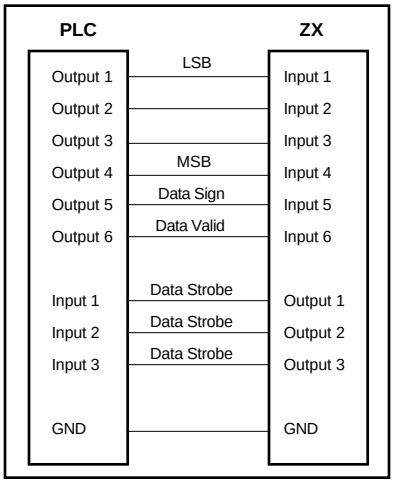
If a sign digit is required, ZX input 5 should be configured as a sign input. Configure outputs #1 - #3 as data strobe outputs.

When the ZX executes a data read command, it cycles its outputs and reads BCD data. The maximum rate that the ZX will cycle through the output levels is set by the Strobe Output Delay Time (**STR**) command. The ZX synchronously cycles through the states at the **STR** time if no data valid line is used. If a data valid line is used, the ZX will maintain its current state until the data valid input is activated. This allows the PLC to control the ZX's data strobe rate.

ZX Outputs			ZX Inputs					
O1	**O2**	**O3**	**Data In (Active Low)**	**I1**	**I2**	**I3**	**I4**	**I5**
low	low	low	MSD (Digit 1)	lsb			msb	± sign bit
high	low	low	Digit 2	"			"	"
low	high	low	Digit 3	"			"	"
high	high	low	Digit 4	"			"	"
low	low	high	Digit 5	"			"	"
high	low	high	Digit 6	"			"	"
low	high	high	Digit 7	"			"	"
high	high	high	LSD (Digit 8)	"			"	"

Output Voltage Levels

☞ ***Helpful Hint:***
This figure shows a possible PLC-to-ZX connection that would allow the ZX to input data from the PLC. *In this configuration, a data valid line (16) controls the ZX's strobe rate and a data sign line is used.*



PLC/ZX Connection

If the ZX runs a Read Distance Via Parallel I/O (**DRD**) command, the following events must occur to transfer distance data from the PLC to the ZX.

☞ ***Helpful Hint:***
This process continues until the ZX reads the eighth digit (LSD). At this point, the ZX enters the eight digits read into its distance register and proceeds with the execution of subsequent commands.

① The ZX executes a **DRD** command and places its outputs #1 - #3 at ØV.

② The PLC places and holds a BCD digit at the ZX's inputs #1 - #4. (This value will be the most significant distance digit). The PLC places a sign value at ZX input #5. (***This sign bit must be the same for each digit read***.)

③ When the data is valid, the PLC should activate the data valid line. The ZX will read the digit and sign values.

④ The PLC should deactivate the data valid line.

⑤ After reading a data valid, the ZX will place its output #1 at high voltage (5 - 24V) and outputs

#2 and #3 at ØV.

⑥ The PLC must place and hold its second BCD digit at the ZX's inputs #1 - #4. The PLC should place the same sign value as that given for input #5.

⑦ When the data is valid, the PLC must activate the data valid line. The ZX will read this second BCD digit as the second significant distance digit.

⑧ The PLC must deactivate the data valid line.

☞ ***Helpful Hint:***
The **XRD** command does not use the z scaling field.

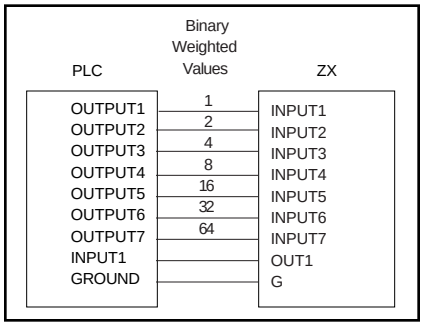
If data scaling is desired, refer to the *Data* section earlier in this chapter. The additional x, y, and z fields may be used. The x field must equal Ø, the y field must equal 7, and z must be greater than or equal to Ø and less than 10. For example, a **DRDØ71** command will cycle the outputs through the sequence and read the data one digit at a time for eight digits. The data however will be multiplied by 10^1 when it is stored in the data's register.

Sequence Select With a PLC

The PLC can execute sequences through two different methods. First, the sequences may be selected by using the inputs as sequence select inputs. In this case, the input lines are binary weighted. The ZX must be in Sequence Scan mode (**SSJ1**) and may either operate in Interrupted mode (**XQ1**) or Continuous Scan mode (**XQØ**).

The PLC activates the lines that will execute the desired sequence.

☞ ***Helpful Hint:***
It may be desirable to have the ZX indicate to the PLC when it has completed a sequence or to indicate to the PLC when it should select another sequence. A programmable output can be used for this handshake to the PLC.



PLC Connection

The PLC may also use the **XRD** command to select and run ZX sequences. Only ZX output #1 should be configured as a strobe output. Inputs #1 - #4 must be configured as data inputs (**INLØ**). A data valid line may also be used.

If the ZX executes the **XRD** command when its I/O is configured as described above, it will set its output #1 to a voltage low for the time defined by the **STR** command or until the data valid line is active. It will then read in the most significant BCD digit of a sequence run command. The ZX will then set the voltage level high for output #7 (again, for the appropriate period) and read in the least significant digit of sequence run data. The ZX will then execute the sequence defined by the PLC if it exits.

ZX Output	ZX Inputs			
Ø1	**I1**	**I2**	**I3**	**I4**
Low	LSB	MS BCD Digit		MSB
High	LSB	LS BCD Digit		MSB

Strobe and Input Sequence

Miscellaneous PLC Control

You can use a PLC to control the activation of the inputs for many input functions that the ZX supports (see the ***Programmable Input*** section). For example, you can use the PLC to stop, kill, go, go home, or reset the ZX.

Front Panel Operation

You can use the ZX's front panel pushbuttons and display to execute sequences. Refer to the ***Alphanumeric Display and Pushbuttons*** section for information on how to use this feature.

Remote Panel Operation

There is an optional remote panel that can be used by the ZX and ZX-F to input data, display messages and prompt the operator for appropriate actions. The RP240 is a panel that communicates over RS-232C with the ZX. It has a 2 line 40 character display, a numeric keypad, and programmable function keys. Contact your distributor or Compumotor for more information on the RP240.

Host Computer Operation

You may use a host computer to execute motion programs via the RS-232C serial interface. The host computer can run interactively from a basic or C program. In this case, the high-level program controls the ZX and acts as an interface to the user (note the following program example).

```
1 '                    ZX.BAS PROGRAM
2 '
3 '  ********************************************************************************
5 ' * This program controls the RS232 Communication line to execute 2 different moves     *
6 ' * using the ZX                                                                       *
********************************************************************************
15 OPEN "COM1:9600,N,8,1,RS,CS,DS,CD" AS #1        ' Open Communication port
20 V$ = "": Q$ = "": ECHO$ = "": LF$ = "":         ' Initialize variables
90 CLS
100 LOCATE 12,15
105 PRINT " PRESS ANY KEY TO START THE PROGRAM "
107 V$ = INKEY$: IF LEN(V$) = 0 THEN 100           ' Wait for input from user
120 Z$ = "K K LD3 "                                ' Reset and disable limits on the ZX
122 PRINT #1,Z$;
124 Q$ = INPUT$(8,1)
900 '  ********************************************************************************
902 ' *  Lines 1000-1060 sends a move down to the first ZX.  Computer waits for the Line *
903 ' *  Feed from the ZX indicating that the motor has finished its move.  Computer      *
904 ' *  will not command second ZX to move until the first move is completed.           *
********************************************************************************
1000 MOVE1$ = "MN A1 V2 D40000 G 1LF "             ' Define move 1
1005 CLS
1007 LOCATE 12,15: PRINT " DOING MOVE 1 "
1010 PRINT #1,MOVE1$                               ' Perform move 1.
1015 ECHO$ = INPUT$(22,1)                          ' Read echoes from ZX.
1020 LF$ = INPUT$(1,1)                             ' Wait for line feed from ZX
1040 IF LF$ <> CHR$(10) GOTO 1020                  ' indicating end of move.
1045 CLS
1047 LOCATE 12,15
1050 PRINT "MOVE 1 DONE"                           ' Let user know that move 1 is done
1060 LOCATE 15,15: PRINT " PRESS ANY KEY TO GO ON TO SECOND MOVE "
1070 V$ = INKEY$: IF LEN(V$) = 0 THEN 1060
1900 '  ********************************************************************************
1902 ' * After axis one is done, we request that you hit any key to go on to second move.*
1903 ' * In real application, we would expect you to go ahead with the process and work on*
1904 ' * on the part before going on to next move.  (i.e., Activate a punch)              *
1905 ' * Now that first move is finished go on to move 2. The ZX also prints a line feed *
1906 ' * after finishing the second move.                                                *
1907 ' * As soon as the computer receives the line feed from ZX, the program will go back*
1908 ' * to the first move.                                                             *
********************************************************************************
2000 MOVE2$ = "H G 1LF "
2005 CLS
2007 LOCATE 12,15: PRINT " DOING MOVE 2 "
2010 PRINT #1,MOVE2$
2015 ECHO$ = INPUT$(8,1)
2020 LF$ = INPUT$(1,1)
2040 IF LF$ <> CHR$(10) GOTO 2020
2045 CLS
2047 LOCATE 12,15
2050 PRINT "MOVE 2 DONE "
2060 FOR I = 1 TO 1000: NEXT I
2070 GOTO 20                                  ' Go back to beginning of program.
```

The second method is for the ZX to run a stored program and prompt the user interactively as a part of the program. To do this, you must use the RS-232C Input (**RSIN**) and Quote (**"**) commands. The Quote command sends messages over the RS-232C link to the host. The message may be status data or a request for information. If it is a request, **RSIN** allows you to enter data into any of the general-purpose variables. The motion program may use these variables. The following program is an example of a stored sequence that interactively requests the number of parts and the size.

```
> 1XE1                          Erase sequence #1
> 1XD1                          Define sequence #1
> 1"Enter_the_number_of_
  parts_to_be_made              Send string to host
> VAR1=RSIN                     Set variable #1 to the value input by the user
> 1"Enter_the_size_
  of_the_parts                  Send string to host
> VAR2=RSIN                     Set variable #2 to the value input by the user
> VAR2=VAR2*25000               Scale variable #2 by 25000 for user units
> D(VAR2)                       Set distance (part size) to variable #2
> L(VAR1)                       Set loop count (part count) to variable #1
> G                             Execute move
> N                             Continue loop
> XT                            Terminate sequence
```

ZX Application Examples

This section provides application examples which illustrate the features and capabilities discussed in this chapter.

Application Development Approach

This section will help you to develop your application program. These steps provide a guideline for identifying and organizing your program's needs. Examples of actual programs demonstrate how the different programming concepts are used together. Use the following steps:

① Identify and define the types of motion required in your application.

Examples

- ❏ Moves to a home switch; go home moves
- ❏ Jogging the motor
- ❏ Preset moves of triangular or trapezoidal velocity profiles
- ❏ Registration moves
- ❏ Moves with changing velocities not based on distance
- ❏ Moves of changing velocities based on distance
- ❏ Motion based on math calculations
- ❏ Motion with outputs turning on at distance points

Once you have identified the motion, determine the gains that will be required for the ZX to perform the moves with the proper performance. To do this, enter the speed , acceleration, and decelerations required for the move and determine the gains necessary for the required performance. To tune the drive, refer to the *Servo Tuning* section in this chapter. After determining the gains, save them with the **SV** command.

② Determine what will cause the motion to occur and when it will cause motion to occur. Choose the interface(s) for execution of the motion programs.

Examples

- ❏ Input states causing motion to occur; triggers
- ❏ Sequence execution
- ❏ RP240 Remote Panel
- ❏ Thumbwheel inputs
- ❏ Power up sequence
- ❏ PLC
- ❏ Variable conditions
- ❏ Time delays

③ Determine what configuration commands are necessary to set the ZX in the proper state for execution of the motion programs.

Examples

- ❏ Sequence scan mode—**SSJ1**, **XQ1**, **SSH1**
- ❏ Input/Output configuration—**IN1A**, **IN4D**, **OUT3J**
- ❏ Motion parameters—**A**, **AD**, **V**, **D**
- ❏ Profile commands—**MC**, **MN**, **MPI**, **MPA**, **MPP**
- ❏ Homing parameters—**GHA**, **GHV**, **GHF**, **GHAD**, **OSB**, **OSG**, **OSH**
- ❏ Other special functions of the **SS** and **FS** commands

Once you determine what set-up commands are required, place them in the power-up sequence, sequence #100. In some applications, the ZX may be required to be in different modes for different parts of the application. For example, some moves may be *Continuous mode* moves and some may be *Preset mode* moves. In this case, you can place the appropriate set-up commands in the sequence in which that particular motion is required. After entering the power-up sequence, cycle power, issue a Reset (**Z**) command or simply run sequence #100 with an **XR10Ø** command to configure the ZX according to the set up commands in the

power-up sequence.

The manner in which the remaining sequences are programmed will be determined largely by the type of interface you choose. There are three typical types of applications that require different methods of executing motion through sequences.

Method ①

The interface may be discrete inputs which select or control sequences. In this case a program of several sequences will have the program flow controlled by the inputs or there may be a small number of sequences selected by discrete inputs as sequence select lines.

Method ②

The interface (thumbwheels or a PLC and sequences) is selected via the inputs. The different motion requirements are usually in each sequence.

Method ③

The third typical application has data or move parameter information entered via the inputs. In this case, there are usually fewer sequences but they require some information from the data entered via the ZX inputs.

These types of applications are not all-inclusive, but they do offer ideas in how you might set up your interface to execute the required motion.

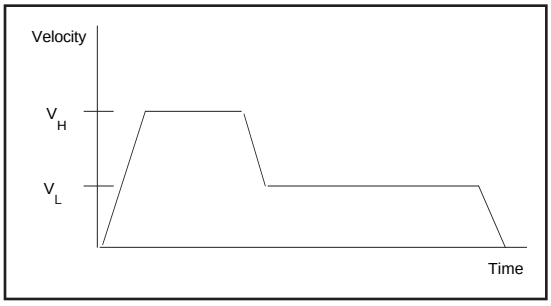
Application Example #1

In this application, parts are formed by a grinder. The part that is being ground moves at a high speed toward the grinder. Just before it hits the grinder, it slows to a lower speed and the grinding operation begins. Because there are different parts of different lengths, different distances are required for the grinding part of the operation. The distance from the point where the part is loaded to the point where the grinding of the part is finished is a fixed distance. What varies is the point at which the grinding begins. Only five different parts are made. The distance from the point of loading the part to the end of the grind is 24 inches. The motor is on a leadscrew with a pitch of 2. The operator interface must be very simple and include part-selection and run/stop switches.

Types of Motion Required

In this application, a preset move must be made. The first part of the distance must move at a high speed and the second part at a low speed. The point at which the move changes from high-speed to low-speed varies with each part type. A homing move positions the machine at the location where parts are loaded. From this location, other grind moves can be made.

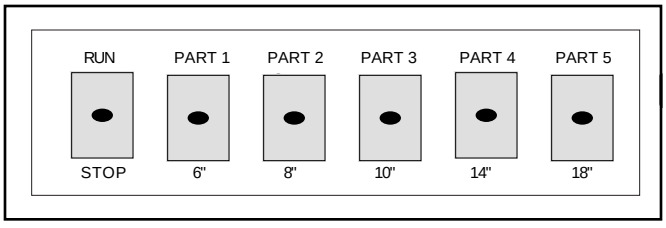
☞ ***Helpful Hint:***
Motion profile of the grinding application



Grinding Application Motion Profile

What Will Cause Motion and When?

It is necessary to have the machine move to its part loading location when it starts so that it is ready to load parts and make grind moves. A home switch will be placed at the loading point to allow the ZX to position itself at the home location. A toggle switch allows the user to select either the Run mode or Stop mode. When the switch is in Run mode, it will continuously make parts of the type selected until the run/stop switch is placed in the Stop mode. The parts are selected by turning on the switch labeled for the particular part. The interface will be 6 switches. One switch will toggle between the run or stop mode. The user can select the part with the other 5 switches. The part-selection switches are on/off switches as opposed to momentary contact switches.



Front Panel of User Interface

What

❑ **SSJ1** Sequence Scan mode

Configuration Commands Are Required?

- ❑ **INnA** Trigger input, n is the input number
- ❑ **INnB** Sequence select input
- ❑ **OSB** Back up to home switch
- ❑ **OSG** Select final home approach direction
- ❑ **OSH** Select edge of home switch which ZX stops on
- ❑ **GHV** Select homing velocity
- ❑ **GHF** Final homing approach velocity

Execution

Part selection switches select sequences that contain the move instructions for specific parts. Motion will not begin until the run/stop switch is in the run position. Parts will continue to be made until the run/stop switch is set to stop. The user can select Stop mode at any time during the grind move. The grind move that was executed at the time the Stop mode was selected will be completed. While in Stop mode, a new part type can be selected. The machine will operate this way as long as it is powered up. Upon power up, the machine will move to the home location to begin operation.

Required Sequences

The resolution set by the Configure Motor Resolution command is 5000 steps/rev (**CMR5000**). This value is stored in battery backed RAM and need not be in the program. With a pitch of 2, the total move distance is 240000 steps. The part sizes are 6, 8, 10, 14, and 18 inches. This makes grind distances of 60000, 80000, 100000, 140000, and 180000 steps.

Sequence #100

Command	Description
XD100	Begins definition of sequence #100
MPI	Incremental move mode
MN	Normal preset mode
GHV2	Initial go home velocity if 2 rps
GHF.3	Final go home velocity of 0.3 rps
OSB1	Back up to home switch
OSH1	Stop on the CCW edge of the switch
OSG0	Final approach direction is CW
GH	Begin go home move
IN1B	Sets up inputs 1-5 as sequence select inputs
IN2B	
IN3B	
IN4B	
IN5B	
IN6A	Sets up input 6 as a trigger
SSJ1	Place the ZX in continuous scan mode
A100	Set the acceleration to 100 rps^2
TR1	Wait until run switch is enabled
XT	Ends definition of sequence #100

Sequence #1

Command	Description
XD1	Begins definition of sequence #1
V4	Defines the fast velocity
D240000	Total move distance
REPEAT	Repeat loop to make parts continuously
MPP	Enters Profiling mode so velocity can be changed on the fly
G	Initiates motion
DP180000	After 180,000 steps decelerate to the grind velocity
V.5	Grind velocity
NG	Ends Profiling mode. No commands will execute until the move is finished
V4	Return for a new part at a high speed
H	Change direction
G	Return to the get a new part
UNTIL(INXXXXXXXX0)	Continue the repeat loop until the run/stop is put in the stop location
TR1	Wait until the run/stop switch is in the run location then finish this sequence and scan the inputs to select another part sequence
XT	Ends definition of sequence #1

Sequence #2

XD2	Begin definition of sequence #2
V4	Defines the fast velocity
D240000	Total move distance
REPEAT	Repeat loop to make parts continuously
MPP	Enters Profiling mode—velocity can be changed on the fly
G	Initiates motion
DP160000	After 160,000 steps decelerate to the grind velocity
V.5	Grind velocity
NG	Ends Profiling mode—no commands will execute until the move is finished
V4	Return for a new part at a high speed
H	Change direction
G	Return to the get a new part
UNTIL(INXXXXXXXX0)	Continue the repeat loop until the run/stop switch is put in the stop location
TR1	Wait until the run/stop switch is in the run location then finish this sequence and scan the inputs to select another part sequence
XT	Ends Sequence definition

Sequence #4

XD4	Begin definition of sequence #4
V4	Defines the fast velocity
D240000	Total move distance
REPEAT	Repeat loop to make parts continuously
MPP	Enters Profiling mode—velocity can be changed on the fly

	G	Initiates motion
	DP140000	After 140,000 steps decelerate to the grind velocity
	V.5	Grind velocity
	NG	Ends Profiling mode—no commands will execute until the move is finished
	V4	Return for a new part at a high speed
	H	Change direction
	G	Return to the get a new part
	UNTIL(INXXXXXXXX0)	Continue the repeat loop until the run/stop switch is put in the stop location
	TR1	Wait until the run/stop switch is in the run location then finish this sequence and scan the inputs to select another part sequence
	XT	Ends Sequence definition
Sequence #8	**XD16**	Begin definition of sequence #16
	V4	Defines the fast velocity
	D240000	Total move distance
	REPEAT	Repeat loop to make parts continuously
	MPP	Enters Profiling mode—velocity can be changed on the fly
	G	Initiates motion
	DP100000	After 100,000 steps decelerate to the grind velocity
	V.5	Grind velocity
	NG	Ends Profiling mode—no commands will execute until the move is finished
	V4	Return for a new part at a high speed
	H	Change direction
	G	Return to the get a new part
	UNTIL(INXXXXXXXX0)	Continue the repeat loop until the run/stop switch is put in the stop location
	TR1	Wait until the run/stop switch is in the run location then finish this sequence and scan the inputs to select another part sequence
	XT	Ends Sequence definition
Sequence #16	**XD2**	Begin definition of sequence #2
	V4	Defines the fast velocity
	D240000	Total move distance
	REPEAT	Repeat loop to make parts continuously
	MPP	Enters the profiling mode so velocity can be changed on the fly
	G	Initiates motion
	DP80000	After 80,000 steps decelerate to the grind velocity
	V.5	Grind velocity
	NG	Ends Profiling mode—no commands will execute until the move is finished
	V4	Return for a new part at a high speed
	H	Change direction
	G	Return to the get a new part
	UNTIL(INXXXXXXXX0)	Continue the repeat loop until the run/stop switch is put in the stop location
	TR1	Wait until the run/stop switch is in the run location then finish this sequence and scan the inputs to select another part sequence
	XT	Ends Sequence definition

Since the sequence select inputs are binary weighted, one switch can select each sequence above. That is why sequences 1, 2, 4, 8, and 16 were chosen. This method is only useful if there are a few sequences (7 or less).

Application Example #2

If example #1 is modified slightly, the same motion can be accomplished using a different user interface. In example #2, we will have the same motion requirements, but 30 different parts will be made. The same run/stop criteria from example #1 are in effect; however, thumbwheels will be used to enter the grind distance data. The data will be entered as the length of the part. The distance of the grind move is the length of the part. After the distance is entered on the thumbwheels, the grind move will run until the stop switch is enabled. A homing move on power up is still required to place the machine in the parts loading position.

Types of Motion Required

The required motion will be the same as example #1.

What Will Cause Motion and When?

Like example #1, a run/stop switch will be used to start and finish the grind moves. A homing move will be required to start the process. This will be done the same way as example 1. The point at which the velocity must slow to a lower velocity will be determined by thumbwheels and some math calculations. The user need only enter the length of the part in inches on the thumbwheels and then select the run mode. The ZX will then perform the required grind move.

What Configuration Commands Are Required?

- ❑ **VARDn** Reads the input data into variable n
- ❑ **INnA** Trigger input, n is the input number
- ❑ **INnN** Data input
- ❑ **OUTnJ** Sets the outputs as strobe outputs
- ❑ **STR** Strobe time for reading the data inputs
- ❑ **OSB** Back up to home switch
- ❑ **OSG** Select final home approach direction
- ❑ **OSH** Select edge of home switch which ZX stops on
- ❑ **GHV** Select homing velocity
- ❑ **GHF** Final homing approach velocity

Execution

The process runs identically to the previous process, with the exception of selecting the grind move. In this case, the grind move is determined by reading data in via the thumbwheels then beginning the move when the run mode is enabled. When Stop mode is enabled, new grind move data can be entered on the thumbwheels. One bank of 8 thumbwheel switches is used so the enable inputs on the TM8 module are set to the enable state rather than to a ZX output.

Required Sequences

The resolution set by the Configure Motor Resolution command is 5000 steps/rev (**CMR5000**).

Sequence #100

Command	Description
XD100	Begins definition of sequence #100
MPI	Incremental move mode
MN	Normal preset mode
GHV2	Initial go home velocity if 2 rps
GHF.3	Final go home velocity of 0.3 rps
OSB1	Back up to home switch
OSH1	Stop on the CCW edge of the switch
OSG0	Final approach direction is CW
GH	Begin go home move
IN1N	Sets up inputs 1 - 4 as data inputs
IN2N	
IN3N	
IN4N	
IN5A	Sets up input 5 as a trigger input
OUT1J	Sets Outputs 1-3 as strobe outputs
OUT2J	
OUT3J	
STR10	Strobe time is 10 milliseconds
A100	Sets acceleration to 100 rps^2
XR1	Runs sequence #1
XT	Ends definition of sequence #100

Sequence #1

Command	Description
XD1	Begins definition of sequence #1
L	Starts an infinite loop
V4	Defines the fast velocity
D240000	Total move distance
VARD1	Reads the thumbwheels for variable 1
VAR1=VAR1*10000	Convert part length inches to steps
VAR1=240000-VAR1	Determine distance point DP where velocity changes to a lower speed
REPEAT	REPEAT loop to repeatedly make parts
MPP	Enters the profiling mode so velocity can be changed on the fly
G	Initiates motion
DP(VAR1)	Loads the distance point determined in variable 1
V.5	Grind velocity
NG	Ends the profiling mode. No commands will execute until the move is finished
V4	Return for a new part at a high speed
H	Change direction
G	Return to the get a new part
UNTIL(INXXXXXX0)	Continue the repeat loop until the run/stop switch is put in the stop location
TR1	Wait until the run/stop switch is set to run then finish this sequence and scan the inputs to select another part sequence
N	Ends the infinite loop
XT	Ends definition of sequence #1

This is the only sequence required. It is an infinite loop where parts are continuously run until the Stop mode is enabled. When this occurs, the program pauses at the trigger command until Run mode is selected. The thumbwheels will be read again so a new grind move can be run. This is more flexible and allows many different grind moves to be executed.

Application Example #3

This application is a process of which the ZX is controlling one axis. A PLC controls the process. In this process the ZX is feeding material at two different speeds. It will either be feeding the material at a high speed or a low speed. The PLC will signal the ZX when to stop the material. The material must be able to switch between the high speed and low speed without stopping. When the PLC stops the material feed, the material is cut and the ZX must return to the point it started to begin the process again. The point at which the material feed begins is the home position for starting motion. On power up, the ZX should move to this position. The PLC controls motion. Three different types of material will require different high and low velocities.

Type of Motion Required

This application will require continuous moves. It will also require a preset move to return to the starting location. On power up a homing move will be made. The PLC controls continuous moves. The moves will always start with the high velocity and will then be toggled between high and low by the PLC. The PLC will stop the operation when it is complete. Different move velocities will be required for the different material. The distance of the preset return move will be determined by the ZX based on the distance it traveled during

the continuous move.

What Will Cause Motion and When?

The PLC will control the entire process. The ZX will home itself after the initial power up. The continuous move will then begin by the PLC selecting one of three materials. Each material will have its own high and low velocities. The continuous move will continue until the PLC issues a stop. Then the ZX will make a preset move back to the start and the PLC will select one of the sequences again.

What Configuration Commands Are Required?

- ❑ **SSJ1** Sequence Scan mode
- ❑ **XQ1** Interrupted Sequence Scan mode
- ❑ **INnA** Trigger input, n is the input number
- ❑ **INnB** Sequence select input
- ❑ **OUTnC** Sequence in progress output
- ❑ **OSB** Back up to home switch
- ❑ **OSG** Select final home approach direction
- ❑ **OSH** Select edge of home switch which ZX stops on
- ❑ **GHV** Select homing velocity
- ❑ **GHF** Final homing approach velocity
- ❑ **MC** Sets the ZX in the Continuous mode
- ❑ **MN** Sets the ZX in the Preset mode

Execution

The ZX will be in Sequence Scan mode. The PLC selects one of the three sequences. High speed or low speed is controlled with one line. One input will also be configured as a stop input. It will set the velocity level input to **1** for high speed and **Ø** for low speed. When a stop is issued, the PLC will wait for the ZX to signal that it is back to the start position and then select the next sequence with the sequence select inputs. An output is set up to turn on when a sequence is in progress. When it is off, the PLC can select a new sequence. The new sequence will not run until the ZX has made the preset move back to the start location. Returning to the start location ends the sequence and allows the next sequence to be scanned.

Sequence #100

Command	Description
XD1ØØ	Begins definition of sequence #100
MPI	Incremental move mode
MC	Sets ZX to Continuous mode
GHV2	Initial go home velocity if 2 rps
GHF.3	Final go home velocity of 0.3 rps
OSB1	Back up to home switch
OSH1	Stop on the CCW edge of the switch
OSGØ	Final approach direction is CW
GH	Begin go home move
IN1B	Sets up inputs 1 and 2 as sequence select inputs
IN2B	
IN3A	High/Low velocity input (trigger input)
IN4A	Stop input
OUT1C	Sets the output of a sequence-in-progress output
XQ1	Enables the interrupted run mode
SSJ1	Place the ZX in Continuous Scan mode
A1ØØ	Set the acceleration to 100 rps^2
XT	Ends definition of sequence #100

Sequence #1

Command	Description
XD1	Begins definition of sequence #1
PZ	Zeroes (clears) the position counter
MC	Sets the ZX in continuous mode
V18.5	Sets the high velocity to 18.5 rps
H+	Sets the direction to CW
MPP	Enables MPP mode
G	Begins motion
REPEAT	Loops until stopped
IF(INXXXXX1)	Checks for high or low velocity
V3	Sets the velocity to the low velocity
ELSE	If not low velocity then
V18.5	Sets the velocity high
NIF	End of IF statement
UNTIL(INXXXXXXZX1)	Checks for the stop signal
NG	Ends the Profiling mode
STOP	Stops the move
VAR1=POS	Sets variable one equal to the position counter
D(VAR1)	Loads the distance with the distance traveled in the Preset move
H	Sets the direction CCW
MN	Sets the ZX in the preset mode
G	Returns to the starting point
XT	Ends definition of sequence #1

Sequence #2

XD1	Begins definition of sequence #1
PZ	Zeroes (clears) the position counter
MC	Sets the ZX in continuous mode
V12	Sets the high velocity to 12 rps
H+	Sets the direction to CW
MPP	Enables MPP mode
G	Begins motion
REPEAT	Loops until stopped
IF(INXXXXX1)	Checks for high or low velocity
V1	Sets the velocity to the low velocity
ELSE	If not low velocity then
V12	Sets the velocity high
NIF	End of IF statement
UNTIL(INXXXXXX1)	Checks for the stop signal
NG	Ends the Profiling mode
STOP	Stops the move
VAR1=POS	Sets variable one equal to the position counter
D(VAR1)	Loads the distance with the distance traveled in the Preset move

	H	Sets the direction CCW
	MN	Sets the ZX in the preset mode
	G	Returns to the starting point
XT		Ends definition of sequence #1

Sequence #3

XD1		Begins definition of sequence #1
	PZ	Zeroes (clears) the position counter
	MC	Sets the ZX in continuous mode
	V9	Sets the high velocity to 9 rps
	H+	Sets the direction to CW
	MPP	Enables MPP mode
	G	Begins motion
	REPEAT	Loops until stopped
	IF(INXXXXXX1)	Checks for high or low velocity
	V.5	Sets the velocity to the low velocity
	ELSE	If not low velocity then
	V9	Sets the velocity high
	NIF	End of IF statement
	UNTIL(INXXXXXX1)	Checks for the stop signal
	NG	Ends the Profiling mode
	STOP	Stops the move
	VAR1=POS	Sets variable one equal to the position counter
	D(VAR1)	Loads the distance with the distance traveled in the Preset move
	H	Sets the direction CCW
	MN	Sets the ZX in the preset mode
	G	Returns to the starting point
XT		Ends definition of sequence #1

C H A P T E R ⑤

ZXF Follower

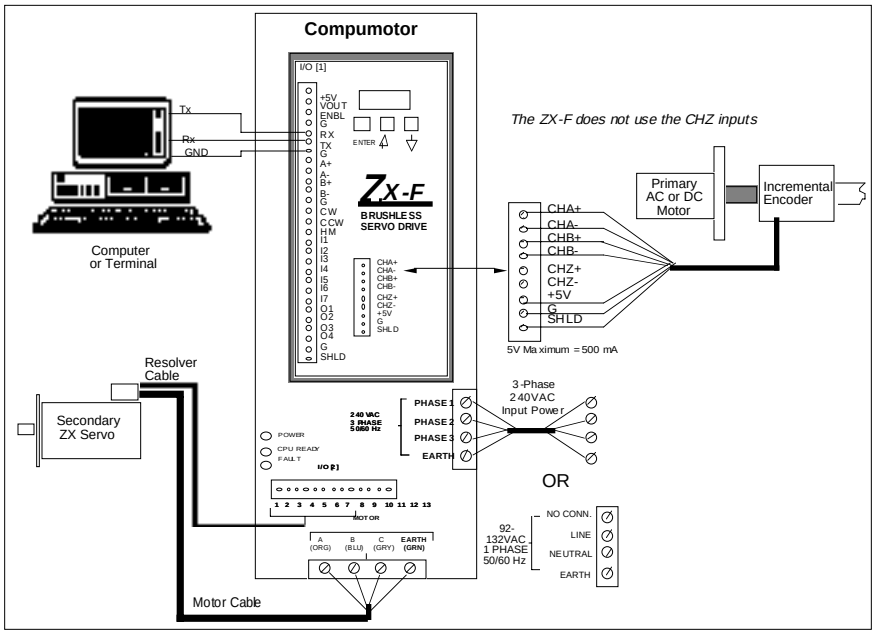
The information in this chapter will enable you to:

- ❏ Understand basic following concepts
- ❏ Understand the basic types of following and their common applications
- ❏ Become familiar with the ZXF commands associated with following

What is Following?

The ZXF can perform velocity following and distance following moves. The ZXF can follow from an incremental encoder input. Unless otherwise noted, all the features that will be presented or have been discussed are also valid in Following mode. The only difference is that you replace the velocity command with a speed ratio and the acceleration with a following acceleration for distance following. Instead of specifying the speed with the **V** command, you will specify the speed with respect to the primary axis using the **FOL** command.

☞ ***Helpful Hint:***
The primary axis is an AC or DC motor. The encoder is mounted on the shaft of the primary motor. The encoder provides the ZXF with the primary motor's position and velocity. The ZXF uses the position and velocity data to move the secondary axis (the secondary motor is following the motion of the primary motor). The ZXF moves the secondary motor based on the primary motor's moves. This concept of following can be used in different forms to satisfy different applications.



Typical Following System Configuration

Types of Following

There are four types of following application motion.

- ❏ Velocity following
- ❏ Velocity and position following
- ❏ Recede and advance while following (includes electronic cam)
- ❏ Phase error correction (synchronization)

The categories are presented roughly in order of increasing complexity. By identifying how the secondary axis motion relates to the primary axis, you will be able to determine your application's type and the applicable programming commands. Several features have been added to address more specific attributes of the various applications.

The ZXF has several features that enhance an application's functionality.

- ❏ Registration while following
- ❏ Jogging in the following mode
- ❏ Following a pulse train and a direction
- ❏ Following encoder pulses and the encoder direction
- ❏ Entering following percentage (**FOL**) via thumbwheels
- ❏ Entering following percentage (**FOL**) via front panel pushbuttons
- ❏ Capability of loading the primary encoder position into a variable

The ZXF can perform all of the ZX's programming functions. The only difference is that the motion profiles are now following moves and motion is based on a primary axis. The ZXF can also be used in Indexer mode. You can easily switch back and forth from follower and indexer functionality and create motion programs in which the ZXF serves as both a follower and an indexer.

☞ ***Helpful Hint:***
The ZXF has the following programming capabilities:

- ❏ Variables, general-purpose & read-only (position [POS], primary encoder position [FEP], etc.)
- ❏ Math
- ❏ Complex branching—**IF ELSE**, **REPEAT UNTIL**, **WHILE**
- ❏ On-the-fly changes—MPP mode
- ❏ Flexible I/O
- ❏ Closed-Loop mode while following

For an explanation of the features that are common to both the ZX and the ZXF, refer to *Chapter ④ Application Design*. You should understand how to set the ZXF for following. You can use one command (**FSI**) to enter and exit the Indexer and Following modes.

Command	Description
FSI1	Enters the following mode
FSIØ	Exits the following mode(indexer mode)

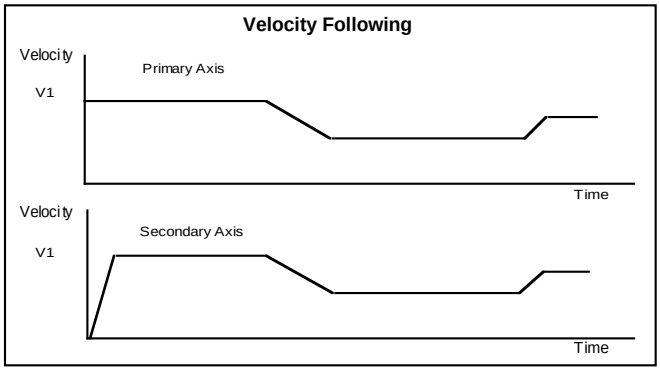
The rest of this chapter explains the four following types. Additional features of ZXF following will also be covered.

Velocity Following

In velocity following, the secondary axis uses the primary axis' speed. The relationship of the primary axis position with respect to the secondary axis is irrelevant. In these applications, the secondary axis accelerates at a specified acceleration up to a ratio of the primary axis' speed. Preset or continuous moves are performed in the same manner as in Indexer mode. Exact distances on the secondary axis can be moved in Preset mode, but instead of moving at a velocity specified by the **V** command, the secondary axis moves at a ratio of the primary axis' velocity specified by the **FOR** and **FOL** commands. The acceleration is independent of the primary axis encoder and is specified by **A** and **AD** (like an indexer).

Once the secondary axis accelerates to the specified following ratio, it tracks the primary axis' speed and position at the specified ratio. During the acceleration, the primary encoder speed and position are not followed. The acceleration ramp is independent of the primary axis. After the secondary axis accelerates to the specified following ratio, it will follow the primary axis (if the primary axis slows down, the secondary axis slows down).

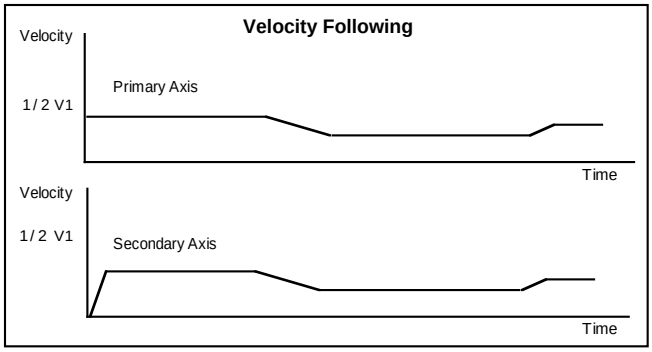
☞ ***Helpful Hint:***
Velocity following is used in dispensing and on-the-fly cutting applications.



Velocity Following

In each of the following categories described, the secondary axis only moves in the direction specified by either the **D** command, the **V** command, or the **H** command. If the primary axis changes direction, the secondary axis will still move in the same direction. Only the number of pulses and the rate of the pulses determines the secondary axis' motion. With position and direction tracking enabled, the secondary axis will follow the primary axis' direction. To illustrate velocity following, detach the encoder from the primary axis and manually move the encoder.

In the figure, if the primary velocity changed after the following ratio was achieved, position and velocity were tracked exactly. If the primary axis' velocity were half as fast as the example in the figure below, a different phase or positional relationship would result during acceleration from rest. This is why it is velocity following. Again, once the secondary axis accelerates to the specified following percentage, it will follow velocity and position exactly.



Primary Axis as Half Speed with Velocity Following

Setting Up Velocity Following

To perform velocity following, you must use two commands.

Command	Description
FOR	Relates the number of secondary motor steps per unit of travel to the corresponding primary encoder steps per unit of travel.
FOL	Relates the speed of the secondary axis to the speed of the primary axis as long as FOR is set correctly. The value is entered as a percentage of the primary axis speed.

The ZXF uses the equation to determine the number of motor steps.

$$\text{Motor Steps} = \text{FOR} \bullet \frac{\textbf{FOL}}{100} \bullet \text{Encoder Steps}$$

These two commands, (**FOR** and **FOL**) remove the complication of having a different resolution for measurement of distance on the primary and secondary axes. Once you relate the number of primary encoder pulses per unit of travel to the number of secondary motor pulses per that same unit of travel, you can relate their speeds. You can specify the secondary axis to move at 50% of the speed of the primary axis (1:2 ratio) or 100% of the speed (1:1 ratio). The encoder is usually mounted on the primary axis or through gearing and will measure a certain number of pulses per inch. The secondary axis would also have some certain number of pulses per inch. The ratio of the secondary axis steps to the primary axis' steps is entered in the **FOR** command. If the primary axis moves at 10 ips (inches per second) and the secondary axis is also set to move at 10 ips, a speed percentage of 100 is all that needs to be programmed (**FOL100**). The remaining commands are standard ZX commands. **Use the following steps to implement the *Velocity Following* feature**.

Step ①

Enter the number of secondary motor steps per unit of travel per number of primary encoder steps per the same unit of travel. In this example, motor step resolution is 25,000 steps/rev and encoder resolution is 4,000 steps/rev. The unit of travel is 1 revolution. The ratio is 25,000/4,000 or 6.25.

Command	Description
> FOR6.25	Sets the motor step to encoder step ratio of 6.25
> OFF	Turns the ZXF off
> CMR25ØØØ	Sets motor resolution to 25000 steps/rev
> ON	Turns the ZXF on

Step ②

Enter the following mode with the **FSI** command.

Command	Description
> FSI1	Enters the following mode

Step ③

Set the speed ratio. If the secondary axis is to move at the same speed as the primary axis (100% of the primary axis' speed), enter **FOL1ØØ**.

Command	Description
> FOL1ØØ	Secondary axis will move at 100% of the primary axis' speed

Step ④

Set the ZXF to Continuous mode and begin motion with the **G** command.

Command	Description
> MC	Enters Continuous mode
> A5ØØ	Sets acceleration
> AD5ØØ	Sets deceleration
> G	Initiates motion

Turn the encoder. The ZXF should move at the same speed that you are turning the encoder. Change the direction that you turn the encoder (note the secondary axis' motion). Follow the steps below.

Step ①

Execute the following commands.

Command	Description
> S	Stops the continuous motion command issued above
> 1FSP1	Enables Direction Tacking mode
> G	Initiates motion

Step ②

Turn the encoder or move the primary axis so that the encoder moves. The ZXF will begin moving. Change the direction that the encoder is moving. The ZXF will change direction. To change the relative direction between the ZXF and the encoder, use the change direction command (**H**±). Now turn the encoder and note that the ZXF's relative direction has changed. Direction tracking and pulses can only be used in Continuous mode. Preset moves can also be performed in Velocity Following mode.

Preset Moves In Velocity Following

A preset move is performed as in the standard Indexer mode. Issue the mode normal command (**MN**) and specify a distance in terms of secondary motor steps. In a preset move:

① The secondary axis accelerates (A) to the desired speed percentage.

② It decelerates at the rate set with the AD command.

③ It moves the specified distance (D) and stop.

If the primary axis' speed varies, the acceleration ramp will be the same, but the distance that the secondary axis travels to reach its following ratio will be different. The positional relationship or phase relationship is not maintained during acceleration. Once the secondary axis achieves the following percentage speed, it will track both velocity and position exactly.

Step ①

Attach the encoder to the primary axis. Start the primary axis moving

Step ②

Enter the following set of commands.

☞ ***Helpful Hint:***
The secondary axis moves 125,000 motor steps at the same speed as the primary axis. Repeat the example, but vary the primary axis' speed. The secondary axis still moves 125,000 steps, and its speed varies with the primary axis.

Command	Description
> MN	Enters Normal mode
> FSPØ	Exits the direction tracking mode
> A5ØØ	Sets acceleration
> AD5ØØ	Sets deceleration
> FOL1ØØ	Sets the following speed percentage to 100
> D125ØØØ	Sets the distance to 125000 steps
> G	Initiates motion

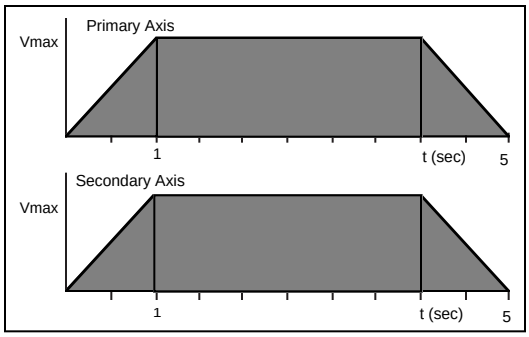
Velocity following is the simplest form of following. All motion in the ZXF is programmed like the ZX. The only exception is that the velocity command is replaced by a following percentage. Direction tracking must be disabled (**FSPØ**) while following in Preset mode (**MN**).

Position and Velocity Following

Position and velocity following is the most common form of following. In this case, the secondary axis must maintain a specific positional relationship with the primary axis. The primary axis moves a specific number of primary encoder pulses while the secondary axis moves a specified number of secondary motor steps. The secondary axis moves at the same speed as the primary axis (a 1:1 speed ratio) in most of these applications. Coil winding applications, however, are an exception. In these applications, the primary (i.e., the spindle axis) and secondary axes (i.e., the traverse axis) relationship is based on a desired pitch. This pitch defines a position and velocity that will arbitrarily set the speed based on the application. The positional relationship is usually defined by a single traverse that corresponds to a specific number of spindle axis turns, thus defining a specific number of primary encoder pulses during which a specific number of secondary pulses must be traveled.

Primary Axis at Rest

There are two variations of position and velocity following. The difference is in the primary axis' motion. The less common form starts the primary axis motion from rest. The secondary axis must follow the primary axis pulse for pulse. In this type of application, the secondary axis is in a continuous move and tracks the primary axis' motion exactly. A web positioning system where two axes are guiding the web (one edge is the primary axis and the other edge is the secondary axis), is an example of such an application. The web can be positioned based on the primary and secondary axes relationship.



Pulse-for-Pulse Following

If the primary axis starts from rest, the secondary axis must track it pulse for pulse (**FSA** command). If position tracking is disabled, the ZXF will only follow the pulse count and will not change direction if the primary axis changes direction. This is acceptable in many applications. In some applications, however, the primary axis may overshoot when it comes to a rest and the encoder will change directions. The secondary axis may mimic the overshoot if position tracking is enabled.

Since the ZXF is following the primary axis' pulse count only, it actually moves the secondary axis too many pulses. By enabling Position Tracking (**FSP1**), you can track the primary encoder's direction and pulse count and not accumulate excess pulses caused by overshoot.

Command	Description
FSA1	Enables Pulse Tracking—secondary axis instantly accelerates between commanded velocities
FSP1	Secondary axis tracks both the direction & pulse count from the primary encoder

Disable Following Synchronized Acceleration (**FSFØ**) and enable Continuous Move mode to use Pulse Tracking (**FSA1**).

Primary Axis in Motion

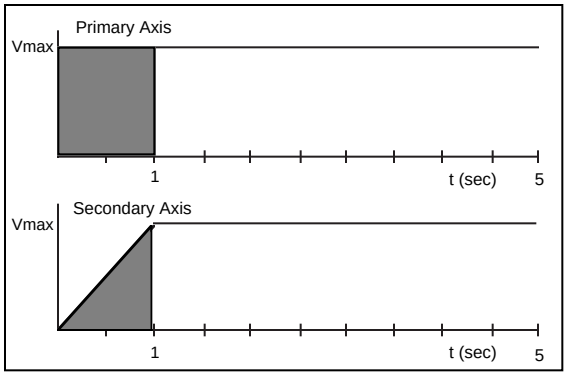
The most common form of position and velocity following begins with the primary axis already in motion. The secondary axis must accelerate to the primary axis' specified speed ratio. In this type of application, the secondary axis must accelerate to a known positional or phase relationship with the primary axis. The primary axis is usually a conveyer or web and the secondary axis performs an operation on the web or parts on the conveyer. The primary axis is always moving, so the secondary axis must move at the same speed and with the correct orientation to perform some operation on the moving primary axis. To maintain positional and velocity relationships, the secondary axis must accelerate over a known distance with respect to the primary axis (i.e., following acceleration is needed). The Set Following Synchronization Rate (**FAC**) and Set Following Synchronization Count (**FEN**) commands are used in conjunction with the velocity following commands.

☞ ***Helpful Hint:***
It does not matter if the speed of the primary axis varies, the secondary axis' acceleration is based on primary encoder steps (not time).

The **A** and **AD** commands used in velocity following are replaced by a following acceleration. A following acceleration is attained by stepping through subsequent ratios from the present following ratio to the final following ratio (**FOL**). The increment between ratios is set by the **FAC** command. Incrementing from one ratio to the next is based on the primary encoder changing by a set number of primary encoder pulses. This number of pulses is set by the **FEN** command. For example, if the value for **FAC** is 1, the value for **FEN** is 10, and the final following speed percentage is 100 (**FOL**), the secondary axis must accelerate from zero speed to 100% of the primary speed. From rest, every 10 primary encoder steps (**FEN**) the following percentage will change by an increment of 1% (**FAC**) until it is equal to the final following percentage of 100% (**FOL**). Therefore, the secondary axis will accelerate over 1000 primary encoder steps.

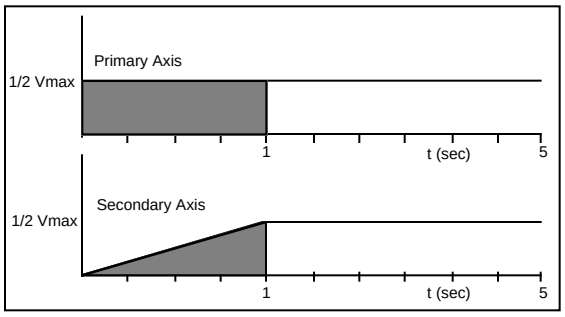
The shaded area indicates the distance moved by both the primary and secondary axes while the secondary axis is accelerating. The secondary axis moves 1/2 the distance that the primary axis moves. This will always be the case when the secondary axis accelerates from rest to the same speed as the primary axis, when the primary axis is already moving.

☞ ***Helpful Hint:***
If the primary axis' speed is 1/2 the maximum velocity, the acceleration ramp will be twice as long. The shaded portions in the top figure are equivalent to the shaded portions in the bottom figure. The primary axis' velocity can change at any time, even during the secondary axis' acceleration ramp, without changing the positional relationship.



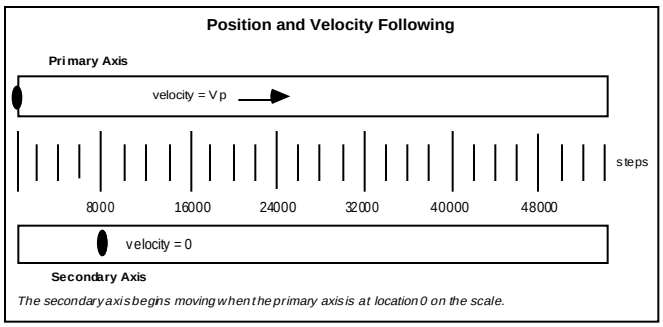
Following Acceleration

☞ ***Helpful Hint:***
This figure shows how the acceleration rate varies if the primary axis' speed varies



Following Acceleration With Primary Axis Velocity Change

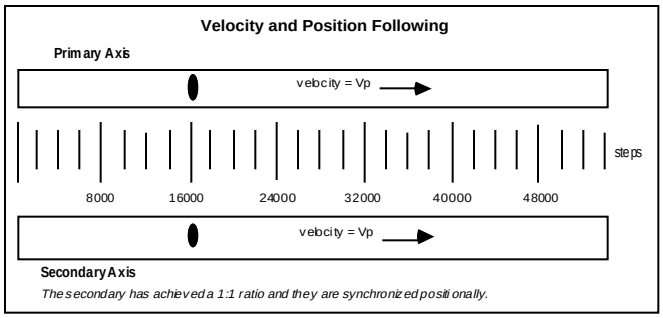
The shaded area for the secondary axis is 1/2 of the shaded area under the primary axis curve. When the primary axis is moving and the secondary axis must start from rest and accelerate to the primary axis' velocity, the secondary axis will always move 1/2 the distance. Using following acceleration, the secondary axis adjusts its acceleration according to the primary axis' velocity (it will always accelerate over the same distance while the primary axis moves a specified distance). To synchronize secondary and primary axes' positions, the secondary axis must start ahead of the primary axis to compensate for the fact that the primary axis is already moving (velocity = Vp).



Starting a Velocity & Position Following Move—1:1 Ratio

The key to position and velocity following is that the Vp value does not matter. Assuming the application in the figure above is programmed with following acceleration (the spots)

while accelerating over 8000 primary encoder steps, will always be at a 1:1 speed ratio after the primary axis moves the 16000 steps (see figure below).



Velocity & Position Following After Acceleration Is Done

Following Acceleration

Accelerating over a known distance with respect to a known primary axis distance allows you to synchronize the exact phase relationship you want between the primary and secondary axes. Following acceleration enables this synchronization.

Command	Description
FAC	The increment of following ratio by which the following ratio changes during acceleration
FEN	The number of encoder pulses that cause the following ratio to increment to the next value
FSF1	Enables Acceleration mode

Calculating FAC & FEN with Axis #1 Data

To determine what **FAC** and **FEN** values, you must know the following:

- ❏ The maximum velocity that the primary axis can travel (the velocity must be in units of primary encoder steps per second)
- ❏ The distance in primary encoder steps that the primary axis will move during which time the secondary axis will accelerate or the maximum acceleration that the secondary axis can accelerate

These parameters will be denoted as follows:

Vp_{max} = Maximum primary velocity in encoder steps per second

Dp_{acc} = Distance primary axis travels while secondary axis accelerates in primary encoder steps

OR

As_{max} = Maximum acceleration of the secondary axis

Based on these equations, the values for **FAC** and **FEN** can be determined:

FEN Equation

$$\textbf{FEN} = Vp_{max} \cdot \frac{TF}{1000} \quad Vp_{max} = \frac{\text{Primary Encoder Counts}}{\text{Second}}$$

TF = Primary Encoder Sample Period in ms

FAC Equation

$$\textbf{FAC} = \frac{\textbf{FOL} \cdot Vp_{max}}{Dp_{acc}} \cdot \frac{TF}{1000} \quad Vp_{max} = \frac{\text{Primary Encoder Counts}}{\text{Second}}$$

TF = Primary Encoder Sample Period in ms

FOL = Following percentage in units of percent

Dp_{acc} = Distance primary axis moves during secondary axis accel in units of primary axis encoder counts

Typically, the application will have the secondary axis start from rest and accelerate up to an **FOL** value of 100 (1:1 ratio). However, the FOL value can be any value that is within the limits of the motor/drive system. D_{acc} is in units of primary encoder steps. Determining the **FAC** and **FEN** values sets the number of primary axis encoder steps over which the secondary axis will accelerate (independent of the primary axis' speed). The secondary axis always travels the same number of motor steps during acceleration while the primary encoder moves Dp_{acc}.

The **TF** command allows you to set the sample period of the primary axis' encoder. It is programmable from 1 - 32 ms. The default is 4 ms. **TF** simply scales the **FAC** and **FEN** values. In the **FAC** and **FEN** equations, **TF** is used in units of ms—the constant of 1000 converts it to seconds so that the units cancel properly. Typically you will want **TF** to be as fast as your system will allow (1 ms). If the primary encoder is moving slowly, you may need to increase the sample rate to more than 1 ms because the actual encoder count does not

change by much in a sample period and thus you have coarser resolution on the changes in encoder counts. For example, if your sample period is 1 ms and the maximum speed is 1 rps, the encoder count only changes by 4 counts each sample period. If there is a slight variation in speed and you read a change of 3 counts, there is a 25% variation. This may cause choppier secondary axis motion. Changing the encoder sample period can have a smoothing effect.

Calculating FAC & FEN with Axis #2 Data

You may know only the maximum acceleration rate of your secondary axis. You can determine Dp_{acc} using A_{max} and the following speed percentage that you are accelerating to.

Dpacc Equation

$$Dp_{acc} = \frac{Vp_{max}^2}{As_{max}} \bullet FOR \bullet \frac{FOL}{100}$$

The acceleration is in units of secondary motor steps/sec^2. The maximum velocity of the primary axis is in primary motor steps/sec.

Using this value for Dp_{acc}, you can use the **FAC** and **FEN** equations. Remember to enable the Following Synchronized Acceleration mode (**FSF1**) to enable following acceleration.

How Following Acceleration Works

The concept of accelerating the secondary axis over a known distance with respect to a known primary axis distance (independent of the primary axis' speed) is based on analogies drawn between following and time-based motion. In time-based motion, the velocity describes the rate of change of the position with respect to a change in time. In following-based motion, the secondary axis moves at a ratio of the primary axis' velocity. This following ratio is of the same units as the velocity it is following, but is scaled (*the following ratio is analogous to a velocity*). Similarly, an acceleration in the time domain is defined as the rate of velocity change. The analogy in following would be to have a following acceleration that is a rate of change of the following ratio. Time-based motion is based on sampled time whereas following is based on the sampled primary axis encoder pulses (for digital systems).

☞ ***Helpful Hint:***
Following Acceleration Example

- ❏ Primary axis encoder fi 4000 counts/revolution
- ❏ Secondary axis fi 4000 steps/revolution
- ❏ Primary axis speed = 1 rps
- ❏ Secondary axis following speed percentage = 100%
- ❏ Distance that secondary axis accelerates = 2000 primary axis encoder steps
- ❏ Primary axis encoder sample period = 1 ms

The secondary axis must now accelerate over 2000 primary encoder steps to a following percentage of 100% or a speed of 1 rps. Specifying the number of primary encoder pulses and the final speed that the secondary axis must attain after acceleration defines the acceleration ramp. If the secondary axis' acceleration is based on time, you can calculate an acceleration ramp that will accelerate the secondary axis properly.

$$\text{Acceleration time} = \frac{2000 \text{ steps}}{4000 \ \frac{\text{steps}}{\text{sec}}} = 0.5 \text{ seconds}$$

Change in velocity = 4000 steps/sec – 0 steps/sec = 4000 steps/sec

$$\text{Acceleration} = \frac{4000 \ \frac{\text{steps}}{\text{sec}}}{0.5 \text{ sec}} = 8000 \ \frac{\text{steps}}{\text{sec}^2} = \frac{8 \ \frac{\text{steps}}{\text{sec}}}{\text{ms}}$$

If velocity is changed by 8 steps/sec every sample period (TF - 1 ms), you will achieve the desired acceleration ramp. *The problem is that the application is time dependent.* If the primary axis' speed changes, the secondary axis would no longer accelerate over 2000 primary encoder steps.

☞ ***Helpful Hint:***
The application requires a following acceleration that is based on encoder pulses rather than time.

Change in Following Percentage = 100% - 0% = 100%

$$\text{Following Acceleration time} = \frac{2000 \text{ steps}}{4000 \ \frac{\text{steps}}{\text{sec}}} = 500 \text{ ms} = 500 \text{ sample periods}$$

$$\text{Following Acceleration} = \frac{100}{500} = 0.2 \text{ (FAC)}$$

The application is still time dependent. To remove the time dependency and make the acceleration dependent on the encoder pulses, replace the time sample period by an encoder period. In the previous example, the following percentage was based on changing the following percentage by 0.2 every sample period. It will take 500 sample periods to achieve a 100% following percentage. At 4000 steps/sec, the primary encoder is changing at a rate of 4 steps per ms or 4 steps/sample period.

1 sample period = 4 steps (encoder period–FEN).

Instead of changing the following percentage every sample period, change it every time the encoder count changes by 4 steps. If the primary encoder is moving at the maximum velocity, the acceleration ramp will be equal to the maximum acceleration provided. However, if the primary encoder velocity is less than the maximum velocity, the acceleration will also be reduced. The distance that the secondary axis accelerates and travels (with respect to the primary axis' moves during this acceleration) will remain unchanged. If the primary axis exceeds the maximum velocity, the acceleration ramp would also increase and exceed the maximum acceleration, which may cause an overcurrent or related servo error.

In the ZXF, the following percentage changes by the **FAC** value for each change in encoder steps of **FEN** steps. Several different combinations of **FAC** and **FEN** can achieve the same acceleration ramp. However, only unique **FAC** and **FEN** values will satisfy a specific maximum velocity and maximum acceleration. **FAC** and **FEN** can easily be determined with the equations and examples discussed earlier.

Decelerating

The ZXF decelerates to zero speed using the **AD** value. Although you may expect that this will diminish the positional relationship, this is usually not a concern at the endpoint in a profile. The following move is usually started by a trigger, which indicates that the primary axis is at a particular location. The move could also be started based on the primary axis' encoder position. The move is typically of a preset distance. ***The important point is that the secondary axis is at a known position with respect to the primary axis when the profile begins***. This positional relationship is maintained during acceleration. After the secondary axis moves the appropriate distance, it will normally return the same distance it just traveled at a high speed to prepare for a repeat move. This is why the deceleration is not important. However, in cases where the ZXF must perform electronic cam profiles, a deceleration's positional relationship may be necessary. In this case, the ZXF can decelerate to a stop using the **FAC** and **FEN** values by setting the following ratio to Ø (**FOLØ**) while in Mode Position Profile mode. To terminate this move, a stop command must be issued after the secondary axis reaches zero speed.

Position and Velocity Following Examples

In this example, you will perform a preset move using following acceleration (with the parameters and motion requirements listed below).

- ❏ Primary axis encoder resolution = 4000 counts/rev
- ❏ Secondary axis motor resolution = 25000 steps/rev
- ❏ Maximum primary encoder speed = 2 rps
- ❏ Distance in primary encoder steps that the secondary axis must accelerate over = 2000 steps
- ❏ Desired speed ratio—**FOL** = 100% (1:1 ratio)
- ❏ Preset secondary axis move distance = 3 motor revolutions
- ❏ Encoder sample period—**TF** = 4 ms

A preset move of 75000 steps will be made. The secondary axis will accelerate over 2000 primary encoder steps up to the same speed as the primary encoder. A trigger will initiate motion. Follow these steps to perform the move profile. The encoder sampling period is set to the default of 4 ms.

Step ① Determine the values for **FAC** and **FEN** from the following equations.

$$\mathbf{FEN} = Vp_{max} \cdot \frac{\mathbf{TF}}{1000} = 2\ \frac{revs}{sec} \cdot 4000\ \frac{counts}{rev} \cdot \frac{4}{1000}\ seconds = 32\ encoder\ counts$$

$$\mathbf{FAC} = \frac{\mathbf{FOL} \cdot Vp_{max}}{Dp_{acc}} \cdot \frac{\mathbf{TF}}{1000} = \frac{100 \cdot 8000\ \frac{counts}{sec}}{2000\ counts} \cdot \frac{4}{1000}\ seconds = 1.6\ percent$$

Step ② Enter the values for **FEN** and **FAC**.

Command	Description
> FEN32	Number of encoder counts of change required to increment the following percentage by **FAC**
> FAC1.6	The amount the following percentage increments for each **FEN** change in encoder counts

If you want to change the primary encoder's sample period to 1 ms, scale both **FEN** and **FAC** by the change in the **TF** value. For example, if you go from 4 ms to 1 ms, divide **FAC** and **FEN** by 4 to get the following values.

FEN = 8
FAC = 0.4

If we went from **TF4** to **TF8**, multiply **FAC** and **FEN** by 2. ***Remember to change* FAC *and* FEN *if you change* TF**.

Step ③ Enable the following acceleration mode.

Command	Description
> FSF1	Enables following acceleration

Step ④ Start the primary axis, then enter the commands below to perform the following acceleration move. If the primary axis' speed exceeds 2 rps, the following acceleration will not work properly.

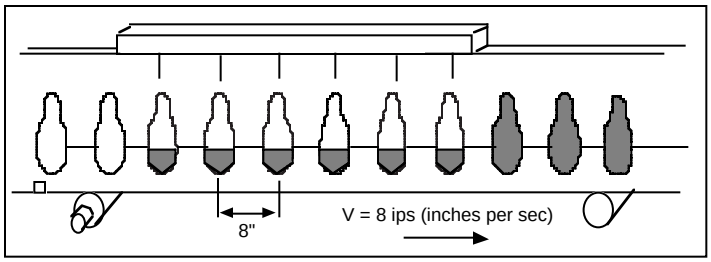
☞ ***Helpful Hint:***
Secondary axis accelerates over 2000 primary encoder steps. Secondary axis moves 1000 • **FOR** or 6250 motor steps over this acceleration ramp.

Command	Description
> MN	Sets ZXF to Normal mode
> FSI1	Enables following
> D75ØØØ	Sets the preset move distance to 75000 steps
> FOL1ØØ	Sets the following percentage to 100%
> G	Starts the following move

Following Acceleration Application: Bottle Filling

Typically, an application that requires position and velocity following will start the secondary axis from rest and accelerate it to 100% of the primary axis' speed (a 1:1 speed ratio). A trigger initiates motion on the secondary axis when a part or product is in a particular location on the primary axis. In this example, a conveyer belt moves bottles on a production line. The secondary axis is a filler that accelerates up to the conveyer's line speed and fills the bottles. It fills six bottles at a time and then returns to the start point to fill six more.

☞ ***Helpful Hint:***
In this application, the rate at which the bottles can be filled determines the maximum rate of the entire dispensing cycle.



Position & Velocity Following—Bottle Filling Application

One cycle of operation consists of the following steps.

① Secondary axis accelerates to the conveyer line speed.

② Secondary axis enables an output that prompts dispenser to fill bottles.

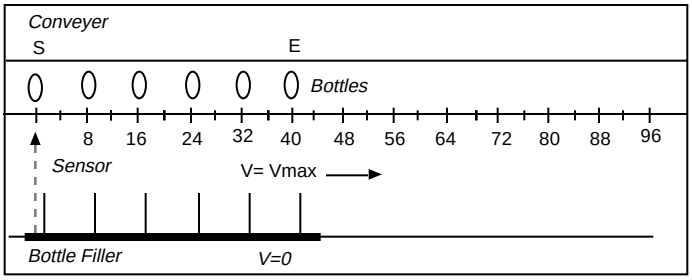
③ Secondary axis decelerates to a stop and returns to the starting point (at a high speed) to begin filling the next set of bottles.

☞ ***Helpful Hint:***
Application Information

Maximum conveyer speed: 8 ips
Time to bill bottles: 2.5 seconds
Primary encoder resolution: 4000 counts/rev
Primary encoder linear resolution: 2 revs/inch = 8000 counts/inch
Distance between bottles: 8 inches
Bottle filler motor resolution: 25000 steps/revolution
Bottle filler linear resolution: 1 revolution/inch = 25000 steps/inch
Distance over which bottle filler accelerates: 2 inches

An output must be activated at the point that the bottle filler is moving at the same speed as the conveyer to initiate the dispensing of fluid into the bottles. A photoelectric sensor detects a bottle and begins the filling cycle. The ZXF waits for the sensor as a trigger. When a bottle is detected, the ZXF accelerates to the line speed, turns on an output, fills the bottle stops, and returns to the starting point to wait for the next trigger. The figure below depicts the conveyer and the filler axis at the start of a filling cycle.

☞ ***Helpful Hint:***
The cycle begins when the bottle marked S crosses the trigger sensor.



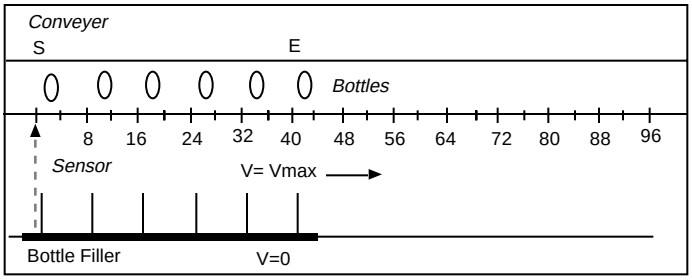
Start of A Filling Cycle

① Secondary axis accelerates to the conveyer line speed.

② Secondary axis enables an output that tells dispenser to begin filling bottles.

③ Secondary axis decelerates to a stop and returns to the starting point (at a high speed) to begin filling the next set of bottles.

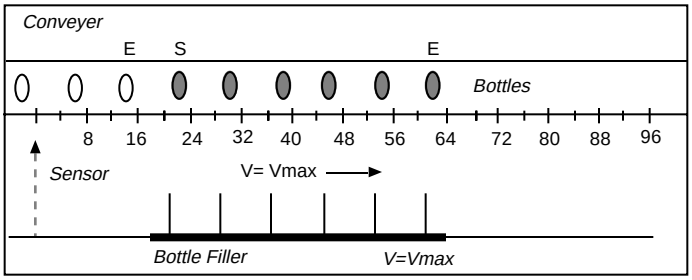
☞ ***Helpful Hint:***
This figure shows the two axes after the bottle filler reaches line speed and is ready to begin dispensing.



Bottle Filler Has Accelerated to Line Speed

A bottle can be filled in 2.5 seconds. At a maximum conveyer speed of 8 ips, the conveyer will move the bottles 20 inches. The figure below shows the bottle's location after the first six bottles have been filled.

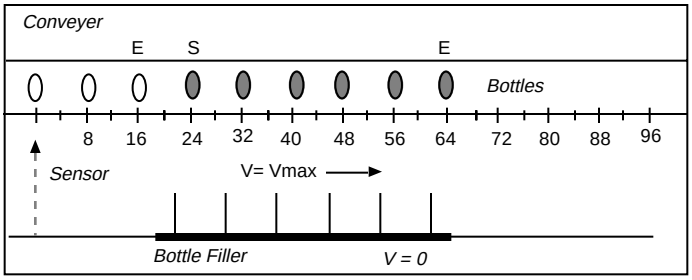
☞ ***Helpful Hint:***
Note the location of the next set of bottles to be filled.



Dispensing is Completed

The bottle filler must now stop and return to the start location before the bottle marked **S** crosses the trigger point. It must now return 22 inches to the start before the bottles have moved 20 inches. The bottle filler will be in place, ready for the next trigger from the next set of bottles. It will arrive in place 4 inches before the next bottle. At 8 ips, 500 ms will elapse before the next bottle.

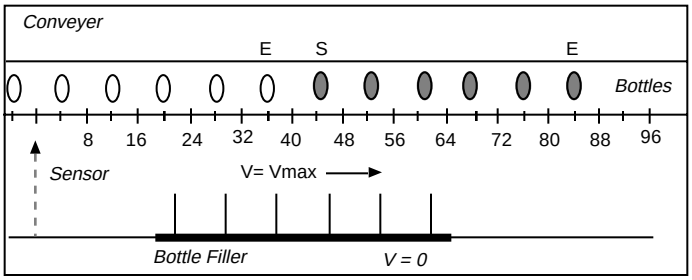
☞ ***Helpful Hint:***
This figure the location of the bottle filler after it has stopped.



Bottle Filler Stops

To get back to the starting point before the conveyer has moved 20 inches, the bottle filler must return at a speed faster than the conveyer. It will accelerate at the same following acceleration set for the first part of the cycle. After the conveyer travels 4 more inches, the cycle will resume.

☞ ***Helpful Hint:***
This figure shows the bottle filler after it returns to the starting point.



Bottle Filler Ready to Start a New Cycle

To program the application, use the following steps.

Step ①

Determine the **FOR** value.

Primary conveyer axis steps per inch = 8000

Secondary bottle filler axis steps per inch = 25000

$$\mathbf{FOR} = \frac{25000}{8000} = 3.125$$

Step ②

Determine the **FAC** and **FEN** following acceleration parameters. The encoder sample period time (**TF**) is 1 ms.

$$\mathbf{FEN} = 8000 \frac{\text{steps}}{\text{inch}} \bullet 8 \frac{\text{inches}}{\text{second}} \bullet \frac{1}{1000} = 64 \text{ steps}$$

$$\mathbf{FAC} = \frac{100\% \bullet 8000 \frac{\text{steps}}{\text{inch}} \bullet 8 \frac{\text{inches}}{\text{second}}}{16000 \text{ steps}} \bullet \frac{1}{1000} = 0.4$$

Step ③

Determine how far the primary and secondary axes will move during the dispensing part of the cycle. This includes the bottle filler's acceleration and deceleration moves when no fluid is dispensed.

$$Vp_{max} = 64000 \frac{\text{steps}}{\text{sec}}$$

The dispensing takes 2.5 seconds. During the time that a constant following percentage occurs, the conveyer will have moved:

$$Dp_{con} = 64000 \frac{\text{steps}}{\text{sec}} \bullet 2.5 \text{ sec} = 160000$$

The secondary axis will move this distance • **FOR**:

$$Ds_{con} = 160000 \text{ primary encoder steps} \bullet 3.125 \frac{\text{secondary motor steps}}{\text{primary encoder steps}} = 500000 \text{ secondary motor steps}$$

The secondary axis or bottle filler axis will accelerate over 2 inches of the conveyer at maximum speed and decelerate over two inches. If the conveyer moves slower, the dispensing part of the process will become a smaller percentage of the total cycle. The distance the secondary axis travels during acceleration will be one half of the distance the conveyer or primary axis travels. The secondary axis starts from rest and accelerates to match the primary axis' speed. The secondary axis will move 1 inch during acceleration and 1 inch during deceleration. The total distance of the bottle filler move is:

$D_{sec} = 500000 + 2 \cdot \left(1 \text{ inch} \cdot 25000 \frac{\text{steps}}{\text{inch}}\right) = 550000$ steps (secondary mtr steps)

The value to be entered for the **AD** command for deceleration is:

$$\textbf{AD} = \frac{V^2}{2 \cdot D} = \frac{\left(8 \frac{\text{inches}}{\text{sec}}\right)^2}{2 \cdot 1 \text{ inch}} = 32 \frac{\text{inches}}{\text{sec}^2} \cdot 1 \frac{\text{revolution}}{\text{inch}} = 32 \frac{\text{revs}}{\text{sec}^2}$$

If the primary axis' speed changes, the ZXF will still decelerate at this rate. The distance that the bottle filler moves will be identical, regardless of the primary axis' speed, so the bottle filler will still have to make a 22-inch move *back* while the primary axis moves 20 inches.

Step ④

You have determined the parameters necessary for the first part of the move. Next, you must determine the following percentage required to move the secondary axis back 22 inches while the primary axis moves 20 inches. Use the following equation to determine **FOL**.

FOL Equation

$$\textbf{FOL} = \frac{D_{prim}}{200 \cdot K} \sqrt{\left(\frac{D_{prim}}{200 \cdot K}\right)^2 \frac{D_{sec}}{\textbf{FOR} \cdot K}} \quad \text{Where } K = \frac{\textbf{FEN}}{100 \cdot \textbf{FAC}}$$

$$\textbf{FOL} = \frac{160000}{200 \cdot 1.6} \sqrt{\left(\frac{160000}{200 \cdot 1.6}\right)^2 \frac{550000}{3.125 \cdot 1.6}} = 500 - 374.2 = 125.8\%$$

125.8% is the return following percentage.

Step ⑤

Enter the sequence below to implement the motion.

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
FOR3.125	Sets motor to encoder steps per unit travel ratio
FAC.4	Sets the following acceleration increment to .4 % per encoder period
FEN64	Sets the encoder period, which increases the percentage to 64 steps
D550000	Sets the secondary axis move for the cycle to 550000 motor steps
FSI1	Enables following mode
AD32	Sets the deceleration to 32 rps^2
IN1A	Defines input 1 as a trigger input
IN2D	Defines input 2 as a stop input
L	Starts a continuous loop
FOL100	Sets the initial following percentage to 100%
TR1	Waits on the input trigger
G	Starts motion
FOL125.8	Sets return move following percentage to 125.8%
H	Changes the direction
G	Starts the return following move
H	Changes direction again
N	Ends the loop—the following cycle will repeat
> XT	Ends the sequence

To decelerate using the **FAC** and **FEN** values, modify the program as follows:

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
SSH1	Sets save buffer on stop
FOR3.125	Sets motor to encoder steps per unit travel ratio
FAC.4	Sets the following acceleration increment to .4 % per encoder period
FEN64	Sets the encoder period, which increases the percentage to 64 steps
D550000	Sets the secondary axis move for the cycle to 550000 motor steps
FSI1	Enables following mode
AD32	Sets the deceleration to 32 rps^2
IN1A	Defines input 1 as a trigger input
IN2D	Defines input 2 as a stop input
L	Starts a continuous loop
FOL100	Sets the initial following percentage to 100%
TR1	Waits on the input trigger
MPP	Enters the profiling mode
G	Starts motion
FP176000	Waits until 176000 encoder pulses have passed
FOL0	Stops the motion of the secondary
FP16000	Waits for the decel ramp distance
STOP	Stops the move itself
FOL125.8	Sets return move following percentage to 125.8%
H	Changes the direction
G	Starts motion
FP139782	Waits until 139782 encoder pulses have passed

FOLØ	Stops the motion of the secondary
FP2Ø218	Waits for the decel ramp distance
STOP	Stops the move itself
H	Changes direction again
N	Ends the loop—the following cycle will repeat
> XT	Ends the sequence

The value for **FP** during the return move is determined by calculating the distance the primary axis will move during acceleration and the constant following percentage portion and determining the distance it moves during the deceleration portion. This is determined from:

$$Dp_{acc} = \mathbf{FOL} \cdot \frac{\mathbf{FEN}}{\mathbf{FAC}} = 125.8\% \cdot \frac{64}{0.4} = 20218 \text{ encoder steps}$$

$$Dp_{dec} = Dp_{acc} = 20218 \text{ inches}$$

$$Dp_{con} = D_{prim} - Dp_{dec} - Dp_{acc} = 119564$$

$$D_{prim} = 20 \text{ inches} = 160000$$

The *first* **FP** = $Dp_{con} + Dp_{acc}$ = 119564 + 20218 = 139782. The *second* **FP** = Dp_{dec} = 20218. The second **FP** (20218 steps) measures the deceleration ramp. After deceleration, the move stops. The acceleration and deceleration ramps are based on the primary axis' speed.

Recede and Advance While Following

Receding and advancing while following requires position and velocity following. In this type of application, the secondary motor follows the primary encoder at a 1:1 ratio or at the same speed. The secondary motor has a specific positional or phase relationship with the primary encoder. This type of application is used when multiple operations (such as welds) must be performed on one moving part. The operations are performed at various places on the part, requiring the secondary axis to advance or recede.

In an ***advance application***, the secondary axis must accelerate and move a specific distance beyond the primary axis, then decelerate to a 1:1 ratio. The secondary axis moves a specific distance with respect to the primary axis while both axes are moving. In a ***recede application***, the secondary axis decelerates until it recedes a specific distance behind the primary axis and then resumes a 1:1 speed ratio with the primary axis.

The point at which the advance or recede move occurs is based on a specific position on the primary axis or an input trigger. This type of application requires the following ratio be changed on-the-fly while based on either an input or the primary encoder's position. It also requires the secondary axis to move a specific distance while the primary axis moves a corresponding specific distance. In this manner, the secondary can advance or recede a specific distance with respect to the primary axis.

To change the following ratio on-the-fly, you must use Motion Profiling (**MPP**) mode. You will need to measure the distance traveled by the primary encoder. Use the set of commands below.

Command	Description
> FPn	Delays command processing for n primary encoder steps
> FPAn	Delays command processing until the absolute count of the primary encoder has reached the value of n
> VAR1=FEP	Allows you to read the Following Encoder Position's value into variable 1

Advance Following Example

In this example, the primary axis has a 4000 count per revolution encoder. The secondary motor has a 4000 step per revolution motor. Therefore, the value for **FOR** is 1. The application requires that the secondary axis:

① Accelerate over 4000 primary encoder steps

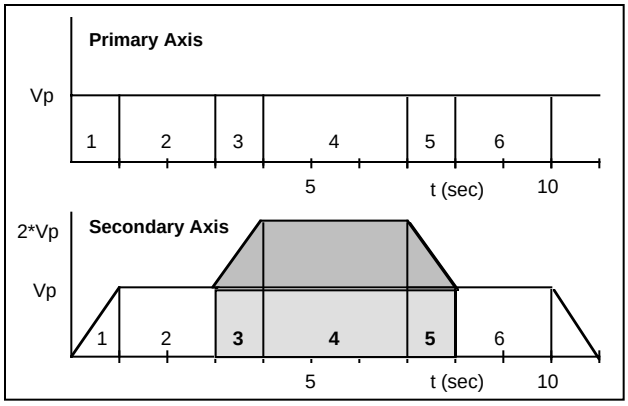
② Move at a 1:1 ratio for 2 primary encoder revolutions

③ Advance (with respect to primary axis) 12000 primary encoder steps

④ Move 2 more primary encoder revolutions at 1:1 after advancing

⑤ Stop

☞ ***Helpful Hint:***
This figure shows the move profiles. The dark shaded area is the distance the secondary axis advances with respect to the primary axis. The light shaded area is the distance that both axes move during the *advance portion* of the profile.



Advancing with Respect to the Primary While Following

In these types of applications, the phase relationship or positional relationship is set with the **FAC** and **FEN** values (for as long as the following acceleration is performed). When the secondary axis accelerates from the 1:1 ratio to the 2:1 ratio, it will again accelerate at the following acceleration set. When it decelerates back to the 1:1 ratio, it decelerates in the same manner as it accelerates (using the **FAC** and **FEN** following acceleration rate). When the secondary axis decelerates back to *rest or zero speed*, it will decelerate at the **AD** rate. If the secondary axis must decelerate to zero speed at the **FAC** and **FEN** values, you must use **FOLØ** and set the following ratio to zero to make it stop.

Before programming this move profile, we will completely analyze the motion of the secondary and primary axes and then describe the sequence of commands necessary for performing the move.

In the figure above, assume that the primary axis is moving at a maximum speed of 4000 steps/second. The area of each 1-second block represents 4000 primary encoder counts. Starting with Section #1, the primary encoder begins the section at a speed of $Vp_{max} = V_p$ = 4000 steps/sec. Therefore, the primary encoder moves 4000 counts during this section. From Section #1 of the secondary profile's plot, you can see that the secondary motor starts from rest and accelerates to a 1:1 speed ratio over 4000 counts of the primary encoder, Dp_{acc} = 4000 steps. You can calculate the desired **FAC** and **FEN** values to create such an acceleration ramp. This following acceleration will be used throughout the profile when changing from one following ratio to another. Set the encoder sample rate, **TF**, to 1 ms.

Vp_{max} = 4000 steps/sec

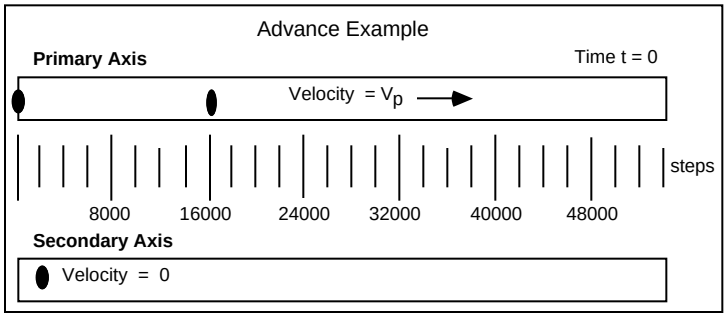
Dp_{acc} = 4000 steps

$$\textbf{FEN} = Vp_{max} \cdot \frac{\textbf{TF}}{1000} = 4000 \frac{\text{steps}}{\text{sec}} \cdot \frac{1}{1000} \text{ sec} = 4 \text{ encoder counts}$$

$$\textbf{FAC} = \frac{FOL \cdot Vp_{max}}{Dp_{acc}} \cdot \frac{\textbf{TF}}{1000} = \frac{100\ \% \cdot 4000 \frac{\text{steps}}{\text{sec}}}{4000 \text{ steps}} \cdot \frac{1}{1000} \text{ sec} = 0.1 \text{ percent}$$

By calculating the area under the secondary axis profile curve, you can determine that the secondary motor has moved 2000 motor steps. If the secondary axis is accelerating to the same speed as the primary axis, it will always travel half of the primary axis' distance. The secondary axis will be physically aligned with this point when it reaches a 1:1 speed ratio. The figure shows two conveyer belts—primary and secondary axes.

☞ ***Helpful Hint:***
This figure shows the locations of spots on the axes at different times during the move profile.



Advance Example at the Start of Section # 1

Usually, you will use an equation, not a graphic, to determine the distance traveled. The Ds_{acc} equation calculates the distance the secondary axis travels for any acceleration, even if the secondary axis starts acceleration while it is already moving at a given ratio to the primary axis.

Ds_{acc} Equation

$$Ds_{acc} = \mathbf{FOR} \cdot \left(\frac{1}{2} \cdot \Delta \mathbf{FOL}^2 \cdot \frac{\mathbf{FEN}}{100 \cdot FAC} + \Delta \mathbf{FOL} \cdot \mathbf{FOL}_1 \cdot \frac{\mathbf{FEN}}{100 \cdot \mathbf{FAC}}\right)$$

FOL = The change in following percentage

$\Delta \mathbf{FOL}_I$ = The initial following percentage

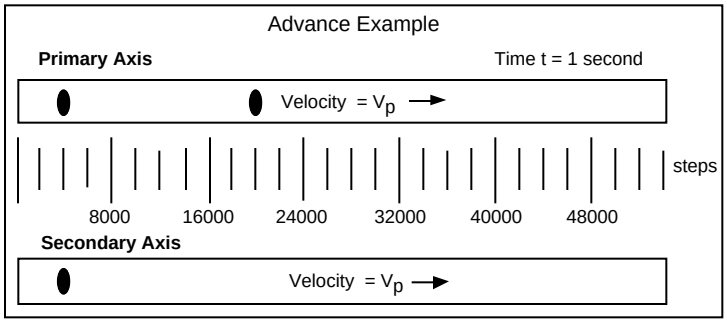
In the Ds_{acc} equation, the change in following percentage is the difference between the final following percentage that you are accelerating to and the following percentage you are starting from. If you are starting from rest, the initial following percentage is Ø and the change in following percentage is **FOLØ** or **FOL**. If you are at **FOL1ØØ** and want to accelerate to a following percentage of **FOL2ØØ**, the change in following percentage is 100 and the initial following percentage is 100. **FAC** and **FEN** are the values calculated from the equations for **FAC** and **FEN** using the Vmax and primary encoder acceleration distance.

The equation has two parts. The first part has the square of the following percentage change and the second part has a single following percentage change term. The first term determines the distance that the secondary axis travels due to the acceleration ramp portion of the curve. The second term gives the distance that the secondary axis travels due to its initial velocity. In this case, the secondary axis starts from rest (the second term contributes zero to the distance traveled). From the plot of the profile (Section # 1), the initial following percentage is Ø and the final percentage is 100, or the same speed as the primary axis. Using the general equation above, you should get 2000 steps, which is the area under the curve.

$$Ds_{acc} = 1 \cdot \left(\frac{100^2}{2} \cdot \frac{4}{100 \cdot 0.1} + 100 \cdot 0 \cdot \frac{4}{100 \cdot 0.1}\right) = 2000 \text{ secondary mtr steps}$$

Therefore, in Section #1, the primary axis moved 4000 encoder counts and the secondary axis moved 2000 motor steps. We have determined the values for **FAC** and **FEN** based on the plots of the primary and secondary axes' profiles and the fact that V_{max} is 4000 steps/sec.

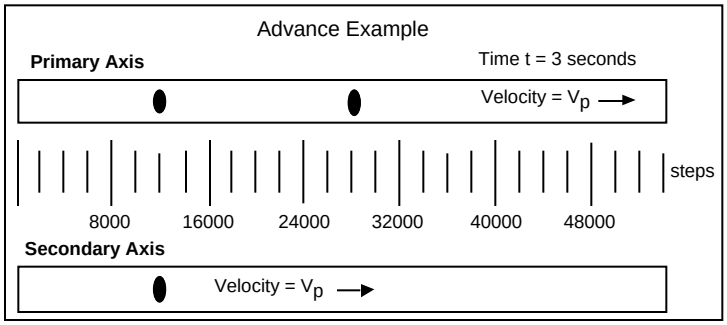
☞ ***Helpful Hint:***
At the end of Section #1, the spots will be in the locations shown in this figure .



Advance Example—End of Section #1

In Section #2, the primary axis is moving at V_{max} and the secondary axis is moving at the same speed because the ratio is 1:1 (**FOL1ØØ**). Section #2 lasts for 2 seconds. The primary axis travels 8000 steps during this section. The secondary axis travels 8000 steps too. This can be determined from the profile plot by calculating the area beneath the curve for the section. The primary and secondary axes are lined up at the start of Section #2 and they travel at a 1:1 ratio for the duration of the section. At the end of Section #2, the primary axis has moved 12000 steps and the secondary axis has moved 10000 steps (thru Sections #1 and #2).

☞ ***Helpful Hint:***
This figure shows the relative location of the spots at the end of Section #2 (t = 3 seconds).



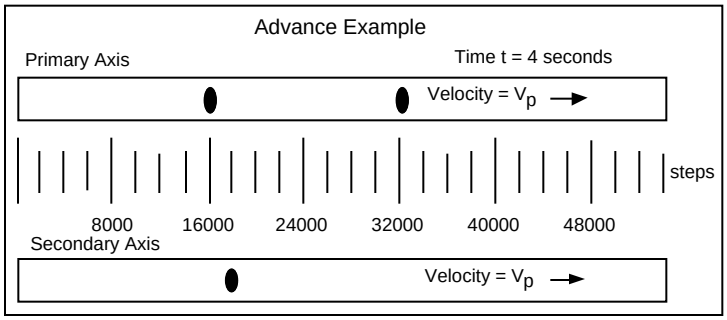
Advance Example—End of Section #2

In Section #3, the advance portion of the secondary profile begins. The secondary axis accelerates from a following percentage of 100% to 200% (a 2:1 ratio). Look at the plot of the profiles to graphically determine the distance that the primary and secondary axes have traveled. The primary axis moves 4000 steps and the secondary axis moves 6000 steps. The secondary axis' distance can also be determined from the equation above.

$$Ds_{acc} = 1 \bullet \left(\frac{1}{2} \bullet 100^2 \bullet \frac{4}{100 \bullet 0.1} + 100 \bullet 100 \bullet \frac{4}{100 \bullet 0.1} \right) = 6000 \text{ steps}$$

At the end of Section # 3, the primary axis has moved 4000 steps since the beginning of the advance portion of the profile. The secondary axis has advanced 2000 steps with respect to the primary axis.

☞ ***Helpful Hint:***
This figure shows the location of the spots after Section #3.



Advance Example—End of Section #3

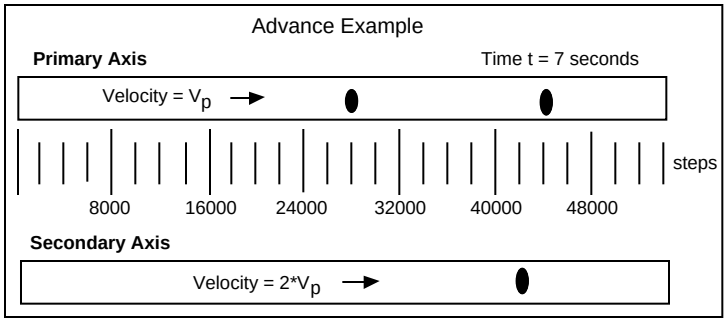
At the start of Section #4, the secondary axis is at a following percentage of 200% and is moving at 2 • Vp (a 2:1 ratio). Section #4 lasts 3 seconds. The primary axis moves 12000 steps, while the secondary axis moves 24000 steps. The distance the secondary axis traveled can be determined from the equation below.

Ds_{con} Equation

$$Ds_{con} = \mathbf{FOR} \bullet \frac{\mathbf{FOL}}{100} \bullet D_{prim} = 1 \bullet \frac{200}{100} \bullet 12000 = 24000$$

Since the start of the advance portion, the primary axis has moved a total of 16000 steps and the secondary axis has moved 30000 steps.

☞ ***Helpful Hint:***
The secondary axis has advanced 14000 steps with respect to the primary axis. This figure shows the location of the spots after Section #4.



Advance Example—End of Section #4

During Section #5, the secondary axis decelerates to a following percentage of 100% (a 1:1 ratio). After it decelerates, it will have completed the advance portion of the profile. The primary axis travels 4000 steps in Section #5. The secondary axis travels 6000 steps. When the secondary axis accelerates from one following percentage to another, then decelerates to the original following percentage (as in this example), ***the secondary axis' acceleration distance will always equal the deceleration distance***. However, the deceleration

distance can also be calculated from the following equation.

Ds_{dec} Equation

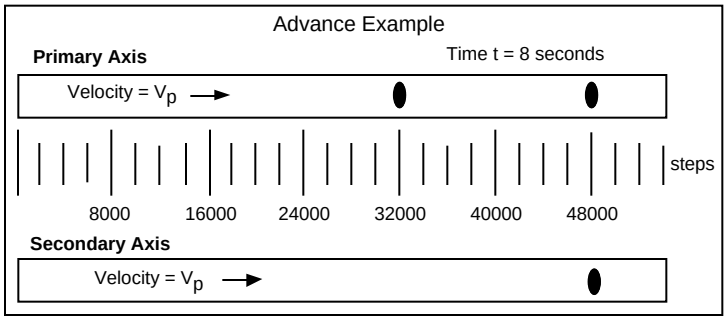
$$Ds_{dec} = FOR \cdot \left(\frac{1}{2} \cdot \Delta FOL^2 \cdot \frac{FEN}{100 \cdot FAC} + FOL \cdot FOL_I \cdot \frac{FEN}{100 \cdot FAC} \right)$$

When decelerating, the initial following percentage is the percentage you are at when you begin deceleration. In this example, it is 200%. The final following percentage is 100%. Therefore, the change in following percentage is a negative number (-100%).

$$Ds_{dec} = 1 \cdot \left(\frac{1}{2} \cdot (-100)^2 \cdot \frac{4}{100 \cdot 0.1} + -100 \cdot 200 \cdot \frac{4}{100 \cdot 0.1} \right) = -2000 + 8000 = 6000 \text{ steps}$$

At the end of Section #5, the advance portion of the move profile is complete. The secondary axis is moving at a 1:1 speed ratio with the primary axis. The secondary axis has moved 36000 steps and the primary axis has moved 20000 steps during the advance portion. The secondary axis has advanced 16000 steps with respect to the primary axis.

☞ ***Helpful Hint:***
This figure shows the locations of the axes after Section #5.



Advance Example—End of Section #5

During Section #5, the secondary axis travels at a 1:1 ratio until the deceleration ramp begins. If the move is preset, it will begin the deceleration ramp as defined by the **AD** command. It will be at rest at the exact distance of the preset move. If the move is continuous, it will decelerate according to the **AD** value when a Stop (**S**) or Kill (**K**) command is reached. From within a sequence, use the buffered Stop command. If you want the deceleration ramp to use the following acceleration value to decelerate, use **FOLØ**.

After the secondary axis rests, a Stop command must be used to terminate the move. In the move profiles, the secondary axis moves at a 1:1 ratio for 8000 more steps, then decelerates to zero. During this time, the primary axis also moves 8000 steps. Programming this profile requires the Following Encoder Distance Point (**FP**) or Following Encoder Absolute Point (**FPA**) command. **FP** is a delay-based on an incremental encoder distance. **FPA** is a delay-based on an absolute encoder distance.

FP Delay Examples

To perform the advance move profile, breakpoints are needed to indicate when the secondary axis should accelerate to new following percentages. The **FP** and **FPA** commands define these breakpoint. These changes are performed on-the-fly and require the Motion Profiling mode. The breakpoints are the points at which acceleration or deceleration begin. This example will show the program using the **FP** command.

Step ①

Set up the ZXF with the proper encoder interface and enable Following mode.

Command	Description
> FSI1	Enables ZXF's Following mode

Step ②

Set up the velocity following portion of the application. The number of secondary motor steps per unit of travel is 4000. The number of primary encoder steps per unit of travel is also 4000. Therefore, the **FOR** value is 1. The **FOL** command will be set up in the sequence for running the profile.

Command	Description
> FOR1	Relates the number of secondary motor steps for a distance to the number of primary encoder pulses for the same distance

Step ③

Set up the following acceleration value and enable Following Acceleration mode. The values are taken from the previous example.

Command	Description
> FAC.1	Increases the following percentage to 0.1 for every change in encoder pulses by FEN
> FEN4	Sets the number of encoder pulses required before the following percentage is incremented by FAC.
> FSF1	Enables the use of following acceleration

Step ④

The breakpoints in this profile occur at the end of Section #2, and the end of Section #4. *If **FAC** and **FEN** are used for deceleration, a breakpoint is also set at the end of Section #6.* If your **FOR** command differs from this example, the distance command is entered in terms of your secondary motor. The following sequence performs the desired profile. Each step of the sequence is explained. Enter the sequence.

Command	Description
> XE1	Erases Sequence #1
> XD1	Begins the definition of Sequence #1
D56000	The total distance the secondary moves is 56000 steps.
FOL100	The first following percentage to accelerate to is 100%.
MPP	Enters Motion Profiling mode so changes can be made on the fly.
G	The secondary motion begins.
FP12000	The first breakpoint occurs after the **primary axis** moves 12000 steps. **FP** causes command processing in the sequence to delay until 12000 primary encoder steps have been counted.
FOL200	After 12000 primary encoder steps the following percentage is changed to 200%. The **secondary axis** begins to accelerate at the following acceleration. Distance is known from previous analysis.
FP16000	Right after the command to begin acceleration to an **FOL** of 200%, the command processing is delayed until the **primary axis** has moved 16000 more pulses (from when **FP** is encountered and thus is an incremental encoder distance).
FOL100	After 16000 more primary encoder pulses, the following percentage changes to 100% and the **secondary axis** decelerates to a 1:1 ratio.
FP12000	Command processing is delayed 12000 more steps.
FOL0	The secondary axis decelerates to an FOL of 0.
FP4000	It takes 4000 primary encoder steps to decelerate
STOP	A Stop command is needed because the ZXF would still think it was in a move and that the current following ratio was set to zero.
NG	Exits the Motion Profiling mode.
XT	Ends the definition of Sequence #1

Step ⑤

In applications that require velocity and position following, the method of decelerating to a stop does not matter (moving from a synchronized state to rest). At the end of such a move, you must reverse direction and return to the starting location to repeat the profile. The optimal profile is as follows:

① Accelerate to a known positional relationship

② Perform the operation required at the synchronized speed

③ When it is complete, decelerate as fast as possible to repeat the cycle

You must decelerate at a rate unrelated to the primary encoder speed. The ZXF allows you to do this. This will also simplify the programming. The same profile is programmed below using the **AD** deceleration value.

Command	Description
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
D56000	The total distance the secondary moves is 56000 steps
FOL100	The first following percentage to accelerate to is 100%
MPP	Enters Motion Profiling mode so changes can be made on the fly
G	The secondary axis' motion begins
FP12000	Delays processing until primary axis moves 12000 encoder pulses
FOL200	Change to 200% following percentage
FPA16000	Delays processing until primary axis moves 16000 encoder pulses
FOL100	Change ratio back to 100%
NG	Exits the profiling mode
XT	Ends Sequence #1 definition

In this sequence, only two breakpoints are needed, the breakpoint to accelerate to 200% and then to decelerate back to 100%. Since the ZXF will decelerate at the value in **AD**, it will automatically decelerate to a distance of exactly 56000 secondary motor steps at the appropriate time.

Step ⑥

This step uses **FPA** instead of **FP**. **FPA** delays processing until the absolute value of the following encoder counter exceeds the **FPA** value.

Command	Description
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
D56000	The total distance the secondary moves is 56000 steps.
PFZ	Zero the following encoder counter.
FOL100	The first following percentage to accelerate to is 100%.
MPP	Enters Motion Profiling mode so changes can be made on the fly.
G	The secondary motion begins.
FPA12000	Delay command processing until primary encoder count exceeds 12000 encoder pulses.
FOL200	Change to 200% following percentage.
FP28000	Delay command processing until primary encoder counter exceeds 28000 encoder pulses or an incremental change of 16000 pulses.
FOL100	Change ratio back to 100%.
NG	Exits the profiling mode.
XT	Ends Sequence #1 definition

In Steps ⑤ and ⑥, the encoder is counting in the positive direction. If the encoder is counting in the negative direction, a negative sign is required for the **FP** and **FPA** commands. Use **FP** or **FPA**, depending on your application.

*Use **FPA** when repetitive cycles of the same move profile are done without a trigger to start each cycle*. By making the delays dependent on an absolute encoder position, there is no accumulative error. In many cases, a trigger input from a sensor is used to start the move profile that is repeated. *If you use a trigger to start the move each time, use **FP** and the trigger will remove any accumulative error*. The following sequence illustrates the uses of the **FPA** command and a variable to perform a repetitive move that does not use a trigger to start it. This case is more like a cam cycle and the position relationship must be maintained while the cycle repeats.

Command	Description
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
VAR1=12000	Sets variable 1 equal to the first breakpoint.
VAR2=28000	Sets variable 2 equal to the second breakpoint
VAR3=0	Sets variable 3 equal to the primary reference point
D56000	Sets distance of secondary axis move to 56000 steps
PFZ	Zeroes the following encoder counter
L	Begins the continuous loop of the profile cycle
FOL100	The first following percentage to accelerate to is 100%
MPP	Enters Motion Profiling mode so changes can be made on-the-fly
FPA(VAR3)	Variable 3 synchronizes the move start with the primary axis. For the first 40000 primary steps, the secondary axis is moving, then it moves back during the next 40000 steps of the primary then it repeats.
G	Begins secondary axis motion.
VAR3=VAR3+80000	Sets variable 3 equal to the start of the next cycle
FPA(VAR1)	Delays command processing until primary encoder count exceeds 12000 + 56000*n* encoder pulses, *n* = # of times through the loop.
FOL200	Changes following percentage to 200%.
VAR1=VAR1+80000	Sets variable 1 equal to the first breakpoint for the next cycle
FPA(VAR2)	Delays command processing until primary encoder count exceeds 28000 + 56000*n* encoder pulses, *n* = the # of times through the loop.
FOL100	Changes ratio back to 100%.
VAR2=VAR2+80000	Sets variable 2 equal to the second breakpoint for the next cycle
NG	Exits the profiling mode and complete the 56000 step move
FOL200	Sets the following ratio to a higher speed to move back to the starting point at a fast speed
H	Changes direction
G	Moves back to the starting point.
H	Changes direction
N	Repeats the cycle
XT	Ends Sequence #1 definition

In this example, the move profile is repeated.

☞ ***Helpful Hint:*** One cycle consists of the following events.

① Secondary axis moves 56000 steps while primary axis moves 40000 steps.

② The secondary axis retreats to the start and after another 40000 primary encoder steps the cycle is repeated.

③ No operation during the secondary axis' retreat.

④ The retreat is set to a high following ratio to get the secondary axis back to the start before the primary axis moves 40000 steps.

This cycle is very similar to a cam cycle (described in the next section). You can load the **FP** command with a variable (like **FPA**). You can check the following encoder counter value any time by loading it into a variable.

Command	Description
> VAR1=FEP	Loads variable 1 with the value of the following encoder counter

Calculating FOL, FP, or FPA For An Advance or Recede Application

The advance example explained how an advance move is made and how the different commands (**FOL**$_I$, **FAC**, and **FEN**) contribute to the move. This section provides some simple formulas that you can use to set up such a profile. To do position and velocity following, the secondary axis must be able to accelerate to a known position with respect to the primary axis. This is what determines the **FAC** and **FEN** following acceleration values. Once you determine these values, you will use them to calculate the acceleration ratio that you must use to make the advance move. The following data will help you understand the move profile in the example.

- **FAC**: Following speed percentage increment
- **FEN**: Change in primary encoder pulses to cause an increment of **FAC**
  - **D**$_{prim}$: The distance the primary axis will travel during the advance portion of the secondary move profile. This is 20000 primary encoder steps in the example above.
  - **D**$_{sec}$: The distance the secondary must advance with respect to the moving primary, measured in primary encoder steps. In the example above this is 16000 primary encoder steps.
  - **FOL**$_I$: The initial following percentage that you will accelerate from to the new following percentage.

In applications that require an advance move, you will usually know the distance that you want to advance with respect to the primary axis and the distance the primary axis will move during the advance. The distance that you want the secondary axis to advance with respect to the primary axis is given in terms of primary encoder steps. The distance can be converted from secondary motor steps to primary motor steps (and vice versa) with the **FOR** command. After determining the parameters listed above, you can use the following formula to determine the following percentage you must accelerate to.

First, determine a following acceleration constant (**K**) to simplify the equations.

Following Constant Equation

$$K = \frac{\mathbf{FEN}}{100 \bullet \mathbf{FAC}}$$

The constant **K** is used in the following equation to determine **FOL**.

FOL Equation

$$\mathbf{FOL} = \mathbf{FOL}_1 + \frac{D_{prim}}{200 \bullet K} - \sqrt{\left(\frac{D_{prim}}{200 \bullet K}\right)^2 - \left(\frac{D_{sec}}{K}\right)}$$

With an advance move, the value of **FOL**I will always be 100. Apply the formula to the example above (the following percentage should be 200%). This is the following percentage that you must attain to advance 16000 steps with respect to the primary axis, while the primary axis moves 20000 steps.

D_{prim} = 20000 primary encoder steps

D_{sec} = 16000 primary encoder steps

FAC = 0.1

FEN = 4

$\mathbf{FOL}_I$ = 100

We will first determine the following acceleration constant K.

$$K = \frac{4}{100 \bullet 0.1} = 0.4$$

We will now determine **FOL**.

FOL is the same as $\mathbf{FOL}_F$ in the equations used earlier for determining the distances traveled by the primary and secondary axes.

$$\mathbf{FOL} = 100 + \frac{20000}{200 \bullet 0.4} - \sqrt{\left(\frac{20000}{200 \bullet 0.4}\right)^2 - \left(\frac{16000}{0.4}\right)}$$

$$= 100 + 250 - \sqrt{(250)^2 - 40000} = 350 - \sqrt{22500} = 350 - 150 = 200 = \mathbf{FOL}_F$$

The following percentage that must be accelerated to is 200%. Now calculate what the breakpoint is for decelerating back to a 1:1 ratio or 100%. The **FP** value will be determined from the following equation:

Following Breakpoint Equation

$$\mathbf{FP} = D_{prim} - \frac{\mathbf{FEN}}{\mathbf{FAC}} \bullet (\mathbf{FOL}_F - \mathbf{FOL}_I) = 20000 - \frac{4}{0.1} \bullet (200 - 100) = 16000 \text{ steps}$$

The value you would use for **FP** is 16000. The breakpoint at which you begin the advance portion of the move profile was not calculated. This value varies from application to application and you may want to use a trigger to begin the advance move rather than **FP**. An example of using a trigger to begin an advance move is described below.

Step ①

Configure an input as a trigger input with the **IN** command.

Command	Description
> IN1A	Configures input #1 as a trigger input
> IN2D	Configures input #2 as a stop input
> XD1	Defines Sequence #1
MC	We will make this a continuous move
FOL100	The first following percentage to accelerate to is 100%
MPP	Enters Profiling mode so changes can be made on-the-fly
G	Initiates motion
TR1	Command processing pauses until input #1 (trigger input) is activated. The secondary axis will move continuously at a speed percentage of 100% (with respect to the primary axis)
FOL200	The following percentage is changed to 200%. Acceleration begins.
FP160000	Command processing will delay 16000 primary encoder steps.
FOL100	The following percentage is changed to 100%. Deceleration begins.
NG	Ends the Profiling mode.
XT	End Sequence #1 definition

In this sequence, the secondary axis begins moving at a 100% speed percentage. When trigger input #1 is activated, the secondary axis will advance 16000 steps with respect to the primary axis. It will then decelerate to a 100% or 1:1 ratio and continue until the stop input (input

#2), is activated (or a Stop [**S**] command is issued).

Recede vs. Advance

Recede moves are similar to advance moves. In the illustration of the spots for the advance example, the secondary axis synchronized with the first primary axis, then receded while the primary axis moved. This motion can be analyzed in the same manner as the advance move with the exception that a different equation is used to determine the required value for **FOL**. Again, you will have to provide the distance that the primary axis will move during the recede move and the distance with respect to the primary axis that the secondary axis must recede (measured in terms of primary encoder steps). For example:

- ❏ A primary encoder has a resolution of 4000 steps/rev (1 rev = 1 inch)
- ❏ The secondary motor has 25000 steps/rev (1 rev = 1 inch)
- ❏ The **FOR** command is set to 6.25.

You will usually know what distance you want the secondary axis to recede. If the secondary is to recede 1.5 inches with respect to the primary axis while the primary axis moves 3.5 inches, set D_{prim} and D_{sec} equal to:

$$D_{prim} = 3.5" = 3.5" \bullet 4000 \frac{steps}{inch} = 14000 \text{ primary steps}$$

$$D_{sec} = 1.5" = 1.5" \bullet 25000 \frac{steps}{inch} = 37500 \text{ secondary mtr steps} = \frac{37500}{6.25} = 6000 \text{ primary enc steps}$$

Both distances are provided in primary encoder steps. The two terms are used to determine the required **FOL** in the following Equation.

FOL Equation

$$FOL = FOL_I - \frac{D_{prim}}{200 \bullet K} + \sqrt{\left(\frac{D_{prim}}{200 \bullet K}\right)^2 - \frac{D_{sec}}{K}}$$

Where K is the following acceleration constant.

$$K = \frac{FEN}{100 \bullet FAC}$$

To calculate the **FOL** value, determine a value for **FAC** and **FEN**. This will depend on your application's maximum velocity and either the maximum acceleration for the secondary axis or the distance the primary axis travels while the secondary axis must accelerate. Use the same **FAC** and **FEN** values from the advance example.

FAC = 0.1

FEN= 4

In an advance or recede application, the initial following percentage FOL_I will always be 100.

☞ ***Helpful Hint:*** The value for FOL produced by the FOL equation is:

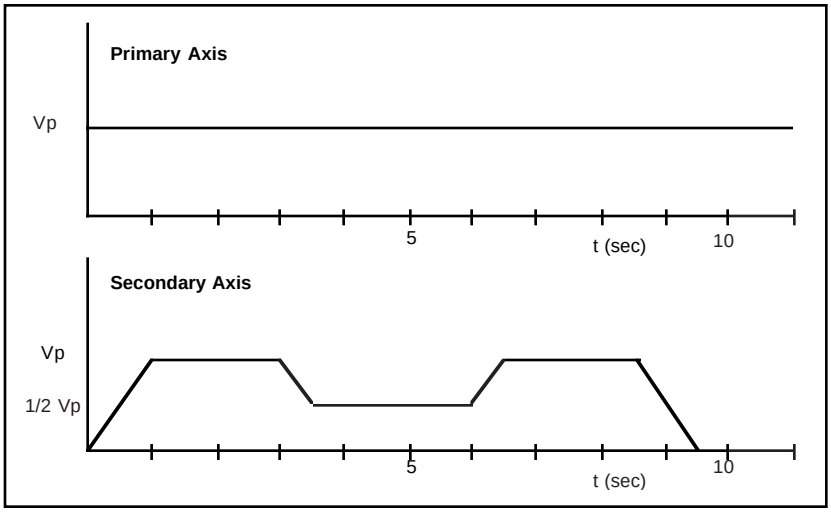
$$K = \frac{4}{100 \bullet 0.1} = 0.4$$

$$FOL = 100 - \frac{14000}{200 \bullet 0.4} + \sqrt{\left(\frac{14000}{200 \bullet 0.4}\right)^2 - \frac{6000}{0.4}} = 100 - 175 + \sqrt{(175)^2 - 15000} = 50\%$$

The value needed for **FP** can be determined from the following equation.

FP Equation

$$FP = D_{prim} + \frac{FEN}{FAC} \bullet (FOL_F - FOL_I) = 14000 + \frac{4}{0.1} \bullet (50 - 100) = 12000 \text{ primary enc. steps}$$



Recede While Following Profile

The sequence that will execute this profile is provided below:

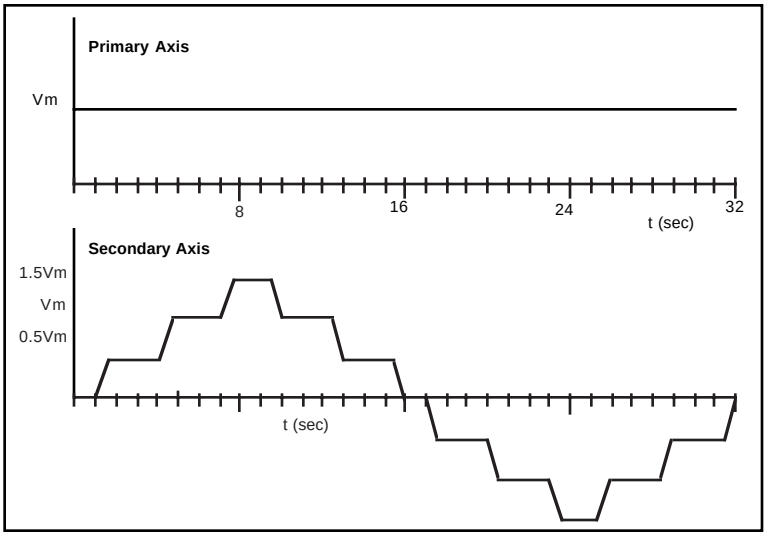
Command	Description
> FAC0.1	Sets the percentage increment to 0.1
> FEN4	Sets the number of encoder counts for an increment to 4
> FSF1	Enables following acceleration
> FOR6.25	Sets the secondary motor steps per unit distance to primary encoder steps per unit distance ratio
> XD1	Defines Sequence #1
MN	Normal mode
D175000	Sets distance to 175000 steps
FOL100	The first following percentage to accelerate to is 100%
MPP	Enters Motion Profiling mode so changes can be made on-the-fly
G	Initiates motion
FP12000	The command processing will pause here until the primary encoder has moved 12000 steps. The secondary will then decelerate to 50%
FOL50	Changes following percentage to 50%—the recede portion begins
FP12000	Command processing will delay 12000 primary encoder steps.
FOL100	The following percentage is changed to 100%—deceleration begins
NG	Ends the Motion Profiling mode
XT	End sequence definition

Cam Following

A common application that requires velocity and position following is the simulation of a cam or an electronic cam. To simulate the motion produced by a cam, you must satisfy the following requirements:

- ❏ Follow both the position and the velocity of a primary encoder.
- ❏ You must also be able to change following ratios during motion and still maintain a positional relationship.
- ❏ Change ratios based on primary encoder distance.
- ❏ Must be able to keep track of the primary encoder position even if the secondary axis is not moving.
- ❏ Must be able to continuously repeat a cam cycle without developing accumulative error.

You can simulate a cam profile electronically using the commands and equations developed earlier. Motion Profiling mode (**MPP**) is required for cam following. For more information on Motion Profiling mode, refer to *Chapter ④ Application Design*.



Cam Following

In this example, the encoder is a 4000 pulse per revolution encoder and it is mounted on the primary axis. The *secondary axis* will perform an electronic cam cycle, which consists of the following steps:

① Sitting at rest for one primary encoder revolution, then executing a step profile of 3 following percentages from 50% to 100%, then to 150%, and back to 0%.

② It will then delay 1 primary encoder revolution and perform the same profile in the opposite direction.

③ This cycle is to be repeated until a stop is issued. The secondary axis will accelerate at its maximum rate when the primary axis is at its maximum velocity. The table below defines the cam cycle for this profile.

Segment	Primary Position		Secondary Position		Following %
	Absolute	Incremental	Absolute	Incremental	
①	0	0	0	0	0%
②	4000	0	0	0	50%
③	16000	12000	5500	5500	100%
④	28000	12000	17000	11500	150%
⑤	38000	10000	31500	14500	100%
⑥	50000	12000	44000	12500	50%
⑦	62000	12000	50500	6500	0%
	68000	6000	51000	500	

Cam Cycle

The following percentage is given for each segment. Each of the distance points is a breakpoint where the following percentage changes. Use the following acceleration to change from one following percentage to the next. The data needed to program this type of profile is listed below.

Vp_{max} : Maximum velocity of the primary axis

As_{max} : Maximum acceleration of the secondary axis

FOR: Relative resolutions per unit of distance for the primary and secondary axes

FAC: Following acceleration value, percentage increments

FEN: Following acceleration value, encoder counts for an increment

For each ratio segment, you must know the distance the primary axis will travel and the corresponding distance the secondary axis will travel. The breakpoints can be determined from the graph. Usually, you will not be able to graphically describe the motion relationship, and will simply know that you want the primary axis to move ***x*** steps and the secondary axis to move ***y*** steps in the same time frame.

After you determine **FAC** and **FEN** from your maximum acceleration and maximum velocity, or from the distance you want the secondary axis to accelerate over and the maximum velocity, you can use **FAC** and **FEN** and the primary and secondary axes' travel distances to determine the following percentages and the values for the breakpoints to change to new following percentages. When accelerating to a higher following percentage, you can use the **FOL ACCEL** Equation to determine the required **FOL** percentage that you must accelerate to using the **FAC** and **FEN** values you have determined.

FOL ACCEL Equation

$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{100 \cdot K} - \sqrt{\left(\frac{D_{prim}}{100 \cdot K}\right)^2 + \frac{2 \cdot \mathbf{FOL}_I \cdot D_{prim}}{100 \cdot K} - \frac{2 \cdot D_{sec}}{\mathbf{FOR} \cdot K}}$$

Where K is the following acceleration constant determined by **FAC** and **FEN**.

$$K = \frac{\mathbf{FEN}}{100 \cdot \mathbf{FAC}}$$

If you are decelerating to a lower following percentage, use the **FOL DECEL** Equation to determine the following percentage you must use to move the secondary axis the specified number of steps for the corresponding motor steps.

FOL DECEL Equation

$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{100 \cdot K} + \sqrt{\left(\frac{D_{prim}}{100 \cdot K}\right)^2 - \frac{2 \cdot \mathbf{FOL}_I \cdot D_{prim}}{100 \cdot K} + \frac{2 \cdot D_{sec}}{\mathbf{FOR} \cdot K}}$$

In both the accelerating and decelerating equations, the terms listed below for the primary encoder steps and the secondary motor steps are required.

D_{prim} = The number of primary encoder steps that the motor will move in the segment.

D_{sec} = The number of corresponding secondary motor steps that the secondary will move during which time the primary encoder will move D_{prim}.

To illustrate the programming of the profile above, we will assign values to the application's requirements. The maximum velocity, acceleration and the **FOR** value are to be determined by the application. The **FAC** and **FEN** values are calculated. The maximum velocity and acceleration are given below. The example below has the following parameters:

- ❏ Primary encoder resolution = 4000 steps/revolution
- ❏ Secondary motor resolution = 25000 steps/revolution
- ❏ 1 encoder rev = 1 motor rev
- ❏ **FOR** 6.25
- ❏ Vp_{max} = 4000 steps/second
- ❏ As_{max} = 1 rev/second ^{2}
- ❏ Encoder sample period TF= 1 ms

From the equations in the Velocity and Position following section **FAC** and **FEN** are determined:

$$\mathbf{FEN} = Vp_{max} \cdot \frac{\mathbf{TF}}{1000}$$

$$\mathbf{FAC} = \frac{\mathbf{FOL} \cdot Vp_{max}}{Dp_{acc}} \cdot \frac{\mathbf{TF}}{1000}$$

Since we are starting with Vp_{max} and As_{max}, we must determine D_{acc} for the equations above.

$$Dp_{acc} = \frac{Vp_{max}^2 \cdot \mathbf{FOR} \cdot \mathbf{FOL}}{As_{max} \cdot 100} = \frac{\left(4000 \frac{steps}{sec}\right)^2 \cdot 6.25 \cdot 100}{25000 \frac{steps}{sec^2} \cdot 100} = 4000 \text{ primary encdr steps}$$

Vp_{max} is in primary enc. steps/sec

As_{max} is in secondary motor steps/sec^2

The **FOR** term converts the acceleration units to primary encoder steps units. **FAC** and **FEN** can now be calculated.

$$\mathbf{FEN} = 4000 \frac{steps}{sec} \cdot \frac{1 \text{ ms}}{1000} = 4 \text{ steps}$$

$$\mathbf{FAC} = \frac{100 \cdot 4000 \frac{steps}{sec}}{4000 \text{ steps}} \cdot \frac{1 \text{ ms}}{1000} = 0.1$$

You can now use these equations to determine the **FOL** value for each segment of primary encoder distance and secondary motor distance. The **FOL** values are already given in the previous table, but it may be necessary to determine the **FOL** required for some of the segments. Evaluate **FOL** for Segments #4 and #6. Typically, you will know the distance you want the primary axis to move and the corresponding distance that you want the secondary axis to move. With this data, you can create a table like the one shown. You will have to enter the **FOL** values for the table from the equations given in this chapter.

Segment ④

D_{prim} = 10000 primary encoder steps

D_{sec} = 90625 secondary motor steps

Using the acceleration equation, we can calculate the value for **FOL** in Segment #4.

The acceleration constant $K = \frac{4}{100 \cdot 0.1} = 0.4$

$$\mathbf{FOL} = 100 + \frac{10000}{100 \cdot 0.4} - \sqrt{\frac{2 \cdot 100 \cdot 10000}{100 \cdot 0.4} + \left(\frac{10000}{100 \cdot 0.4}\right)^2 - \frac{2 \cdot 90625}{6.25 \cdot 0.4}}$$

$$= 100 + 250 - \sqrt{50000 + (250)^2 - 72500} = 150\%$$

The breakpoint is given automatically by the table and is 10000 for FP and is 38000n for **FPA** where n is the number of cycles completed thus far.

Segment ⑥

D_{prim} = 12000 primary encoder steps

D_{sec} = 40625 secondary motor steps

Using the deceleration equation, we can calculate the value for **FOL** in Segment #6.

$$\mathbf{FOL} = 100 - \frac{12000}{100 \cdot 0.4} + \sqrt{\left(\frac{12000}{100 \cdot 0.4}\right)^2 - \frac{2 \cdot 12000 \cdot 100}{100 \cdot 0.4} + \frac{2 \cdot 40625}{6.25 \cdot 0.4}}$$

$$= 100 - 300 + \sqrt{(300)^2 - 60000 + 32500} = 50\%$$

The breakpoints for **FP** and **FPA** are 12000 and 62000***n*** steps respectively. In many cases involving a cam cycle, a trigger is not used to start each cycle and the repetition of the cycle is based on the primary encoder. In these situations, use the **FPA** command (it is based on the following encoder's absolute count). The absolute count comes from a hardware counter that can be accessed by assigning it to a variable:

> VAR1 = FEP **FEP** is the value in the hardware counter, it is a read only value

The following sequence will perform the cam profile. The secondary axis will be put in a continuous move. Two parts occur in a cycle. The first part moves the stepped profile in one direction. The second part reverses direction and returns to the start to repeat the cycle.

Command	Description
> FOR6.25	Sets the secondary motor steps to primary encoder steps ratio
> FAC0.1	Sets the change in following percentage for following acceleration
> FEN4	Sets the number of encoder pulses required to change by **FAC**
> FSF1	Enables following acceleration
> FSI1	Enables following
> SSH1	Saves buffer on stop
> VAR1=64000	Variable for the incrementing the cycle
> VAR2=4000	Breakpoint 1
> VAR3=16000	Breakpoint 2
> VAR4=28000	Breakpoint 3
> VAR5=38000	Breakpoint 4
> VAR6=50000	Breakpoint 5
> VAR7=62000	Breakpoint 6
> VAR8=64000	Breakpoint 7
> 1XE1	Erases Sequence #1
> 1XD1	Defines Sequence #1
MC	Enables continuous mode
FOLØ	Sets the current following percentage to 0%
L	Begins the loop cycle
MPP	Enters the Motion Profiling Mode
G	Initiates motion
FPA(VAR2)	Pauses execution until absolute primary encoder counter value exceeds breakpoint 1
FOL5Ø	Following % is changed to 50% or 1/2 as fast as the primary motor
VAR2=VAR2+VAR1	Set variable 2 to the breakpoint value for the next cycle
FPA(VAR3)	Pauses execution until primary encoder counter absolute value exceeds breakpoint 2
FOL1ØØ	Speed ratio is changed to 1:1
VAR3=VAR3+VAR1	Set variable 3 to the breakpoint value for the next cycle
FPA(VAR4)	Pauses execution until primary encoder counter absolute value exceeds breakpoint 3
FOL15Ø	Speed ratio is changed to 1.5:1
VAR4=VAR4+VAR1	Set variable 4 to the breakpoint value for the next cycle
FPA(VAR5)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 4
FOL1ØØ	Speed ratio is changed to 1:1
VAR5=VAR5+VAR1	Set variable 5 to the breakpoint value for the next cycle
FPA(VAR6)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 5
FOL5Ø	Speed ratio is changed to 5:1
VAR6=VAR6+VAR1	Set variable 6 to the breakpoint value for the next cycle
FPA(VAR7)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 6
FOLØ	Speed ratio is changed to 0
VAR7=VAR7+VAR1	Set variable 7 to the breakpoint value for the next cycle
FPA(VAR8)	Pauses execution the absolute value of the primary encoder counter exceeds breakpoint 7
STOP	Ends the move (this is required)
NG	Ends Motion Profiling mode
VAR8=VAR8+VAR1	Set variable 8 to the breakpoint value for the next cycle
H	change the direction
N	Ends the loop cycle
XT	Ends Sequence #1 definition

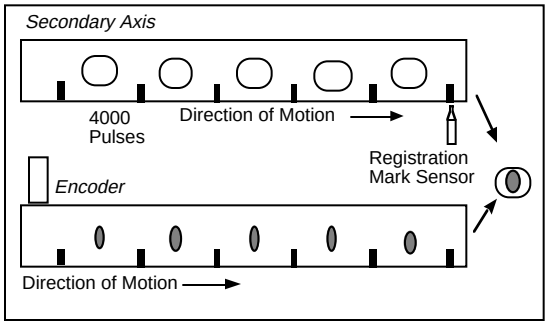
This sequence is an example of a complex following profile. Position and velocity are synchronized and the positional relationship is maintained.

Synchronization

The ZXF can synchronize its speed and phase with respect to a primary axis. In many applications, it is necessary to have the position and speed of a secondary axis synchronized with the speed and position of the primary axis with registration marks on the secondary axis parts or material. These marks must be evenly spaced so that at a constant speed (with respect to the primary axis) the number of primary axis encoder steps recorded between registration marks is an expected constant number. If these marks should come further apart (e.g., the material stretches) the ZXF will adjust the speed ratio to correct for the error between the registration marks. The synchronization figures illustrate this process.

In the figure below, a secondary axis has parts that are to be synchronized to the primary axis' parts. The registration sensor detects the location of the parts with respect to the primary axis. This sensor goes to the ZXF. It indicates the start of the part.

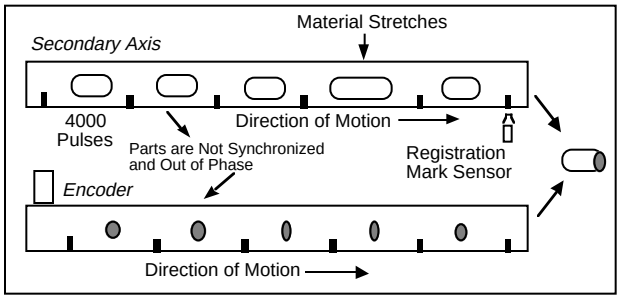
☛ Helpful Hint:
The ZXF then counts the encoder pulses from the primary axis that occur between registration marks.



Secondary Parts Synchronized With the Primary Axis

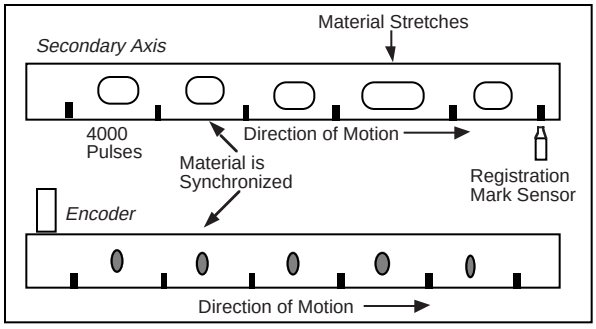
If the material on the secondary axis stretches, as indicated in the figure below, all parts after the stretched material will no longer be synchronized with the primary axis' parts.

☛ Helpful Hint:
The figure below depicts the result of not using Synchronization mode.



Material Stretches—Parts Out of Sync

With the ZXF's Synchronization mode, the secondary axis accelerates to re-synchronize with the primary axis. It removes the phase shift between the two axes. The ZXF detects that the number of pulses between the registration marks has increased (due to stretching). The speed ratio is increased, so the secondary axis speeds up. The material after the stretched portion is good material, so the pulses between the next two registration marks will be slightly less than 4000 because the speed ratio is now higher. The speed will now be reduced. The secondary and primary axes will be synchronized again. Every time a registration mark is encountered, a new actual count is latched and the speed ratio is adjusted to synchronize the axis. Corrections will continue until the secondary axis again has the expected number of pulses between registration marks.



ZXF Synchronizes Parts (After Stretching)

You can determine and program the amount of correction that is applied to the speed ratio between each registration mark to fit your application. Use the following commands to program a synchronization application.

Command	Description
> FC	Expected encoder count between Registration marks
> FBS	Normalizing count for determining new speed ratio to move at
> FIN	Increment used to determine the new following speed
> FSL	Enables the Synchronization mode
> FSK	Enables the expected encoder count teach mode

> INnI	Defines an input as the Registration mark synchronization input
> TF	Sets the sample rate of the encoder input
> FOR	Motor to encoder count ratio
> FOL	Primary to Secondary Speed ratio
> FSI	Enables following mode

The **FOL** and **FOR** commands determine the number of secondary motor steps that will be commanded for the encoder pulses that are received. The **FBS** and **FIN** commands determine the amount of correction that will be applied to the **FOL** and **FOR** motor to encoder ratio. **FC** is the number of encoder pulses that are expected between the registration marks if the speeds are synchronized properly. This number is compared to the actual number of pulses that are counted between each registration mark. The difference between these two values represents the error. The **FIN** and **FBS** numbers are applied to this error to determine the new speed ratio.

The **INnI** command configures an input to the ZXF for accepting the signal from the sensor that detects the registration marks on the parts or material. The **TF** command is the rate at which the encoder interface is sampled. The **FSL** command enables the Synchronization mode. When Synchronization mode is enabled the first time, the Synchronization input is toggled, the encoder counter is started. The next time it is toggled, the count is latched and the counter is reset to zero. The latched count is the actual number of encoder pulses counted between registration inputs. This is compared to the expected value (**FC**). The difference is multiplied by the correction factor (**FIN** and **FBS**) to determine the speed ratio to run at until the synchronization input is toggled again. This process is continuous as long as the synchronization mode (**FSL**) is enabled.

If you do not know the expected encoder count between registration marks, use the Self Learn mode (**FSK**) to determine the expected count (**FC**). To determine the expected count using Self Learn mode, you must start the process at the speed ratio that you want to run at, turn on Self Learn mode. The ZXF will count the pulses between the registration marks. When the secondary axis stops, the last recorded number will be placed in the expected count (**FC**) number. This will be used when you are in Synchronization mode.

As an example, the process in the previous figures use an encoder that has 4000 pulses per revolution and the secondary motor has 25000 steps per revolution. The **FOR** command is set to 6.25. The motor is mounted on the secondary conveyer belt so that one revolution is 4 inches. The encoder is mounted on the primary axis so that one encoder revolution is 4 inches. The **FOL** command must be set to 100% (**FOL100**) for the secondary axis to move at the same speed as the primary axis. If the primary axis moves at 1 rps, the secondary axis must move at 1 rps. In the first synchronization figure, the two axes start moving at the same speed. The registration marks are 4 inches apart on the secondary axis. Use the Self Learn mode to determine how many encoder pulses are between the registration marks on the material on the secondary axis.

☞ ***Helpful Hint:*** The following steps show how to program the sample registration application.

① Set up the Self-Learn mode.

Command	Description
> **FOR6.25**	Set the motor to encoder ratio to 6.25
> **FOL100**	Set the motor to encoder ratio speed percentage to 100%
> **FSI1**	Enable the Following mode
> **FSK1**	Enable the Self Learn mode
> **TF1**	Set the encoder sample rate to 1 ms
> **A500**	Set the acceleration of 500 rps^2
> **AD500**	Set the deceleration of 500 rps^2
> **MC**	Place the ZXF in the continuous mode
> **IN1I**	Defines input number 1 as a synchronization input

② The registration sensor should be wired to the synchronization input. The primary and secondary axes should now be started. After the ZXF has passed more than 3 registration marks, it can be stopped. The number of encoder pulses between registration marks can be checked with the **FC** command. In this example, the number that is determined is 4000 counts.

Command	Response
> **1FC**	*4000

If you know the number of encoder pulses you expect to record between registration marks, this number can be entered directly for the **FC** command and will override the number determined in the self learn mode.

Command	Description
> **1FC4000**	Manually entering the expected count

③ The ZXF now has the number of counts expected between registration marks. The next step is to determine the correction gain desired. The correction will be applied to the difference between the expected count that was just determined and the actual counts that will be counted during actual operation.

Correction Equation

$$\text{Correction} = \frac{(\text{Actual encoder count} - \text{Expected encoder count}) \bullet \textbf{FIN}}{\textbf{FBS}}$$

This motor-to-encoder step ratio is determined by the following equation.

Speed Ratio Equation

New Speed Ratio = **FOR** • **FOL** + Correction

To determine the number of motor steps that will be commanded for the number of encoder pulses received, use the following equation.

Motor Step Correction Equation

Motor Steps = Encoder Steps • (**FOR** • **FOL** + Correction)

You must determine the amount of correction you want to have for a given amount of error. Once this has been determined, you can enter the **FIN** and **FOR** commands.

Command	Description
> **1FIN3.12**	The following increment is 3.12
> **1FBS100**	The following base number is 100

④ Disable Self Learn mode and enable Synchronization mode.

Command	Description
> **1FSK0**	Disable Self Learn mode
> **1FSL1**	Enable Synchronization mode

⑤ Orient the primary and secondary axes to attain the desired phase relationship. Start both axes at once, or start the ZXF first and then the primary axis.

The ZXF will correct any errors in the phase relationship between the two axes and maintain a synchronized speed. The new speed ratio will be applied for the entire period between registration marks. The time between the registration marks is effectively the sample period. A correction is made for each sample period.

Another method for synchronization is to use the inputs to the ZXF for increasing and decreasing the following speed ratio. The ZXF inputs can be defined to increase or decrease the following ratio. By setting one input for increasing the following ratio and one input for decreasing the following ratio, synchronization can be achieved. In this case, use an external circuit to determine whether the secondary axis should accelerate (increase ratio) or decelerate (decrease ratio) the secondary axis.

Define the input with the **IN** command. **INnX** defines the input for increasing following ratio, **INnY** defines the input for decreasing the following ratio. The following ratio will be increased/decreased while the input is active. During the ZXF's sample periods, the ratio will increase or decrease while the inputs are active. The inputs have a 2 ms debounce time. If the input is active for 4 ms, the following ratio will be increased/decreased twice. The amount that the following ratio is increased/decreased is determined by the **FIN** command. If **FIN** is 1, **FOL** is increased/decreased by 1 during each sample period.

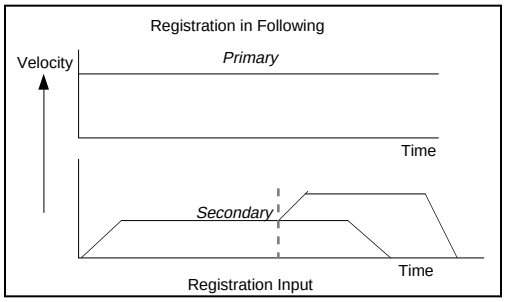
Other Following Features

This section discusses following features that the ZXF provides for special following requirements.

Registration in Following Mode

With the ZXF, registration can be performed in Following mode. It is like programming registration in the indexer version, *but the velocity term is replaced by* **FOL** *for the desired speed*. The graph below illustrates registration in the Following mode.

☞ **Helpful Hint:** Registration can only be performed if Following Synchronized Acceleration mode is disabled (FSFØ). For this example, re-attach the encoder to the primary axis.



Registration in Following Mode

① Repeat steps ① - ⑤ of the Velocity Following Example.

② Set up input #7 as a registration input

Command	Description
> IN7Q	Defines input #7 as a registration input

③ Define registration move #1 as follows:

> REG1,A1Ø,AD1Ø,FOL2ØØ,D25ØØØ

④ Begin motion on the primary axis, then begin motion with the ZXF.

Command	Description
> MC	Change to continuous mode
> G	Initiate motion

⑤ Toggle input #7. The motor will begin following at a speed ratio of 2:1 for 25,000 motor steps.

Jogging in Following Mode

In some applications, you may want to move the motor manually while in Following mode. This allows you to follow the primary axis at a following ratio by toggling a switch. You can configure the ZXF to allow you to follow the primary motor manually with the **IN** command. Jogging in Following mode does not require the **JHV** or **JVL** commands. In this case, you will jog at the **FOL** command's speed ratio. To use the inputs, you can either configure the input as a CW or CCW jog as with the preset indexer jog. However, you cannot use the high-speed/low-speed jog input because you can only jog at the **FOL**'s speed ratio.

Therefore, you use the two jog input functions: CW Jog input (**IN#J**) and CCW Jog Input (**IN#K**). You must also enable the jogging feature with the **OSE1** command. Once you set these parameters, you can attach a switch to the jog inputs (predefined) and jog the motor(s). The **#** character represents digits 1 - 13, which you enter. You must have the ZXF in Following mode to jog at a speed ratio of the primary. The following example shows you how to define power-up sequence #100 to jog.

① Define a power-up sequence. Position Tracking mode must be disabled (FSPØ) to enable direction jogging.

Command	Description
> XE1ØØ	Erases sequence #100
> XD1ØØ	Defines sequence #100
> LD3	Disables the limits(*not needed if you have limit switches installed*)
> JA25	Sets Jog Acceleration to 25 rps^2
> JAD25	Sets Jog Deceleration to 25 rps^2
> OSE1	Enables Jog function
> JVL.5	Sets low-speed jog velocity to 0.5 rps
> JVH5	Sets high-speed jog velocity to 5 rps
> IN1J	Sets **IN 1** as a CW jog input
> IN2K	Sets **IN 2** as a CCW jog input
> FOR6.25	Sets the motor to encoder steps
> FOL75	Secondary moves at 75% of the primary speed
> FSI1	Enables the following mode
> XT	Ends Sequence Definition

② Reset the ZXF. Move the primary or primary axis.

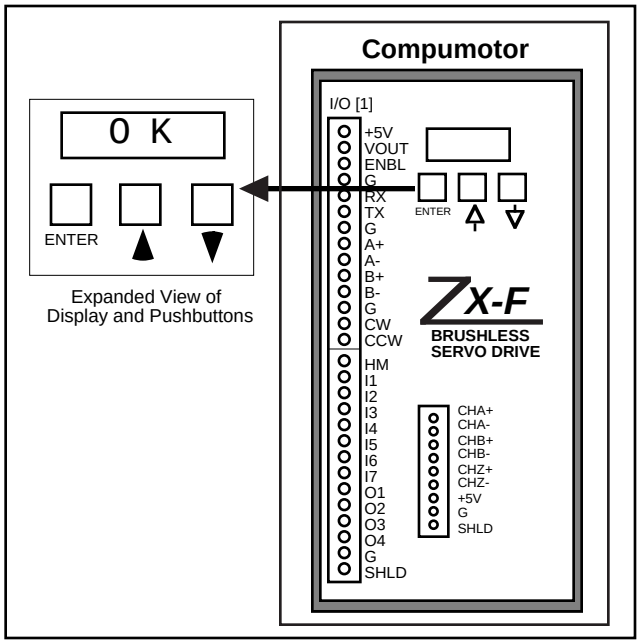
Command	Description
> **Z**	Resets the ZXF

③ Turn on **IN 1** to move the motor CW at 75% of primary axis' speed (until **IN 1** is turned off).

④ Turn on **IN 2** to move the motor CCW at 75% of primary axis' speed (until **IN 2** is turned off).

Changing Following Ratio Via Front Panel Pushbuttons

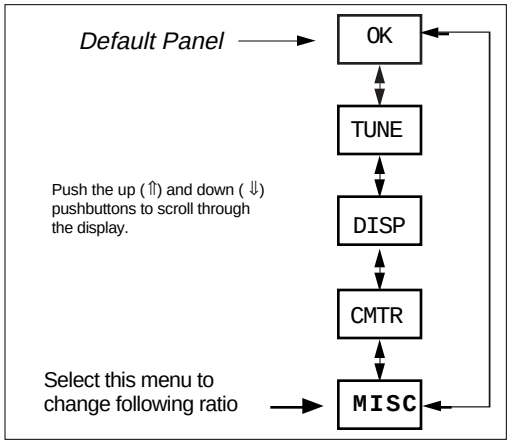
The ZX has a four-character, dot-matrix, alphanumeric display. You can modify the following ratio with the pushbuttons.



ZX Indexer/Drive Display and Pushbuttons

Pushbutton Operation

You can change the following ratio under the **MISC** menu. Only one panel is shown on the display at a time.



Main Menu Panel (Overview)

OK is the default message. Press the **DOWN** pushbutton to access the **MISC** menu. The table below shows the main menus and sub-menus. Select the **FOLL** option under the **MISC** menu.

O K	T U N E	D I S P	C M T R	**M I S C**
Home Panel	P P n n	D V E L	6 0 5	S A V E
	P I n n	D E R R	6 0 6	R F S
	P D n n	D C A	6 1 0	B R m m
	T G n n		6 2 0	A D p p
	V P n n		6 3 0	**F O L L**
	V I n n		6 4 0	S E Q U
	V F n n		F M C A	R E V #
				J O G

Main Menu Panel

To change the following ratio on the fly, press **ENTER** when **FOLL** is displayed. The four least significant digits of the **FOL** command will be displayed. The least significant digit is the following ratio in tenths (it should be blinking). This indicates that changes to the following ratio will be in 1/10 increments. To select a higher magnitude of ratio change, press **UP** or **DOWN** (moving the cursor left or right respectively). Any changes to the following ratio will be at the new digit's magnitude.

100s	10s	1s	0.1s	FOL weighting
a	b	c	d	Blinking digit location

To change the ratio at the blinking digits magnitude, simultaneously press **ENTER** with **UP**

or **DOWN**. *Press **UP** and **DOWN** simultaneously to return to the Main menu.*

Following a Step and/or Direction Signal

The ZXF can follow a step (or pulse) and a direction signal rather than quadrature encoder pulses. The same incremental encoder interfaces used for quadrature following are used for the step and direction following. The **Phase A+** and **Phase A-** inputs are now the **Step+** and **Step-** inputs. The **Phase B+** and **Phase B-** inputs are now the **Direction+** and **Direction-** inputs. The only other requirement is to put the ZXF in the pulse and direction mode. This is accomplished with the **FSN** command. By typing **FSN1** the pulse and direction capability is enabled and the **Phase A** and **Phase B** inputs are now step and direction inputs. Once the pulse and direction capability is added, any following applications are performed exactly as if the input were quadrature signals. If your application requires pulse and direction, enable the Pulse and Direction mode and repeat the procedures in this following section.

Following Equation and Command Summary

This section provides a reference for the following equations and the ZXF software commands that are associated with following.

Following Command Summary

For more information on these commands, refer to the ***ZX Software Reference Guide***. *Set-up commands are required for any following application.*

FSA	Followed by a 1—enables instant acceleration between commanded velocities for each resulting velocity change after sampling the encoder and determining the ratio.
FSF	Followed by a 1—enables the use of following acceleration as determined by **FAC** and **FEN**.
FSI	Followed by a 1—enables following mode versus indexer mode.
FSJ	Followed by a 1—enables the encoder port labeled ABS/INC ENCODER for following.
FSK	Followed by a 1—enables the calculation of **FC** for Synchronization mode.
FSL	Followed by a 1—enables Synchronization mode.
FSN	Followed by a 1—enables a step/direction signal to be followed or just a pulse train if direction is not used in conjunction with **FSP**.
FSP	Followed by a 1—enables position tracking.
FOR	The number of secondary motor pulses per unit of travel divided by the primary encoder pulses per the same unit of travel.
FOL	The % of the primary encoder speed that the secondary axis moves at.
FAC	The change in following % for each change in encoder pulse count of **FEN** during following acceleration.
FEN	The number of encoder pulses that the encoder count must change by to increment the following percentage by **FAC.**
FP	In Motion Profiling (MPP) mode, command execution is paused for the number of following encoder steps set by the **FP** command.
FPA	In Motion Profiling (MPP) mode, the execution of commands pauses until the value in the encoder counter exceeds the **FPA** value.
FIN	The amount by which the following % changes when changed by inputs or by the pushbuttons.
FBS	In synchronization, used with **FIN** to determine the amount of following percentage correction.
FC	In synchronization, the expected number of encoder counts between registration marks.
TF	The following encoder sample period.
VARn=FEP	**FEP** is a read only variable of the actual encoder count. Set it equal to the variable to get the current value of the encoder counter.
PF	Gives you a report back of the encoder count.
PFZ	Clears the encoder counter.
SSP	Followed by a 1—enables you to modify the following percentage by the front panel pushbuttons.
INnX	An input function that lets you increase the following percentage by **FIN**.
INnY	An input function that lets you to decrease the following percentage by **FIN**.
INnI	An input function used for the registration mark sensor of the synchronization mode.

The commands are categorized according to the applications the support.

Velocity Following

FOR
FOL
FSI
FSJ
TF

Velocity and Position Following

FOR	FSF	FSN
FOL	FEN	SSP
FSI	FAC	FSM
FSJ	FSA	PF
TF	FSP	PFZ

Special Function

Advance and Recede

FOR	FSF
FOL	FAC
FSI	FEN
FSJ	FP
TF	FPA

Synchronization

FOR	FSF	FBS
FOL	FAC	FSK
FSI	FEN	FSL
FSJ	FC	INnI
TF	FIN	INnX
INnY		

Following Equation Summary

The following equations were discussed throughout this chapter. They are provided again here for reference and convenience

Velocity Following

Secondary Motor steps = Primary encoder count • **FOR** • $\frac{\textbf{FOL}}{100}$

Velocity and Position Following

Given V_{max} and D_{acc} :

FEN = $V_{max} \bullet \frac{\textbf{TF}}{1000}$ $V_{max} = \frac{\text{Primary encoder counts}}{\text{second}}$

TF = Primary Encoder Sample Period in ms

FAC = $\frac{\textbf{FOL} \bullet V_{max}}{D_{acc}} \bullet \frac{\textbf{TF}}{1000}$ $V_{max} = \frac{\text{Primary encoder counts}}{\text{second}}$

TF = Primary Encoder Sample Period in units of ms

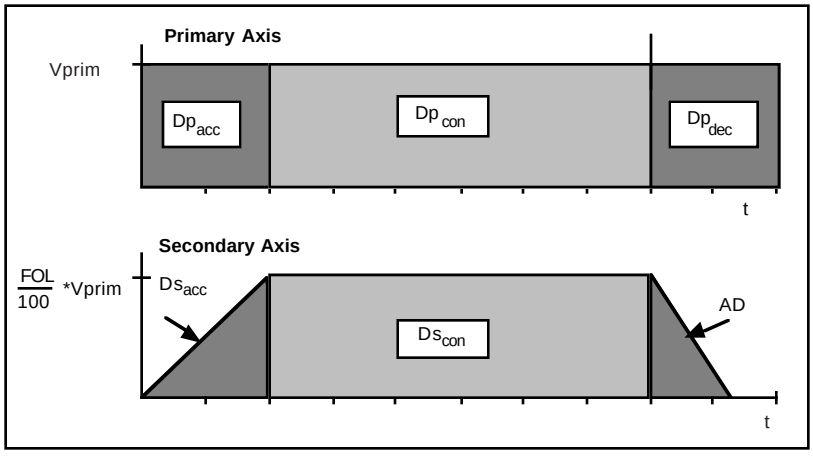
FOL = Following percentage in units of percent

D_{acc} = Distance primary axis moves during secondary axis acceleration in units of primary encoder counts

Given V_{max} and A_{max} Determine D_{acc} from $D_{acc} = \frac{V_{max}^2}{A_{max}} \bullet \textbf{FOR} \bullet \frac{\textbf{FOL}}{100}$

The acceleration is in units of secondary motor steps/sec^2 . The maximum velocity of the primary is in primary motor steps/sec.

The figure below illustrates the motion profiles of a secondary and the primary. The different parameters are shown on the profiles and the equations to determine the parameters are given below. **AD** is used for the deceleration.



Velocity and Position Following With AD Decel

$$Ds_{acc} = \textbf{FOR} \bullet \left(\frac{1}{2} \bullet \Delta\textbf{FOL}^2 \bullet \frac{\textbf{FEN}}{100 \bullet \textbf{FAC}} + \Delta\textbf{FOL} \bullet \textbf{FOL}_I \bullet \frac{\textbf{FEN}}{100 \bullet \textbf{FAC}} \right)$$

Δ**FOL** = The change in following percentage

$\mathbf{FOL_I}$ = The initial following percentage

The equations above can be simplified by defining a following acceleration constant determined by **FAC** and **FEN**.

$$K = \frac{\mathbf{FEN}}{100 \cdot \mathbf{FAC}}$$

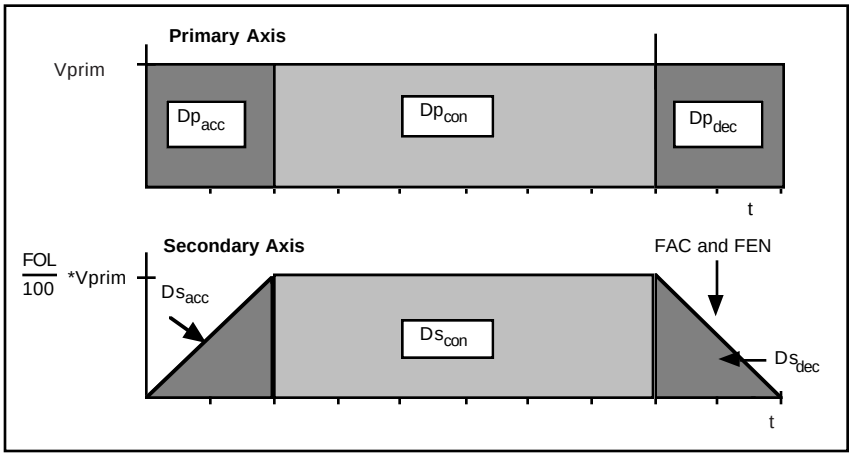
The equations can now be written as:

$$D_{sec} = \mathbf{FOR} \cdot K \cdot \left(\frac{1}{2} \cdot \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL} \cdot FOL_I \right)$$

$$D_{sec\ (Deceleration)} = - \mathbf{FOR} \cdot K \cdot \left(\frac{1}{2} \cdot \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL} \cdot FOL_I \right)$$

Use the equations/parameters in the following graph to make a trapezoidal move.

☞ ***Helpful Hint:*** Deceleration is done according to the FAC and FEN commands.



Velocity and Position Following With Following Decel

$$D_{prim} = Dp_{acc} + Dp_{con} + Dp_{dec}$$

$$D_{sec} = Ds_{acc} + Ds_{con} + Ds_{dec}$$

$$Ds_{acc} = \mathbf{FOR} \cdot K \cdot \left(\frac{1}{2} \cdot \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL} \cdot \mathbf{FOL_I} \right)$$

$$Ds_{dec} = - \mathbf{FOR} \cdot K \cdot \left(\frac{1}{2} \cdot \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL} \cdot \mathbf{FOL_I} \right)$$

$$Dp_{acc} = \Delta\mathbf{FOL} \cdot \frac{\mathbf{FEN}}{\mathbf{FAC}}$$

$$Ds_{con} = Dp_{con} \cdot \mathbf{FOR} \cdot \frac{\mathbf{FOL}}{100}$$

$$\mathbf{FOL} = \frac{D_{prim}}{200 \cdot K} - \sqrt{\left(\frac{D_{prim}}{200 \cdot K}\right)^2 - \left(\frac{D_{sec}}{K}\right)}$$

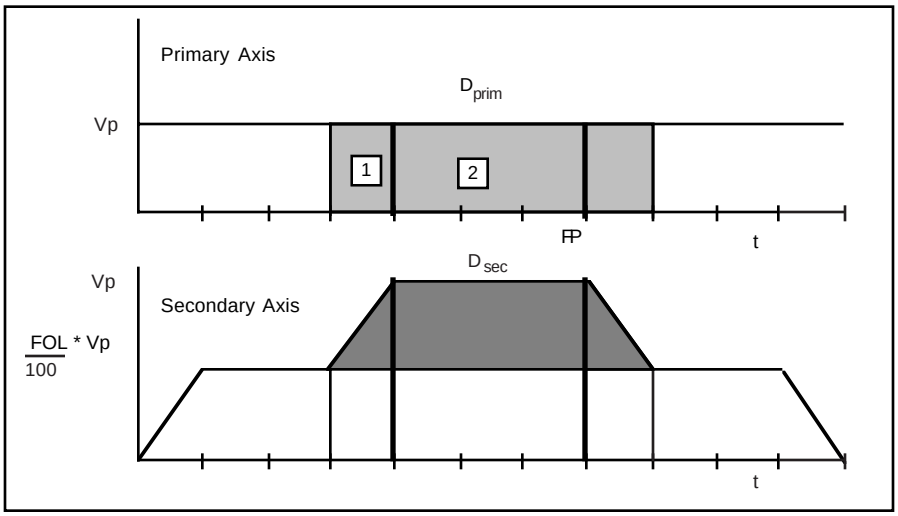
K is the following acceleration constant, D_{prim} and D_{sec} are the distances that the primary axis and secondary axes will move, respectively. $\mathbf{FOL_I}$ is the initial following percentage. In the case of a trapezoidal move, it will always be 0. The required sequence is:

Command	Description
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
FSI1	Enables Following mode
FORn	Sets motor to encoder steps per unit travel ratio
FACn	Sets the following acceleration increment
FENn	Sets the encoder period
Dn	Sets the secondary axis distance to n motor steps
FOLn	Sets the initial following percentage to n
TR1	Waits on the input trigger
MPP	Enters the Motion Profiling mode
G	Starts motion
FPa	Waits until a encoder pulses have passed
FOLØ	Stops the motion of the secondary
FPb	Waits for the decel ramp distance
STOP	Stops the move itself
> XT	Ends the Sequence #1 definition

The value for a in the first **FP** = Dp_{acc} + Dp_{con}

The value for b in the second **FP** = Dp_{dec}

Advance and Recede



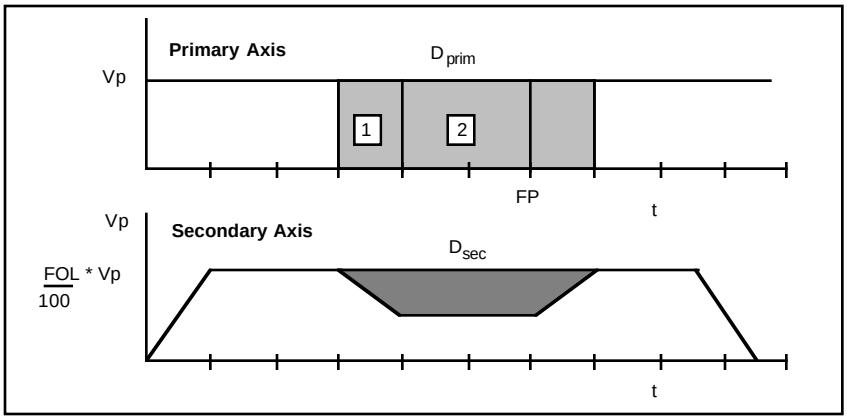
Advance Profile

In this figure, D_{prim} is the distance that the primary axis moves in encoder steps during the advance portion of the profile. It is the lightly shaded region. D_{sec} is the distance in secondary motor steps that the secondary axis advances with respect to the moving primary axis. This distance is the darkly shaded region.

$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{200 \cdot K} - \sqrt{\left(\frac{D_{prim}}{200 \cdot K}\right)^2 - \left(\frac{D_{sec}}{\mathbf{FOR} \cdot K}\right)}$$

The breakpoint to decelerate to **FOL1ØØ** is entered for the **FP** value in the advance sequence. **FP** is a distance equal to the sum of areas 1 and 2 in the primary profile.

$$\mathbf{FP} = D_{prim} - \frac{\mathbf{FEN}}{\mathbf{FAC}} \cdot \left(\mathbf{FOL}_F - \mathbf{FOL}_I\right)$$



Recede Profile

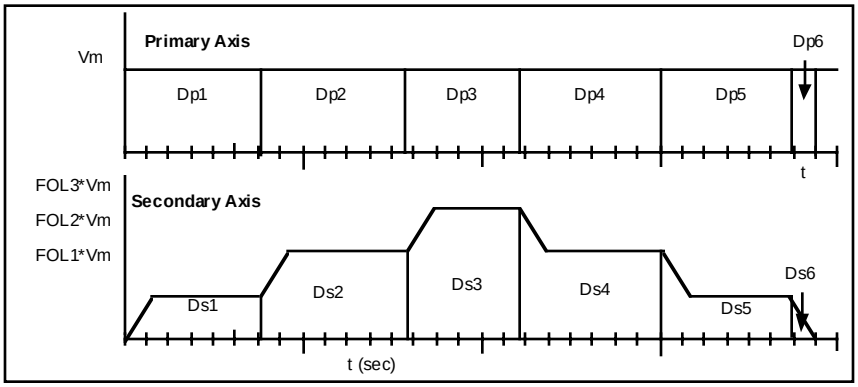
The dark shaded area is the distance that the secondary will recede with respect to the moving primary. The lightly shaded area is the distance the primary moves while the secondary recedes.

$$FOL = FOL_I - \frac{D_{prim}}{200 \cdot K} + \sqrt{\left(\frac{D_{prim}}{200 \cdot K}\right)^2 - \frac{D_{sec}}{FOR \cdot K}}$$

K is the following acceleration constant and is equal to: $\frac{FEN}{100 \cdot FAC}$

FP is equal to: $D_{prim} + \frac{FEN}{FAC} \cdot (FOL_F - FOL_I)$

Cam Following



Cam Profile

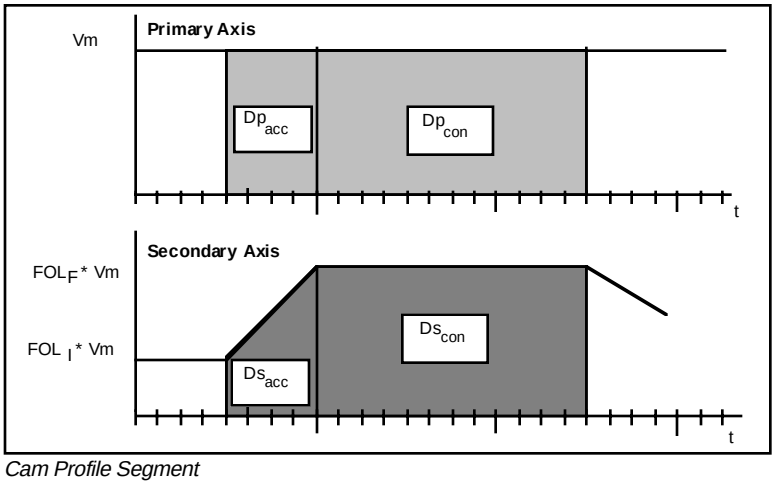
Each segment of the secondary and primary move profile is marked by a primary distance Dpn and a secondary distance Dsn. The following percentage required for each segment is determined from the equations below. D_{prim} and D_{sec} are equal to Dpn and Dsn for each segment. When accelerating to a higher percentage use:

$$FOL = FOL_I + \frac{D_{prim}}{100 \cdot K} - \sqrt{\left(\frac{D_{prim}}{100 \cdot K}\right)^2 + \frac{2 \cdot FOL_I \cdot D_{prim}}{100 \cdot K} - \frac{2 \cdot D_{sec}}{FOR \cdot K}}$$

When decelerating to a lower percentage, use:

$$FOL = FOL_I - \frac{D_{prim}}{100 \cdot K} + \sqrt{\left(\frac{D_{prim}}{100 \cdot K}\right)^2 - \frac{2 \cdot FOL_I \cdot D_{prim}}{100 \cdot K} + \frac{2 \cdot D_{sec}}{FOL \cdot K}}$$

FP or **FPA** are equal to Dpn or the accumulative Dpn respectively. Each cam segment can be broken down and described with the equations below.



Cam Profile Segment

$$D_{prim} = Dp_{acc} + Dp_{con}$$

$$D_{sec} = Ds_{acc} + Ds_{con}$$

$$Ds_{acc} = FOR \cdot K \cdot \left(\frac{1}{2} \cdot (FOL_F - FOL_I)^2 + (FOL_F - FOL_I) \cdot FOL_I \right)$$

$$Dp_{acc} = (FOL_F - FOL_I) \cdot \frac{FEN}{FAC}$$

$$Ds_{con} = Dp_{con} \cdot FOR \cdot \frac{FOL}{100}$$

The distance for **FP** is $Dp_{acc} + Dp_{con}$, for **FPA** is the absolute value of the primary distance since the start of the cam cycle.

C H A P T E R ⑥

Z Series Shunt Regulator

Chapter Objectives

The information in this chapter will enable you to:

- ❏ Understand the shunt regulator's purpose
- ❏ Install the shunt regulator
- ❏ Use this chapter as a quick reference for shunt regulator specifications

What Is A Shunt Regulator?

The Z Shunt Regulator is an option for the ZX/ZXF 600, 800, and 900 series motor/drives. Two versions of the Z Shunt Regulator are available.

- ❏ The ZSR-400W version dissipates 400 watts
- ❏ The ZSR-800W version dissipates 800 watts

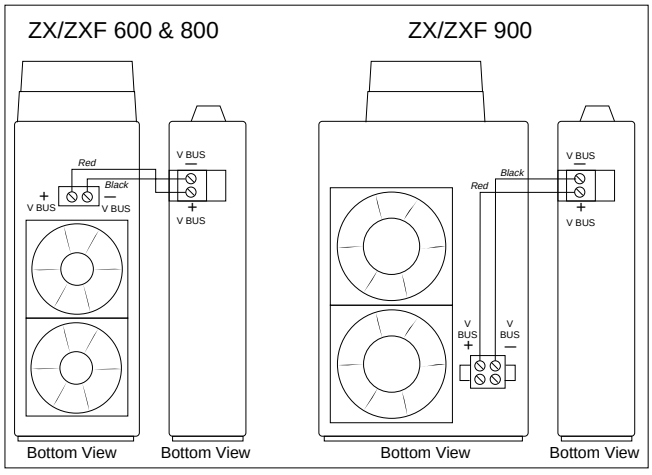
☞ ***Helpful Hint:***
The Z Shunt Regulator is designed to operate from 0°C - 50°C and dump power at a voltage level of 365VDC

Compumotor recommends that you use this option if your load reflects more energy than the ZX/ZXF can absorb. This typically happens when the drive is decelerating a large inertial load with little or no friction. In this situation, the ZX/ZXF will fault with error code **Ø4** (Drive Overvoltage). The shunt regulator dissipates the reflected energy (heat).

Shunt Regulator Installation

① Make sure that power is off before proceeding with this procedure.
② Connect V BUS(+) on the ZX/ZXF to V BUS(+) on the shunt regulator.
③ Connect V BUS(-) on the ZX/ZXF to V BUS(-) on the shunt regulator.

☞ ***Helpful Hint:***
For best performance, the shunt regulator should be as close to the ZX/ZXF as possible. Use 12 AWG, twisted pair wires (20A - 30A @ 350VDC pulsed) to connect the terminals. The wire should be no longer than 2 feet.

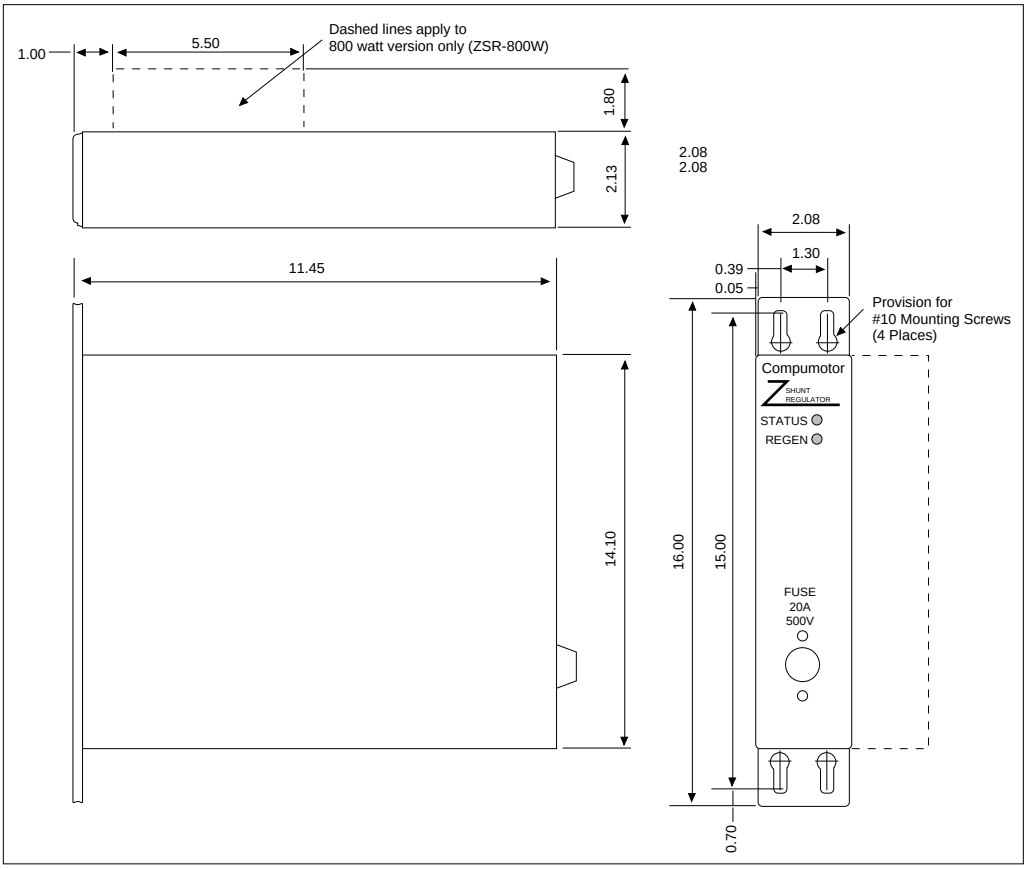


Shunt Regulator Installation

④ Mount the shunt regulator so air flows from the bottom to the top of the unit.

⑤ Apply power to the ZX/ZXF.

If the shunt regulator is properly configured, the unit's status LED will be illuminated. When the units is power dumping, the yellow **REGEN** LED will be illuminated. When power is removed from the ZX/ZXF, the **STATUS** LED will remain on until the bus voltage drops to approximately 60V.



Z Series Shunt Regulator Dimensions

As the motor decelerates the load, the voltage on the DC bus inside the ZX increases as the energy is reflected to the drive from the load. The capacitors inside the drive store energy; however, the capacitors' storage capacity is limited. If the energy in the system exceeds the capacitors' storage capacity, the excess energy must be dumped into a shunt regulator. Shunt regulators dump excess energy into resistors. If the bus voltage in the ZX rises above 365VDC, **Overvoltage Warning #03** occurs. At a bus voltage of 405VDC the drive faults with an **Overvoltage Fault #04**. A shunt regulator can eliminate these conditions.

Selecting a Z Series Shunt Regulator

To determine which shunt regulator is appropriate for your system, complete the following steps.

① Determine how much energy is in the mechanical system. Use the following equation:

$Mech_{energy}$ (Joules) $= \frac{1}{2}$ J ω^2: Where J is the *system* inertia in Kg-m^2 and ω is the maximum motor velocity in radians/sec.

② Determine if the ZX capacitors can absorb the mechanical system's energy. Use the following equation depending on the ZX type:

$Elec_{energy}$ (Joules) $= \frac{1}{2}$ C V^2: where C is the capacitance of the ZX in Farads. In the Z600s and Z800s, C = 4200mF and in the Z900s, C = 8400mF. The V that you use to calculate the energy the drive can absorb is the maximum ZX voltage (405VDC) for ELEC/ENERGY1 and the AC line voltage • $\sqrt{2}$ for ELEC/ENERGY2.
The absorption capability = ELEC/ENERGY1 — ELEC/ENERGY2

☞ ***Helpful Hint:***
Capacitor Energy Absorption Calculation

Line voltage is 208VAC applied to a Z620

$$\Delta \text{ ELEC/ENERGY1} = \text{ELEC/ENERGY1} - \text{ELEC/ENERGY2}$$

$$= \frac{1}{2} \text{ C } (405)^2 - \frac{1}{2} \text{ C } (208 \cdot \sqrt{2})^2$$

$$= \frac{1}{2} (0.0042) \ [405^2 - (208 \cdot \sqrt{2})^2]$$

$$= 0.0021 \ (164{,}025\text{-}86{,}528)$$

$$= 162.7 \text{ Joules}$$

If the amount of mechanical energy in the system is less than the amount of electrical energy that the ZX can store, you do not need a shunt regulator. You may run your load at any velocity and decelerate at any rate to a stop without causing an overvoltage fault.

☞ ***Helpful Hint:***
Fast decelerations require more power to be dumped than slow decelerations.

If the system's mechanical energy exceeds the amount that can be absorbed, a shunt regulator may be required. There are several losses in the system that help lower the amount of energy the drive needs to absorb. Friction, drive inefficiencies and I^2R losses in the motor are some examples. These losses can skew your ability to determine whether or not a shunt regulator is needed (if your calculations indicate that there is more mechanical energy than electrical energy in the system). If, however, the mechanical energy exceeds the electrical energy that can be absorbed and you want to decelerate the load quickly, you will need a regulator. To select the proper regulator (400W or 800W), additional calculations must be made. Determine the amount of power (not energy) that must be dumped. The power that must be dissipated increases as deceleration time decreases (assuming the same load and peak velocity for the move). Since the average power is the change in energy over time, and the energy in this example is fixed, the only variable is time.

☞ ***Helpful Hint:***
To calculate the time required to decelerate, use the following equation:

$$\text{Time}_{\text{decel}} \text{ (Seconds)} = \frac{(V_{\text{init}} - V_{\text{final}})}{\text{Decel rate}}$$

Where V is in revs/sec and decel rate is in revs/sec^2

The shunt regulator for a 240V ZX is only active when the bus voltage exceeds 365VDC. The ZX absorbs some energy before the shunt becomes active and continues to absorb energy if the bus voltage continues to climb while the regulator is active. Therefore, the amount of electrical energy that the drive can absorb from the mechanical system should be subtracted before calculating the power that must be dumped. For a Z600 or Z800 operating at 240VAC input, subtract 102 joules. For a Z900 at 240VAC, subtract 204 joules. The calculation for a Z600 is shown below:

$$\text{Elec}_{\text{energy}} = \tfrac{1}{2} \ (0.0042) \ [405^2 - (240 * \sqrt{2})^2] = 102 \text{ Joules}$$

Now divide the mechanical energy by the decel time to get the average power in joules that will dissipate over the linear deceleration portion of the move.

$$\text{Power}_{\text{avg}} = \frac{\text{Mech}_{\text{energy}}}{\text{Time}_{\text{decel}}}$$

$$\text{Power}_{\text{peak}} = \text{Power}_{\text{avg}} \cdot 2$$

☞ ***Helpful Hint:***
For moves with a *linear deceleration ramp*, peak power is twice the average power.

The shunt regulator chosen must be able to dump the **peak** power of the move. If the peak power is too high, lower the deceleration rate or the move velocity. A lower velocity will reduce the stored mechanical energy in the system.

The table below indicates the maximum load-to-rotor inertia ratio for the various motor sizes that you can have without requiring a shunt regulator. These calculations are based on the motors being operated at rated speed and an input power of 240V to the ZX. The assumption is that there are no losses in the drive or motor. These numbers are conservative.

Motor	Rated Speed	Load:Rotor Inertia Ratio
605	103.33	3.9:1
606	60.00	7.3:1
610	116.67	0.52:1
620	61.67	1.1:1
630	41.67	2.2:1
635	50.00	1.0:1.0

Motor	Rated Speed	Load:Rotor Inertia Ratio
640	26.67	2.6:1
910	83.33	0.61:1
920	52.5	0.86:1
930	38.33	1.34:1
940	25.00	0.98:1

As long as the load to rotor inertia ratio is equal to or less than what is in the table, you may run at any velocity and use any deceleration rate (0<decel<∞) to stop the load without getting an over voltage **fault** on the drive. If your inertia ratio is greater than the listing, you will have to limit either your deceleration rate or lower the rotor speed. The greater the inertial mismatch, the lower the allowable deceleration rate or velocity.

The table below gives the maximum load-to-rotor-inertia ratio you can use having an infinite deceleration rate without a shunt regulator at various percentages of the rated motor velocity. Again, these numbers are based on 240VAC input power to the ZX. Losses are not accounted for, so the numbers are conservative.

Motor	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
605	3.91	5.06	6.68	9.02	12.64	18.65	29.70	53.58	121.80	490.21
606	7.39	9.36	12.11	16.13	22.31	32.57	51.46	92.25	208.82	838.28
610	0.53	0.89	1.39	2.12	3.24	5.11	8.55	15.98	37.20	151.79
620	1.09	1.59	2.27	3.27	4.82	7.38	12.09	22.27	51.35	208.41
630	2.24	3.00	4.06	5.61	8.00	11.96	19.24	34.99	79.98	322.90
635	1.0	2.48	3.10	4.10	5.60	8.00	12.58	22.4	50.3	201.3
640	2.61	3.46	4.64	6.37	9.03	13.45	21.57	39.13	89.29	360.18
910	0.62	1.00	1.53	2.31	3.50	5.48	9.12	16.99	39.49	160.95
920	0.86	1.30	1.91	2.80	4.18	6.45	10.65	19.71	45.59	185.37
930	1.34	1.89	2.65	3.77	5.50	8.36	13.62	24.99	57.47	232.89
940	0.99	1.46	2.11	3.06	4.53	6.96	11.43	21.10	48.73	197.92

The table is calculating theoretical maximums. It would be difficult, if not impossible, to tune a servo with inertia mismatches greater than 25:1. If your application is not within the ranges in the table, you may need a shunt regulator.

The tables below show the maximum allowable deceleration rates using different wattage shunt regulators at various speeds and with variable inertia ratios. The calculations do not include any losses in the system (the results are conservative).

The load-to-rotor inertia ratios are on the top row of each table. Each table uses a different wattage shunt regulator. These tables can help determine the right shunt regulator for your application (400W or 800W version).

Deceleration rates using a 400W Shunt Regulator

Deceleration rates at 100% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	103.33	∞	∞	∞	6972.4	862.7	459.8	237.7	160.3	88.4	61.0	46.6
606	60.00	∞	∞	∞	∞	∞	∞	1398.9	362.0	126.9	76.9	55.2
610	116.67	707.5	232.2	138.9	99.1	77.0	63.0	46.2	36.4	23.9	17.7	14.1
620	61.67	∞	269.7	129.9	85.5	63.8	50.8	36.1	28.1	18.0	13.2	10.5
630	41.67	∞	∞	328.8	145.7	93.6	69.0	45.2	33.6	20.5	14.7	11.5
635	50	∞	199.2	99.1	65.9	49.4	39.5	28.2	21.9	14.1	10.38	8.21
640	26.67	∞	∞	437.0	130.9	77.0	54.5	34.4	25.2	15.0	10.7	8.3
910	83.33	328.8	93.6	54.6	38.5	29.8	24.2	17.7	13.9	9.1	6.7	5.4
920	52.50	606.1	82.0	44.0	30.1	22.8	18.4	13.3	10.4	6.7	5.0	3.9
930	38.33	∞	126.7	51.6	32.4	23.6	18.6	13.0	10.0	6.4	4.7	3.7
940	25.00	1487.8	46.7	23.7	15.9	12.0	9.6	6.9	5.3	3.4	2.5	2.0

Deceleration rates at 75% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	77.50	∞	∞	∞	∞	∞	∞	3624.8	555.7	178.3	106.2	75.6
606	45.00	∞	∞	∞	∞	∞	∞	∞	∞	1049.1	208.7	115.8
610	87.50	∞	1471.8	351.0	199.3	139.1	106.9	73.0	55.5	34.6	25.2	19.8
620	46.25	∞	∞	1051.8	253.5	144.1	100.7	62.8	45.6	27.1	19.3	15.0

630	31.25	∞	∞	∞	∞	1147.3	267.6	105.6	65.8	33.9	22.8	17.2
635	37.5	∞	∞	618.0	184.4	108.4	76.7	48.4	35.4	21.2	15.1	11.7
640	20.00	∞	∞	∞	∞	∞	383.9	94.0	53.6	25.8	17.0	12.7
910	62.50	∞	1147.3	151.5	81.1	55.4	42.0	28.4	21.4	13.3	9.6	7.5
920	39.38	∞	∞	175.0	73.4	46.5	34.0	22.1	16.4	9.9	7.1	5.6
930	28.75	∞	∞	∞	130.7	61.4	40.2	23.7	16.8	9.8	6.9	5.3
940	18.75	∞	∞	128.1	42.8	25.7	18.4	11.7	8.6	5.1	3.7	2.9

Deceleration rates at 50% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	51.67	∞	∞	∞	∞	∞	∞	∞	∞	∞	1258.3	299.9
606	30.00	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
610	58.33	∞	∞	∞	∞	∞	724.2	234.0	139.6	69.5	46.2	34.7
620	30.83	∞	∞	∞	∞	∞	∞	701.2	184.5	64.9	39.4	28.3
630	20.83	∞	∞	∞	∞	∞	∞	∞	∞	164.4	64.0	39.7
635	25.0	∞	∞	∞	∞	∞	∞	412.0	133.3	49.5	30.4	21.9
640	13.33	∞	∞	∞	∞	∞	∞	∞	∞	218.5	55.7	31.9
910	41.67	∞	∞	∞	∞	∞	442.0	101.0	57.0	27.3	17.9	13.4
920	26.25	∞	∞	∞	∞	∞	∞	116.7	52.3	22.0	13.9	10.2
930	19.17	∞	∞	∞	∞	∞	∞	∞	99.6	25.8	14.8	10.4
940	12.50	∞	∞	∞	∞	∞	∞	85.4	30.8	11.9	7.3	5.3

Deceleration rates at 25% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	25.83	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
606	15.00	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
610	29.17	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	760.3
620	15.42	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
630	10.42	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
635	12.5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
640	6.67	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
910	20.83	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1419.0
920	13.13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
930	9.58	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
940	6.25	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Deceleration rates using an 800W Shunt Regulator

Deceleration rates at 100% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	103.33	∞	∞	∞	∞	1725.3	919.5	475.4	320.6	176.7	122.0	93.1
606	60.00	∞	∞	∞	∞	∞	∞	2797.7	724.0	253.7	153.8	110.4
610	116.67	1414.9	464.5	277.8	198.2	154.0	126.0	92.3	72.9	47.7	35.5	28.2
620	61.67	∞	539.4	259.7	171.0	127.5	101.6	72.3	56.1	36.0	26.5	20.9
630	41.67	∞	∞	657.6	291.5	187.2	137.9	90.3	67.2	40.9	29.4	23.0
635	50.0	∞	398.6	198.2	131.8	98.8	79.0	56.4	43.9	28.2	20.7	16.4
640	26.67	∞	∞	874.1	261.7	153.9	109.0	68.8	50.3	30.1	21.4	16.7
910	83.33	657.6	187.2	109.2	77.0	59.5	48.5	35.4	27.9	18.2	13.5	10.7
920	52.50	1212.1	164.1	88.0	60.1	45.7	36.8	26.5	20.7	13.4	9.9	7.9
930	38.33	∞	253.3	103.1	64.7	47.2	37.1	26.0	20.0	12.7	9.3	7.3
940	25.00	2975.6	93.4	47.5	31.8	23.9	19.2	13.7	10.7	6.9	5.1	4.0

Deceleration rates at 75% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	77.50	∞	∞	∞	∞	∞	∞	7249.6	1111.4	356.6	212.4	151.2
606	45.00	∞	∞	∞	∞	∞	∞	∞	∞	2098.3	417.3	231.7
610	87.50	∞	2943.5	702.1	398.6	278.3	213.7	146.0	110.9	69.2	50.3	39.5
620	46.25	∞	∞	2103.6	506.9	288.2	201.3	125.6	91.3	54.2	38.6	29.9
630	31.25	∞	∞	∞	∞	2294.5	535.2	211.2	131.6	67.7	45.6	34.4
635	37.5	∞	∞	1236.0	368.8	216.7	153.5	96.9	70.8	42.3	30.1	23.4
640	20.00	∞	∞	∞	∞	∞	767.8	188.1	107.2	51.6	34.0	25.4
910	62.50	∞	2294.5	302.9	162.2	110.7	84.1	56.7	42.8	26.5	19.2	15.1
920	39.38	∞	∞	350.1	146.9	92.9	68.0	44.2	32.8	19.9	14.3	11.1
930	28.75	∞	∞	∞	261.4	122.9	80.3	47.4	33.7	19.5	13.7	10.6
940	18.75	∞	∞	256.1	85.6	51.4	36.7	23.4	17.1	10.3	7.3	5.7

Deceleration rates at 50% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	51.67	∞	∞	∞	∞	∞	∞	∞	∞	∞	2516.6	599.8
606	30.00	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
610	58.33	∞	∞	∞	∞	∞	1448.5	468.0	279.1	138.9	92.5	69.3
620	30.83	∞	∞	∞	∞	∞	∞	1402.4	369.1	129.9	78.8	56.6
630	20.83	∞	∞	∞	∞	∞	∞	∞	∞	328.8	127.9	79.4
635	25.0	∞	∞	∞	∞	∞	∞	824.0	266.7	99.1	60.9	43.9
640	13.33	∞	∞	∞	∞	∞	∞	∞	∞	437.0	111.4	63.8
910	41.67	∞	∞	∞	∞	∞	884.0	202.0	114.0	54.6	35.9	26.7
920	26.25	∞	∞	∞	∞	∞	∞	233.4	104.7	44.0	27.9	20.4
930	19.17	∞	∞	∞	∞	∞	∞	∞	199.2	51.6	29.6	20.8
940	12.50	∞	∞	∞	∞	∞	∞	170.8	61.6	23.7	14.7	10.6

Deceleration rates at 25% of rated speed

Motor	RPS	1:1	2:1	3:1	4:1	5:1	6:1	8:1	10:1	15:1	20:1	25:1
605	25.83	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
606	15.00	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
610	29.17	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1520.6
620	15.42	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
630	10.42	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
635	12.5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
640	6.67	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
910	20.83	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2837.9
920	13.13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
930	9.58	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
940	6.25	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

C H A P T E R ⑦

Hardware Reference

The information in this chapter will enable you to:

- ❑ Use this chapter as a quick-reference for system performance specifications

ZX600 Electrical Specifications

Electrical specifications for the ZX Drive's input and output power are provided in this section.

Input Power

Voltage (Nominal)	120VAC (1 or 3-phase) **or** 120VAC (1 or 3-phase)
Voltage (Range)	92-130VAC (1 or 3-phase) **or** 205-252VAC (1 or 3-phase)
Frequency (Range)	47-66 Hz
Current (Max. cont.)	15A (rms) 3-phase **or** 26A (rms) 1-phase
Power (Max. cont.)	6.2 KVA
Fuses	20A slow blow—Not user accessible
Isolation transformer	Not required

The actual input power and current is a function of the motor's operating point (speed and torque) and the duty cycle. You can de-rate the fuses and isolation transformer by scaling the above numbers by your actual requirements. The numbers above reflect the servo motor and drive operating at rated speed and rated torque at 100% duty.

Output Power

Voltage	405 VDC (maximum)
Frequency	0 - 400Hz fundamental (7 kHz PWM)
Current	20A continuous per phase sinusoidal (14.14Arms)
	40A peak per phase sinusoidal (28.3Arms)
Regen/power dump	Optional accessory

ZX600 Motor/Drive Configuration

The ZX600's hardware is pre-configured to control Z600 motors. The ZX600 series drives only Z600 motors, the ZX800 series drives only Z800 motors, and the ZX900 series drives only Z900 motors. *Be sure that your drive type matches your motor type (Z600, Z800, or Z900).* If you have questions about the ZX motor/drive configuration, call your local Automation Technology Center (ATC) or distributor.

	Units	ZX-605	ZX-606	ZX-610	ZX-620	ZX-630	ZX-635	ZX-640
Continuous Stall Torque*	oz-in	346	633	867	1743	2475	2458	4114
	lb-in	22	40	54	109	155	154	257
	lb-ft	1.8	3.3	4.5	9.1	12.9	12.8	21.4
	Nm	2.4	4.5	6.1	12.3	17.5	17.4	29.1
Peak Torque (±10%)	oz-in	1083	1954	1733	3486	4951	7008	8228
	lb-in	68	122	108	218	309	438	514
	lb-ft	5.6	10.2	9.0	18.2	25.8	36.5	42.9
	Nm	7.7	13.8	12.2	24.6	35.0	49.5	58.1
Rated Torque (±10%)	oz-in	321	576	616	1538	2172	2054	3729
	lb-in	20	36	39	96	136	128	233
	lb-ft	1.7	3.0	3.2	8.0	11.3	10.7	19.4
	Nm	2.3	4.1	4.4	10.9	15.3	14.5	26.3
Rated Power	hp	2.0	2.1	4.3	5.6	5.4	6.1	5.9
	kWatts	1.5	1.5	3.2	4.2	4.0	4.5	4.4
Rated Speed	rpm	6200	3600	7000	3700	2500	3000	1600
	rps	103	60	117	62	42	50	27
Rated Current (line)	A (rms)	5.0	5.3	14.1	14.1	14.1	14.1	14.1
Peak Current (3.3 seconds max)	A (rms)	16.6	17.2	28.2	28.2	28.2	28.2	28.2
Max. Cont. AC Input Current (3 phase 240VAC)	A (rms)	6	6	15	15	15	15	15
Rotor Inertia	$oz\text{-}in^2$ (mass)	5.45	9.45	13.73	35.87	50.79	56.21	111.21
	$oz\text{-}in\text{-}sec^2$	0.01	0.02	0.04	0.09	0.13	0.146	0.29
	$Kg\text{-}m^2$ x 1E-6	99.6	172.9	251.2	656.0	929.0	1028.0	2034.0
Motor Weight	lbs	10	13	16	29	32	37	51
	kg(f)	5	6	7	13	15	16	23
Shipping Weight	lbs	51	55	58	70	73	78	92
	kg(f)	23	25	26	32	33	35	42

ZX600 Series Indexer/Drive Performance Specifications

Positional Repeatability

Repeatability: ±0.088 degrees, unloaded

Positional Accuracy

Resolver Accuracy: ±7 arc minutes

Resolver-to-Digital Converter Accuracy: ±8 arc minutes *(For finer accuracies, contact Compumotor—800-722-2282.)*

Motor/Drive Compatibility

Different motors can take different amounts of current and require different tuning parameters for typical loads. The **CMTR** (Configure Motor Type) command sets up a drive for a particular motor. By issuing **CMTR**, motor current levels and default parameters are recalled from memory. Do not exceed the current level specified for the motor, excessive current levels will damage the motor.

The following information is provided in case you must modify the motor/drive configuration. This command sequence will set up a drive for a particular motor size and perform the commutation (refer to the ***ZX Indexer/Drive Software Reference Guide*** for more on these commands).

> ***WARNING***
>
> This commutation procedure causes violent motor motion. All loads should be removed from the motor shaft before you begin this procedure.

Command	Description
`> 1OFF`	Turns drive off
`> 1CMTRxxx`	Sets drive for the motor; xxx = 605, 606, 610, 620, 630, 635, or 640
`> 1ON`	Enables the drive

Motor Brakes

Motor brakes are mounted directly behind the motor and are pre- assembled at the factory. When ordering the brake option, specify the motor type.

Brake Characteristics	ZX605/606/610	ZX620/630	ZX640	Units
Supply voltage	24	24	24	VDC
Supply current	0.57	0.93	1.27	A
Static braking torque	960	1152	6720	oz-in

Z600 Motor Brake Characteristics

Motor Data

The following pages provide data on each of the motor frame sizes of ZX600 series systems (ZX605, ZX606, ZX610, ZX620, ZX630, ZX635, ZX640). The data reflecting motor torque does not assume operation from a ZX600 drive. The torque specifications reflect the motor's capabilities. In most cases, the motor windings match the drive's output power with an additional safety margin.

	Motor Size:	**Z605**	Value	Units	Tolerance
1	Constant (s):	Torque	68.7	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	29.4	V rms/Krpm	± 10%
3		Electrical Time	10.68	milliseconds	nominal
4		Mechanical Time	1.46	milliseconds	nominal
5		Thermal	32	minutes	nominal
6	Torque (s):	Continuous, Stall	367	oz-in	min. [1]
7		Continuous, Stall	346	oz-in	min. [2]
8		Continuous, Rated	321	oz-in	min. [2]
9		Peak, Max w/o Saturation	1085	oz-in	min. [1]
10		Static Friction	0.96	oz-in	max.
11		Ripple (of Rated Torque)	5	percent	max. [3]
12	Speed:	Rated	6200	rpm	reference
13		Maximum	6200	rpm	reference
14	Frequency	Rated	207	Hz	max.
15	Current:	Rated	5	A rms	max. [1]
16		Peak	16.6	A rms	nominal
17	Voltage:	Rated	240	V rms	reference
18		Max	250	V rms	maximum
19	Output Power	Rated	1.5 (2.0)	kWatts (hp)	min. [1]
20	Inductance	Terminal (line-line)	25	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	2.3	ohms	± 10 % [1]
22	Acceleration at Rated Torque		76870	rads/sec^2	Theoretical
23	Rotor Inertia		99.6	kgm^2 * 1E-6	nominal
24	Damping		0.96	oz-in / krpm	nominal
25	Weight		10	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5° C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10° C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance-to-Frame		0.00122	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (Front to Back)	1.4E-5/8E-6	in/lb	reference
35		Material [5]	RC-#30	—	reference
36		Magnet Type	NdFeB	—	—
37		Loading [6] 1000 rpm	85.4	lbs.	max. [7]
		2000 rpm	67.8	lbs.	max. [7]
		3000 rpm	59.1	lbs.	max. [7]
		4000 rpm	53.8	lbs.	max. [7]
		5000 rpm	50	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2(P/2)		
42	Stator Phase Sequence		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011774-xx		
47	Options:	Brake—24VDC (0.57A)—960 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25° C Ambient
[2] 40° C Ambient
[3] Measured at 60 rpm (1 rps) in Velocity Mode
[4] Rated for 20,000 Hours or 40,000 Hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z605 Motor Specifications

	Motor Size:	**Z606**	Value	Units	Tolerance
1	Constant (s):	Torque	120	oz-in/A rms	± 10%
2		Voltage (sinusoidal)	51.2	V rms/Krpm	± 10%
3		Electrical Time	15.32	milliseconds	nominal
4		Mechanical Time	0.896	milliseconds	nominal
5		Thermal	34	minutes	nominal
6	Torque (s):	Continuous, Stall	672	oz-in	min. [1]
7		Continuous, Stall	634	oz-in	min. [2]
8		Continuous, Rated	576	oz-in	min. [2]
9		Peak, Max w/o Saturation	1957	oz-in	min. [1]
10		Static Friction	0.96	oz-in	max
11		Ripple (of Rated Torque)	5	percent	max. [3]
12	Speed:	Rated	3600	rpm	reference
13		Maximum	3600	rpm	reference
14	Frequency	Rated	120	Hz	max.
15	Current:	Rated	5.3	A rms	max. [1]
16		Peak	17.2	A rms	nominal
17	Voltage:	Rated	240	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	1.6 (2.1)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	38	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	2.48	ohms	± 10% [1]
22	Acceleration at Rated Torque		80000	rads/sec^2	Theoretical
23	Rotor Inertia		172.9	kgm^2 * 1E-6	nominal
24	Damping		1.344	oz-in / krpm	nominal
25	Weight		13.4	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.00201	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	1.4E-5/8E-6	in/lb	reference
35		Material [5]	RC-#30	—	reference
36		Magnet Type	NdFeB	—	—
37		Loading [6] 1000 rpm	90.1	lbs.	max. [7]
		2000 rpm	71.6	lbs.	max. [7]
		3000 rpm	62.4	lbs.	max. [7]
		4000 rpm	N/A	lbs.	max. [7]
		5000 rpm	N/A	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2(P/2)		
42	Stator Phase Sequence		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011774-xx		
47	Options:	Brake—24VDC (0.57A)—960 oz-in Holding Torque IP67 Classification Incremental Encoder Tachometer No Keyway			

[1] 25° C Ambient
[2] 40° C Ambient
[3] Measured at 60 rpm (1 rps) in Velocity Mode
[4] Rated for 20,000 Hours or 40,000 Hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z606 Motor Specifications

	Motor Size:	**Z610**	Value	Units	Tolerance
1	Constant (s):	Torque	61.4	oz-in/A rms	±10%
2		Voltage (Sinusoidal)	26.2	V rms/Krpm	±10%
3		Electrical Time	13.16	milliseconds	nominal
4		Mechanical Time	0.762	milliseconds	nominal
5		Thermal	36	minutes	nominal
6	Torque (s):	Continuous, Stall	977	oz-in	min. [1]
7		Continuous, Stall	921.6	oz-in	min. [2]
8		Continuous, Rated	653	oz-in	min. [2]
9		Peak, Max w/o Saturation	2630	oz-in	min. [1]
10		Static Friction	0.96	oz-in	max
11		Ripple (of Rated Torque)	5	percent	min. [3]
12	Speed:	Rated	7000	rpm	reference
13		Maximum	7000	rpm	reference
14	Frequency	Rated	233	Hz	max.
15	Current:	Rated	15	A rms	max. [1]
16		Peak	45	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	3.3 (4.5)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	5	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	0.38	Ohms	± 10% [1]
22	Acceleration at Rated Torque		73934	rads/sec^2	Theoretical
23	Rotor Inertia		251.2	kgm^2 * 1E-6	nominal
24	Damping		1.728	oz-in / krpm	nominal
25	Weight		16.35	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance-to-Frame		0.00205	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	1.4E-5/8E-6	in/lb	reference
35		Material [5]	RC-#30	—	—
36		Magnet Type	NdFeB	—	max. [7]
37		Loading [6] 1000 rpm	93.5	lbs.	max. [7]
		2000 rpm	74.2	lbs.	max. [7]
		3000 rpm	64.8	lbs.	max. [7]
		4000 rpm	59	lbs.	max. [7]
		5000 rpm	54.7	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2(P/2)		
42	Stator Phase Sequence		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-01		
46	Standard Motor Cable Part Number		71-011775-01		
47	Options:	Brake—24VDC (0.57A)—960 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25° C Ambient
[2] 40° C Ambient
[3] Measured at 60 rpm (1 rps) in Velocity Mode
[4] Rated for 20,000 Hours or 40,000 Hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z610 Motor Specifications

	Motor Size:	**Z620**	Value	Units	Tolerance
1	Constant (s):	Torque	124.2	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	53	V rms/Krpm	± 10%
3		Electrical Time	23.4	milliseconds	nominal
4		Mechanical Time	0.82	milliseconds	nominal
5		Thermal	40	minutes	nominal
6	Torque (s):	Continuous, Stall	1974	oz-in	min. [1]
7		Continuous, Stall	1862	oz-in	min. [2]
8		Continuous, Rated	1632	oz-in	min. [2]
9		Peak, Max w/o Saturation	5299	oz-in	min. [1]
10		Static Friction	25	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	min. [3]
12	Speed:	Rated	3700	rpm	reference
13		Maximum	3700	rpm	reference
14	Frequency	Rated	123	Hz	max.
15	Current:	Rated	15	A rms	max. [1]
16		Peak	45	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	4.5 (6)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	15	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	0.64	Ohms	± 10% [1]
22	Acceleration at Rated Torque		57025	rads/sec^2	Theoretical
23	Rotor Inertia		656	kgm^2 * 1E-6	nominal
24	Damping		2.496	oz-in / krpm	nominal
25	Weight		29	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance-to-Frame		0.0034	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play	2E-5/7E-6	in/lb	reference
35		Material [5]	RC-#30		reference
36		Magnet Type	NdFeB	—	—
37		Loading [6] 1000 rpm	154.7	lbs.	max. [7]
		2000 rpm	122.8	lbs.	max. [7]
		3000 rpm	107.2	lbs.	max. [7]
		4000 rpm	N/A	lbs.	max. [7]
		5000 rpm	N/A	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2(P/2)		
42	Stator Phase Sequence		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011775-xx		
47	Options:	Brake—24VDC (0.93A)—1140 oz-in Holding Torque IP67 Classification Incremental Encoder Tachometer No Keyway			

[1] 25° C Ambient
[2] 40° C Ambient
[3] Measured at 60 rpm (1 rps) in Velocity Mode
[4] Rated for 20,000 Hours or 40,000 Hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z620 Motor Specifications

	Motor Size:	**Z630**	Value	Units	Tolerance
1	Constant (s):	Torque	175.3	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	74.9	V rms/Krpm	± 10%
3		Electrical Time	26.7	milliseconds	nominal
4		Mechanical Time	0.68	milliseconds	nominal
5		Thermal	43	minutes	nominal
6	Torque (s):	Continuous, Stall	2788	oz-in	min. [1]
7		Continuous, Stall	2630	oz-in	min. [2]
8		Continuous, Rated	2304	oz-in	min. [2]
9		Peak, Max w/o Saturation	7488	oz-in	min. [1]
10		Static Friction	40.7	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	min. [3]
12	Speed:	Rated	2500	rpm	reference
13		Maximum	2500	rpm	reference
14	Frequency	Rated	83	Hz	max.
15	Current:	Rated	15	A rms	max. [1]
16		Peak	45	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	4.3 (5.7)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	20	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	0.75	Ohms	± 10% [1]
22	Acceleration at Rated Torque		56934	rads/sec^2	Theoretical
23	Rotor Inertia		929	kgm^2 * 1E-6	nominal
24	Damping		2.88	oz-in / krpm	nominal
25	Weight		32	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		132	°C (Celsius)	± 5 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0038	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play	2E-5/7E-6	in/lb	reference
35		Material [5]	RC-#30	—	reference
36		Magnet Type	NdFeB	—	—
37		Loading [6] 1000 rpm	160	lbs.	max. [7]
		2000 rpm	127.1	lbs.	max. [7]
		3000 rpm	N/A	lbs.	max. [7]
		4000 rpm	N/A	lbs.	max. [7]
		5000 rpm	N/A	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2(P/2)		
42	Stator Phase Sequence—CW rotor rotation		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011775-xx		
47	Options:	Brake—24VDC (0.93A)—8.0 Nm Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C (Celsius) ambient
[2] 40 °C (Celsius) ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z630 Motor Specifications

	Motor Size:	**Z635**	Value	Units	Tolerance
1	Constant (s):	Torque	175.3	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	70	V rms/Krpm	± 10%
3		Electrical Time	0.77	milliseconds	nominal
4		Mechanical Time	20.8	milliseconds	nominal
5		Thermal	28	minutes	nominal
6	Torque (s):	Continuous, Stall	2605	oz-in	min. [1]
7		Continuous, Stall	2458	oz-in	min. [2]
8		Continuous, Rated	2054	oz-in	min. [2]
9		Peak, Max w/o Saturation	7008	oz-in	min. [1]
10		Static Friction	69	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	min. [3]
12	Speed:	Rated	3000	rpm	reference
13		Maximum	3000	rpm	reference
14	Frequency	Rated	150	Hz	max.
15	Current:	Rated	15	A rms	max. [1]
16		Peak	45	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	4.5 (6.1)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	14	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	0.647	Ohms	± 10 % [1]
22	Acceleration at Rated Torque		48945	rads/sec^2	Theoretical
23	Rotor Inertia		1028	kgm^2 * 1E-6	nominal
24	Damping		2.88	oz-in / krpm	nominal
25	Weight		37	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 5 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0038	µF	max.
33	IP Classification		65	rated	standard
34	Shaft:	Radial-Play	2E-5/7E-6	in/lb	reference
35		Material [5]	RC-#30	—	reference
36		Magnet Type	NdFeB	—	—
37		Loading [6] 1000 rpm	243.5	lbs.	max. [7]
		2000 rpm	193.3	lbs.	max. [7]
		3000 rpm	168.8	lbs.	max. [7]
		4000 rpm	N/A	lbs.	max. [7]
		5000 rpm	N/A	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2(P/2)		
42	Stator Phase Sequence—CW rotor rotation		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011775-xx		
47	Options:	Brake—24VDC (0.93A)—8.0 Nm holding torque IP67 Classification Incremental Encoder Tachometer No Keyway			

[1] 25 °C Ambient
[2] 40 °C Ambient
[3] Measured at 60 rpm (1 rps) in Velocity Mode
[4] Rated for 20,000 hours or 40,000 hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.

Z635 Motor Specifications

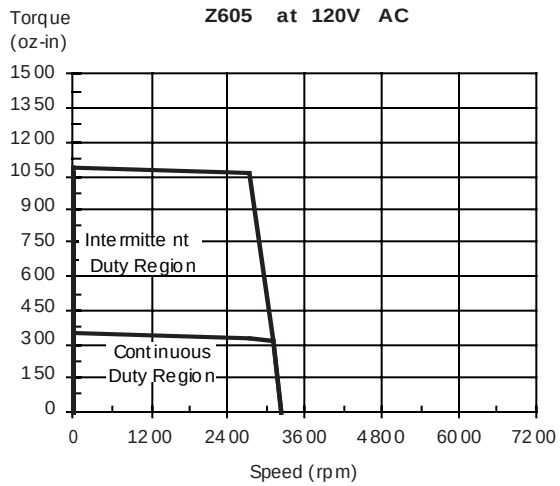
	Motor Size:	**Z640**	Value	Units	Tolerance
1	Constant (s):	Torque	291.5	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	124.5	V rms/Krpm	± 10%
3		Electrical Time	26.2	milliseconds	nominal
4		Mechanical Time	0.55	milliseconds	nominal
5		Thermal	33	minutes	nominal
6	Torque (s):	Continuous, Stall	4640	oz-in	min. [1]
7		Continuous, Stall	4378	oz-in	min. [2]
8		Continuous, Rated	3955	oz-in	min. [2]
9		Peak, Max w/o Saturation	12461	oz-in	min. [1]
10		Static Friction	73	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	max. [3]
12	Speed:	Rated	1600	rpm	reference
13		Maximum	1600	rpm	reference
14	Frequency	Rated	80	Hz	max.
15	Current:	Rated	15	A rms	max. [1]
16		Peak	45	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	4.7 (6.3)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	20	mH	± 30%
21	D.C. Resistance	Terminal (line-line)	0.763	Ohms	± 10 % [1]
22	Acceleration at Rated Torque		43667	rads/sec^2	Theoretical
23	Rotor Inertia		2034	kgm^2 * 1E-6	nominal
24	Damping		15.36	oz-in / krpm	nominal
25	Weight		51	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0082	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play	1E-5/4E-6	in/lb	reference
35		Material [5]	RC-#30	—	reference
36		Magnet Type	NdFeB	—	—
37		Loading [6] 1000 rpm	255.6	lbs.	max. [7]
		2000 rpm	N/A	lbs.	max. [7]
		3000 rpm	N/A	lbs.	max. [7]
		4000 rpm	N/A	lbs.	max. [7]
		5000 rpm	N/A	lbs.	max. [7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI #2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 3(P/2)		
42	Stator Phase Sequence—CW rotor rotation		A-C-B (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco # 21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011776-xx		
47	Options:	Brake—24VDC (1.27A)—6816 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer No Keyway			

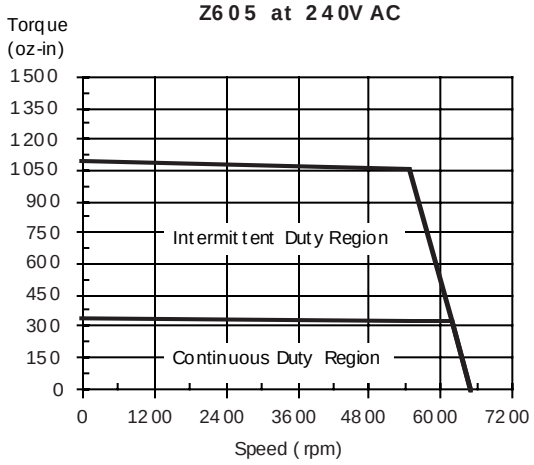
[1] 25 °C (Celsius) ambient
[2] 40 °C (Celsius) ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155° C
[5] Rotor steel is rated as *fatigue proof*
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

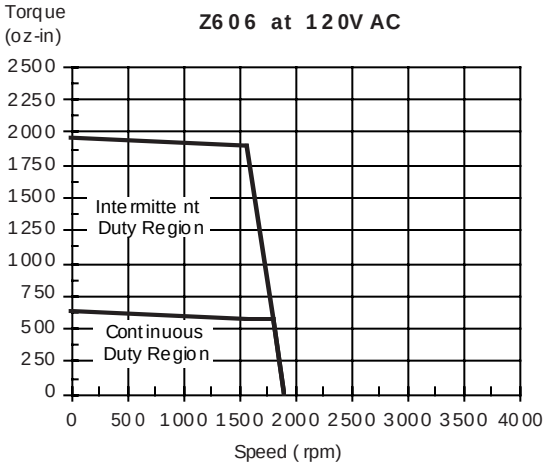
Z640 Motor Specifications

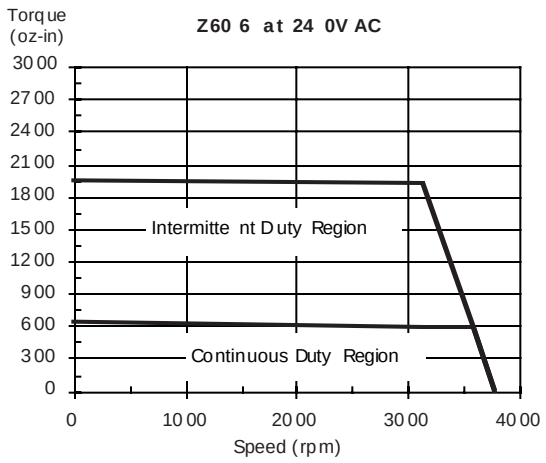
Speed/Torque Curves

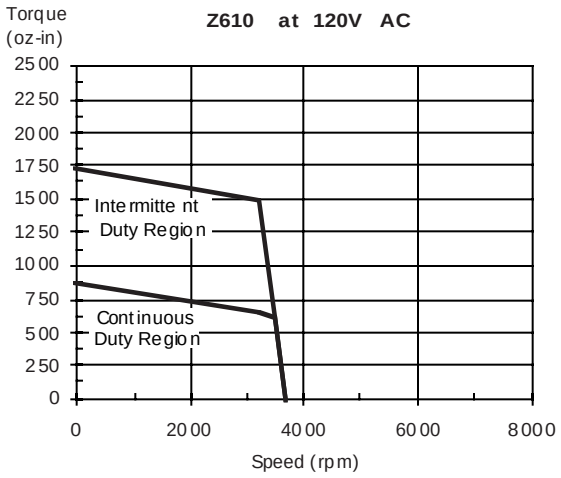
The speed/torque curves below represent available shaft torque at different speeds. Operation at 120VAC and 240VAC is shown for each motor size. Actual motor torque may vary ±10% due to motor manufacturing variances. For operation from a 1-phase 120VAC, the output torque stays relatively constant and the top-end speed falls off at the ratio of the input voltage (i.e., A Z610 operating at 240VAC has a rated speed of 7000 rpm; operating at 120VAC, it will have a rated speed of 3500 rpm). **Continuous Duty** means steady state operation for drive ambient temperatures of 0°C to 50°C. **Intermittent Duty** means operation for 3.3 seconds or less.

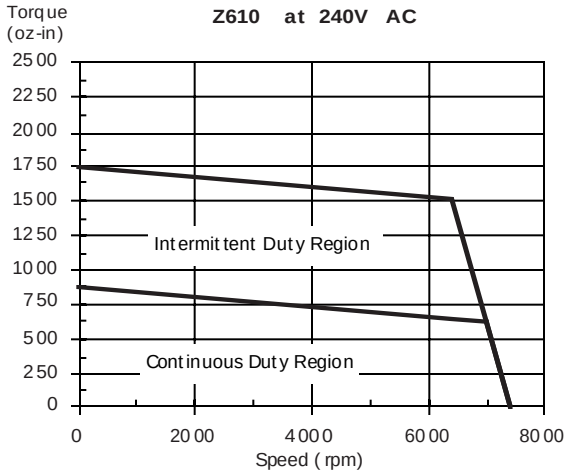


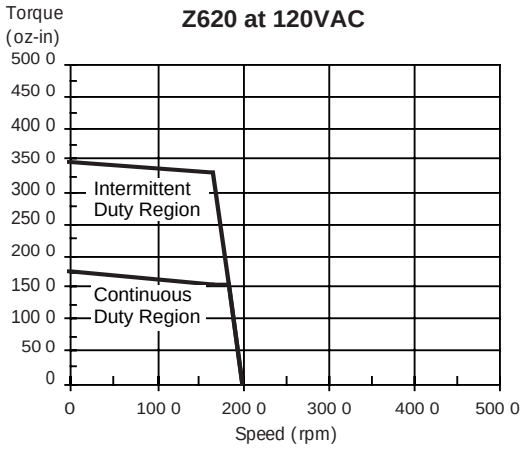


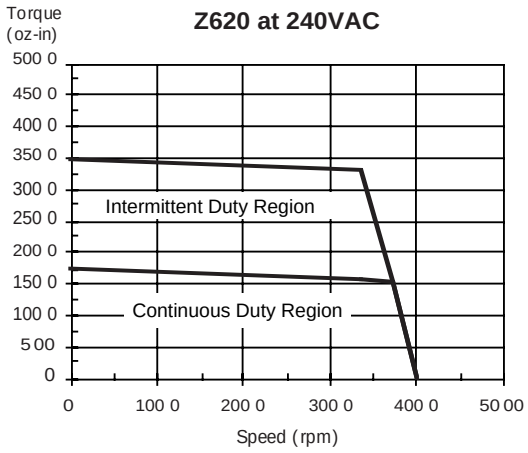


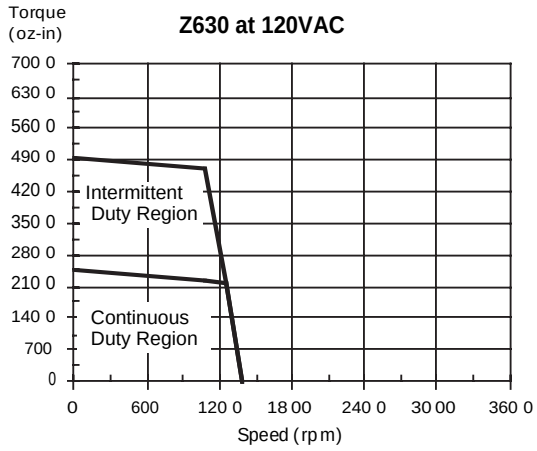


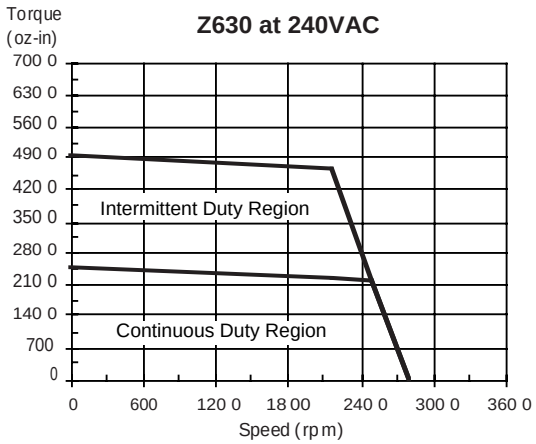


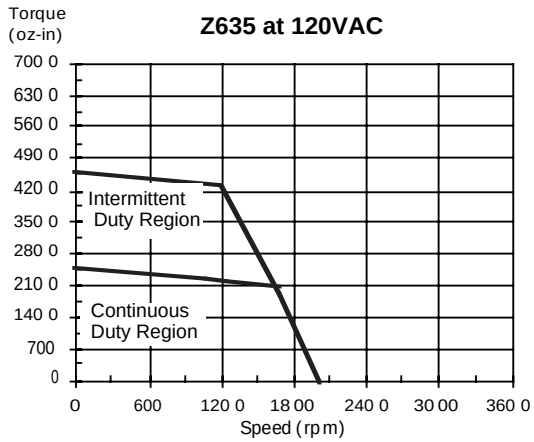


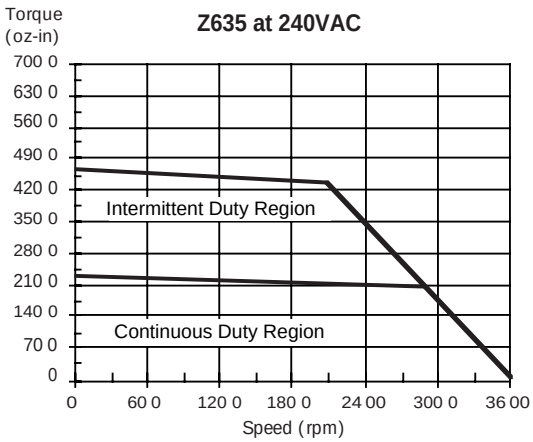

Z620 at 120VAC
Torque (oz-in)
500 0
450 0
400 0
350 0
300 0
250 0
200 0
150 0
100 0
50 0
0
Intermittent Duty Region
Continuous Duty Region
0
100 0
200 0
300 0
400 0
500 0
Speed (rpm)

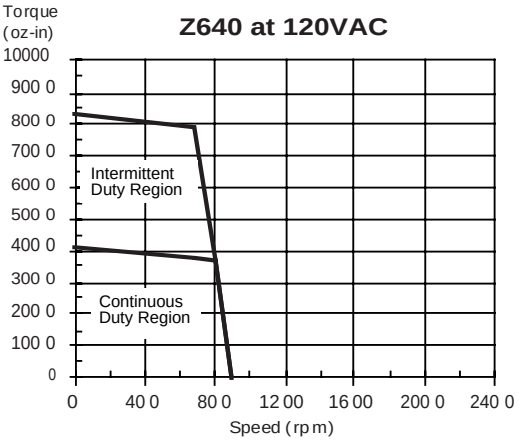

Z620 at 240VAC
Torque (oz-in)
500 0
450 0
400 0
350 0
300 0
250 0
200 0
150 0
100 0
5 00
0
Intermittent Duty Region
Continuous Duty Region
0
100 0
200 0
300 0
400 0
50 00
Speed (rpm)

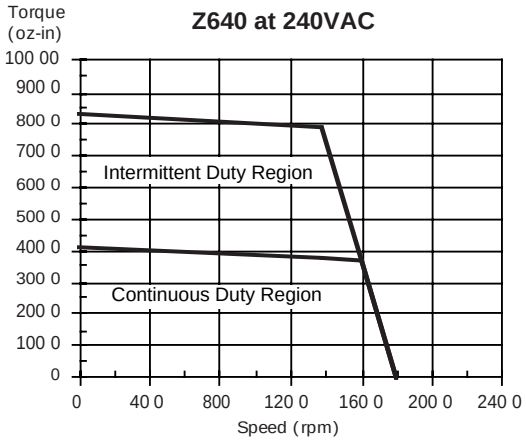

Z630 at 120VAC
Torque (oz-in)
700 0
630 0
560 0
490 0
420 0
350 0
280 0
210 0
140 0
700
0
Intermittent Duty Region
Continuous Duty Region
0
600
120 0
18 00
240 0
30 00
360 0
Speed (rpm)


Z630 at 240VAC
Torque (oz-in)
700 0
630 0
560 0
490 0
420 0
350 0
280 0
210 0
140 0
700
0
Intermittent Duty Region
Continuous Duty Region
0
600
120 0
18 00
240 0
300 0
360 0
Speed (rpm)


Z635 at 120VAC
Torque (oz-in)
700 0
630 0
560 0
490 0
420 0
350 0
280 0
210 0
140 0
700
0
Intermittent Duty Region
Continuous Duty Region
0
600
120 0
18 00
240 0
30 00
360 0
Speed (rpm)


Z635 at 240VAC
Torque (oz-in)
700 0
630 0
560 0
490 0
420 0
350 0
280 0
210 0
140 0
70 0
0
Intermittent Duty Region
Continuous Duty Region
0
60 0
120 0
180 0
24 00
300 0
36 00
Speed (rpm)


Z640 at 120VAC
Torque (oz-in)
10000
900 0
800 0
700 0
600 0
500 0
400 0
300 0
200 0
100 0
0
Intermittent Duty Region
Continuous Duty Region
0
40 0
80 0
12 00
16 00
200 0
240 0
Speed (rpm)


Z640 at 240VAC
Torque (oz-in)
100 00
900 0
800 0
700 0
600 0
500 0
400 0
300 0
200 0
100 0
0
Intermittent Duty Region
Continuous Duty Region
0
40 0
800
120 0
160 0
200 0
240 0
Speed (rpm)

ZX800 Electrical Specifications

Electrical specifications for the ZX800 series drive's input and output power are provided in this section.

Input Power

Voltage (Nominal)	120VAC (1 or 3-phase) **or** 120VAC (1 or 3-phase)
Voltage (Range)	92-130 VAC (1 or 3-phase) **or**
	205-252 VAC (1 or 3-phase)
Frequency (Range)	47-66 Hz
Current (Max. cont.)	15A (rms) 3-phase **or** 26A (rms) 1-phase
Power (Max. cont.)	6.2 KVA
Fuses	20A slow blow—Not user accessible
Isolation transformer	Not required

The actual input power and current is a function of the operating point of the motor (speed and torque) and the duty cycle. You can de-rate the fuse and the isolation transformer by scaling the above numbers by your actual requirements. The data above reflects the servo motor and drive operating at rated speed and at rated torque at 100% duty.

Output Power

Voltage	405 VDC (maximum)
Frequency	0 - 400Hz fundamental (7 kHz PWM)
Current	20A continuous per phase sinusoidal (14.1Arms)
	40A peak per phase sinusoidal (28.3Arms)
Regen/power dump	Optional accessory

Motor/Drive Configuration

The ZX800's hardware is pre-configured to control ZX800 series motors. The ZX800 drives only ZX800 series motors, the ZX600 drives only ZX600 motors, and the Z900 drives only Z900 series motors. *Be sure that your drive type matches your motor type (Z600, Z800, or Z900)*. If you have questions about the ZX Series motor/drive configuration, contact your local Automation Technology Center (ATC) or distributor.

Technical Data ZX800 Series

	Units	ZX-805	ZX-806	ZX-810	ZX-820	ZX-830	ZX-840
Continuous Stall Torque (*±10%)	oz-in lb-in lb-ft Nm	346 22 1.8	680 42 3.5 4.8	991 62 5.2 7.0	1997 125| 10.4 14.1	3101 194 16.2 21.9	4093 256 21.3 28.9
Peak Torque (*±10%)	oz-in lb-in lb-ft Nm	1005 63 5.2 7.1	1997 125 10.4 14.1	1983 124 10.3 14.1	3993 250 20.8 28.2	6203 388 32.3 43.8	8185 512 42.6 57.8
Rated Torque (*±10%)	oz-in lb-in lb-ft Nm	340 21 1.8 2.4	680 42 3.5 4.8	769 48 4.0 5.4	1651 103 8.6 11.7	2553 160 13.3 18.0	3554 222 18.5 25.1
Rated Power	hp kWatts	1.8 1.3	1.8 1.3	4.6 3.4	4.4 3.3	4.6 3.4	4.6 3.4
Rated Speed	rpm rps	5200 87	2600 43	6000 100	2700 45	1800 30	1300 22
Rated Current (line)	A (rms)	4.4	4.4	14.1	14.1	14.1	14.1
Peak Current (3.3 seconds max)	A (rms)	13.2	13.2	28.2	28.2	28.2	28.2
Max. Cont. AC input Current (3 phase 240VAC)	A (rms)	7	7	15	15	15	15
Rotor Inertia	oz-in^2 (mass) oz-in-sec^2 kg-m^2 x 1E-6	12.03 0.03 220	19.14 0.05 350	26.24 0.07 480	114.82 0.30 2100	166.76 0.43 3050	210.50 0.55 3850
Motor Weight	lbs kg(f)	11 5	15 7	20 9	37 17	48 22	59 27
Shipping Weight	lbs kg(f)	52 24	56 26	61 28	78 36	89 41	100 45

ZX800 Indexer/Drive Performance Specifications

Positional Repeatability

Repeatability: ±0.088 degrees, unloaded

Positional Accuracy

Resolver Accuracy: ±15 arc minutes

Resolver-to-Digital Converter Accuracy: ±8 arc minutes *(For finer accuracies, contact Compumotor—800-722-2282.)*

Motor/Drive Compatibility

Different motors can take different amounts of current and require different tuning parameters for typical loads. The **CMTR** (Configure Motor Type) command sets up a drive for a particular motor. By issuing **CMTR**, motor current levels and default parameters are recalled from memory. Do not exceed the current level specified for the motor, excessive current levels will damage the motor.

The following information is provided in case you must modify the motor/drive configuration. This command sequence will set up a drive for a particular motor size and perform the commutation (refer to the ***ZX Indexer/Drive Software Reference Guide*** for more on these commands).

> ***WARNING***
>
> This commutation procedure causes violent motor motion. All loads should be removed from the motor shaft before you begin this procedure.

Command	Description
> 1OFF	Turns drive off
> 1CMTRxxx	Sets drive for the motor; xxx = 805, 806, 810, 820, 830, or 840
> 1ON	Enables the drive

Motor Brakes

These brakes are mounted directly behind the motor and come completely assembled from the factory. When ordering the brake option, please specify the motor type.

Brake Characteristics	ZX805/806/810	ZX820/830/840	Units
Supply voltage	24	24	VDC
Supply current	1	1	A
Static braking torque	1345	4956	oz-in
Dynamic braking torque	1133	3965	oz-in
Inertia	0.007	0.063	oz-in/sec^2
Build-up time	90	140	ms
Release time	60	80	ms
Mass	3.9	11	lbs

Z800 Motor Brake Characteristics

Motor Data

The following pages provide data on each of the motor frame sizes of ZX800 series systems (ZX805, ZX806, ZX810, ZX820, ZX830, ZX840). The data reflecting motor torque does not assume operation from a ZX800 drive. The torque specifications reflect the motor's capabilities. In most cases, the motor windings match the drive's output power with an additional safety margin.

	Motor Size	**Z805**	Value	Units	Tolerance
1	Constant (s):	Torque	82.0	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	35.2	V rms/Krpm	± 10%
3		Electrical Time	2.8	milliseconds	nominal
4		Mechanical Time	2.3	milliseconds	nominal
5		Thermal	41	minutes	nominal
6	Torque (s):	Continuous, Stall	357	oz-in	min. [1]
7		Continuous, Stall	340	oz-in	min. [2]
8		Continuous, Rated	340	oz-in	min. [1]
9		Peak, Max w/o Saturation	1020	oz-in	min. [1]
10		Static Friction	—	oz-in	max
11		Ripple (of Rated Torque)	6	percent	max
12	Speed:	Rated	5300	rpm	reference
13		Maximum	5300	rpm	reference
14	Frequency:	Rated	353	Hz	max.
15	Current:	Rated	4.4	A rms	max. [1]
16		Peak	13.2	A rms	nominal
17	Voltage:	Rated	220	V rms	reference
18		Max	240	V rms	maximum
19	Output Power:	Rated	1.4 (1.9)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	5.6	mH	± 10%
21	DC Resistance	Terminal (line-line)	2.0	Ohms	± 10% [1]
22	Acceleration at Rated Torque		10900	rads/sec^2	Theoretical
23	Rotor Inertia		220	kgm^2 * 1E-6	nominal
24	Damping		--	oz-in / krpm	nominal
25	Weight		11.0	lbs.	max.
26	Winding Temperature		150	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		125	°C (Celsius)	reference
28	Insulation Class		F	—	reference
29	Thermostat TRIP Temperature		145	°C (Celsius)	± 5°C
30	Thermostat RESET Temperature		130	°C (Celsius)	± 10°C
31	Dielectric Strength, (Winding-to-Frame)		1000	VAC	min.
32	Winding Capacitance to Frame		--	µF	max.
33	IP Classification		54	rated	standard
34	Shaft:	Radial-Play (front to back)	0.001	in/lb	reference
35		Material [3]	UNI 5332	—	reference
36		Magnet Type	SmCo	—	—
37		Loading [4] 1000 rpm	123	lbs.	max.
		2000 rpm	101	lbs.	max.
		3000 rpm	81	lbs.	max.
		4000 rpm	74	lbs.	max.
		5000 rpm	67	lbs.	max.
38	Bearing Class, Internal/External		3/1	ABEC/AFBMA	reference
39	Bearing Grease		—	Manufacturer	reference
40	Shaft Seal Pressure		—	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 4 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 15 arc min.		
44	Resolver Manufacturer/Model #		Clifton Precision JSMBH-21-K-3		
45	Standard Resolver Cable Part Number		71-011449-xx		
46	Standard Motor Cable Part Number		71-007200-xx		
47	Options:	Brake—24VDC (1A)—1345 oz-in Holding Torque IP65 Classification Incremental Encoder Tachometer No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Material 38 Ni Cr Mo 4

[4] Loads centered 1 inch from mounting flange
Life expectancy = 20000 hours

Z805 Motor Specifications

	Motor Size	**Z806**	Value	Units	Tolerance
1	Constant (s):	Torque	164.0	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	70.4	V rms/Krpm	± 10%
3		Electrical Time	3.8	milliseconds	nominal
4		Mechanical Time	1.3	milliseconds	nominal
5		Thermal	40	minutes	nominal
6	Torque (s):	Continuous, Stall	714	oz-in	min. [1]
7		Continuous, Stall	680	oz-in	min. [2]
8		Continuous, Rated	680	oz-in	min. [1]
9		Peak, Max w/o Saturation	2040	oz-in	min. [1]
10		Static Friction	—	oz-in	max.
11		Ripple (of Rated Torque)	5	percent	max.
12	Speed:	Rated	2600	rpm	reference
		Maximum	2600	rpm	reference
14	Frequency	Rated	173	Hz	max.
15	Current:	Rated	4.4	A rms	max. [1]
16		Peak	13.2	A rms	nominal
17	Voltage:	Rated	220	V rms	reference
18		Max.	240	V rms	maximum
19	Output Power:	Rated	1.3 (1.7)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	11.3	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	3.0	Ohms	± 10% [1]
22	Acceleration at Rated Torque		13700	rads/sec^2	Theoretical
23	Rotor Inertia		350	kgm^2 * 1E-6	nominal
24	Damping		—	oz-in / krpm	nominal
25	Weight		15.5	lbs.	max.
26	Winding Temperature		150	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		125	°C (Celsius)	reference
28	Insulation Class		F	—	reference
29	Thermostat TRIP Temperature		145	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		130	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1000	VAC	min.
32	Winding Capacitance to Frame		—	µF	max.
33	IP Classification		54	rated	standard
34	Shaft:	Radial-Play (front to back)	0.001	in/lb	reference
35		Material [3]	UNI 5332	—	reference
36		Magnet Type	SmCo	—	—
37		Loading [4] 1000 rpm	135	lbs.	max.
		2000 rpm	105	lbs.	max.
		3000 rpm	90	lbs.	max.
		4000 rpm	—	lbs.	max.
		5000 rpm	—		max.
38	Bearing Class, Internal/External		3/1	ABEC/AFBMA	reference
39	Bearing Grease		--	Manufacturer	reference
40	Shaft Seal Pressure		--	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 4 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 15 arc min.		
44	Resolver Manufacturer/Model #		Clifton Precision JSMBH-21-K-3		
45	Standard Resolver Cable Part Number		71-011449-xx		
46	Standard Motor Cable Part Number		71-007200-xx		
47	Options:	Brake—24VDC (1A)—1345 oz-in Holding Torque			
		IP65 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Material 38 Ni Cr Mo 4
[4] Loads centered 1 inch from mounting flange
Life expectancy = 20000 hours

Z806 Motor Specifications

	Motor Size	**Z810**	Value	Units	Tolerance
1	Constant (s):	Torque	74.0	oz-in/A rms	± 10%
2		Voltage ()	31.7	V rms/Krpm	± 10%
3		Electrical Time	8.5	milliseconds	nominal
4		Mechanical Time	1.8	milliseconds	nominal
5		Thermal	36	minutes	nominal
6	Torque (s):	Continuous, Stall	1086	oz-in	min. [1]
7		Continuous, Stall	1034	oz-in	min. [2]
8		Continuous, Rated	770	oz-in	min. [1]
9		Peak, Max w/o Saturation	3100	oz-in	min. [1]
10		Static Friction	—	oz-in	max.
11		Ripple (of Rated Torque)	15	percent	max.
12	Speed:	Rated	6000	rpm	reference
13		Maximum	6000	rpm	reference
14	Frequency	Rated	400	Hz	max.
15	Current:	Rated	14.0	A rms	max. [1]
16		Peak	42.0	A rms	nominal
17	Voltage:	Rated	220	V rms	reference
18		Max	240	V rms	maximum
19	Output Power:	Rated	3.4 (4.5)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	1.4	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.6	Ohms	± 10% [1]
22	Acceleration at Rated Torque		15200	rads/sec^2	Theoretical
23	Rotor Inertia		480	kgm^2 * 1E-6	nominal
24	Damping		—	oz-in/krpm	nominal
25	Weight		19.5	lbs.	max.
26	Winding Temperature		150	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		125	°C (Celsius)	reference
28	Insulation Class		F	—	reference
29	Thermostat TRIP Temperature		145	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		130	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1000	VAC	min.
32	Winding Capacitance to Frame		—	µF	max.
33	IP Classification		54	rated	standard
34	Shaft:	Radial-Play (front to back)	0.001	in/lb	reference
35		Material [3]	UNI 5332	—	reference
36		Magnet Type	SmCo	—	—
37		Loading [4] 1000 rpm	146	lbs.	max.
		2000 rpm	112	lbs.	max.
		3000 rpm	92	lbs.	max.
		4000 rpm	85	lbs.	max.
		5000 rpm	81	percent	max.
38	Bearing Class, Internal/External		3/1	ABEC/AFBMA	reference
39	Bearing Grease		—	Manufacturer	reference
40	Shaft Seal Pressure		—	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 4 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 15 arc min.		
44	Resolver Manufacturer/Model #		Clifton Precision JSMBH-21-K-3		
45	Standard Resolver Cable Part Number		71-011449-xx		
46	Standard Motor Cable Part Number		71-007130-xx		
47	Options:	Brake—24VDC (1A)—1345 oz-in Holding Torque			
		IP65 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Material 38 Ni Cr Mo 4
[4] Loads centered 1 inch from mounting flange
Life expectancy = 20000 hours

Z810 Motor Specifications

	Motor Size	**Z820**	Value	Units	Tolerance
1	Constant (s):	Torque	156.0	oz-in/A rms	± 10%
2		Voltage ()	66.5	V rms/Krpm	± 10%
3		Electrical Time	8.5	milliseconds	nominal
4		Mechanical Time	1.8	milliseconds	nominal
5		Thermal	63	minutes	nominal
6	Torque (s):	Continuous, Stall	2171	oz-in	min. [1]
7		Continuous, Stall	2068	oz-in	min. [2]
8		Continuous, Rated	1652	oz-in	min. [1]
9		Peak, Max w/o Saturation	6200	oz-in	min. [1]
10		Static Friction	—	oz-in	max.
11		Ripple (of Rated Torque)	3	percent	max.
12	Speed:	Rated	2700	rpm	reference
13		Maximum	2700	rpm	reference
14	Frequency	Rated	180	Hz	max.
15	Current:	Rated	14.0	A rms	max. [1]
16		Peak	42.0	A rms	nominal
17	Voltage:	Rated	220	V rms	reference
18		Max	240	V rms	maximum
19	Output Power:	Rated	3.3 (4.4)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	5.1	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.6	Ohms	± 10% [1]
22	Acceleration at Rated Torque		7000	rads/sec^2	Theoretical
23	Rotor Inertia		2100	kgm^2 * 1E-6	nominal
24	Damping		—	oz-in / krpm	nominal
25	Weight		37.8	lbs.	max.
26	Winding Temperature		150	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		125	°C (Celsius)	reference
28	Insulation Class		F	—	reference
29	Thermostat TRIP Temperature		145	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		130	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1000	VAC	min.
32	Winding Capacitance to Frame		—	µF	max.
33	IP Classification		54	rated	standard
34	Shaft:	Radial-Play (front to back)	0.001	in/lb	reference
35		Material [3]	UNI 5332	—	reference
36		Magnet Type	SmCo	—	—
37		Loading [4] 1000 rpm	360	lbs.	max.
		2000 rpm	281	lbs.	max.
		3000 rpm	236	lbs.	max.
		4000 rpm	—	lbs.	max.
		5000 rpm	—	lbs.	max.
38	Bearing Class, Internal/External		3/1	ABEC/AFBMA	reference
39	Bearing Grease		—	Manufacturer	reference
40	Shaft Seal Pressure		—	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 4 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 15 arc min.		
44	Resolver Manufacturer/Model #		Clifton Precision JSMBH-21-K-3		
45	Standard Resolver Cable Part Number		71-011449-xx		
46	Standard Motor Cable Part Number		71-007130-xx		
47	Options:	Brake—24VDC (1.7A)—4956 oz-in Holding Torque IP65 Classification Incremental Encoder Tachometer No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Material 38 Ni Cr Mo 4
[4] Loads centered 1 inch from mounting flange
Life expectancy = 20000 hours

Z820 Motor Specifications

	Motor Size	**Z830**	Value	Units	Tolerance
1	Constant (s):	Torque	234.0	oz-in/A rms	± 10%
2		Voltage ()	99.8	V rms/Krpm	± 10%
3		Electrical Time	9.6	milliseconds	nominal
4		Mechanical Time	1.6	milliseconds	nominal
5		Thermal	67	minutes	nominal
6	Torque (s):	Continuous, Stall	3271	oz-in	min. [1]
7		Continuous, Stall	3115	oz-in	min. [2]
8		Continuous, Rated	2554	oz-in	min. [1]
9		Peak, Max w/o Saturation	9345	oz-in	min. [1]
10		Static Friction	—	oz-in	max.
11		Ripple (of Rated Torque)	6	percent	max.
12	Speed:	Rated	1800	rpm	reference
13		Maximum	1800	rpm	reference
14	Frequency	Rated	120	Hz	max.
15	Current:	Rated	14.0	A rms	max. [1]
16		Peak	42.0	A rms	nominal
17	Voltage:	Rated	220	V rms	reference
18		Max	240	V rms	maximum
19	Output Power:	Rated	3.4 (4.6)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	7.7	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.8	Ohms	± 10 % [1]
22	Acceleration at Rated Torque		7200	rads/sec^2	Theoretical
23	Rotor Inertia		3050	kgm^2 * 1E-6	nominal
24	Damping		—	oz-in / krpm	nominal
25	Weight		48.0	lbs.	max.
26	Winding Temperature		150	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		125	°C (Celsius)	reference
28	Insulation Class		F	—	reference
29	Thermostat TRIP Temperature		145	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		130	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1000	VAC	min.
32	Winding Capacitance to Frame		—	µF	max.
33	IP Classification		54	rated	standard
34	Shaft:	Radial-Play (front to back)	0.001	in/lb	reference
35		Material [3]	UNI 5332	—	reference
36		Magnet Type	SmCo	—	max
37		Loading [4] 1000 rpm	370	lbs.	max
		2000 rpm	281	lbs.	max
		3000 rpm	—	lbs.	max
		4000 rpm	—	lbs.	max
		5000 rpm	—	lbs.	max
38	Bearing Class, Internal/External		3/1	ABEC/AFBMA	reference
39	Bearing Grease		—	Manufacturer	reference
40	Shaft Seal Pressure		—	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 4 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 15 arc min.		
44	Resolver Manufacturer/Model #		Clifton Precision JSMBH-21-K-3		
45	Standard Resolver Cable Part Number		71-011449-xx		
46	Standard Motor Cable Part Number		71-007130-xx		
47	Options:	Brake—24VDC (1.7A)—4956 oz-in Holding Torque			
		IP65 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Material 38 Ni Cr Mo 4
[4] Loads centered 1 inch from mounting flange
Life expectancy = 20000 hours

Z830 Motor Specifications

	Motor Size	**Z840**	Value	Units	Tolerance
1	Constant (s):	Torque	312.0	oz-in/A rms	± 10%
2		Voltage ()	133.0	V rms/Krpm	± 10%
3		Electrical Time	10.2	milliseconds	nominal
4		Mechanical Time	1.4	milliseconds	nominal
5		Thermal	68	minutes	nominal
6	Torque (s):	Continuous, Stall	4311	oz-in	min. [1]
7		Continuous, Stall	4106	oz-in	min. [2]
8		Continuous, Rated	3554	oz-in	min. [1]
9		Peak, Max w/o Saturation	12318	oz-in	min. [1]
10		Static Friction	—	oz-in	max.
11		Ripple (of Rated Torque)	8	percent	max.
12	Speed:	Rated	1300	rpm	reference
13		Maximum	1300	rpm	reference
14	Frequency	Rated	86.7	Hz	max.
15	Current:	Rated	14.0	A rms	max. [1]
16		Peak	42.0	A rms	nominal
17	Voltage:	Rated	220	V rms	reference
18		Maximum	240	V rms	maximum
19	Output Power:	Rated	3.5 (4.7)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	10.2	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	1.0	Ohms	± 10 % [1]
22	Acceleration at Rated Torque		7500	rads/sec^2	Theoretical
23	Rotor Inertia		3850	kgm^2 * 1E-6	nominal
24	Damping		--	oz-in / krpm	nominal
25	Weight		59.0	lbs.	max.
26	Winding Temperature		150	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		125	°C (Celsius)	reference
28	Insulation Class		F	—	reference
29	Thermostat TRIP Temperature		145	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		130	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1000	VAC	min.
32	Winding Capacitance to Frame		—	µF	max.
33	IP Classification		54	rated	standard
34	Shaft:	Radial-Play (front to back)	0.001	in/lb	reference
35		Material [3]	UNI 5332	lbs.	maximum
36		Magnet Type	SmCo	lbs.	maximum
37		Loading [4] 1000 rpm	393	lbs.	maximum
		2000 rpm	—	lbs.	maximum
		3000 rpm	—	lbs.	maximum
		4000 rpm	—	lbs.	maximum
		5000 rpm	—	lbs.	maximum
38	Bearing Class, Internal/External		3/1	ABEC/AFBMA	reference
39	Bearing Grease		—	Manufacturer	reference
40	Shaft Seal Pressure		—	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 4 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 15 arc min.		
44	Resolver Manufacturer/Model #		Clifton Precision JSMBH-21-K-3		
45	Standard Resolver Cable Part Number		71-011449-xx		
46	Standard Motor Cable Part Number		71-007130-xx		
47	Options:	Brake—24VDC (1.7A)—4956 oz-in Holding Torque IP65 Classification Incremental Encoder Tachometer No Keyway			

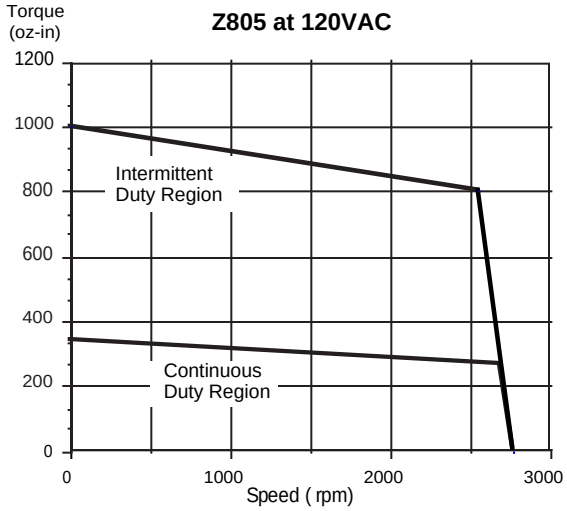
① 25 °C ambient
② 40 °C ambient
③ Material 38 Ni Cr Mo 4
④ Loads centered 1 inch from mounting flange
Life Expectancy = 20000 hours

Z840 Motor Specifications

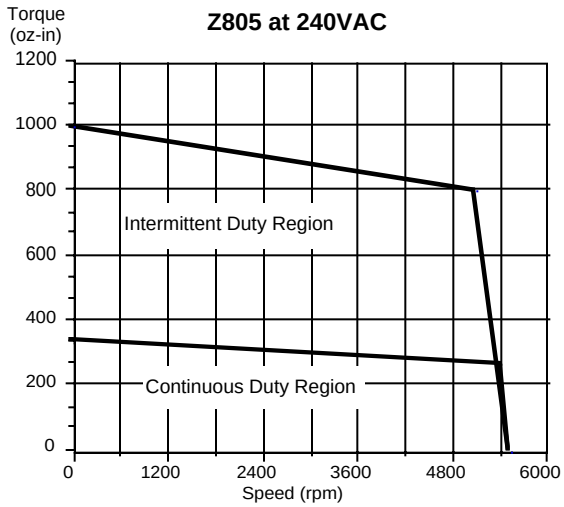
Speed/Torque Curves

The speed/torque curves below represent available shaft torque at different speeds. Operation at 120VAC and 240VAC is shown for each motor size. Actual motor torque may vary ±10% due to motor manufacturing variances. For operation from a 1-phase 120VAC, the output torque stays relatively constant and the top-end speed falls off at the ratio of the input voltage (i.e., A Z810 operating at 240VAC has a rated speed of 6000 rpm; operating at 120VAC, it will have a rated speed of 3000 rpm). **Continuous Duty** means steady state operation for drive ambient temperatures of 0°C to 50°C. **Intermittent Duty** means operation for 3.3 seconds or less.

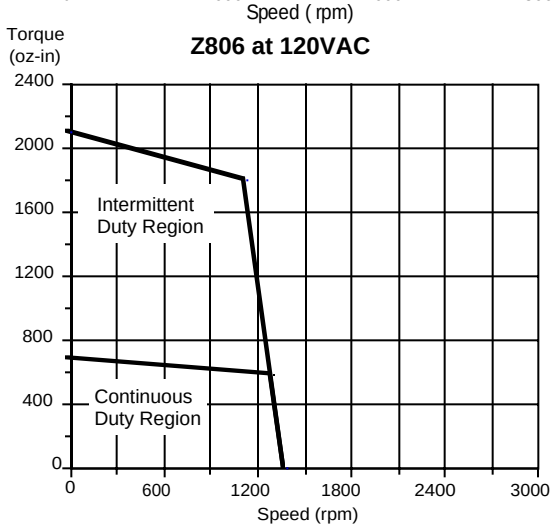
Z805 at 120VAC



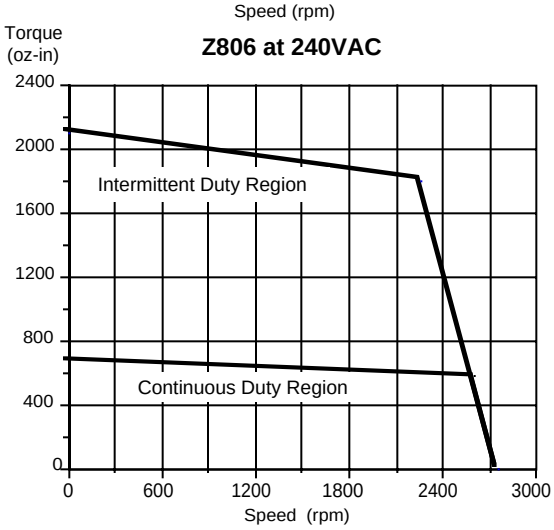
Z805 at 240VAC



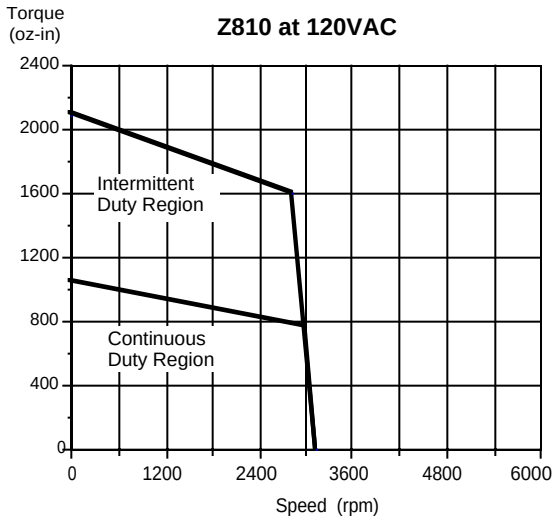
Z806 at 120VAC



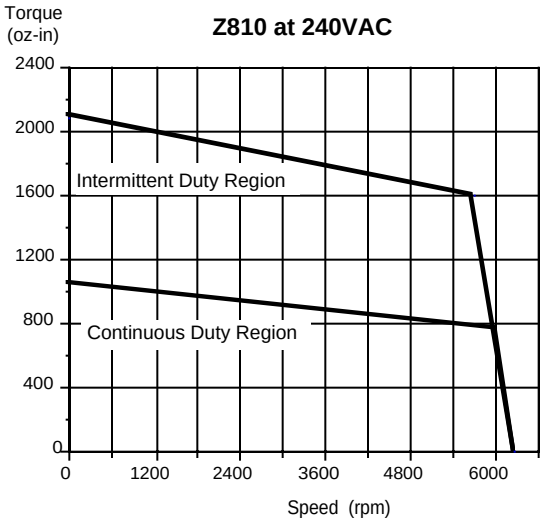
Z806 at 240VAC

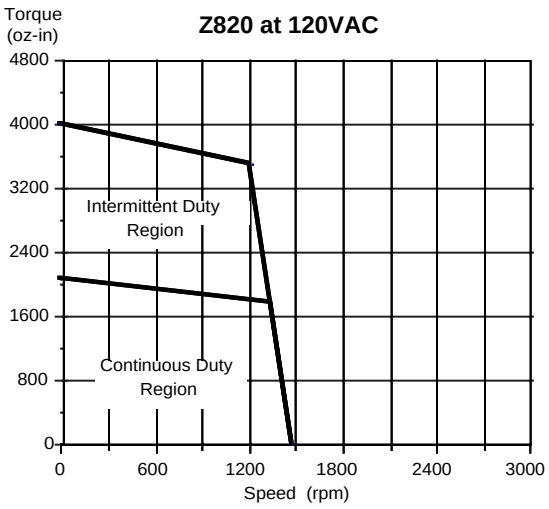


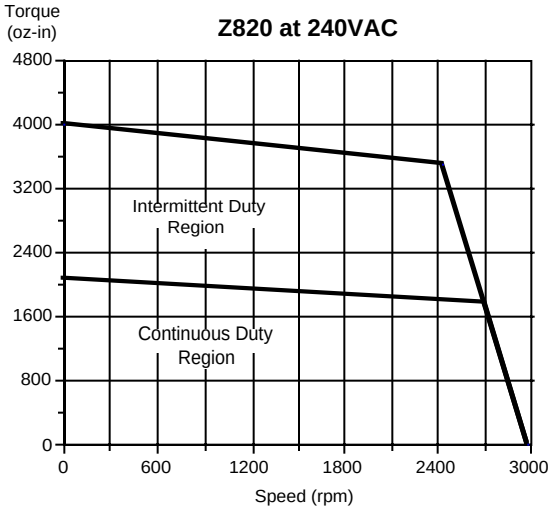
Z810 at 120VAC

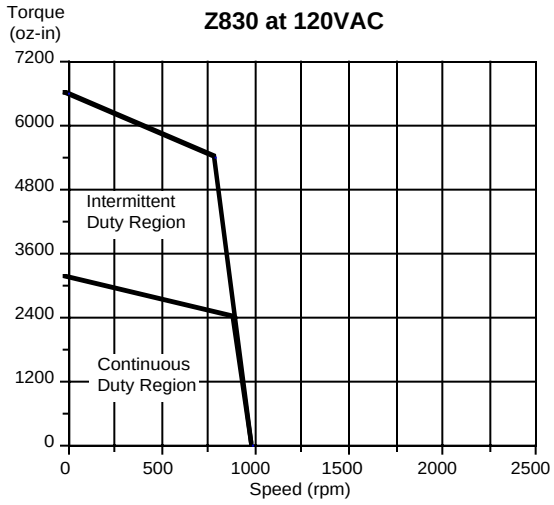


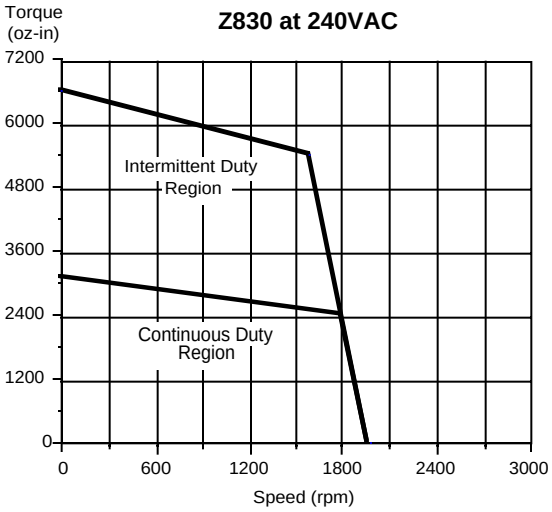
Z810 at 240VAC

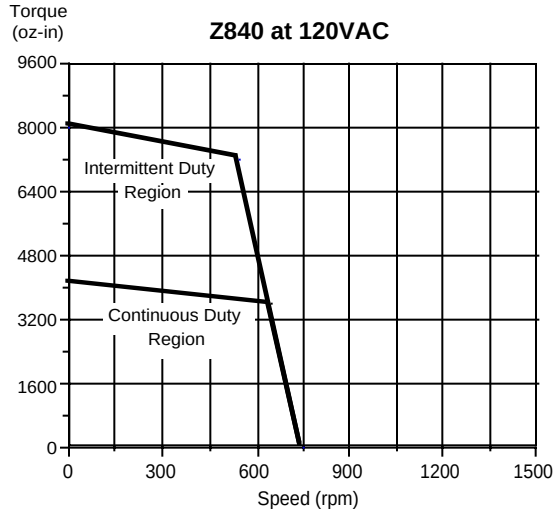


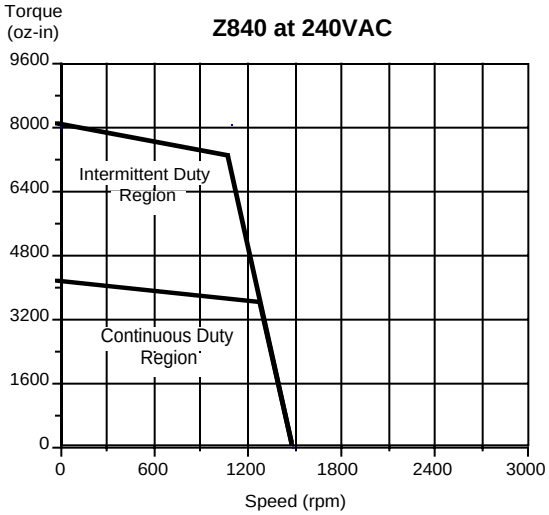

Z820 at 120VAC
Torque (oz-in)
4800
4000
3200
2400
1600
800
0
Intermittent Duty Region
Continuous Duty Region
0
600
1200
1800
2400
3000
Speed (rpm)


Z820 at 240VAC
Torque (oz-in)
4800
4000
3200
2400
1600
800
0
Intermittent Duty Region
Continuous Duty Region
0
600
1200
1800
2400
3000
Speed (rpm)


Z830 at 120VAC
Torque (oz-in)
7200
6000
4800
3600
2400
1200
0
Intermittent Duty Region
Continuous Duty Region
0
500
1000
1500
2000
2500
Speed (rpm)


Z830 at 240VAC
Torque (oz-in)
7200
6000
4800
3600
2400
1200
0
Intermittent Duty Region
Continuous Duty Region
0
600
1200
1800
2400
3000
Speed (rpm)


Z840 at 120VAC
Torque (oz-in)
9600
8000
6400
4800
3200
1600
0
Intermittent Duty Region
Continuous Duty Region
0
300
600
900
1200
1500
Speed (rpm)


Z840 at 240VAC
Torque (oz-in)
9600
8000
6400
4800
3200
1600
0
Intermittent Duty Region
Continuous Duty Region
0
600
1200
1800
2400
3000
Speed (rpm)

ZX900 Electrical Specifications

Electrical specifications for the ZX900 series drive's input and output power are provided in this section.

Input Power

Voltage (Nominal)	240VAC (1-phase or 3-phase)
Voltage (Range)	205-252 VAC (1 or 3-phase)
Frequency (Range)	47-66 Hz
Current (Max. cont.)	30A (rms) 3-phase
Power (Max. cont.)	12.4 KVA
Fuses	40A slow blow—User accessible
Isolation transformer	Not required

The actual input power and current is a function of the operating point of the motor (speed and torque) and the duty cycle. You can de-rate the fuse and the isolation transformer by scaling the above numbers by your actual requirements. The data above reflects the servo motor and drive operating at rated speed and at rated torque at 100% duty.

Output Power

Voltage	405 VDC (maximum)
Frequency	0 - 400Hz fundamental (7 kHz PWM)
Current	40A continuois per phase sinusoidal (28.28 Arms) 80A per phase peak (56.56 Arms)
Regen/power dump	Optional accessory

Motor/Drive Configuration

The ZX900's hardware is pre-configured to control ZX900 series motors. The ZX900 series drives only ZX900 series motors, the ZX800 series drives only Z800 motors, and ZX600 series drives only ZX600 motors. *Be sure that your drive type matches your motor type (Z600, Z800, or Z900).* If you have questions about the ZX Series motor/drive configuration, contact your local Automation Technology Center (ATC) or distributor.

Technical Data ZX900 Series

	Units	ZX-910	ZX-920	ZX-930	ZX-940
Continuous Stall Torque (*±10%)	oz-in lb-in lb-ft Nm	2407 150 12.5 17.0	4263 266 22.2 30.1	5990 374 31.2 42.3	9021 564 47.0 63.7
Peak Torque (*±10%)	oz-in lb-in lb-ft Nm	5205 325 27 37	8525 533 44 60	11980 749 62 85	18041 1128 94 127
Rated Torque (*±10%)	oz-in lb-in lb-ft Nm	1939 121 10.1 13.7	3341 209 17.4 23.6	4830 302 25.2 34.1	7480 467 39.0 52.8
Rated Power	hp kWatts	9.6 7.2	10.4 7.8	11.0 8.2	11.1 8.3
Rated Speed	rpm rps	5000 83.3	3150 52.5	2300 38.3	1500 25.0
Rated Current (line)	A (rms)	27.2	27.7	28.3	28.3
Peak Current (3.3 seconds max)	A (rms)	56.6	56.6	56.6	56.6
Max. Cont. AC input Current (3 phase 240VAC)	A (rms)	30	30	30	30
Rotor Inertia	oz-in^2 (mass) oz-in-sec^2 kg-m^2 x 1E-6	50.79 0.132 92.9	111.21 0.288 2034	166.21 0.431 3040	459.48 1.190 8404
Motor Weight	lbs kg(f)	32 15	57 26	65 29	112 51
Shipping Weight	lbs kg(f)	89 40	114 52	122 55	169 77

ZX900 Indexer/Drive Performance Specifications

Positional Repeatability

Repeatability: ±0.088 degrees, unloaded

Positional Accuracy

Resolver Accuracy: ±7 arc minutes

Resolver-to-Digital Converter Accuracy: ±8 arc minutes *(For finer accuracies, contact Compumotor—800-358-9070.)*

Motor/Drive Compatibility

Different motors can take different amounts of current and require different tuning parameters for typical loads. The **CMTR** (Configure Motor Type) command sets up a drive for a particular motor. By issuing **CMTR**, motor current levels and default parameters are recalled from memory. Do not exceed the current level specified for the motor, excessive current levels will damage the motor.

The following information is provided in case you must modify the motor/drive configuration. This command sequence will set up a drive for a particular motor size and perform the commutation (refer to the ***ZX Indexer/Drive Software Reference Guide*** for more on these commands).

> ***WARNING***
>
> This commutation procedure causes violent motor motion. All loads should be removed from the motor shaft before you begin this procedure.

Command	Description
> 1OFF	Turns drive off
> 1CMTRxxx	Sets drive for the motor; xxx =910, 920, 930, or 940
> 1ON	Enables the drive

Motor Brakes

These brakes are mounted directly behind the motor and come completely assembled from the factory. When ordering the brake option, please specify the motor type.

Brake Characteristics	Z910	Z920/930	Z940	Units
Supply voltage	24	24	24	VDC
Supply current	0.93	1.27	*	A
Static braking torque	1152	6720	*	oz-in

**Same as Z920/930 Characteristics*

Z900 Motor Brake Characteristics

Motor Data

The following pages provide data on each of the four motor frame sizes of ZX900 series systems (ZX910, ZX920, ZX930, ZX940). The data reflecting motor torque does not assume operation from a ZX900 drive. The torque specifications reflect the motor's capabilities. In most cases, the motor windings match the drive's output power with an additional safety margin.

	Motor Size	**Z910**	Value	Units	Tolerance
1	Constant (s):	Torque	88.32	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	26.7	V rms/Krpm	± 10%
3		Electrical Time	24	milliseconds	nominal
4		Mechanical Time	0.672	milliseconds	nominal
5		Thermal	43	minutes	nominal
6	Torque (s):	Continuous, Stall	2544	oz-in	min. [1]
7		Continuous, Stall	2400	oz-in	min. [2]
8		Continuous, Rated	1939	oz-in	min. [2]
9		Peak, Max w/o Saturation	6835	oz-in	min. [1]
10		Static Friction	40.7	oz-in	max
11		Ripple (of Rated Torque)	4.5	percent	max [3]
12	Speed:	Rated	5000	rpm	reference
13		Maximum	5000	rpm	reference
14	Frequency:	Rated	166.6	Hz	max.
15	Current:	Rated	27.2	A rms	max. [1]
16		Peak	81.4	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	7.2 (9.6)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	4.8	mH	± 30%
21	DC Resistance	Terminal (line-line)	0.2	Ohms	± 10% [1]
22	Acceleration at Rated Torque		51971	rads/sec^2	Theoretical
23	Rotor Inertia		929	kgm^2 * 1E-6	nominal
24	Damping		1.728	oz-in / krpm	nominal
25	Weight		32	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		135	°C (Celsius)	± 5°C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10°C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.00205	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	2E-5/7E-6	in/lb	reference
35		Material [3]	EC#30	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	160	lbs.	max.[7]
		2000 rpm	127.1	lbs.	max.[7]
		3000 rpm	110.9	lbs.	max.[7]
		4000 rpm	100.8	lbs.	max.[7]
		5000 rpm	93.3	lbs.	max.[7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		oz-in/A rms ±7 arc min.		
44	Resolver Manufacturer/Model #		Facso #21-BRCX-334-J39		
45	Standard Resolver Cable Part Number		71-011776-xx		
46	Standard Motor Cable Part Number		71-011776-xx		
47	Options:	Brake—24VDC (1A)—1152 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

	Motor Size	**Z920**	Value	Units	Tolerance
1	Constant (s):	Torque	154.2	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	46.6	V rms/Krpm	± 10%
3		Electrical Time	26.9	milliseconds	nominal
4		Mechanical Time	0.6	milliseconds	nominal
5		Thermal	46	minutes	nominal
6	Torque (s):	Continuous, Stall	4518	oz-in	min. [1]
7		Continuous, Stall	4262	oz-in	min. [2]
8		Continuous, Rated	3341	oz-in	min. [2]
9		Peak, Max w/o Saturation	12422	oz-in	min. [1]
10		Static Friction	65	oz-in	max
11		Ripple (of Rated Torque)	4.5	percent	max [3]
12	Speed:	Rated	3150	rpm	reference
13		Maximum	3150	rpm	reference
14	Frequency:	Rated	157.5	Hz	max.
15	Current:	Rated	27.7	A rms	max. [1]
16		Peak	84.8	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	7.7 (10.4)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	6.3	mH	± 30%
21	DC Resistance	Terminal (line-line)	0.234	Ohms	± 10% [1]
22	Acceleration at Rated Torque		43133	rads/sec^2	Theoretical
23	Rotor Inertia		2034	kgm^2 * 1E-6	nominal
24	Damping		15.36	oz-in / krpm	nominal
25	Weight		51	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5°C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10°C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.00034	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	2E-5/7E-6	in/lb	reference
35		Material [3]	RC#30	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	255.6	lbs.	max.[7]
		2000 rpm	202.9	lbs.	max.[7]
		3000 rpm	177.2	lbs.	max.[7]
		4000 rpm	N/A	lbs.	max.[7]
		5000 rpm	N/A	lbs.	max.[7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 3 (P/2)		
42	Stator Phase Sequence		A-B-C (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited;±7 arc min.		
44	Resolver Manufacturer/Model #		Facso #21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011777-xx		
47	Options:	Brake—24VDC (1.27A)—6720 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

	Motor Size	**Z930**	Value	Units	Tolerance
1	Constant (s):	Torque	212	oz-in/A rms	± 10%
2		Voltage ()	64	V rms/Krpm	± 10%
3		Electrical Time	31.1	milliseconds	nominal
4		Mechanical Time	0.53	milliseconds	nominal
5		Thermal	50	minutes	nominal
6	Torque (s):	Continuous, Stall	6513	oz-in	min. [1]
7		Continuous, Stall	6144	oz-in	min. [2]
8		Continuous, Rated	4954	oz-in	min. [1]
9		Peak, Max w/o Saturation	17357	oz-in	min. [1]
10		Static Friction	133.2	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	max.
12	Speed:	Rated	2300	rpm	reference
13		Maximum	2300	rpm	reference
14	Frequency	Rated	115	Hz	max.
15	Current:	Rated	29	A rms	max. [1]
16		Peak	86.2	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	8.4 (11.3)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	8.1	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.26	Ohms	± 10% [1]
22	Acceleration at Rated Torque		40357	rads/sec^2	Theoretical
23	Rotor Inertia		3040	kgm^2 * 1E-6	nominal
24	Damping		2.88	oz-in/krpm	nominal
25	Weight		65	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0038	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	2E-5/7E-6	in/lb	reference
35		Material [3]	RC#30	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	263	lbs.	max.
		2000 rpm	208.8	lbs.	max.
		3000 rpm	N/A	lbs.	max.
		4000 rpm	N/A	lbs.	max.
		5000 rpm	N/A	percent	max.
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco #21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-01		
46	Standard Motor Cable Part Number		71-012979-01		
47	Options:	Brake—24VDC (1.27A)—6720 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

	Motor Size	**Z940**	Value	Units	Tolerance
1	Constant (s):	Torque	319	oz-in/A rms	± 10%
2		Voltage ()	96.3	V rms/Krpm	± 10%
3		Electrical Time	56.5	milliseconds	nominal
4		Mechanical Time	0.57	milliseconds	nominal
5		Thermal	56	minutes	nominal
6	Torque (s):	Continuous, Stall	12007	oz-in	min. [1]
7		Continuous, Stall	11328	oz-in	min. [2]
8		Continuous, Rated	9388	oz-in	min. [2]
9		Peak, Max w/o Saturation	32870	oz-in	min. [1]
10		Static Friction	128.6	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	max.[3]
12	Speed:	Rated	1500	rpm	reference
13		Maximum	1500	rpm	reference
14	Frequency	Rated	75	Hz	max.
15	Current:	Rated	35.5	A rms	max. [1]
16		Peak	108.5	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	10.4 (14.0)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	13	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.23	Ohms	± 10% [1]
22	Acceleration at Rated Torque		27613	rads/sec^2	Theoretical
23	Rotor Inertia		8400	kgm^2 * 1E-6	nominal
24	Damping		15.36	oz-in / krpm	nominal
25	Weight		112	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0082	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	1E-5/4E-6	in/lb	reference
35		Material [3]	RC#3	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	365.9	lbs.	max.
		2000 rpm	N/A	lbs.	max.
		3000 rpm	N/A	lbs.	max.
		4000 rpm	N/A	lbs.	max.
		5000 rpm	N/A	lbs.	max.
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 3 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Faxco #21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-01		
46	Standard Motor Cable Part Number		71-012979-01		
47	Options:	Brake—24VDC (1.27A)—6816 oz-in Holding Torque			
		IP65 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

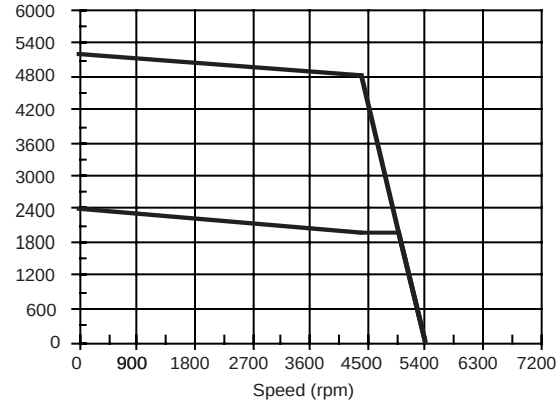
[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z820 Motor Specifications

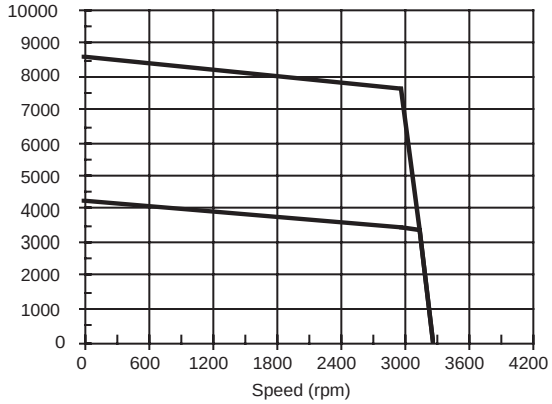
Speed/Torque Curves

The following speed/torque curves represent the available shaft torque at different operating speeds. Operation at 240VAC is shown for each motor size. Actual motor torque may vary ±10% due to motor manufacturing variances.

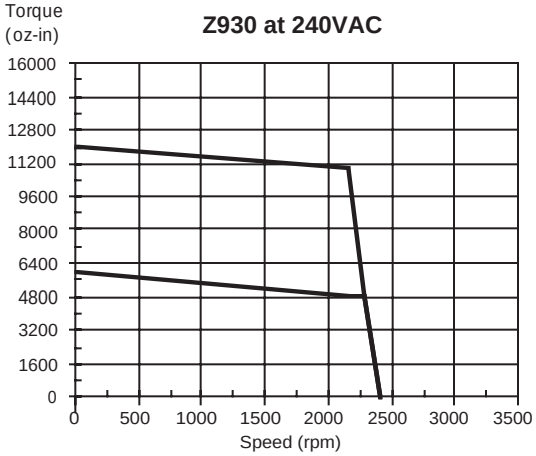
Z910 at 240VAC



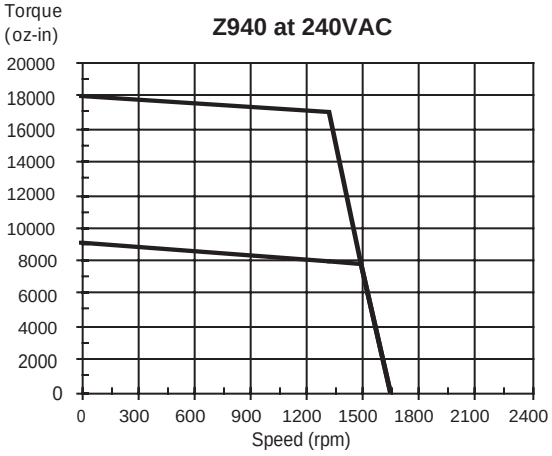
Z920 at 240VAC



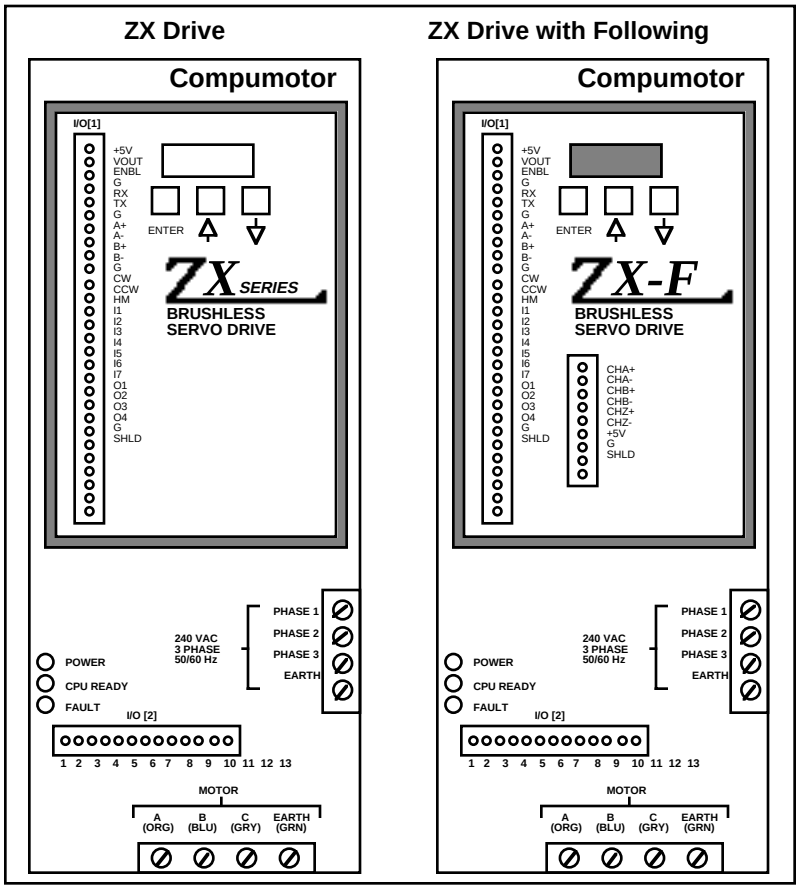
Z930 at 240VAC



Z940 at 240VAC



This section defines the inputs and ouputs that are on the ZX's front panel. Refer to the figure below for the location of the inputs and outputs described in this section.



ZX & ZXF Front Panels

Pin #	Function	Color
1	Shield	—
2	Stator 1	Red
3	Stator 2	Black
4	Stator 3	Green
5	Stator 4	Blue
6	Rotor 1	Brown
7	Rotor 2	White
8	Motor Temp (+)	Yellow
9	Motor Temp (-)	Orange

Resolver Cable Pin-Out

Motor Connector Pin	Color
A	Orange
B	Blue
C	Grey
D	Green

Motor Cable Pin-Out

Programmable Inputs

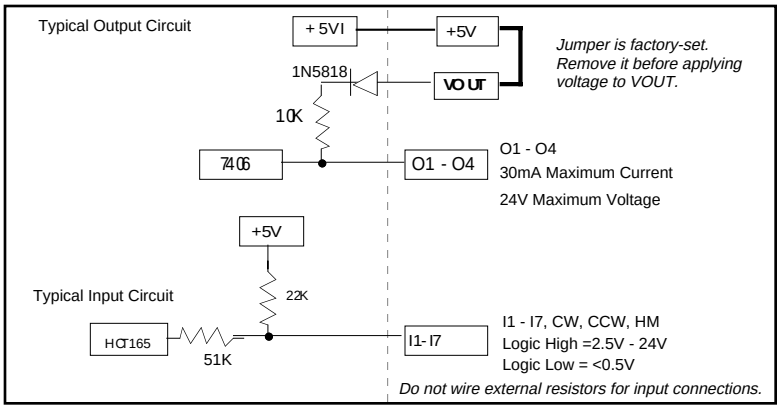
There are 7 inputs, one I/O ground, and one earth ground pin associated with **PROG INPUTS**. The first three inputs are **CCW**, **CW**, and **HM**. These inputs are electronically identical tooinputs **I1** - **I17**. The primary difference between these inputs is that **CCW**, **CW**, and **HM** are dedicated inputs, while **I1**-**I17** have programmable input functions. These

inputs have an internal isolated 5V supply. The figure below represents a typical input circuit. *Input Voltage: 0 - 24VDC (Current limiting resistor not needed).*

Programmable Outputs

The ZX has four programmable outputs that are jumpered at the factory and pulled up to 5VDC through an internal 10K resistor. With the VOUT input, you can pull these outputs up to 24VDC and sink up to 30 mA (max).

Helpful Hint: Output: 5 - 25V and can sink up to 30 mA

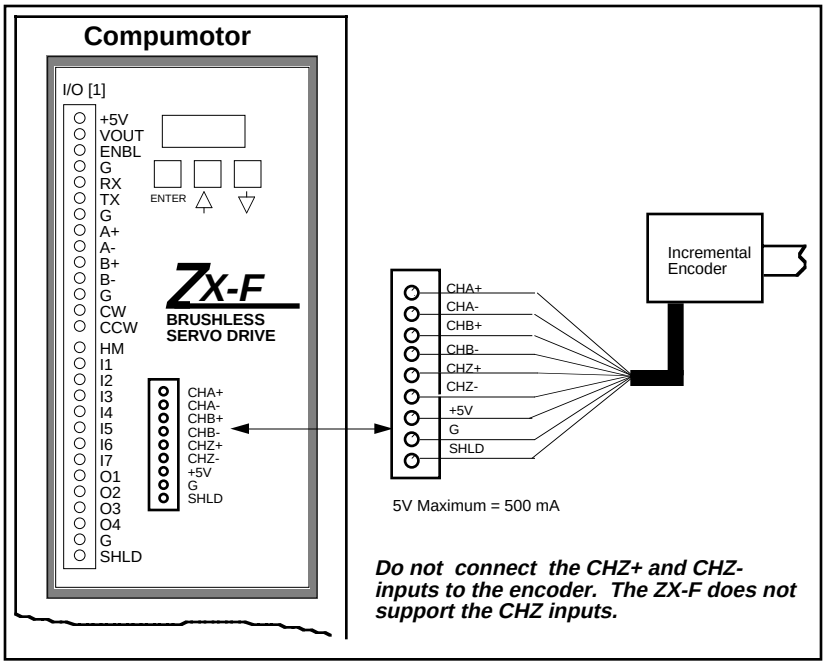


Typical Model Input & Output Circuit

INC Encoder

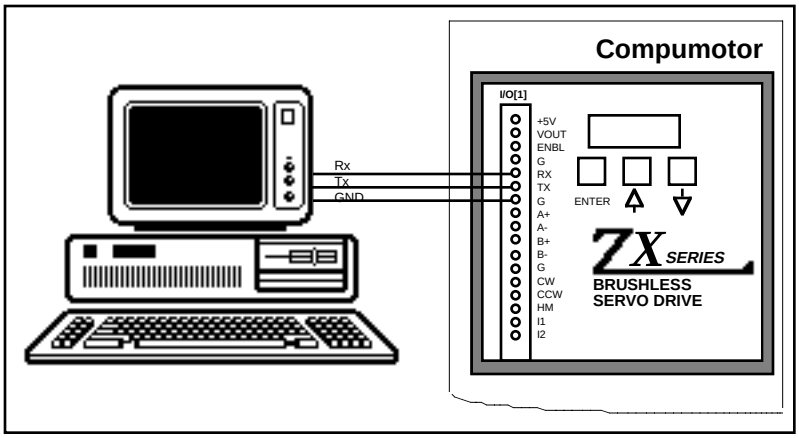
The ZXF's Incremental encoder connector is a 13-pin phoenix connector. The incremental encoder interface accepts two-phase quadrature incremental encoders with differential or single-ended outputs (+5V TTL-compatible). The ZX provides the 5VDC at 500 mA supply for the encoder. The maximum frequency per channel is 800Khz.

Helpful Hint: The ZXF does not use the Z Channel



Incremenal Encoder Connector Description

You can communicate with the ZX via a computer or terminal through RS-232C inputs. You must wire the Tx (Transmit), Rx (Receive), and GND (Logic Ground) inputs as shown in the figure below. The ZX accepts standard EIA RS-232C signals from +15VDC to -15VDC. You can change the baud rate, but the other parameters are fixed. You must configure your computer or terminal to the proper setting.



ZX RS-232C Interface

Factory Default Settings

This section contains the default settings for various ZX/ZXF functions. Your product will be shipped to you with these default settings.

RS-232C Communications

- ❑ Baud Rates: 9600 (default setting), 4800, 2400, 1200, 600, 300
- ❑ Data Bits: 8
- ❑ Parity: None
- ❑ Stop Bits: 1
- ❑ Full Duplex

ZX Device Address

The default address setting is **01**. Refer to *Chapter ② Getting Started* for steps on displaying and modifying the address setting via pushbuttons.

Hardware Interfaces

- ❑ Front panel pushbuttons are enabled
- ❑ Sequence and Ratio Select are disabled
- ❑ RS-232C is enabled
- ❑ Inputs active low and configured as triggers, with dedicated home, CW, and CCW limits
- ❑ Outputs active low, configured as programmable outputs

Limits

- ❑ Hardware limits enabled
- ❑ Software limits disabled

Motion Parameters

- ❑ Acceleration = 10 rps^2
- ❑ Deceleration = 10 rps^2
- ❑ Velocity = 1 rps
- ❑ Motor Resolution = 5,000
- ❑ Distance = 25,000
- ❑ Encoder Resoultion = 4,000

- Use the DR command to display the present configuration of the ZX

ZX900 Electrical Specifications

Electrical specifications for the ZX900 series drive's input and output power are provided in this section.

Input Power

Voltage (Nominal)	240VAC (1-phase or 3-phase)
Voltage (Range)	205-252 VAC (1 or 3-phase)
Frequency (Range)	47-66 Hz
Current (Max. cont.)	30A (rms) 3-phase
Power (Max. cont.)	12.4 KVA
Fuses	40A slow blow—User accessible
Isolation transformer	Not required

The actual input power and current is a function of the operating point of the motor (speed and torque) and the duty cycle. You can de-rate the fuse and the isolation transformer by scaling the above numbers by your actual requirements. The data above reflects the servo motor and drive operating at rated speed and at rated torque at 100% duty.

Output Power

Voltage	405 VDC (maximum)
Frequency	0 - 400Hz fundamental (7 kHz PWM)
Current	40A continuois per phase sinusoidal (28.28 Arms) 80A per phase peak (56.56 Arms)
Regen/power dump	Optional accessory

Motor/Drive Configuration

The ZX900's hardware is pre-configured to control ZX900 series motors. The ZX900 series drives only ZX900 series motors, the ZX800 series drives only Z800 motors, and ZX600 series drives only ZX600 motors. *Be sure that your drive type matches your motor type (Z600, Z800, or Z900).* If you have questions about the ZX Series motor/drive configuration, contact your local Automation Technology Center (ATC) or distributor.

Technical Data ZX900 Series

	Units	ZX-910	ZX-920	ZX-930	ZX-940
Continuous Stall Torque (*±10%)	oz-in lb-in lb-ft Nm	2407 150 12.5 17.0	4263 266 22.2 30.1	5990 374 31.2 42.3	9021 564 47.0 63.7
Peak Torque (*±10%)	oz-in lb-in lb-ft Nm	5205 325 27 37	8525 533 44 60	11980 749 62 85	18041 1128 94 127
Rated Torque (*±10%)	oz-in lb-in lb-ft Nm	1939 121 10.1 13.7	3341 209 17.4 23.6	4830 302 25.2 34.1	7480 467 39.0 52.8
Rated Power	hp kWatts	9.6 7.2	10.4 7.8	11.0 8.2	11.1 8.3
Rated Speed	rpm rps	5000 83.3	3150 52.5	2300 38.3	1500 25.0
Rated Current (line)	A (rms)	27.2	27.7	28.3	28.3
Peak Current (3.3 seconds max)	A (rms)	56.6	56.6	56.6	56.6
Max. Cont. AC input Current (3 phase 240VAC)	A (rms)	30	30	30	30
Rotor Inertia	oz-in^2 (mass) oz-in-sec^2 kg-m^2 x 1E-6	50.79 0.132 92.9	111.21 0.288 2034	166.21 0.431 3040	459.48 1.190 8404
Motor Weight	lbs kg(f)	32 15	57 26	65 29	112 51
Shipping Weight	lbs kg(f)	89 40	114 52	122 55	169 77

ZX900 Indexer/Drive Performance Specifications

Positional Repeatability

Repeatability: ±0.088 degrees, unloaded

Positional Accuracy

Resolver Accuracy: ±7 arc minutes

Resolver-to-Digital Converter Accuracy: ±8 arc minutes *(For finer accuracies, contact Compumotor—800-358-9070.)*

Motor/Drive Compatibility

Different motors can take different amounts of current and require different tuning parameters for typical loads. The **CMTR** (Configure Motor Type) command sets up a drive for a particular motor. By issuing **CMTR**, motor current levels and default parameters are recalled from memory. Do not exceed the current level specified for the motor, excessive current levels will damage the motor.

The following information is provided in case you must modify the motor/drive configuration. This command sequence will set up a drive for a particular motor size and perform the commutation (refer to the ***ZX Indexer/Drive Software Reference Guide*** for more on these commands).

> ***WARNING***
>
> This commutation procedure causes violent motor motion. All loads should be removed from the motor shaft before you begin this procedure.

Command	Description
> 1OFF	Turns drive off
> 1CMTRxxx	Sets drive for the motor; xxx =910, 920, 930, or 940
> 1ON	Enables the drive

Motor Brakes

These brakes are mounted directly behind the motor and come completely assembled from the factory. When ordering the brake option, please specify the motor type.

Brake Characteristics	Z910	Z920/930	Z940	Units
Supply voltage	24	24	24	VDC
Supply current	0.93	1.27	*	A
Static braking torque	1152	6720	*	oz-in

**Same as Z920/930 Characteristics*

Z900 Motor Brake Characteristics

Motor Data

The following pages provide data on each of the four motor frame sizes of ZX900 series systems (ZX910, ZX920, ZX930, ZX940). The data reflecting motor torque does not assume operation from a ZX900 drive. The torque specifications reflect the motor's capabilities. In most cases, the motor windings match the drive's output power with an additional safety margin.

	Motor Size	**Z910**	Value	Units	Tolerance
1	Constant (s):	Torque	88.32	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	26.7	V rms/Krpm	± 10%
3		Electrical Time	24	milliseconds	nominal
4		Mechanical Time	0.672	milliseconds	nominal
5		Thermal	43	minutes	nominal
6	Torque (s):	Continuous, Stall	2544	oz-in	min. [1]
7		Continuous, Stall	2400	oz-in	min. [2]
8		Continuous, Rated	1939	oz-in	min. [2]
9		Peak, Max w/o Saturation	6835	oz-in	min. [1]
10		Static Friction	40.7	oz-in	max
11		Ripple (of Rated Torque)	4.5	percent	max [3]
12	Speed:	Rated	5000	rpm	reference
13		Maximum	5000	rpm	reference
14	Frequency:	Rated	166.6	Hz	max.
15	Current:	Rated	27.2	A rms	max. [1]
16		Peak	81.4	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	7.2 (9.6)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	4.8	mH	± 30%
21	DC Resistance	Terminal (line-line)	0.2	Ohms	± 10% [1]
22	Acceleration at Rated Torque		51971	rads/sec^2	Theoretical
23	Rotor Inertia		929	kgm^2 * 1E-6	nominal
24	Damping		1.728	oz-in / krpm	nominal
25	Weight		32	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		135	°C (Celsius)	± 5°C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10°C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.00205	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	2E-5/7E-6	in/lb	reference
35		Material [3]	EC#30	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	160	lbs.	max.[7]
		2000 rpm	127.1	lbs.	max.[7]
		3000 rpm	110.9	lbs.	max.[7]
		4000 rpm	100.8	lbs.	max.[7]
		5000 rpm	93.3	lbs.	max.[7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		oz-in/A rms ±7 arc min.		
44	Resolver Manufacturer/Model #		Facso #21-BRCX-334-J39		
45	Standard Resolver Cable Part Number		71-011776-xx		
46	Standard Motor Cable Part Number		71-011776-xx		
47	Options:	Brake—24VDC (1A)—1152 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z805 Motor Specifications

	Motor Size	**Z920**	Value	Units	Tolerance
1	Constant (s):	Torque	154.2	oz-in/A rms	± 10%
2		Voltage (Sinusoidal)	46.6	V rms/Krpm	± 10%
3		Electrical Time	26.9	milliseconds	nominal
4		Mechanical Time	0.6	milliseconds	nominal
5		Thermal	46	minutes	nominal
6	Torque (s):	Continuous, Stall	4518	oz-in	min. [1]
7		Continuous, Stall	4262	oz-in	min. [2]
8		Continuous, Rated	3341	oz-in	min. [2]
9		Peak, Max w/o Saturation	12422	oz-in	min. [1]
10		Static Friction	65	oz-in	max
11		Ripple (of Rated Torque)	4.5	percent	max [3]
12	Speed:	Rated	3150	rpm	reference
13		Maximum	3150	rpm	reference
14	Frequency:	Rated	157.5	Hz	max.
15	Current:	Rated	27.7	A rms	max. [1]
16		Peak	84.8	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	7.7 (10.4)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	6.3	mH	± 30%
21	DC Resistance	Terminal (line-line)	0.234	Ohms	± 10% [1]
22	Acceleration at Rated Torque		43133	rads/sec^2	Theoretical
23	Rotor Inertia		2034	kgm^2 * 1E-6	nominal
24	Damping		15.36	oz-in / krpm	nominal
25	Weight		51	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5°C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10°C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.00034	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	2E-5/7E-6	in/lb	reference
35		Material [3]	RC#30	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	255.6	lbs.	max.[7]
		2000 rpm	202.9	lbs.	max.[7]
		3000 rpm	177.2	lbs.	max.[7]
		4000 rpm	N/A	lbs.	max.[7]
		5000 rpm	N/A	lbs.	max.[7]
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 3 (P/2)		
42	Stator Phase Sequence		A-B-C (viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited;±7 arc min.		
44	Resolver Manufacturer/Model #		Facso #21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-xx		
46	Standard Motor Cable Part Number		71-011777-xx		
47	Options:	Brake—24VDC (1.27A)—6720 oz-in Holding Torque			
		IP67 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z805 Motor Specifications

	Motor Size	**Z930**	Value	Units	Tolerance
1	Constant (s):	Torque	212	oz-in/A rms	± 10%
2		Voltage ()	64	V rms/Krpm	± 10%
3		Electrical Time	31.1	milliseconds	nominal
4		Mechanical Time	0.53	milliseconds	nominal
5		Thermal	50	minutes	nominal
6	Torque (s):	Continuous, Stall	6513	oz-in	min. [1]
7		Continuous, Stall	6144	oz-in	min. [2]
8		Continuous, Rated	4954	oz-in	min. [1]
9		Peak, Max w/o Saturation	17357	oz-in	min. [1]
10		Static Friction	133.2	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	max.
12	Speed:	Rated	2300	rpm	reference
13		Maximum	2300	rpm	reference
14	Frequency	Rated	115	Hz	max.
15	Current:	Rated	29	A rms	max. [1]
16		Peak	86.2	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	8.4 (11.3)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	8.1	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.26	Ohms	± 10% [1]
22	Acceleration at Rated Torque		40357	rads/sec^2	Theoretical
23	Rotor Inertia		3040	kgm^2 * 1E-6	nominal
24	Damping		2.88	oz-in/krpm	nominal
25	Weight		65	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0038	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	2E-5/7E-6	in/lb	reference
35		Material [3]	RC#30	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	263	lbs.	max.
		2000 rpm	208.8	lbs.	max.
		3000 rpm	N/A	lbs.	max.
		4000 rpm	N/A	lbs.	max.
		5000 rpm	N/A	percent	max.
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 2 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Fasco #21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-01		
46	Standard Motor Cable Part Number		71-012979-01		
47	Options:	Brake—24VDC (1.27A)—6720 oz-in Holding Torque IP67 Classification Incremental Encoder Tachometer No Keyway			

[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z810 Motor Specifications

	Motor Size	**Z940**	Value	Units	Tolerance
1	Constant (s):	Torque	319	oz-in/A rms	± 10%
2		Voltage ()	96.3	V rms/Krpm	± 10%
3		Electrical Time	56.5	milliseconds	nominal
4		Mechanical Time	0.57	milliseconds	nominal
5		Thermal	56	minutes	nominal
6	Torque (s):	Continuous, Stall	12007	oz-in	min. [1]
7		Continuous, Stall	11328	oz-in	min. [2]
8		Continuous, Rated	9388	oz-in	min. [2]
9		Peak, Max w/o Saturation	32870	oz-in	min. [1]
10		Static Friction	128.6	oz-in	max.
11		Ripple (of Rated Torque)	4.5	percent	max.[3]
12	Speed:	Rated	1500	rpm	reference
13		Maximum	1500	rpm	reference
14	Frequency	Rated	75	Hz	max.
15	Current:	Rated	35.5	A rms	max. [1]
16		Peak	108.5	A rms	nominal
17	Voltage:	Rated	230	V rms	reference
18		Max	250	V rms	maximum
19	Output Power:	Rated	10.4 (14.0)	kWatts (hp)	min. [1]
20	Inductance:	Terminal (line-line)	13	mH	± 10%
21	D.C. Resistance	Terminal (line-line)	0.23	Ohms	± 10% [1]
22	Acceleration at Rated Torque		27613	rads/sec^2	Theoretical
23	Rotor Inertia		8400	kgm^2 * 1E-6	nominal
24	Damping		15.36	oz-in / krpm	nominal
25	Weight		112	lbs.	max.
26	Winding Temperature		170 [4]	°C (Celsius)	max.
27	Winding Temperature Rise (Above Ambient) [1]		145	°C (Celsius)	reference
28	Insulation Class		H	—	reference
29	Thermostat TRIP Temperature		170	°C (Celsius)	± 5 °C
30	Thermostat RESET Temperature		135	°C (Celsius)	± 10 °C
31	Dielectric Strength, (Winding-to-Frame)		1750	VAC	min.
32	Winding Capacitance to Frame		0.0082	µF	max.
33	IP Classification		65 (Shaft [8])	rated	standard
34	Shaft:	Radial-Play (front to back)	1E-5/4E-6	in/lb	reference
35		Material [3]	RC#3	—	reference
36		Magnet Type	NbFeB	—	—
37		Loading [4] 1000 rpm	365.9	lbs.	max.
		2000 rpm	N/A	lbs.	max.
		3000 rpm	N/A	lbs.	max.
		4000 rpm	N/A	lbs.	max.
		5000 rpm	N/A	lbs.	max.
38	Bearing Class, Internal/External		1/Class 3	ABEC/AFBMA	reference
39	Bearing Grease		SRI#2	Manufacturer	reference
40	Shaft Seal Pressure		0.21 (3)	kg/cm^2 (psi)	max.
41	Basic Motor Design		3 phase *wye* connected 3 (P/2)		
42	Stator Phase Sequence		A-B-C (CW viewed from front face plate)		
43	Resolver Type/Accuracy		Single-Speed; Rotor-Excited; ± 7 arc min.		
44	Resolver Manufacturer/Model #		Faxco #21-BRCX-335-J39		
45	Standard Resolver Cable Part Number		71-011777-01		
46	Standard Motor Cable Part Number		71-012979-01		
47	Options:	Brake—24VDC (1.27A)—6816 oz-in Holding Torque			
		IP65 Classification			
		Incremental Encoder			
		Tachometer			
		No Keyway			

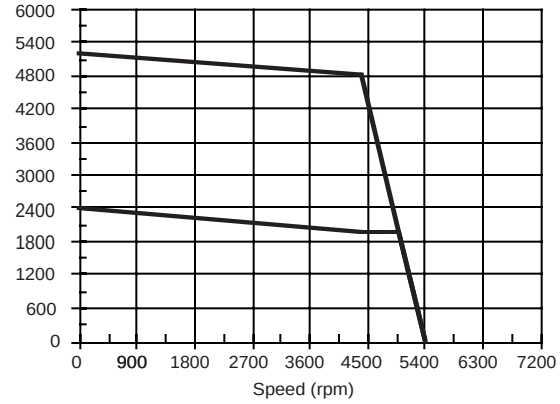
[1] 25 °C ambient
[2] 40 °C ambient
[3] Measured at 60 rpm (1 rps) in velocity mode
[4] Rated for 20,000 hours or 40,000 hours @ 155°C
[5] Rotor steel is rated af fatigue proof
[6] Loads centered 1 inch from mounting flange
[7] Loads may be radial and axial such that the sum of the radial and two times the axial does not exceed this figure.
[8] The motor shaft is IP30 rated.

Z820 Motor Specifications

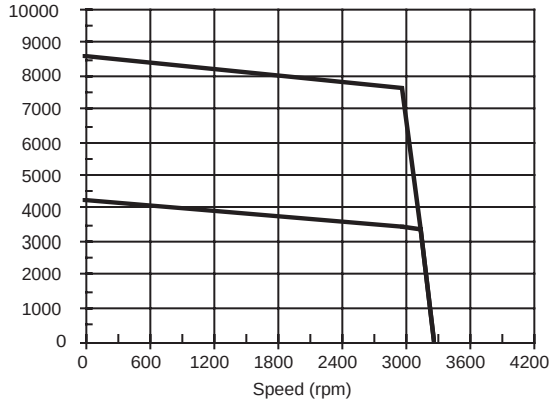
Speed/Torque Curves

The following speed/torque curves represent the available shaft torque at different operating speeds. Operation at 240VAC is shown for each motor size. Actual motor torque may vary ±10% due to motor manufacturing variances.

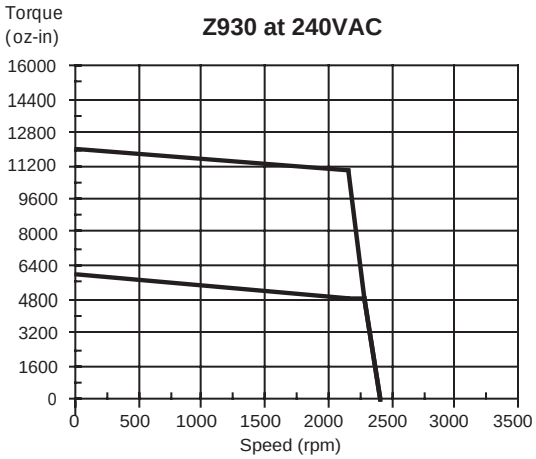
Z910 at 240VAC



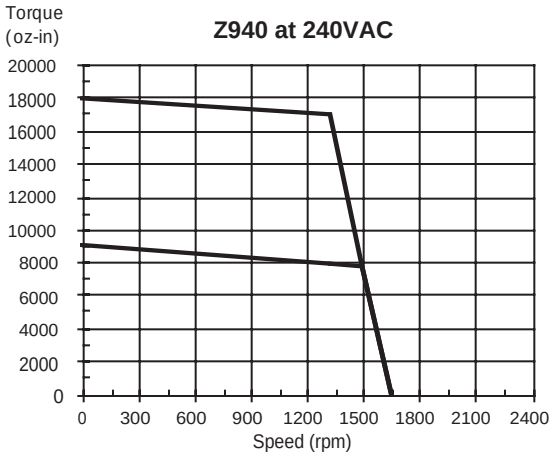
Z920 at 240VAC



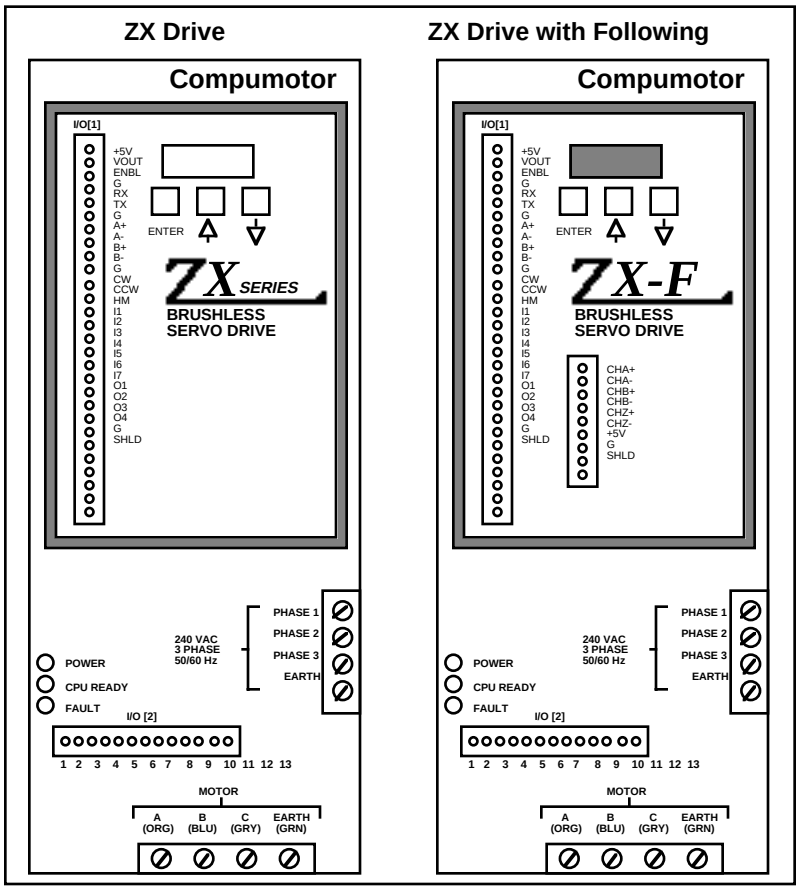
Z930 at 240VAC



Z940 at 240VAC



This section defines the inputs and ouputs that are on the ZX's front panel. Refer to the figure below for the location of the inputs and outputs described in this section.



ZX & ZXF Front Panels

Pin #	Function	Color
1	Shield	—
2	Stator 1	Red
3	Stator 2	Black
4	Stator 3	Green
5	Stator 4	Blue
6	Rotor 1	Brown
7	Rotor 2	White
8	Motor Temp (+)	Yellow
9	Motor Temp (-)	Orange

Resolver Cable Pin-Out

Motor Connector Pin	Color
A	Orange
B	Blue
C	Grey
D	Green

Motor Cable Pin-Out

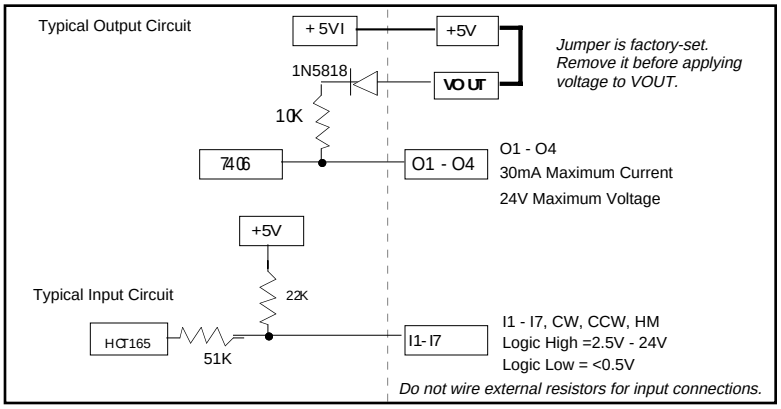
Programmable Inputs

There are 7 inputs, one I/O ground, and one earth ground pin associated with **PROG INPUTS**. The first three inputs are **CCW**, **CW**, and **HM**. These inputs are electronically identical tooinputs **I1** - **I17**. The primary difference between these inputs is that **CCW**, **CW**, and **HM** are dedicated inputs, while **I1**-**I17** have programmable input functions. These inputs have an internal isolated 5V supply. The figure below represents a typical input circuit. *Input Voltage: 0 - 24VDC (Current limiting resistor not needed).*

Programmable Outputs

The ZX has four programmable outputs that are jumpered at the factory and pulled up to 5VDC through an internal 10K resistor. With the VOUT input, you can pull these outputs up to 24VDC and sink up to 30 mA (max).

Helpful Hint: Output: 5 - 25V and can sink up to 30 mA

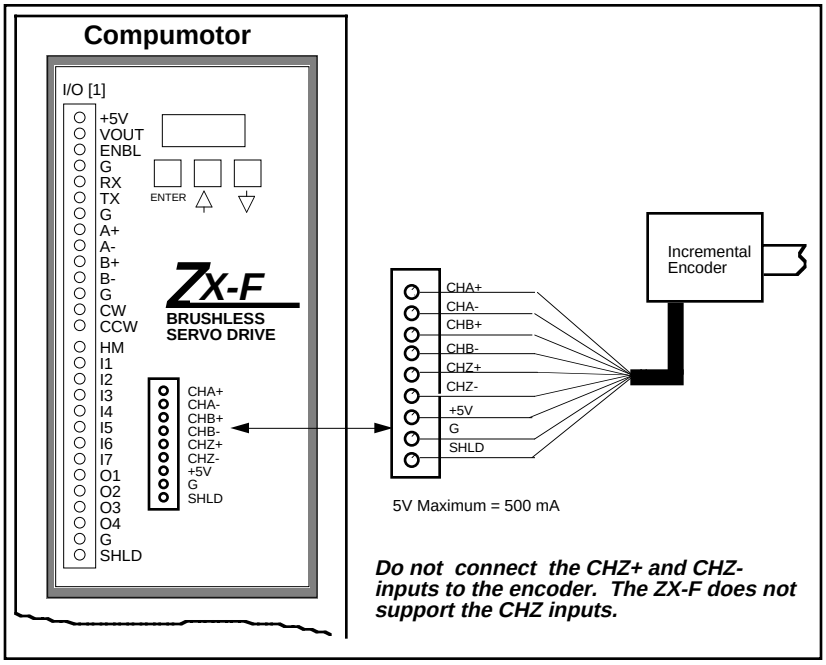


Typical Model Input & Output Circuit

INC Encoder

The ZXF's Incremental encoder connector is a 13-pin phoenix connector. The incremental encoder interface accepts two-phase quadrature incremental encoders with differential or single-ended outputs (+5V TTL-compatible). The ZX provides the 5VDC at 500 mA supply for the encoder. The maximum frequency per channel is 800Khz.

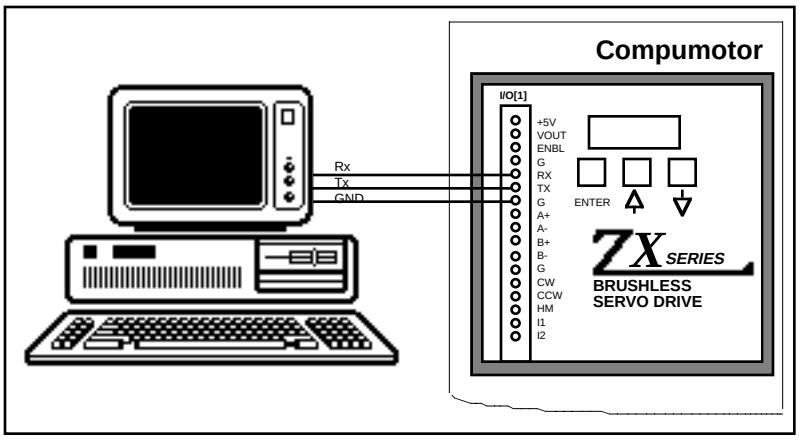
Helpful Hint: The ZXF does not use the Z Channel



Incremenal Encoder Connector Description

RS-232C

You can communicate with the ZX via a computer or terminal through RS-232C inputs. You must wire the Tx (Transmit), Rx (Receive), and GND (Logic Ground) inputs as shown in the figure below. The ZX accepts standard EIA RS-232C signals from +15VDC to -15VDC. You can change the baud rate, but the other parameters are fixed. You must configure your computer or terminal to the proper setting.



ZX RS-232C Interface

Factory Default Settings

This section contains the default settings for various ZX/ZXF functions. Your product will be shipped to you with these default settings.

RS-232C Communications

- ❑ Baud Rates: 9600 (default setting), 4800, 2400, 1200, 600, 300
- ❑ Data Bits: 8
- ❑ Parity: None
- ❑ Stop Bits: 1
- ❑ Full Duplex

ZX Device Address

The default address setting is **01**. Refer to *Chapter ② Getting Started* for steps on displaying and modifying the address setting via pushbuttons.

Hardware Interfaces

- ❑ Front panel pushbuttons are enabled
- ❑ Sequence and Ratio Select are disabled
- ❑ RS-232C is enabled
- ❑ Inputs active low and configured as triggers, with dedicated home, CW, and CCW limits
- ❑ Outputs active low, configured as programmable outputs

Limits

- ❑ Hardware limits enabled
- ❑ Software limits disabled

Motion Parameters

- ❑ Acceleration = 10 rps^2
- ❑ Deceleration = 10 rps^2
- ❑ Velocity = 1 rps
- ❑ Motor Resolution = 5,000
- ❑ Distance = 25,000
- ❑ Encoder Resoultion = 4,000
- ❑ Use the DR command to display the present configuration of the ZX

C H A P T E R ⑧

Maintenance & Troubleshooting

The information in this chapter will enable you to:

- ❑ Maintain the system's components to ensure smooth, efficient operation
- ❑ Isolate and resolve system hardware and software problems
- ❑ Use this chapter as a quick-reference for a description of system error codes

Motor Maintenance

Inspect the motor regularly to ensure that bolts and couplings have not loosened during operation. This will prevent minor problems from becoming major problems. The motor is brushless and the bearings are sealed. ***No maintenance, other than regular visual inspection, is necessary***.

ZX Maintenance

The ZX requires minimal maintenance, but the NEMA cabinet should be inspected regularly to ensure that the air filters are kept clean. The ZX requires good air flow for proper cooling.

Spare Parts

Description	Function	Part Number
5-Pin Connector	I/O [1] Connectors	43-008128- 01
9-Pin Connector	I/O [1] Connectors	43-078755-01
Battery	Maintains RAM memory	47-007709-01

Spare Parts Table

Battery

☞ ***Helpful Hint:***
The battery is located in the ZX's front module.

The ZX's nonvolatile memory is a battery-backed RAM (Random Access Memory). The lithium battery keeps the RAM powered at all times. *The life of this battery is approximately 10 years.* When the battery runs low, the ZX will be unable to hold a program in nonvolatile memory. *This error may also occur if the Sequence Termination (**XT**) command is not placed at the end of each sequence.* If the battery back up is not functioning, error code **3Ø** (Check Sum Error) will be displayed when power is cycled. To verify that the battery is OK, type **1XC**, then cycle power to the ZX. If the battery is OK, error code **3Ø** should be gone. If the message persists, replace the battery (part #47-007709-01). Call your local distributor to order the battery.

☞ **Helpful Hint:**
To change the battery, complete the following steps.

① You must remove the enclosure to reach the battery.
② Remove the four screws holding the visible circuit board.
③ Remove the visible circuit board to reach the ***middle*** circuit board.
④ The battery is labeled **BT1** and is the size of a quarter. Remove the battery from the ZX and ***immediately replace it with the new battery***.

> ***WARNING***
> Battery may explode if mistreated. Do not recharge, disassemble or dispose of in fire.

If your system malfunctions, you must identify and isolate the problem. The first step is to isolate each component and ensure that each component functions properly when it is run independently. If you have additional units available, you may want to use them to replace existing components in your system to help identify the source of the problem.

☞ ***Helpful Hint:***
Refer to the problem solutions in this chapter. If the problem persists, contact Compumotor's Applications Engineering Department (800-358-9070).

Determine if the problem is mechanical, electrical, or software-related. Can you repeat or re-create the problem? Random events may appear to be related, but they may not be contributing factors to your problem. Carefully investigate and decipher the events that occurred before the subsequent system problem.

You may be experiencing more than one problem. Solve one problem at a time. Log (document) all testing and problem isolation procedures. You may need to review and consult these notes later. This will also prevent you from duplicating your testing efforts.

CW, CCW, HM

If you are have problems using the Trigger (**TR**), Home (**HM**), CW, CCW, or Sequence Select inputs, check your wiring for proper installation. Use an ohmmeter to properly connect the switches and inputs.

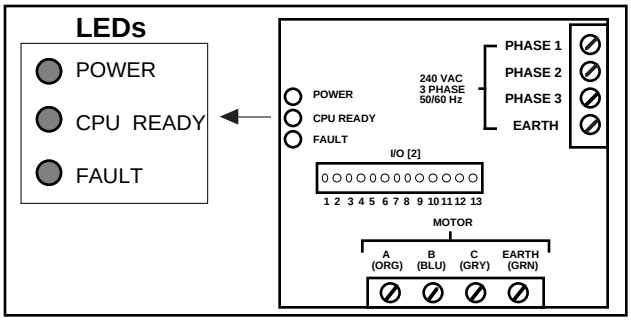
If the hardware connections are correct, use the **IS** command to check the hardware status of each input manually and determine if the ZX recognizes the input change. If the status does not change, check the hardware settings.

Remote Sequencing (Binary Weighted)

If you want to run sequences from binary weighted interfaces, verify that the hardware interface is wired properly. Use an ohmmeter to verify proper wiring. Enter the **IS** command to read the status of the inputs. Change the input setting and check the input status (**IS**) again to ensure that the ZX recognizes the change in the sequence select input. Ensure that the binary weighted input is calling the proper sequences. Check *Chapter ④ Application Design* for the Sequence Select Table. If you have a problem running a sequence from the remote input, run the sequence using the **XR** command before attempting to run it using BCD input.

Diagnostic LEDs

The ZX has three diagnostic LEDs on its front panel.



ZX LEDs

POWER	Indicates that the drive is powered-up (normally green).
CPU READY	Indicates that the processor is properly functioning (normally green).
FAULT	Indicates a drive fault or error (red when a fault exists; normally off).

In addition to these LED indicators, the ZX has a front panel display that scrolls fault messages when a fault occurs. *Chapter ④ Application Design* describes all operations and capabilities of the front panel display.

ZX Warning Codes

The following warning codes will be displayed on the alphanumeric display before an overvoltage fault occurs.

Code	Warning	Description
03	**Over Voltage**	The internal bus voltage has passed the warning threshold. This warning will not fault the drive; the error threshold (Error 04) **will** fault the drive. The warning message will remain present until an **ON** command or a system reset is executed.
60	**Commanded Shutdown**	The ZX has been commanded off by the **OFF** command.

ZX Warning Codes

ZX Fault Codes

The following fault codes and messages will be displayed on the alphanumeric display when the respective faults occur. The Report Servo Errors (**RSE**) command also reports faults and warnings. Except where noted, the drive can be re-enabled after a fault has occurred by correcting the fault (if applicable) and typing **ON**.

Code	Fault	Description
04	**Overvoltage**	The internal bus voltage has exceeded its limit. Verify the input power voltage. This error may also be caused by rapid decelerations. This error overrides the over voltage warning.
18	**Low Voltage**	The input power to the drive is either not present, or at a lower voltage than 92V. Verify the input power voltage and make sure that the power is correctly connected.
19	**over-current (Short Circuit)**	A short circuit between motor phases, or between any phase and ground will cause the drive to fault. This may be due to improper motor wiring to the drive or due to a bad motor. In some cases an extremely high acceleration may cause this error. Power down the complete ZX, correct the wiring, and power-up the system. This fault cannot be removed by typing the **ON** command.
20	**Position Error Exceeded**	The user defined position error limit has been exceeded. The **CPE** (Configure Position Error) command can alter the error limit. The factory default is one revolution (5,000 motor steps). A position error may be caused by a rapid acceleration rate or possibly bad servo tuning. The user must decide whether to increase the position error limit or to change his motion profile so that the drive no longer faults out.
21	**Rolling Average Current Fault**	The user specified average current limit has been exceeded. The **CCA** (Configure Current Average) command alters the average current limit. If the drive continues to fault out, you may have to reduce the acceleration rate of your move. The average current of the drive is calculated over a rolling 3-second time period.
22	**Average Current Limit Exceeded**	The drive's average current limit has been exceed. A hardware detect circuit has turned the drive off due to too much current.
23	**Drive Enable Input Not Active**	The drive enable input on the screw terminal connector I/O[1] has been deactivated. For drive operation, a normally closed switch or wire jumper is necessary from ENBL to ground. The ZX is shipped with a jumper installed.
27	**Velocity Error Exceeded**	The user-defined velocity error limit has been exceeded. The **CVE** (Configure Velocity Error) command can alter the error limit. A velocity error may be caused by a rapid acceleration rate or bad servo tuning.
30	**Z RAM Failure**	This error should occur only during factory initialization of the ZX/ZXF or *if the Sequence Termination (**XT**) command is not placed at the end of each sequence.* . Call Compumotor (800-358-9070) if it occurs at any other time.
41	**CW Hardware Limit**	The ZX has reached a CW hardware limit.
42	**CCW Hardware Limit**	The ZX has reached a CCW hardware limit.
43	**CW Software Limit**	The ZX has reached a CW software limit.
44	**CCW Software Limit**	The ZX has reached a CCW software limit.
55	**Power Supply Failure**	This error indicates that the isolated power supply for the ZX's amplifier has failed. Call Compumotor (800-358-9070).
61	**Incoming Indexer Pulses**	When the ZX is reset or commanded **ON**, no step pulses are accepted when the amplifier is first enabled (this would constitute infinite acceleration).
62	**Current Integration Fault**	Currently this fault is not enabled.
70	**Resolver Disconnected**	The resolver feedback signal from the motor to the drive is missing. This feedback is required to operate the drive. Make sure the resolver connection is secure.
77	**Resolver Resolution Fault**	For a given resolver resolution there is a maximum velocity at which the resolver-digital converter can operate. If the drive faults due to this error, the maximum velocity commanded to the drive exceeds the capabilities of the resolver-digital converter. See the **CRR** command.
91	**Heatsink Overtemperature**	The internal heatsink of the ZX has reached its temperature limit. Make sure the ambient temperature for the drive is within specifications.
92	**Motor Overtemperature**	The motor has reached its temperature limit. Make sure the motor's ambient temperature is within specifications. Be sure the drive is configured correctly for the motor used (**CMTR** command).
93	**PCB Overtemperature**	The Printed Circuit Board temperature limit has been reached. Make sure the ambient temperature for the drive is within specifications.

Drive Fault Codes

RS-232C Problems

Use the following procedure to troubleshoot communication problems that you may have with the ZX.

① Be sure the host computer's transmit (**Tx**) wire is wired to the peripheral's receive (**Rx**) connection, and the host computer's receive (**Rx**) wire is wired to the peripheral's transmit (**Tx**) connection. Switch the receive and transmit wires on either the host or peripheral if the problem persists.

② Confirm that the host and peripheral are configured for the same baud rate, 8 data bits, 1 stop bit, and no parity.

③ If you receive double characters, for instance typing **A** and receiving **AA**, the computer is set for half duplex mode. Change the setup to full duplex mode.

④ Use DC common or signal ground as a reference, not earth ground.

⑤ Cable lengths should not exceed 50 ft. unless you are using some form of line driver, optical coupler, or shield. As with any control signal, be sure to shield the cable-to-earth ground at one end only.

⑥ To test the terminal or terminal emulation software and the RS-232C cable for proper three-wire communication, unhook the ZX and enter a character. You should not receive an echoed character. If you do, you are in half duplex mode. Connect the host's transmit and receive lines together and send another character. You should receive the echoed character. If not, consult the manufacturer of the host's serial interface for proper pin outs.

Software Debugging Tips

This section offers helpful tips for debugging your programs. The ZX has several tools that can be used to aide in the determination of a problem in the system design.

☞ **Helpful Hint:** **The software tools are listed below**

XTR	Trace Mode
DIN,DOUT	I/O Simulation
XST	Single Step
XS	Sequence Execution Status
DF	Displays indexer Status
X	Displays state of the Indexer
DR	Displays Interface Option Status
FS	Displays Indexer Status Options
OS	Displays Homing/Jog Status
	Report backs

Trace Mode

Trace mode is displays what is occurring as you execute your sequence. By running the Trace mode you can see what commands are being executed and if the program stops running, you can see what command was last executed. Trace mode and Interactive mode (**SSI**) will help you to find commands that the indexer may not recognize. The trace mode is enabled and disabled using the **XTR** command. When enabled you will execute sequences as you normally would using the **XR** command. As the sequence is running the commands are displayed on the screen. **XTR1** enables Trace mode, **XTRØ** disables it. See *Chapter ④ Application Design*, for a thorough explanation of Trace mode.

I/O Simulation

I/O simulation can be done without actually physically toggling the inputs or outputs using the **DIN** and **DOUT** commands. These commands can be used to simulate the input or output state you desire so that you can test portions of your sequences and program. See ***Sequence Debugging Tools*** in *Chapter ④ Application Design* for more details.

Displaying ZX Status

There are several commands that you can use to check the ZX's status. You can report back the value or setting of almost all of the ZX commands by typing the device address followed by the command then a carriage return or a space bar. In this way, you can find out what values you have entered in different commands. The **DR** command reports the current state of the ZX. Use this to verify that the unit is configured as you want it. You can use four other report back commands to determine the state of the unit. These commands report a binary number. Each bit of the report corresponds to possible ZX functions or modes.

DFX Command Report Back

The **DFX** command reports the ZX 's current states and conditions.

```
32  31  3Ø  29  23  27  26  25  24  23  22  21  20  19  18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  Ø
*Ø   Ø   Ø   Ø__Ø   Ø   Ø   Ø   Ø__Ø   Ø   Ø   Ø__Ø   Ø   Ø   Ø__Ø   Ø   Ø   Ø__Ø   Ø   Ø  Ø__Ø  Ø  Ø  Ø__Ø  Ø  Ø  Ø
```

Bit representations —25-32,13 reserved

Bit

24 Mode Profile: no = Ø, yes = 1

23 Mode alternate: no = Ø, yes = 1

21 Hit a software CCW limit no = Ø; yes = 1
20 Hit a software CW limit no = Ø; yes= 1
19 Home limit not found = Ø; found = 1
18 Jogging no = Ø; yes = 1
17 Queued for **RM** mode no = Ø; yes = 1
16 Run sequence on power up no=Ø; yes=1
15 **U** command Ø=not waiting; 1=waiting for continue
14 Waiting for a trigger no = Ø; yes = 1
12 Back up to home limit Ø = no; 1 = yes
11 High-speed portion of home move no = Ø; in process = 1
10 Execute a sequence no = Ø; yes = 1
9 Wait on a timer no = Ø; yes = 1
8 Hit a CCW limit no = Ø; yes = 1
7 Hit a CW limit no = Ø; yes = 1
6 **PS** command: Ø = not waiting; 1 = waiting for continue
5 Absolute move direction Ø = CW 1= CCW
4 Incremental/absolute Ø = **MPI**; 1 = **MPA**
3 Mode preset = Ø; Continuous = 1
2 Commanded move direction: Ø = CW; 1 = CCW
1 Preset move in progress: Ø = not moving; 1 = moving
Ø Continuous move: Ø = not moving; 1 = moving

The **FS** command reports a binary word that has various interface options associated with each bit. The **OS** command reports the homing options and the Jog enable option. The **SS** command reports various indexer software options. If **1** is in the bit location, the feature or mode is enabled. Refer to the example below.

```
      A B C D  _ E F G H _I J K L_M N O P _Q R S T
SS    *Ø Ø Ø Ø  _ Ø Ø Ø Ø _Ø Ø Ø Ø_Ø Ø Ø Ø _Ø Ø Ø Ø
FS    *Ø Ø Ø Ø  _ Ø Ø Ø Ø _Ø Ø Ø Ø_Ø Ø Ø Ø
OS    *Ø Ø Ø Ø  _ Ø Ø Ø Ø
```

SSA RS232 Echo 0 = Echo on 1 = Echo off
SSD Alternate Mode Stop 0 = end of cycle 1 = immediately
SSG Clear/Save buffer on limit 0 = clear 1 = Save
SSH Clear/Save buffer on stop 0 = clear 1 = Save
SSI Enable/Disable Interactive Mode
SSJ Enable/Disable Continuous scan mode
SSL Resume execution enable
SSN Enable error message mode
SSQ Enable Drive Fault indicator

FSD Enable/Disable Stop on Stall
FSI Enable/Disable Following
FSK Enable following learn mode
FSL Enable following self correction mode
FSN Enable Pulse and Direction following
FSP Enable position tracking

OSB Enable back-up to home switch
OSC Define Active edge of Home switch 1=Active high signal
OSD Enable Encoder Z channel Input for Homing
OSE Enable Jogging
OSG Define Final Home approach direction 1=CCW
OSH Define Active edge of home switch to stop on 1=CCW

Error Code 30

Error code 30 indicates that a checksum that was calculated on a reset or from cycling power does not match the previously calculated checksum. Error code 30 will occur in the following situations:

- ❏ You are changing PROMs.
- ❏ If memory is corrupted during operation or when the ZX is off.
- ❏ If the battery fails (memory cannot be saved).

In most cases, you can use the Create Checksum (**XC**) command to clear error code 30. Select the proper motor size before you issue the **ON** command. **ON** clears error code 30 from the front panel display. If the error occurs each time you power up your unit, consult Compumotor's Application Engineering Department (800-358-9070).

Reducing Electrical Noise

For detailed information on reducing electrical noise in your system, refer to the current Compumotor Catalog.

Common Mistakes

Some common mistakes that have made with Compumotor's X language and remedies for these mistakes are discussed below.

Problem A

A ZX move is commanded and no motion occurs.

Possible Errors

- ❏ Following mode may be enabled when you try to make a move as an indexer.
- ❏ A limit may be enabled and active.
- ❏ You may be in Absolute mode and are already at the position you are commanding the motor to move to.

Solution A

Check your configuration and motion settings.

Problem B

A following move is commanded and no motion occurs.

Possible Errors

- ❏ The unit may not be in Following mode (**FSI**).
- ❏ Limits are enabled, you may be in Absolute mode and are already at the position you are commanding the motor to move to.

Solution B

Set the ZX to Following mode. Check your configuration and motion settings.

Problem C

Motion is jittery in Following mode.

Solution C

Smooth motion by decreasing the sampling of the encoder interface (increase TF).

Problem D

The unit may appear to not be responding to commands.

Possible Error

- ❏ If you defined a sequence and never issued an **XT** command, the ZX still thinks you are defining a sequence.

Solution D

Issue an XT command at the end of the sequence to end sequence definition.

Returning The System

If your ZX Indexer/Drive system is faulty, you must return the drive and motor for replacement or repair. A failed drive can damage motors, and a faulty motor can damage subsequent drives. If you must return your ZX System to effect repairs or upgrades, use the following steps:

Step ①

Get the serial number and the model number of the defective unit(s), and a purchase order number to cover repair costs in the event the unit is determined by Parker Compumotor to be out of warranty.

Step ②

Before you ship the drive to Parker Compumotor, have someone from your organization with a technical understanding of the ZX and its application include answers to the following questions:

- ❏ What is the extent of the failure/reason for return?
- ❏ How long did it operate?
- ❏ How many units are still working?
- ❏ How many units failed?
- ❏ What was happening when the unit failed (i.e., installing the unit, cycling power, starting other equipment, etc)?

- ❑ How was the product configured (in detail)?
- ❑ What, if any, cables were modified and how?
- ❑ With what equipment is the unit interfaced?
- ❑ What was the application?
- ❑ What was the system sizing (speed, acceleration, duty cycle, inertia, torque, friction, etc.)?
- ❑ What was the system environment (temperature, enclosure, spacing, unit orientation, contaminants, etc.)?
- ❑ What upgrades, if any, are required (hardware, software, user guide)?

Step ③ Call Parker Compumotor [(800) 358-9070] for a Return Material Authorization (RMA) number. Returned products cannot be accepted without an RMA number.

Step ④ Ship the unit to:

Parker Compumotor Corporation
5500 Business Park Drive
Suite D
Rohnert Park, CA 94928
Attn: RMA # xxxxxxx

A P P E N D I X

Command Listing

"—Quote Command
;—Comment Field
#—Step Sequence

A—Acceleration
AD—Deceleration

B—Buffer Status Report
BCCA—Buffered Configure Current Average
BCCP—Buffered Configure Current Peak
BCDB—Buffered Configure Deadband
BCPD—Buffered Configure Position Derivative
BCPE—Buffered Configure Position Error
BCPI—Buffered Configure Position Integral
BCPP—Buffered Configure Position Proportional
BCPPM—Buffered Configure Pos Proportional Max.
BCRR—Buffered Configure Resolver Resolution
BCTC—Buffered Configure Time Constant
BCTG—Buffered Configure Tach Gain
BCTGM—Buffered Configure Tach Gain Maximum
BCVF—Buffered Configure Velocity Feed-Forward
BCVFM—Buffered Configure Vel Feed-Forward Max.
BCVI—Buffered Configure Velocity Integral
BCVIM—Buffered Configure Velocity Integral Maximum
BCVP—Buffered Configure Velocity Proportional
BCVPM—Buffered Configure Vel Proportional Max.
BL—Backlash
BS—Buffer Status Report

C—Continue
CCA—Configure Current Average
CCP—Configure Current Peak
CDB—Configure Deadband
CMR—Configure Motor Resolution
CMTR—Configure Motor
CPB—Configure Pushbuttons
CPD—Configure Position Derivative
CPDM—Configure Position Derivative Maximum
CPE—Configure Position Error
CPI—Configure Position Integral
CPIM—Configure Position Integral Maximum
CPP—Configure Position Proportional
CPPM—Configure Position Proportional Max.
CRR—Configure Resolver Resolution
CTG—Configure Tach Gain
CTGM—Configure Tach Gain Maximum
CVE—Configure Velocity Error
CVF—Configure Velocity Feed-Forward
CVFM—Configure Velocity Feed-Forward Max.
CVI—Configure Velocity Integral
CVIM—Configure Velocity Integral Maximum
CVP—Configure Velocity Proportional
CVPM—Configure Velocity Proportional Maximum
CR—Carriage Return

D—Distance
DCA—Display Current Average
DCI—Display Current Instantaneous
DCP—Display Current Peak
DFS—Display Flags for Servo Parameters
DFX—Display Flags for Indexer Status
DIN—Disable Inputs
DOUT—Disable Outputs
DP—Distance Point
DPA—Display Actual Position
DPE—Display Position Error
DPS—Display Setpoint Position
DR—Display Parameters
DRD—Read Distance Via Parallel I/O
DSP—Display Servo Picture
DVA—Display Actual Velocity
DVS—Display Velocity Setpoint

E—Enable RS-232C Communication Interface
ELSE—Else

F—Disable RS-232C Communication Interface
FAC—Set Following Synchronization Rate
FBS—Following Base
FC—Following Learn Count
FEN—Set Following Synchronization Count
FIN—Following Increment
FMCA—Find Motor Commutation Angle
FOL—Following Percent
FOR—Following Ratio
FP—Following Encoder Distance Point
FPA—Following Encoder Absolute Point
FRD—Read Following Via Parallel I/O
FS—Encoder Function Report
FSA—Enable Following Mimic Mode
FSF—Enable Following Synchronized Accel
FSI—Enable/Disable Following Mode
FSK—Enable/Disable Following Learn Mode
FSL—Enable/Disable Correction Mode
FSN—Enable/Disable Pulse Following Mode
FSP—Enable/Disable Tracking Mode

G—Go
GD—Go Predefined Move
GDEF—Move Definition
GH—Go Home
GHA—Go Home Acceleration
GHAD—Go Home Deceleration
GHF—Go Home Final Velocity
GHV—Go Home Velocity
GOSUB—GOSUB Sequence
GOTO—GOTO Sequence

^H—Backspace
H—Set Direction
HALT—Halt

ID—Immediate Distance
IF—If
IN—Set Input Functions
INL—Set Active Input Level
IO—Immediate Output
IS—Input Status Report
IV—Immediate Velocity

JA—Jog Acceleration
JAD—Jog Deceleration
JVH—Jog Velocity (High)
JVL—Jog Velocity (Low)

K—Kill

L—Loop
LAD—Limit Deceleration
LD—Limit Disable
LF—Line Feed
LRD—Read Loop Count via Parallel I/O

MA—Mode Alternate
MC—Mode Continuous
MN—Mode Normal
MPA—Mode Position Absolute
MPI—Mode Position Incremental
MPP—Mode Position Profile

N—End of Loop
NG—End Position Profile
NIF—End of IF Command
NWHILE—End of While

O—Output
OFF—Off
ON—On
OS—Function Set-Up Report
OSA—Set Limit Input Active Level
OSB—Backup to Home Switch
OSC—Define Active State of Home Switch
OSD—Enable Resolver Z Channel Input

OSE—Jog Enable
OSF—Acknowledge STOP and KILL inputs on Power Up
OSG—Final Homing Direction
OSH—Reference Edge of Home Switch
OSI—Save Sequence Scan Mode on Stop
OUT—Set Output Functions
OUTL—Set Active Output Level
OUTP—Output on Position

PF—Follower Position Report
PFZ—Set Follower Position to Zero
PR—Absolute Position Report
PS—Pause
PU—Configure Square Wave
PUL—Activate Square Wave Output
PZ—Set Absolute Counter to Zero

R—Request ZX Status
RA—Limit Switch Status Report
RB—Loop, Pause, Shutdown, Trigger Status Report
REG—Configure Registration Move
REPEAT—Repeat
RFS—Return Servo Parameters to Factory Settings
RG—Go Home Status Report
RIFS—Return to Factory Settings
RS—Report Status of Sequence Execution
RSE—Report Servo Errors
RSIN—Set Variable Interactively
RV—Revision Level Report

S—Stop
SFL—Set User Flag
SL—Software Limits Enable
SLD—Software Limits Disable
SN—Scan Delay Time
SP—Set Position Absolute
SS—Function Set-Up Report
SSA—RS-232 Echo Control
SSD—Alternate Stop Mode
SSG—Clear/Save the Command Buffer on Limit
SSH—Clear/Save the Command Buffer on Stop
SSI—Enable/Disable Interactive Mode
SSJ—Enable/Disable Continuous Scan Mode
SSL—Enable/Disable Resume Execution
SSN—Enable/Disable Message Mode
SSU—Enable/Disable Warning Display Mode
ST—Shutdown
STOP—Stop
STR—Set Strobe Output Delay Time
SV—Save

T—Time
TD—Set Input Debounce Time
TDR—Set Registration Debounce
TEST—System Test
TF—Set Following Time
TM—Move Time Report
TR—Wait for Trigger
TRD—Read Timer from Parallel I/O
TS—Trigger Input Status
TX—Transmit Variable and String

U—Pause and Wait for Continue
UNTIL—Until

V—Velocity
VAR—Variables
VARD—Read Variables from Parallel I/O
VRD—Read Velocity from Parallel I/O
VS—Set Start/Stop Velocity

W1—Signed Binary Position Report
W2—Hexadecimal Position Report
W3—Signed Hexadecimal (±) Position Report
WHEN—Set WHEN COndition
WHILE—While

XBS—Report Sequence Memory Available
XC—Sequence Checksum Report
XD—Sequence Definition
XDIR—Sequence Directory
XE—Sequence Erase
XEALL—Erase All Sequences
XFK—Set Fault or Kill Sequence
XG—GOTO Sequence
XQ—Sequence Interrupted Run Mode
XR—Run Sequence
XRD—Read Sequence via Parallel I/O
XRP—Sequence Run with Pause
XS—Sequence Execution Status
XSD—Sequence Status Definition Report
XSR—Sequence Status Run Report
XSS—Sequence Status Report
XST—Sequence Step Mode
XT—Sequence Termination
XTR—Set Trace Mode
XU—Upload Sequence
XWHEN—When Sequence

Y—Stop Loop

Z—Reset

I N D E X

A

ACCELERATION MODE 101
ACCURACY 48, 164
ADVANCE
  ENCODER SAMPLE RATE 110
  INITIAL FOLLOWING PERCENTAGE 111
ALPHANUMERIC DISPLAY 45
ANALOG CONTROL LOOP 41
APPLICATION EXAMPLES 88

B

BACKLASH 53
BACKLASH COMPENSATION 53
BATTERY 173
BENCH TEST 6
  ZX600/ZX800 7
  ZX900 7
BOOLEAN COMPARISONS 69
BRANCHING 68
BRANCHING COMMANDS 65
BREAKPOINT 122
BREAKPOINTS 114

C

COMMANDS
  BUFFERED 56
  FAC 100, 101, 111, 120
  FEN 100, 101, 111, 120
  FOL 100, 111
  FOLI 117
  FOR 120
  FP 114
  FPA 114
  IMMEDIATE 56
  SEQUENCE STATUS COMMANDS 55
CONDITIONAL BRANCHING 70
CONDITIONAL LOOPING 69
CONNECTIONS
  ENCODER 34
  MOTOR CABLE PIN-OUT 170
  RESOLVER CABLE PIN-OUT 170
  RESOLVER PIN-OUT 28
  RS-232C 10
  ZX600/800 .I.RESOLVER 8
  ZX600/800 MOTOR CABLE CONNECTION 9
  ZX600/800 POWER 35
  ZX600/800 POWER CABLE 10
  ZX900 MOTOR CABLE CONNECTION 9
  ZX900 POWER 35
  ZX900 POWER CABLE 10
  ZX900 RESOLVER CABLE 8
CONNECTORS
  ENCODER 171
  MOTOR 26
  RESOLVER 26
CONTROL LOOPS 41
COUPLERS 26

D

DEBOUNCE TIME 52
DIAGNOSTIC LEDS 174
DIGITAL CONTROL LOOP 41
DIMENSIONS
  ZX600/ZX800 INDEXER/DRIVE 14
  ZX900 INDEXER/DRIVE 14
DISPLAY AND PUSHBUTTONS 45
DRIVE/MOTOR CONFIGURATION 5
  120VAC VERSION 5
  240VAC VERSION 5

E

ENCODER
  PHASE A 129
  PHASE B 129
  QUADRATURE SIGNALS 129
  Z CHANNEL 54
ENCODER CONNECTION 34
ENCODER INPUTS 34
ENVIRONMENTAL CONSIDERATIONS 13
ERROR FLAG 69
ERROR MESSAGES 64

F

FACTORY DEFAULT SETTINGS 172
FAULT
  MESSAGES 45
FAULT CODES 175
FOLLOWING 97, 117
  ADVANCE 109, 132
  ADVANCE PROFILE 132
  BREAKPOINT 117
  BREAKPOINTS 122
  CAM 133
  CAM CYCLE 120
  CAM FOLLOWING 120
  CAM PROFILE 133
  CONSTANT 117
  EQUATIONS 130
  EXAMPLES 104, 120
  FOLLOWING ACCELERATION 100
  JOGGING 127
  POSITION AND VELOCITY 99
  RATIO 128
  RECEDE 109, 132
  RECEDE PROFILE 132
  REGISTRATION 126
  SOFTWARE COMMANDS THAT ARE ASSOCIATED WITH FOLLOWING. 129
  SYSTEM CONFIGURATION 95
  TIME-BASED MOTION 103
  VELOCITY FOLLOWING 97
FOLLOWING ENCODER ABSOLUTE POINT 114
FOLLOWING? 95
HARDWARE INTERFACES 172

H

HOMING 39, 53
  HM INPUT 32
HOST COMPUTER 87

I

I/O[1] CONNECTORS 29
  CCW 32
  CW 32
  ENBL INPUT 32
  HOME LIMIT 32
  PROGRAMMABLE INPUTS 33
  PROGRAMMABLE OUTPUTS 33
  RS-232C 29
  VOUT INPUT 32
I/O[2] CONNECTOR

ANALOG MONITOR OUTPUT 34
DRIVE FAULT RELAY OUTPUT I/O[2] 34
I/O[2] SCREW TERMINAL 33
IMMEDIATE TERMINAL MODE 66
INCREMENTAL ENCODER 171
INPUT & OUTPUT CIRCUIT 171
INPUT FUNCTIONS 36, 77

J

JOGGING 52
JOG ACCELERATION 127

L

LIMIT SWITCHES 38
LIMITS 172
CCW 38
CW 38
LOOPING 68

M

MAIN MENU 45
MATH OPERATIONS 67
MENUS 46
MODES
ABSOLUTE 50
CONTINUOUS 49
ERROR MESSAGE MODE 64
FOLLOWING MODE 95
INCREMENTAL 50
INTERACTIVE 31
MOTION PROFILING 49, 72
PREDEFINED 49
PRESET 49
PULSE AND DIRECTION 129
PULSE TRACKING 100
SELF LEARN 125
SEQUENCE SCAN 74, 77
SEQUENCE SELECT 77
SINGLE-STEP 62
SYNCHRONIZATION 124
TRACE 61, 176
MOTION PROFILING 114, 119
MOTION PROFILING MODE 72
MOTOR BRAKING
WIRING 28
MOTOR BRAKING (DYNAMIC) 27
MOTOR DIMENSIONS
Z605 - Z610 16
Z620 & Z630 17
Z635 & Z640 18
Z805 & 806 19
Z810 20
Z820, 830, & 840 21
Z910 23
Z920 & 930 24
Z940 25
MOTOR MAINTENANCE 173
MOTOR SPECIFICATIONS
Z605 144
Z606 145
Z610 146
Z620 147
Z630 148
Z635 149
Z640 150
Z806 156
Z810 157
Z820 158
Z830 159
Z840 160
Z910 165
Z920 166
Z930 167
Z940 168
MOUNTING
AIR FLOW 15
DRIVE 13
ENCLOSURE 15
MOTORS 16
PANEL LAYOUT 15
PANEL LAYOUT GUIDELINES 15

O

OUTPUT FUNCTIONS 76
OVERSHOOT 100

P

PLC 85
CONNECTION 85
DATA STROBE RATE 85
ENTER DATA 85
EXECUTE SEQUENCES 86
POSITION MODE TUNING 44
POWER 34, 35
PRODUCT
LABELS 6
PROGRAMMABLE
INPUTS 36
OUTPUTS 38, 76
PROGRAMMABLE INPUTS 76, 171
PROGRAMMABLE OUTPUTS 76, 171
PROGRAMMING
CONDITIONAL STATEMENTS 69
CONSTANTS 66
VARIABLES 65
PUSHBUTTONS 45, 128

R

REGISTRATION 51, 52
REGISTRATION INPUT 127
REGISTRATION IN FOLLOWING MODE 127
REMOTE PANEL 87
RP240 87
REPAIR 178
REPEATABILITY 48
RESOLVER ELECTRICAL DRAWING 4
RESOLVER MECHANICAL DRAWING 4
RETURN MATERIAL AUTHORIZATION (RMA) 179
RP240 87
RS-232C 10, 172, 176
BAUD RATE 10, 30
DAISY CHAIN 31
DATA BITS 10
DEVICE ADDRESS 31, 172
ENBL INPUT 11
FULL DUPLEX MODE 10
INPUT 30
PARITY 10
STOP BITS 10

S

SEQUENCE
FAULT 58
POWER UP 60
WHEN 59
SEQUENCE SCAN MODE 74
SHIP KIT 5
SHUNT REGULATOR 135
DIMENSIONS 136
DRIVE OVERVOLTAGE 135
MECHANICAL ENERGY 137
SHUNT REGULATOR INSTALLATION 135
SPARE PARTS 173
SWITCHES 77
SYNCHRONIZATION 123
SYSTEM WIRING 26

T

TEMPERATURE
OPERATING 13
SENSORS 15
STORAGE 13
THUMBWHEELS 80, 83
WIRING 81
TM8 MODULE 80
TRACE MODE 176
TRANSFORMER 34
TRIGGERS 77
TROUBLESHOOTING 174
TUNING 43
COMMANDS 44
PARAMETERS 42
RS-232C INTERFACE 42
TUNING PARAMETER COMMANDS 42

U

USER FLAG 69

V

VELOCITY FOLLOWING 96

W

WARNING CODES 175

Z

ZX600
ELECTRICAL SPECIFICATIONS 141
MOTOR/DRIVE CONFIGURATION 142
PERFORMANCE SPECIFICATIONS 142
REPEATABILITY 142
RESOLVER ACCURACY 142
RESOLVER-TO-DIGITAL CONVERTER ACCURACY 142
SPEED/TORQUE CURVES 151
Z600 MOTOR BRAKE CHARACTERISTICS 143
Z605 MOTOR SPECIFICATIONS 144
Z606 MOTOR SPECIFICATIONS 145
Z610 MOTOR SPECIFICATIONS 146
Z620 MOTOR SPECIFICATIONS 147
Z630 MOTOR SPECIFICATIONS 148
Z635 MOTOR SPECIFICATIONS 149
Z640 MOTOR SPECIFICATIONS 150
ZX800
ELECTRICAL SPECIFICATIONS 153
PERFORMANCE SPECIFICATIONS 153
REPEATABILITY 154
RESOLVER ACCURACY 154
RESOLVER-TO-DIGITAL CONVERTER ACCURACY 154
SPEED/TORQUE CURVES 161
Z800 MOTOR BRAKE CHARACTERISTICS 154
Z805 MOTOR SPECIFICATIONS 155
Z806 MOTOR SPECIFICATIONS 156
Z810 MOTOR SPECIFICATIONS 157
Z820 MOTOR SPECIFICATIONS 158
Z830 MOTOR SPECIFICATIONS 159
Z840 MOTOR SPECIFICATIONS 160
ZX900
ELECTRICAL SPECIFICATIONS 163
INPUT POWER 163
PERFORMANCE SPECIFICATIONS 163
REPEATABILITY 164
RESOLVER ACCURACY 164
RESOLVER-TO-DIGITAL CONVERTER ACCURACY 164
SPEED/TORQUE CURVES REPRESENT THE AVAILABLE SHAFT TORQUE AT DIFFERENT OPERATING SPEEDS. OPERATION AT 240VAC IS SHOWN FOR EACH MOTOR SIZE. ACTUAL MOTOR TORQUE MAY VARY ±10% DUE TO MOTOR MANUFACTURING VARIANC 169
Z900 MOTOR BRAKE CHARACTERISTICS 164
Z910 MOTOR SPECIFICATIONS 165
Z920 MOTOR SPECIFICATIONS 166
Z930 MOTOR SPECIFICATIONS 167
Z940 MOTOR SPECIFICATIONS 168

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

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

ARTISAN® TECHNOLOGY GROUP