CHAPTER 2

Installation

Complete the following installation steps before you use the OEM770X drive.

Installation Steps:

1. Verify shipment is correct.
2. Install selectable resistors.
3. Mount the drive.
4. Mount the motor.
5. Connect the motor to the drive.
6. Connect inputs, outputs, and controller.
7. Connect a power supply to the drive.
8. Tune the drive.

The sections in this chapter give basic instructions about how to complete each of these steps.

OEM770X Ship Kit

Inspect the OEM770X upon receipt for obvious damage to its shipping container. Report any damage to the shipping company. Parker Compumotor cannot be held responsible for damage incurred in shipment. You should receive one or more drives, depending upon what you ordered. Compare your order with the units shipped.

Component	Part Number
OEM770X Drive	OEM770X
Resistor Kit	73-018496-01
Accessories	
OEM770X User Guide	88-018468-01
Heatsink	OEM -HS1

User guides are not sent with each product. They are available upon request. Please order user guides as needed.

The following SM and NeoMetric Series servo motors are designed to be used with the OEM770X. Compare your order with the motors shipped.

Motor Size	Part Number
Size 16	SM160A, SM160B, SM161A, SM161B SM162A, SM162B
Size 23	SM230A, SM230B, SM231A, SM231B, SM232A, SM232B, SM233A, SM233B
Size 34	NO341D, NO341F, NO342E, NO342F JO341D, JO341F, JO342E, JO342F
Size 70mm	NO701D, NO701F, NO702E, NO702F JO701D, JO701F, JO702E, JO702F

Installing Selectable Resistors and Jumper JU1

You must install four resistors into sockets on the OEM770X's circuit board. Three of these are *foldback resistors*; they determine the parameters for the current foldback circuit, which can protect your motor from overheating due to prolonged high currents. The fourth resistor is a *response resistor*—it affects the gain and frequency response of the current loop.

The OEM770X ships with resistors and jumper installed. These resistors are not appropriate for most applications. You *must* select other resistors and install them in the drive.

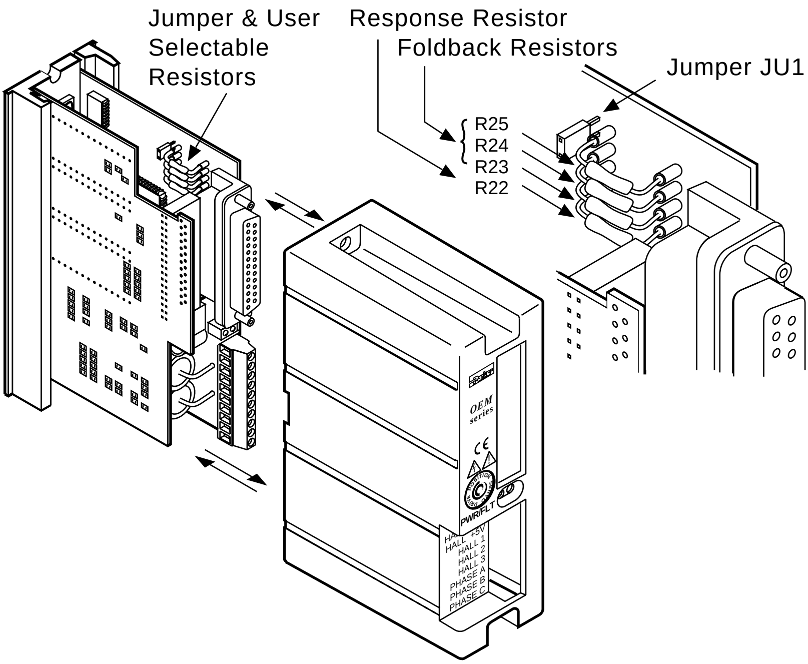
You can also install jumper JU1, located near the resistors, to adjust drive performance for your particular motor.

A resistor kit for use with Compumotor SM and NeoMetric Series motors is included with the drive. If the resistors are color coded, a key to the code is included in the kit. If the resistors have a numerical code, the first three digits are resistance values; the fourth digit is a multiplier.

Example: 3013 = 301 x 10^3 = 301KΩ
6492 = 649 x 10^2 = 64.9 KΩ
Note: zero ohm resistors may be color coded (black band)

To install resistors or the jumper, remove the drive's molded plastic cover. Apply pressure to the D-connector while you hold the cover's sides. The circuit board will slide out. The resistors and jumper are located at the corner of the board, near the 25 pin D-connector, as shown in the next drawing.

WARNING
Remove power from the OEM770X before installing selectable resistors.



Selectable Resistor and Jumper Locations

Remove any resistors that are in the sockets, and install those that you have selected. The next table shows recommended resistors for Compumotor SM and NeoMetric Series motors. For full details on further customizing the response and foldback circuits, or choosing resistors for non-Compumotor motors, see *Chapter 4 Special Internal Circuits*.

The next table also shows jumper position—installed or removed—for Compumotor motors.

RESISTOR & JUMPER SELECTION FOR COMPUMOTOR MOTORS

Use the table below to select resistors and jumper position for Compumotor motors. (The next section shows default values.)

OEM770X – Resistor and Jumper Settings for Motors at 75VDC*

Motor	R22 ($R_{response}$)	R23 ($T_{c\text{-}therm}$)	R24 ($I_{pk\text{-}tune}$)	R24 ($I_{pk\text{-}final}$)	R25 (I_{fold})	Jumper Installed
SM160A	249 KΩ	5.1 MΩ	348 KΩ (5 A)	150 KΩ (7.5 A)	1.2 MΩ (2.2 A)	no
SM160B	750 KΩ	10 MΩ	64.9 KΩ (10 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
SM161A	301 KΩ	5.1 MΩ	450 KΩ (4 A)	249 KΩ (6 A)	1.2 MΩ (2.2 A)	no
SM161B	750 KΩ	10 MΩ	124 KΩ (8 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
SM162A	205 KΩ	5.1 MΩ	450 KΩ (4 A)	249 KΩ (6 A)	1.2 MΩ (2.2 A)	no
SM162B	402 KΩ	10 MΩ	124 KΩ (8 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
SM230A	402 KΩ	5.1 MΩ	348 KΩ (5 A)	150 KΩ (7.5 A)	1.2 MΩ (2.2 A)	no
SM230B	1 MΩ	10 MΩ	64.9 KΩ (10 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
SM231A	402 KΩ	5.1 MΩ	450 KΩ (4 A)	249 KΩ (6 A)	1.2 MΩ (2.2 A)	no
SM231B	604 KΩ	10 MΩ	124 KΩ (8 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
SM232A	205 KΩ	5.1 MΩ	450 KΩ (4 A)	249 KΩ (6 A)	1.2 MΩ (2.2 A)	no
SM232B	500 KΩ	10 MΩ	124 KΩ (8 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
SM233A	30.1 KΩ	5.1 MΩ	450 KΩ (4 A)	249 KΩ (6 A)	1.2 MΩ (2.2 A)	yes
SM233B	700 KΩ	10 MΩ	124 KΩ (8 A)	0 Ω (12 A)	124 KΩ (3.0 A)	no
NO701D/NO341D	205 KΩ	10 MΩ	249 KΩ (6 A)	90.9 KΩ (9 A)	1.2 MΩ (2.2 A)	yes
NO701F/NO341F	750 KΩ	10 MΩ	90.9 KΩ (9 A)	0 Ω (12 A)	124 KΩ (3.0 A)	yes
NO702E/NO342E	750 KΩ	10 MΩ	182 KΩ (7 A)	64.9 KΩ (10 A)	1.2 MΩ (2.2 A)	yes
NO702F/NO342F	604 KΩ	10 MΩ	90.9 KΩ (9 A)	0 Ω (12 A)	124 KΩ (3.0 A)	yes

* For supply voltages less than 75VDC, calculate R22 using the following equation: $R22_{new} = (R22_{old} \bullet V_{bus})/75$, where $R22_{old}$ is the value from the table above (at 75VDC). R23, R24, R25 remain the same as for 75VDC.

R24 – "pk-tune" and "pk-final"

Note that there are two values recommended for R24. Use the first value (*pk-tune*) when you begin your tuning procedure. This keeps peak currents low, to avoid the damaging currents that instability during tuning can cause. As you refine your tuning settings, replace R24 with the second value (*pk-final*), if your application requires more torque.

RESISTOR & JUMPER SELECTION FOR NON-COMPUMOTOR MOTORS

The following sections describe how to choose resistor values and jumper position for other motors.

Selecting Foldback Resistors

The OEM770X ships with resistors already installed.

Default Foldback Resistors (as shipped)

Res. #:	Function	Value
R25	Foldback Current	23.7 KΩ (6A)
R24	Peak Current	Ø Ω (12A)
R23	Time Constant	5.1 MΩ

The default values above may not be suitable for your application. If your system cannot withstand the peak torque, you should determine foldback resistor values appropriate to your application and install them in your drive.

For full details about how to choose foldback resistor values, and about how the foldback circuit works, see *Chapter 4 Special Internal Circuits*.

Selecting a Response Resistor

The OEM770X ships with a response resistor already installed.

Default Response Resistor (as shipped)

Res. #:	Function	Value
R22	Optimize gain and frequency response	100 KΩ

If your motor is not well matched to the default resistor, your system might not perform as well as you expect. In this case, improve your system's performance by selecting an appropriate response resistor, and installing it in the drive.

For full details about how to choose a value for the response resistor, and about how the circuit works, see *Chapter 4 Special Internal Circuits*.

Selecting Jumper Position for Non-Compumotor Motors

You can adjust the performance of the OEM770X's internal error amplifier by installing or removing jumper JU1. The drive ships with the jumper installed.

For motors with long electrical time constants (L/R), such as Compumotor's NeoMetric motors, install the jumper. Remove the jumper for motors with short time constants, such as Compumotor's SM motors (except SM233A).

Jumper Position Selection Procedure

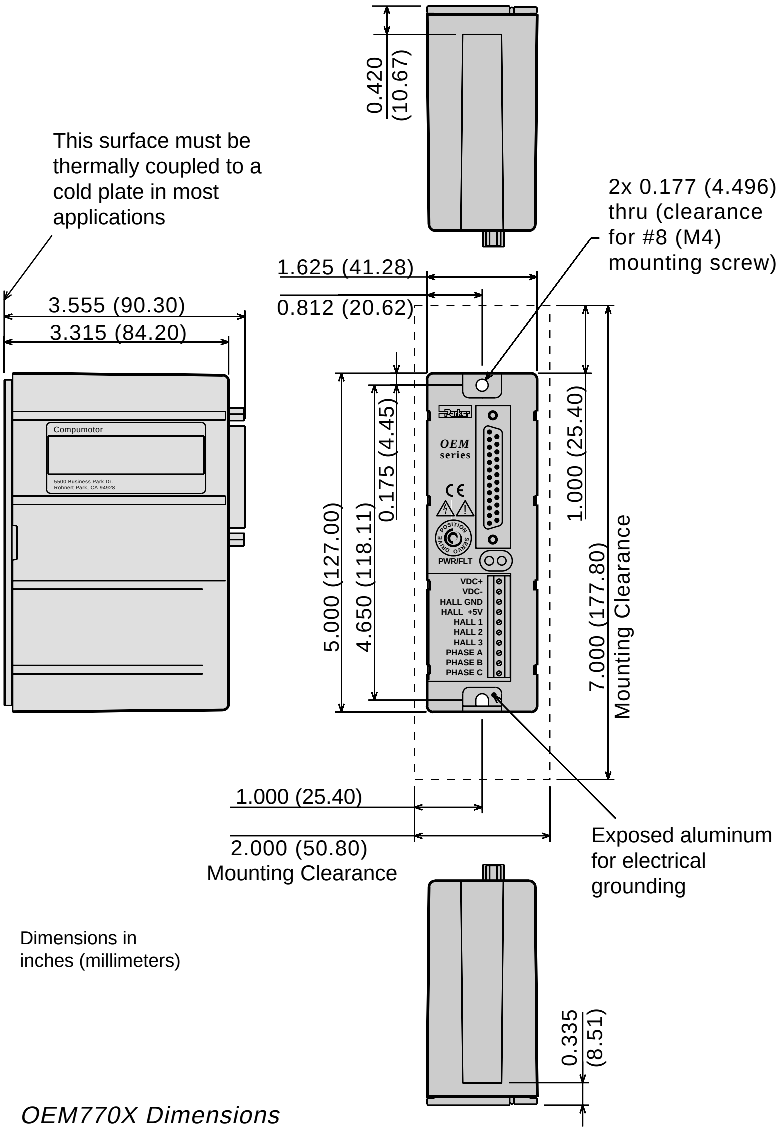
1. **Adjust R22 with Jumper JU1 Installed**
Starting with a high value, adjust R22 for optimum system response. For adjustment instructions, see *Selecting a Response Resistor* in *Chapter 4 Special Internal Circuits.*

2. **If Unable to Obtain an Optimum Response:**
Chapter 4 Special Internal Circuits describes optimum responses. If you could not obtain an optimum response in *Step 1*—your adjustments produced overdamped or underdamped responses, with no range of optimum responses in between—then:
   - Replace R22 with a high value, to limit oscillations during *Step 3* below.
   - Remove Jumper JU1.

3. **Adjust R22 with Jumper JU1 Removed**
With Jumper JU1 removed, adjust R22 to achieve an optimum system response.

For further help, provide your motor's inductance (*L*) and resistance (*R*) values to Compumotor's Applications Department. We can calculate a recommended jumper position and R22 value, based on your motor's values.

Drive Mounting



OEM770X Dimensions

DRIVE DIMENSIONS

The OEM770X is designed to minimize panel area, or footprint, in an equipment cabinet. Dimensions are shown in the drawing. You can mount the drive in a “minimum depth” configuration if you use an optional heatsink. (See below.)

PANEL LAYOUT

Move profiles and loads affect the amount of heat dissipated by the OEM770X. Applications with low average power (less than 3 Amps continuous motor current) and mild ambient temperatures may not require a heatsink.

The OEM770X is designed to operate within the following temperature guidelines:

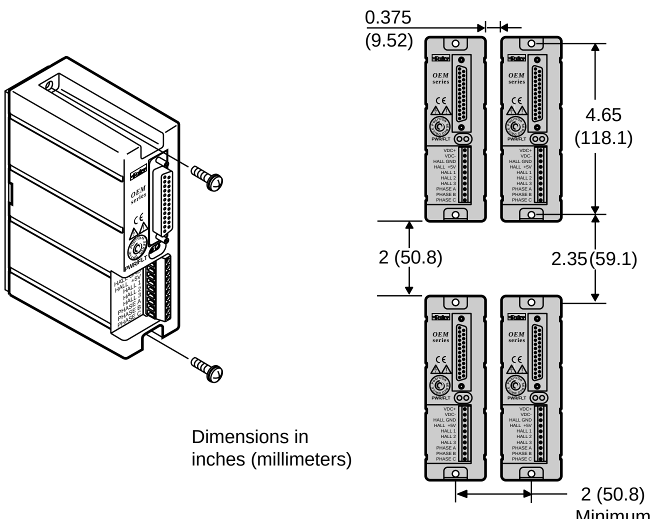
- Maximum Ambient Temperature: 45°C (113°F)
- Maximum Heatsink Temperature 45°C (113°F)

For applications with higher power or elevated ambient temperatures, you may need to mount the drive in a way that removes heat from it. The drive uses a heatplate design as a pathway to dissipate its excess heat; it should be mounted to a heatsink or a suitable heat sinking surface.

The OEM770X is overtemperature protected. (See *Chapter 4 Special Internal Circuits* for more information.)

Mounting Without a Heatsink

The next drawing shows the recommended panel layout for mounting the OEM770X without a heatsink.



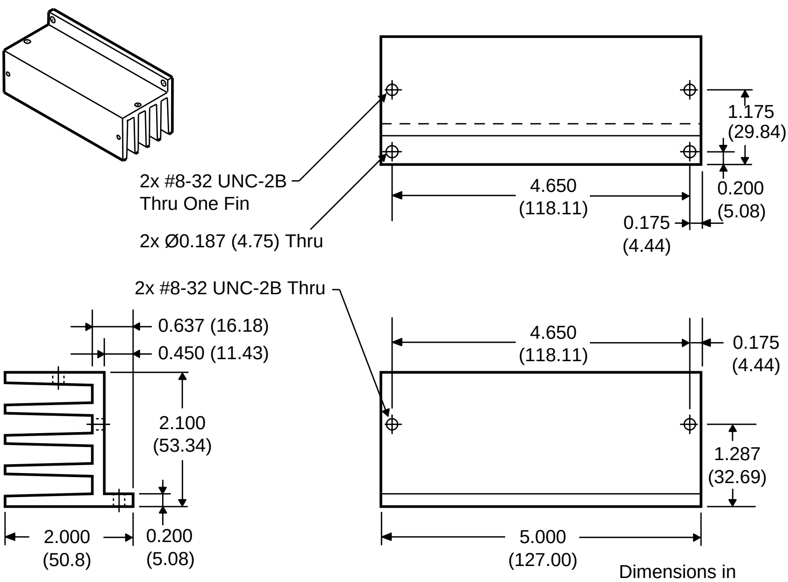
Panel Layout (Without a Heatsink)

Mounting With Compumotor Heatsink OEM-HS1

A heatsink designed to work with the OEM770X can be purchased from Compumotor (Part Number OEM-HS1). This heatsink is sufficient for most applications operating in 45°C (113°F) or lower ambient temperatures.

The drive may be mounted in two different configurations. One configuration uses a minimum amount of mounting area (*minimum area*). The other configuration uses a minimum amount of mounting depth (*minimum depth*).

Heatsink dimensions are shown in the next drawing.



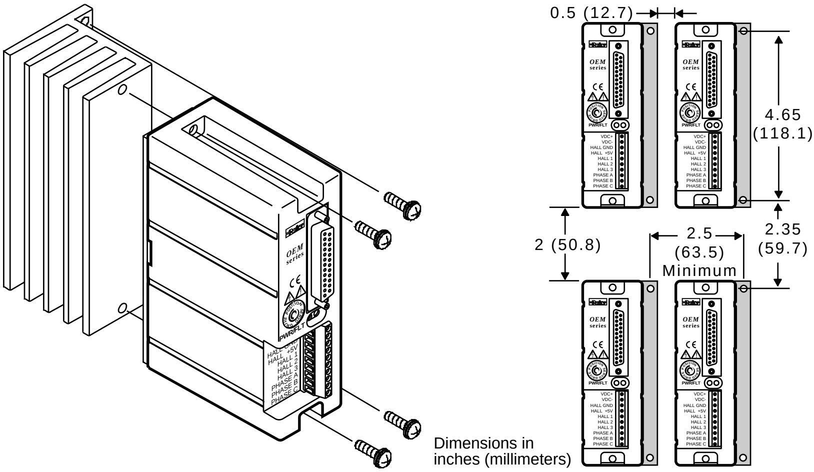
OEM-HS1 Heatsink Dimensions

Two #8-32 screws are needed to mount the OEM770X to the OEM-HS1 heatsink. Use a *star washer* on the bottom screw to ensure proper electrical grounding. Use two #8 screws to mount the OEM-HS1 to the cabinet.

Do not use a star washer between the back of the OEM770X heatplate and the mounting surface. The mounting surface must be flat. Use silicone thermal joint compound or thermal pads to facilitate heat transfer from the drive's heatplate to your mounting surface.

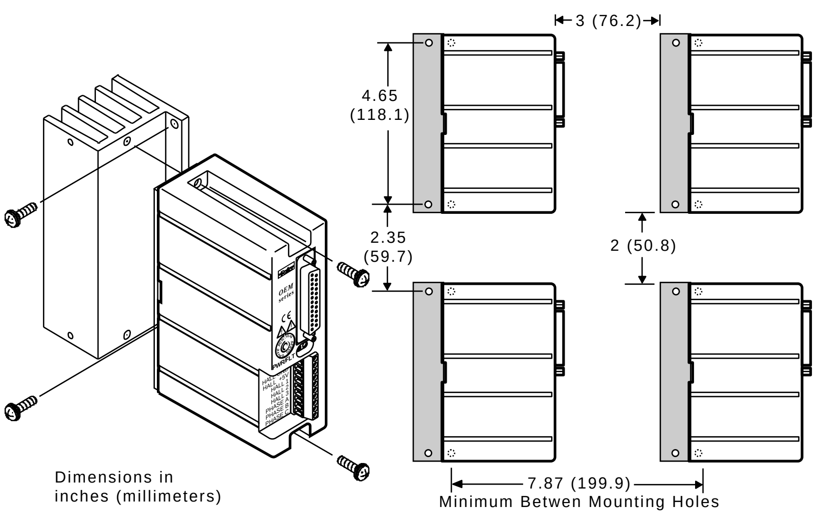
A heatsink with holes tapped for metric screws is available. Its part number is OEM-HS1-M4. See your Compumotor sales guide for more information.

The next drawing shows the panel layout for minimum area.



OEM-HS1 Minimum Area Panel Layout

The following drawing shows dimensions for a minimum depth panel layout.



OEM-HS1 Minimum Depth Panel Layout

Motor Mounting

The following guidelines present important points about motor mounting and its effect on performance.

For mechanical drawings of SM and NeoMetric Series servo motors, see *Chapter ¬ Specifications*

WARNING
Improper motor mounting can reduce system performance and jeopardize personal safety.

Servo motors used with the OEM770X can produce large torques and high accelerations. This combination can shear shafts and mounting hardware if the mounting is not adequate. High accelerations can produce shocks and vibrations that require much heavier hardware than would be expected for static loads of the same magnitude.

The motor, under certain move profiles, can produce low-frequency vibrations in the mounting structure. These vibrations can cause metal fatigue in structural members if harmonic resonances are induced by the move profiles you are using. A mechanical engineer should check the machine design to ensure that the mounting structure is adequate.

CAUTION
Consult a Compumotor Applications Engineer (800-358-9070) before you machine the motor shaft. Improper shaft machining can destroy the motor's bearings. Never disassemble the motor.

Servo motors should be mounted by bolting the motor's face flange to a suitable support. Foot mount or cradle configurations are not recommended because the motor's torque is not evenly distributed around the motor case. Any radial load on the motor shaft is multiplied by a much longer lever arm when a foot mount is used rather than a face flange.

Motor Heatsinking

Performance of a servo motor is limited by the amount of current that can flow in the motor's coils without causing the motor to overheat. Most of the heat in a brushless servo motor is dissipated in the stator—the outer shell of the motor. Performance specifications usually state the maximum allowable case temperature. Exceeding this temperature can permanently damage the motor.

If yours is a demanding application, your motor may become quite hot. The primary pathway through which you can remove the heat is through the motor's mounting flange. Therefore, mount the motor with its flange in contact with a suitable heatsink.

Specifications for Compumotor SM and NeoMetric Series servo motors apply when the motor is mounted to a ten inch by ten inch aluminum mounting plate, 1/4 inch thick. To get rated performance in your application, you must mount the motor to a heatsink of at least the same thermal capability. Mounting the motor to a smaller heatsink may result in decreased performance and a shorter service life. Conversely, mounting the motor to a larger heatsink can result in enhanced performance.

ATTACHING THE LOAD

Your mechanical system should be as stiff as possible. Because of the high torques and accelerations of servo systems, the ideal coupling between a motor and load would be completely rigid. Rigid couplings require perfect alignment, however, which can be difficult or impossible to achieve. In real systems, some misalignment is inevitable. Therefore, a certain amount of flexibility may be required in the system. Too much flexibility can cause resonance problems, however.

These conflicting requirements are summarized below.

- Maximum Stiffness (in the mechanical system)
- Flexibility (to accommodate misalignments)
- Minimum Resonance (to avoid oscillations)

The best design solution may be a compromise between these requirements.

MISALIGNMENT & COUPLERS

The type of misalignment in your system will affect your choice of coupler.

Parallel Misalignment
The offset of two mating shaft center lines, although the center lines remain parallel to each other.

Angular Misalignment
When two shaft center lines intersect at an angle other than zero degrees.

End Float
A change in the relative distance between the ends of two shafts.

There are three types of shaft couplings: single-flex, double-flex, and rigid. Like a hinge, a single-flex coupling accepts angular misalignment only. A double-flex coupling accepts both angular and parallel misalignments. Both single-flex and double-flex, depending on their design, may or may not accept endplay. A rigid coupling cannot compensate for any misalignment.

Single-Flex Coupling

When a single-flex coupling is used, one and only one of the shafts must be free to move in the radial direction without constraint. ***Do not use a double-flex coupling in this situation:*** it will allow too much freedom and the shaft will rotate eccentrically, which will cause large vibrations and catastrophic failure. ***Do not use a single-flex coupling with a parallel misalignment:*** this will bend the shafts, causing excessive bearing loads and premature failure.

Double-Flex Coupling

Use a double-flex coupling whenever two shafts are joined that are fixed in the radial and angular direction. (This is the most common situation. It results from a combination of angular and parallel misalignment).

Rigid Coupling

As mentioned above, rigid couplings would be ideal in servo systems, but are not generally recommended because of system misalignment. They should be used only if the motor or load is on some form of floating mounts that allow for alignment compensation. Rigid couplings can also be used when the load is supported entirely by the motor's bearings. A small mirror connected to a motor shaft is an example of such an application.

RESONANCE ISSUES

A coupler that is too flexible may cause a motor to overshoot its commanded position. When the encoder sends a position feedback signal, the controller will command a correction move in the opposite direction. If the resonant frequency of the system is too low (too flexible), the motor may overshoot again and again. In extreme cases, the system could become an oscillator.

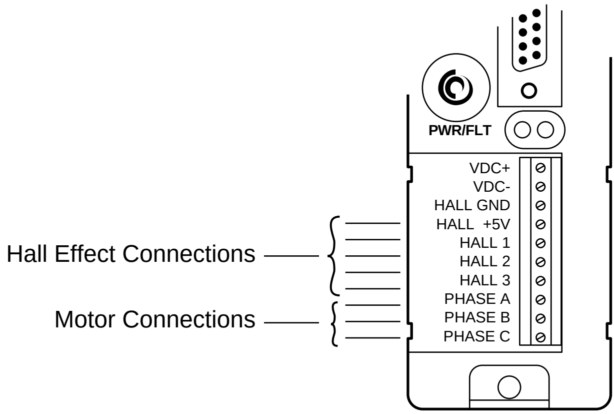
To solve resonance problems, increase the mechanical stiffness of the system to raise the resonant frequency so that it no longer causes a problem.

If you use a servo as a direct replacement for a step motor, you may need to modify your mechanical coupling system to reduce resonance. For example, we recommend using a bellows-style coupler with servo motors, rather than the helical-style coupler that is often used with step motors. Helical couplers are often too flexible, with resonant frequencies that can cause problems. Bellows couplers are stiffer, and perform better in servo systems.

Connecting a Motor to the Drive

The OEM770X drive is designed to work with three-phase brushless motors equipped with Hall effect sensors or equivalent feedback signals. The typical motor has a permanent-magnet rotor with four poles (two pole pairs).

Connect your motor's phase wires and Hall effect sensor wires to the 10-pin screw terminal on the OEM770X. Each terminal is labeled with the name of the wire you should connect to it.



10-Pin Screw Terminal

14 AWG (2.5 mm^2) is the maximum wire size that can fit in the connector.

CAUTION

Do not turn on power unless the motor's Hall effect sensors, Hall +5, and Hall GND are connected to the drive. The motor may be destroyed by overheating if these connections are not made.

If the Hall effects are not connected, the drive determines that it is configured to run a *brushed* servo motor. With power and a command input applied, the drive will send the commanded DC current through the motor. If the motor is a *brushless* motor, it will not turn. Full current may flow in the motor and cause overheating, or destroy the motor within a short period of time.

CONNECTING COMPUMOTOR SM AND NEOMETRIC SERIES MOTORS

To connect a Compumotor SM or NeoMetric Series motor to the OEM770X, follow the color code shown below for flying lead or cable versions. (These motors have additional wires not used by the OEM770X. See *Chapter 3 Specifications* for colors and functions of the additional wires.)

Function	Wire Color
Hall Ground	White/Green
Hall +5V	White/Blue
Hall 1	White/Brown
Hall 2	White/Orange
Hall 3	White/Violet
Phase A	Red/Yellow
Phase B	White/Yellow
Phase C	Black/Yellow

Connect each motor wire to its appropriate screw terminal on the OEM770X. Wire sizes used for Compumotor motors are:

Phase:	18 AWG (O.75 mm^2)
Hall/Encoder:	24 AWG (0.25 mm^2)

CONNECTING MOTORS FROM OTHER VENDORS

Before connecting a motor from another vendor, you must determine which motor phase wires correspond to Phase A, Phase B, and Phase C on the OEM770X. Similarly, you must determine which Hall effect wires correspond to Hall 1, Hall 2, and Hall 3.

Connect each wire to its appropriate terminal on the OEM770X. Ensure that the Hall effect sensors accurately transmit information about rotor position, and that motor current is commutated to the correct motor phases. See *Chapter 5 Hall Effect Sensors* for more information.

If your drive arrived with a response resistor installed, you should consider using a different response resistor. See *Chapter 4 Special Internal Circuits* for details about selecting a response resistor to improve your system’s perfor-mance.

CONNECTING A BRUSHED DC SERVO MOTOR

You can use the OEM770X as a drive for brushed DC servo motors. Follow these steps:

1. Connect HALL 1 and HALL 2 to HALL GND.
2. Make no connections to HALL 3.
3. Connect the drive's Phase A to your motor's positive input.
4. Connect the drive's Phase C to your motor's negative input.

Under these conditions, the drive's internal logic determines that a brushed motor is connected. DC current will flow out of Phase A, through the motor, and back into the drive through Phase C. The amount and polarity of the current will be determined by the command input signal.

SHIELDED MOTOR CABLES

Prevent electrical noise from interfering with the signals that the Hall effect sensors send to the drive. Position the motor as close to the drive as possible. If you need to connect a long cable between the drive and motor, we recommend you use a shielded cable for the Hall wires (Hall 1, Hall 2, Hall 3, +5V, GND). Run the power wires (phase A, B, and C) separately from the Hall wires.

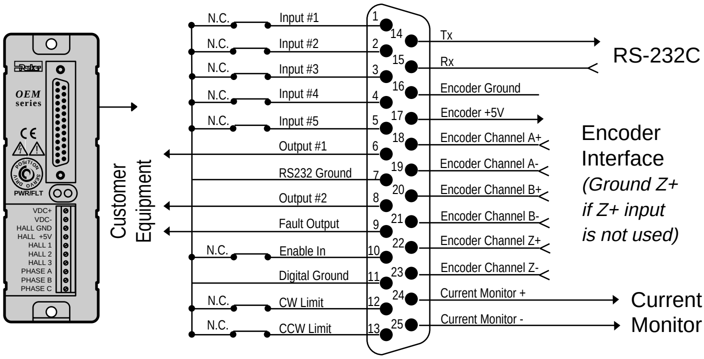
MOTOR GROUNDING

For safety reasons, the motor should be grounded. Often, the motor can be grounded through the equipment to which it is mounted. This requires a good electrical connection between the motor's mounting flange and the equipment, and that the equipment be connected to ground. Check with the National Electrical Code (NEC) and your local electrical code to ensure you use proper grounding methods.

Proper grounding can also reduce electrical noise.

OEM770X Inputs and Outputs

Inputs and outputs are located on the 25 pin D-connector.



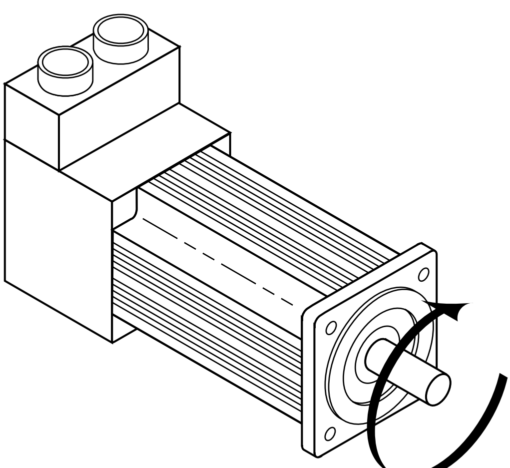
OEM770X Inputs & Outputs

CAUTION

I/O is not OPTO isolated, I/O GND (pins 7, 11, 16 & 24) are common to VDC- and HALL GND. For greater noise immunity, we recommend using optical isolation modules. For added noise immunity, the OEM770X has a digital filter; each input must be true for three successive cycles before it recognizes a given state.

CLOCKWISE AND COUNTERCLOCKWISE – DEFINITIONS

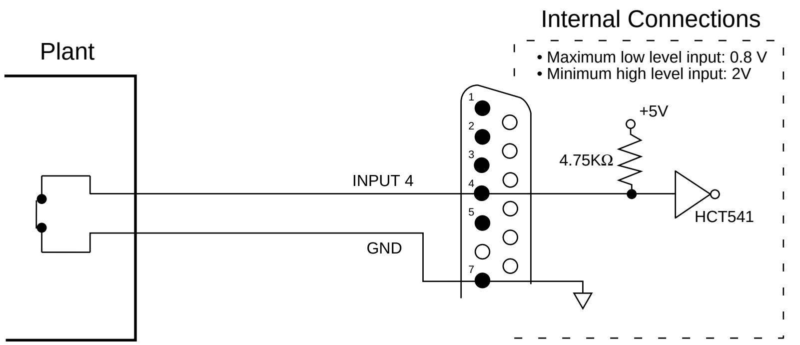
Shaft rotation is defined as the direction the shaft rotates, as viewed from the mounting flange end of the motor.



Clockwise Shaft Rotation

General Purpose Inputs (Signal 1-Signal 5)

The OEM770X has 5 general purpose inputs. Each of these inputs may be configured to match the application needs. The figure represents a typical configuration of one of these inputs.



Internal Connections for Pins 1-5 are identical

General Purpose Input Connected to a Switch

The **IN** command is used to configure the inputs to the following functions:

Trigger Input

The OEM770X can dedicate up to five Trigger inputs. These inputs are pulled up internally. These inputs are used with the Trigger (**TR**) command to control the OEM770X's trigger function. Minimum pulse width is 1 ms.

Home Position Input

The OEM770X can dedicate up to one Home input. The Home input allows you to establish a home reference position. This input is not active during power-up. Refer to the Go Home (**GH**) command for more information on setting up and using this function. Minimum pulse width is 1 ms.

Sequence Select Input

The OEM770X can dedicate up to three Sequence Select inputs that allow you to control seven different sequences. Sequences are executed remotely by using one of the following logic patterns in conjunction with the **XP** command.

Sequence #	**Ø**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
SEQ Input #1	Ø	1	Ø	1	Ø	1	Ø	1
SEQ Input #2	Ø	Ø	1	1	Ø	Ø	1	1
SEQ Input #3	Ø	Ø	Ø	Ø	1	1	1	1

Ø = low, pulled to ground
1 = high, 5VDC

Stop or Kill Input
The OEM770X can dedicate up to one Stop and one Kill input. The Stop or Kill input is identical in function to the effect of the **S** or **K** command respectively.

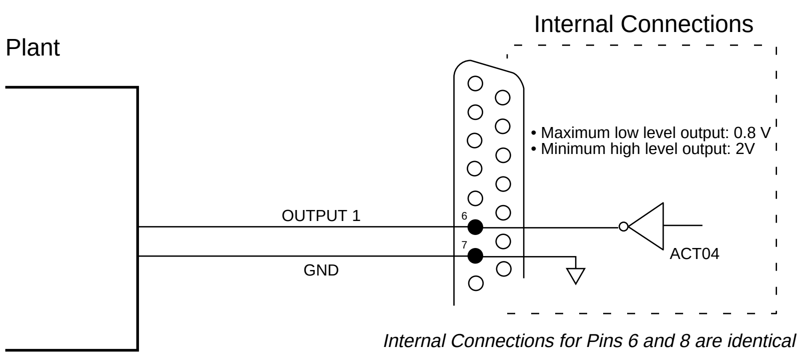
Go Input
The OEM770X can dedicate up to one Go input. The active state is high. The Go input is identical in function to the effect of the GO (**G**) command.

CAUTION
Unless configured otherwise (**SSH** command), the controller will dump the commands following the IN command in the buffer. Please pay special attention to the state of the inputs before entering the IN command.

OUTPUT #1 (SIGNAL 6) AND OUTPUT #2 (SIGNAL 8)

The OEM770X has two dedicated programmable +5 volt outputs. They may be used to signal peripheral devices upon the start or completion of a move. The default state for Outputs #1 and #2 is logic low. Refer to the Output (**O**) command for information on using these outputs.

The next drawing shows the schematic for one of the outputs.

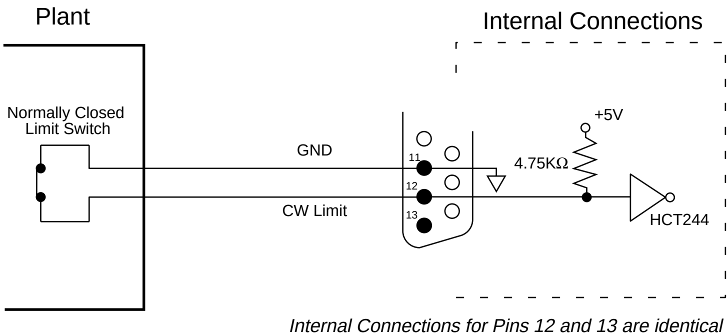


General Purpose Outputs

CW (SIGNAL 12) & CCW (SIGNAL 13) LIMIT INPUTS

The OEM770X has two dedicated hardware end-of-travel limits (CCW and CW). When you power up the OEM770X, these inputs are enabled (high). To test the OEM770X without connecting the CCW and CW limits, you must disable the limits with the **LD3** command. You can use the Limit Switch

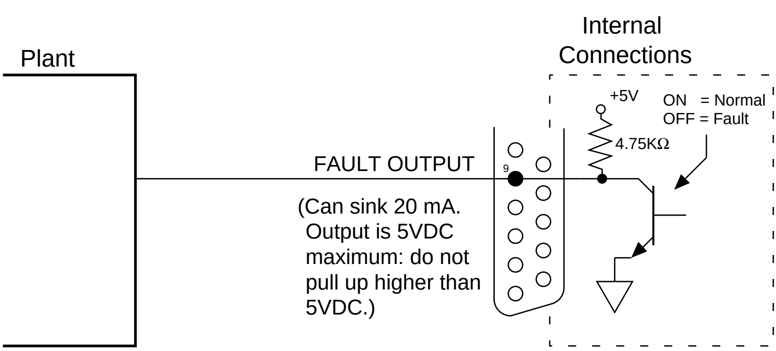
Status Report (**RA**) and Input Status (**IS**) commands to monitor the limits' status. The figure represents a typical configuration of these inputs. Minimum pulse width is 1 ms.



Limit Switch Inputs

Dedicated Fault Output (Signal 9)

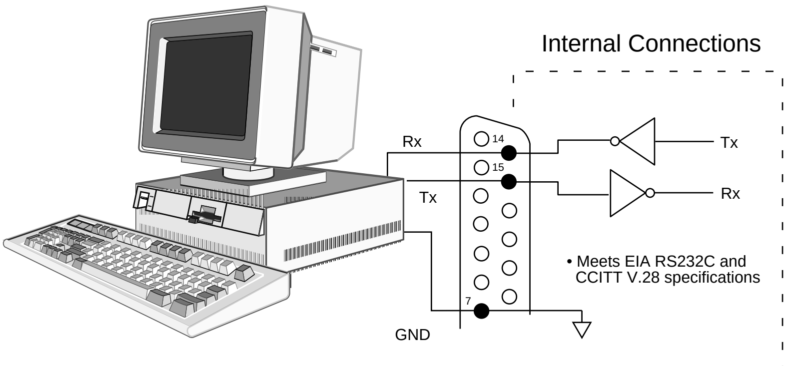
The OEM770X has one dedicated fault output**.** This output may be used to signal peripheral devices if a drive fault occurs. The Fault output's default state is logic high. When the front panel LED signals a drive fault, this signal will be active high. The **RSE** command will report the controller's error conditions. The next figure represents a typical configuration of this output.



Fault Output

RS-232C—TX (SIGNAL 14), RX (SIGNAL 15), AND GROUND (SIGNAL 7)

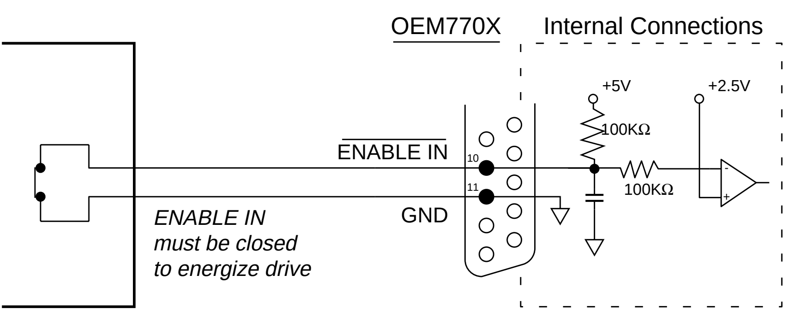
The OEM770X uses RS-232C as its communication medium. The OEM770X does not support handshaking. A typical three-wire (Rx, Tx, and Signal Ground) configuration is used. The figure represents a typical RS-232C configuration.



RS232 Input

ENABLE INPUT (SIGNAL 10)

The OEM770X has an enable input that must be connected to ground for the drive to be enabled. As the next drawing shows, you can connect the enable input to a switch for a hardware method of disabling the drive. When the switch is opened, the drive will be disabled.



Enable Input Connected to a Switch

If you need an emergency stop, do not use the hardware switch alone. There will be no torque on the motor when the drive is disabled. If the motor is moving, it will freewheel until it slows to a stop. Loads in vertical applications may drop due to gravity. Use a mechanical brake, in conjunction with the hardware disable switch, as an emergency stop.

WARNING

Disabling the drive with a switch is not an emergency stopping method. There is no torque on the motor when the drive is disabled. Therefore, you must also use a mechanical brake or some other means to stop the motor.

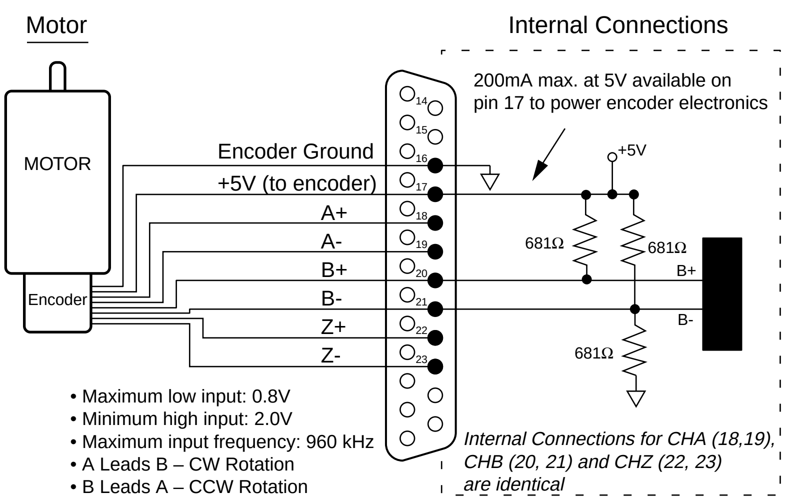
If you do not need to disable the drive by a hardware switch, connect a jumper wire from pin 10 to ground. The drive will then be enabled when it powers up. The OEM770X can also be disabled by software commands. For more information, see the shutdown command, **ST**, and the **OFF** command in the software reference section.

ENCODER INPUTS +5V, A, B, Z, GND (SIGNALS 16 - 23)

The OEM770X has six dedicated inputs for use with a differential incremental encoder. These inputs provide the position information for the servo loop.

CAUTION

If you do not connect an encoder Z-channel output to the OEM770X, then you must ground the Z+ input on the OEM770X. To do this, connect a jumper wire between the Z+ input (pin 22), and the nearest available ground (pin 16 or 11).

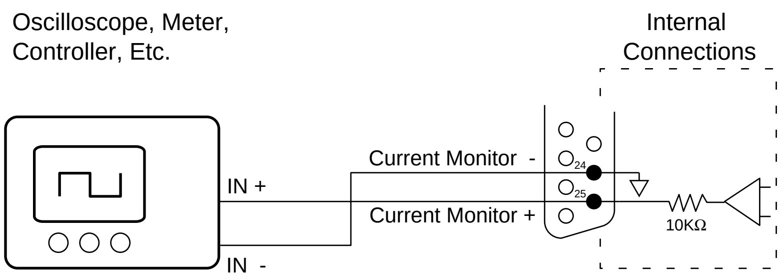


Encoder Input

CURRENT MONITOR

You can use the OEM770X's current monitor output to measure motor current. Connect pin 25 to the positive input of your oscilloscope, meter, etc. Use pin 24 as a signal ground for your oscilloscope or meter.

The OEM770X monitors the actual motor current. It puts out a voltage on pin 25 that is proportional to current, with 1 volt out = 1.2 amps of motor current. Positive voltages correspond to clockwise rotation (as viewed from the mounting flange end of the motor). Negative voltages correspond to counter-clockwise rotation.

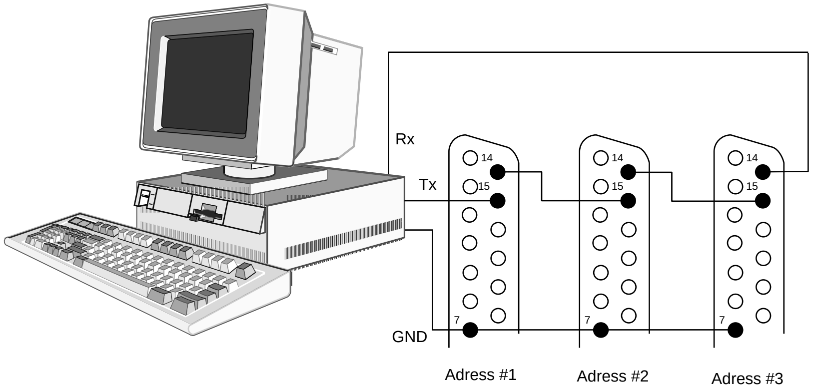


Current Monitor Output Connections

DAISY CHAINING

You may daisy chain up to 255 OEM770Xs. Individual drive addresses are set with the **#** (Address Numbering) command. When daisy chained, the units may be addressed individually or simultaneously. You should establish a unique device address for each OEM770X.

Refer to the figure below for OEM770X daisy chain wiring.



Daisy Chain of 3 OEM770Xs

Commands prefixed with a device address control only the unit specified. Commands without a device address control 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 1 to go.

When you use a single communications port to control more than one OEM770X, 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 OEM770X 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.

Connecting a Power Supply

The OEM770X requires a single external power supply with these features:

- 24VDC to 75VDC
- Fast Transient Response (can quickly supply enough current to meet your application's requirements)
- Power Dump (not required for all applications)

The power dump may be required if your system produces excess regenerated energy. To avoid damage, dissipate the regenerated energy in a power resistor, store it in extra capacitance (a blocking diode may be needed), or provide some other means to absorb regenerated energy.

For information about power supply selection, regeneration, and power dump methods, see *Chapter 6 Power Supply Selection*. The following table briefly lists the type of power supply you can use for different applications.

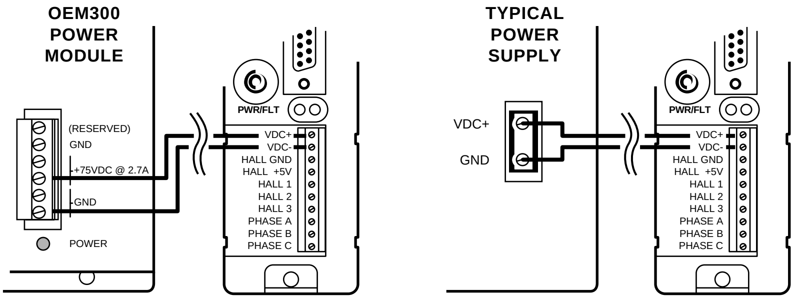
Application	**Recommended Power Supply**
Very Low Power (low regen)	24-48VDC Switching Power Supply 24-48VDC Linear Unregulated Supply OEM300 Power Module
Low Power (with regen)	Switching Power Supply with blocking diode and extra capacitance. Linear Unregulated Supply OEM300 Power Module
High Power (low regen)	Linear Unregulated Supply with Transformer OEM1000 Power Supply
High Power (with regen)	Linear Unregulated Supply with added Capacitance or added Power Dump OEM1000 Power Supply

The Compumotor OEM300 Power Module is a single unit that contains a 75VDC/300W power supply, integral power dump, and several protective circuits.

The Compumotor OEM1000 Power Supply is a linear power supply that can provide 1000W/15A at 70VDC.

CONNECTING THE POWER SUPPLY

Connect your power supply to the 10 pin screw terminal on the OEM770X. The next drawing shows connections for a typical power supply, and for an OEM300 Power Module.



Power Supply Connections

- Connect the positive DC terminal of your power supply to the VDC+ input on the OEM770X's 10-pin screw terminal.
- Connect the ground terminal of your power supply to VDC- on the OEM770X.

To reduce electrical noise, minimize the length of the power supply wires and twist them tightly together.

Grounding

Internally, VDC- is connected to the Hall Ground and the grounds on the 25 pin D-connector (pins 7, 11, 16, 24). Do not connect your power supply's ground to these pins, however. Connect it only to VDC-.

The shell of the 25 pin D-connector and the heatplate are connected internally. They are not connected to VDC-, Hall Ground, or the D-connector grounds (pins 7, 11, 16, 24).

Wire size

Use 18 AWG (0.75 mm^2) or greater diameter wire for power connections. For applications that use high peak power, use larger diameter wires. 14 AWG (2.5 mm^2) wire is the biggest wire that will fit in the 10-pin screw terminal.

Tuning

The OEM770X uses a digital *Proportional Integral Derivative* (PID) filter to compensate the control loop. For best performance, you must tune the filter's parameters. A properly tuned system will exhibit smooth motor rotation, accurate tracking, and fast settling time.

All tuning is performed via RS232-C communications.

Saving the Settings

Tuning parameters are saved in memory on the OEM770X. The –M2 option (battery backed RAM) allows gains to be saved when the drive is turned off:

–M2 Option Installed	Gains are saved in nonvolatile memory
–M2 Option **NOT** *Installed*	Gains are **not saved** when drive is off

If you do not have the –M2 option, you must reload the gains each time power is cycled to the OEM770X.

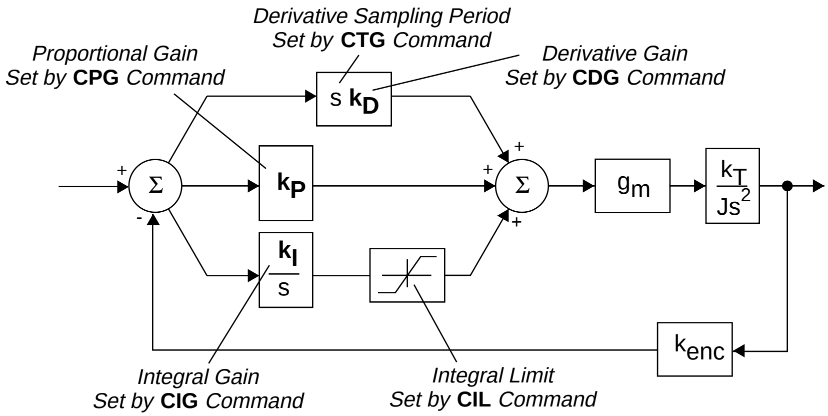
PID TUNING

In the procedure described below, you will systematically vary the tuning parameters until you achieve a move that meets your requirements for accuracy and response time.

The OEM770X generates a move profile based upon the user supplied acceleration, velocity, and distance commands (**A**, **V**, and **D**). At each servo sampling period (every 266 microseconds), the OEM770X calculates the position the motor should reach as it follows the move profile. This is called *commanded position,* and is one of two inputs to a summing node. Position information from the encoder, which is called *actual position,* is the other input to the summing node. During a typical move, actual position will differ from commanded position by at least a few encoder counts. When actual position is subtracted from commanded position at the summing node, an error signal is produced. The error signal is the input to the PID filter.

The position specified by the distance command (**D**) is called the *target position*. During the move, commanded position is not the same as target position. The commanded position is incremented each sampling period. When it finally matches the target position, the move is over.

The servo block diagram is shown on the following page.



As the figure shows, you can adjust five different parameters to tune the PID filter. The relevant commands are:

CPG	Configure Proportional Gain
CDG	Configure Derivative Gain
CTG	Configure Derivative Sampling Period
CIG	Configure Integral Gain
CIL	Configure Integral Limit

To tune the system, you will iteratively increase **CDG** and **CTG** to their optimum values, then increase **CPG** to its optimum value. If necessary, you will also increase **CIG** and **CIL**.

In general, you will set **CDG** and **CPG** as high as possible, and **CIG** as low as possible. Trade-offs between response time, stability, and final position error will dictate the values you select. For loads that vary during operations, you can download new parameters by using buffered versions of the five tuning commands (**BCPG, BCDG, BCTG, BCIG, BCIL**).

WARNING

During servo tuning, the system can undergo accidental and violent movement due to improper gain settings and programming errors. Please use extreme caution while prototyping.

Each tuning parameter is described in the following sections.

CPG – *Proportional Gain*

Proportional gain provides a torque that is directly proportional to the *magnitude* of the error signal. Proportional gain is similar to a spring—the larger the error, the larger the restoring force. It determines the stiffness of the system and affects the following error. High proportional gain gives a stiff, responsive system, but can result in overshoot and oscillation. Damping—provided by derivative gain—can reduce this overshoot and oscillation.

CDG – Derivative Gain; CTG – Derivative Sampling Period

Derivative gain provides a torque that is directly proportional to the *rate of change* of the error signal. The previous error is subtracted from the present error each sampling period. The difference represents the error's instantaneous rate of change, or *derivative*. The difference is multiplied by the value set by the **CDG** command, and the product contributes to the motor control output.

Derivative gain opposes rapid changes in velocity. It will dampen the resonance effects of proportional gain. With higher derivative gain, you can use higher proportional gain.

You can use the **CTG** command to make the derivative sampling period longer than the system's sampling period. The system sampling period—266 µsec—is the period between updates of position error, and cannot be changed. The derivative sampling period is an integer multiple of the system sampling period. It can range from 266 µsec to 68 msec, in increments of 266 µsec (for example: CTG0 = 266 µs, CTG1 = 532 µs, CTG2 = 798 µs, etc.).

With a longer derivative sampling period, more time elapses between error measurements. The difference between previous and present error is still multiplied by the CDG value. The product contributes to the motor control output every *system* sampling period, but is only updated every *derivative* sampling period. This gives a more constant derivative term and improves stability. Low velocity systems in particular can benefit from a longer sampling period.

CIG – Integral Gain; CIL – Integral Limit

Integral gain provides a torque that is directly proportional to the sum, over time, of the error values—the *integral* of the error. The controller reads the error value every sampling period, and adds it to the sum of all previous error values. The sum is multiplied by the value set by the **CIG** command, and the product contributes to the motor control output every system sampling period.

Integral gain can remove steady state errors that are due to gravity or a constant static torque. Integral gain can also correct velocity lag that can occur in a constant velocity system.

If error persists during a move, the sum of the error values may be quite high at the end of the move. In this case, the torque provided by the integral gain can also be very high, and can cause an overshoot. This effect is called *integral windup*. You can use **CIL**, the integral limit command, to set a maximum value for integral gain. The integral limit constrains the integral term to values less than or equal to **CIL**, which will reduce the overshoot caused by integral windup.

TUNING PROCEDURE

You can manually tune the OEM770X by varying the tuning parameters while you empirically evaluate the system response. This manual method works well in most applications. You can also connect an oscilloscope to the OEM770X's current monitor output, and observe the motor current waveform. This will give you information about the system's step response. You can then adjust the parameters until you obtain the step response you desire. Both of these methods are explained below.

Tuning Procedure – Manual Method

You will achieve best results by making a consistent, repetitive move that is representative of your application.

1. ***Issue a RETURN TO FACTORY SETTINGS command (RFS)***
   The RFS command will reset the gains to their default values (CDG240, CTG0, CPG16, CIG2, CIL2, CPE4000)

2. ***Decrease CPG***
   Decrease **CPG** to zero (**CPG0**). If your system has very little friction, internal offsets in the drive may cause the motor to run away (spin faster and faster) with **CPG** set to zero. If the motor runs away, issue an **OFF** command. When the motor stops, increase **CPG** by one unit (**CPG1**), and issue an **ON** command. If the motor continues to run away, repeat this procedure—incrementing **CPG** by one unit—until the motor remains stopped.

3. ***Decrease CIG***
   Set the integral gain to zero (**CIG0**).

4. ***Increase Derivative Gain (CDG) and Derivative Sampling Period (CTG)***
   Determine **CDG** and **CTG** iteratively. Increase **CDG** until the shaft begins high frequency oscillations, then increase **CTG** by one. With a higher **CTG**, the oscillations should be damped. Again increase **CDG** until oscillations occur, then increase **CTG** by one. Repeat this process until **CTG** reaches a value appropriate for the system.

   In general, you will want values for **CDG** and **CTG** that are as large as possible, without producing unacceptably high motor vibrations. However, many systems will require a low **CTG** value, to ensure that the derivative sampling period is shorter than one tenth of the system mechanical time constant. Therefore, start with a low **CTG** and gradually increase it, rather than immediately trying a large **CTG** value.

5. ***Increase Proportional Gain (CPG)***
   Determine the **CPG** value iteratively. Increase **CPG**, and evaluate the system damping. Repeat until the system is critically damped. You should increase **CPG** to the largest value that does not cause overshoot or ringing. Because proportional gain and derivative gain affect each other, you may need to repeat step 4 and step 5 several times to arrive at optimum values for **CPG** and **CDG**.

6. ***Determine Integral Gain (CIG) and Integral Limit (CIL) Values***
   High values for **CIG** will make the system respond quickly, but can cause other problems. In general, you should set **CIG** to the *lowest* value that will correct following errors and static position errors, but not increase overshoot or settling time. In a system without static torque loading, a **CIG** of zero may be appropriate.

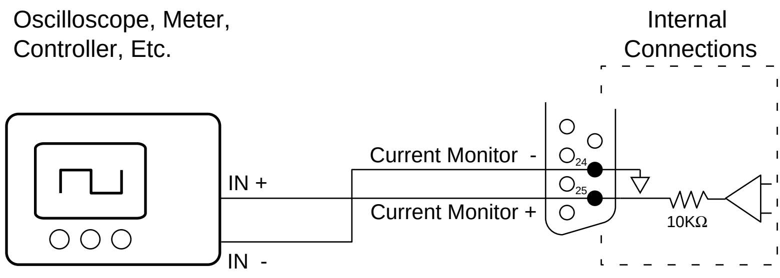
   CIL limits **CIG**—therefore, before you increase **CIG** to a particular value, you must first increase **CIL** to an equal or higher value.

Tuning Procedure – Step Response Method

You can use the OEM770X’s current monitor output to view a signal that is a scaled replica of motor current. Use this method if you were unable to obtain satisfactory results from the manual tuning method described above, or if you need more performance information to help you tune your system.

The step response for a motion control system is a graph of actual position versus a small and instantaneous change in commanded position. An oscilloscope, connected as described below, does not show position—it shows motor current. However, in most applications motor current is directly related to position. Viewing the current waveform, therefore, can give you insight into the system’s step response.

Connect an oscilloscope to the current monitor, as shown below.

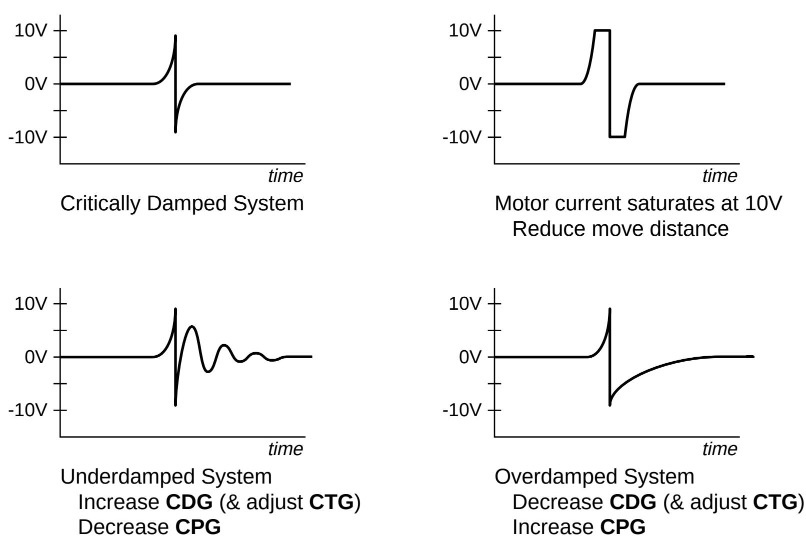


Current Monitor Output Connections

The voltage on pin 25 can range from -10VDC to +10VDC. It is scaled to be proportional to motor current, with 1 volt out = 1.2 amps of motor current. Positive voltages correspond to clockwise rotation of the motor shaft.

Command the longest move that does not cause the motor current to saturate (keep the current monitor output below 10V). Typical move distances are 250 to 500 encoder counts. Use a high velocity and high acceleration.

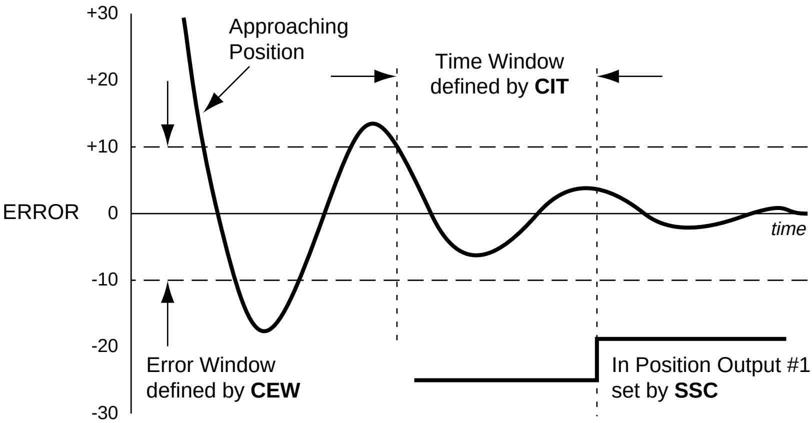
The next drawing shows typical response waveforms, along with suggestions on how to adjust the tuning parameters to achieve a critically damped system.



Step Response Waveforms

CONFIGURING AN IN POSITION WINDOW

You can define an In Position Window, and use it to indicate that the preceding move is done. Two commands—**CEW** and **CIT**—determine the height and width of the window. A third command—**SSC**—can turn on output #1 when the In Positon criteria are met.



As the drawing shows, **CEW** defines the position error window at the end of a move. **CIT** specifies the length of time the motor must be within the error window. The motor is In Position when three conditions are satisfied:

1. The controller algorithm is finished (no input position command)
2. Position error is less than that specified by the **CEW** command
3. Condition *2* above has been true for the time period specified by **CIT**

If **SSC** has been set to 1, output #1 will turn on when these three conditions have been met. You can use output #1 to trigger external hardware from the In Position condition. The output will stay on until the next move command is issued, such as **GO** or **GO HOME**.

(*Note:* If the motor is held (mechanically, or against an end stop), and **CPE** is greater than **CEW**, the motor may become “trapped” between **CPE** and **CEW**: it will not execute the next move. In this rare situation, two things are happening:

1. **CPE** is not violated, and therefore no position error fault occurs;
2. in position criteria are not met.

If you were to execute a **1R**, the response would be ***B**, which means the drive is “busy” waiting for the move to be over. Why doesn’t the drive force the motor to finish the move? The motor is somehow held. To correct this situation, try touching the motor; this may complete the move, and the drive may execute the next move. Or, execute a **DPA** to read actual position, and verify that the move is not complete. You can also execute a **KILL** to reset the positions, and then do the next series of moves.)