ICONT

Application Continuous Current:
This variable sets the system continuous current. This variable is used in the foldback algorithm (see FOLD and FOLDMODE). The default value of this variable is the minimum of DICONT (Drive Continuous Current) and MICONT (Motor Continuous Current), unless that value exceeds IMAX, in which case ICONT will be set equal to IMAX. This variable will be reset to its default whenever DICONT or MICONT is changed. The user can override the default.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... % of DIPEAK * 0.1
Range: .... 0 to IMAX
Default: .... Minimum of DICONT and MICONT
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

IENCSTART

This variable sets the B-C phase current for the ENCSTART encoder initialization process.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... % of MICONT
Range: .... 1 to 100
Default: 25
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

IFRIC

This variable is the Coulomb Friction constant for the current loop.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... 0.1% of DIPEAK
Range: .... 0 to 500
Default: 0
EEPROM: .... yes
OPMODES: ...... All
Drive Status: ..... en/dis

IGRAV

This variable is the Gravity constant for the current loop.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... 0.1% of DIPEAK
Range: .... -500 to 500
Default: 0
EEPROM: .... yes
OPMODES: ...... All
Drive Status: ..... en/dis

ILIM

This variable sets the application current limit, allowing the user to limit the drive's peak current. This variable limits the current command that will be accepted from the user (using the T command in Opmode 2) or issued by the control loops (in Opmodes 0, 1, 3, and 4). This variable is an independent variable that is not calculated from hardware parameters and is not tied to any other variables. ILIM is similar to VLIM (which is used in Opmodes 0 and 1) and can be used to protect delicate load equipment.

Firmware Versions: all
Type: .... variable (R/W)
Units: ....% of DIPEAK * 0.1
Range: ....0 to IMAX
Default: .... IMAX
EEPROM: .... yes
OPMODES: .... all
Drive Status: .....en/dis

ILSBMODE

sets the mode of operation of the inter-LSB algorithm, which interpolates feedback between least significant bits (LSB's) of the feedback device. Enabling this algorithm will improve performance when the RDRES resolution is low (12 bits), BW is high, and the commanded velocity is low.

ILSBMODE = 0; algorithm disabled
ILSBMODE = 1; algorithm enabled for velocity feedback.
ILSBMODE = 2; reserved for future enhancements

Firmware Versions: 2.1.0 and later
Type: .... switch mode (R/W)
Units: ....n/a
Range: 0, 1, 2
Default: 0
EEPROM: ....yes
OPMODES: 0, 1, 4
Drive Status: ......en

IMAX

Maximum System Current:
This variable displays the system current maximum for a drive and motor combination. This variable is actually the minimum of the Drive Peak Current (DIPEAK) and the Motor Peak Current (MIPEAK).

Firmware Versions: All
Type: ....Standard Variable (Read Only)
Units: ....% of DIPEAK * 0.1
Range: ....0 to 1000
Default: ....Minimum of DIPEAK and MIPEAK
EEPROM: ....No
OPMODES: ....All
Drive Status: ......Enabled / Disabled

IN1

This variable is a redundant name for CWLIM and is used to read the state of the hardware input on user connector C3 Pin 9. Note that IN1 and CWLIM have the opposite polarity from each other.

Firmware Versions: V2.1.0 and later
Type: .... switch (R)
Units: .... n/a
Range: .... 0 (off), 1 (on)
Default: .... n/a
EEPROM: .... n/a
Opmodes: .... all
Drive Status: ...... en/dis

IN1MODE

This variable sets the functionality of the IN1 input. The function list is:
IN1MODE=0: No function.
IN1MODE=1: CW limit switch
IN1MODE=2: CCW limit switch
IN1MODE=3: Gear disable input
IN1MODE=4: Gear mask input
IN1MODE=5: Gear A input
IN1MODE=6: Gear B input
IN1MODE=7-16: Reserved
IN1MODE=17: Trigger active disable
IN1MODE=18: Reserved
IN1MODE=19: Hold position

Firmware Versions: 2.1.0 and later
Type: .... switch mode (R/W)
Units: .... n/a
Range: .... 0 to 19
Default: 1
EEPROM: .... yes
OPMODES: ...... dependent
Drive Status: ..... en/dis

IN2

This variable is a redundant name for CCWLIM and is used to read the state of the hardware input on user connector C3 Pin 10. Note that IN2 and CCWLIM have the opposite polarity from each other.

Firmware Versions: V2.1.0 and later
Type: .... switch (R)
Units: .... n/a
Range: .... 0 (off), 1 (on)
Default: .... n/a
EEPROM: .... n/a
Opmodes: .... all
Drive Status: ...... en/dis

IN2MODE

This variable sets the functionality of the IN2 input. The function list is:
IN2MODE=0: No function.
IN2MODE=1: CW limit switch
IN2MODE=2: CCW limit switch
IN2MODE=3: Gear disable input
IN2MODE=4: Gear mask input
IN2MODE=5: Gear A input
IN2MODE=6: Gear B input
IN2MODE=7-16: Reserved
IN2MODE=17: Trigger active disable
IN2MODE=18: Reserved
IN2MODE=19: Hold position

Firmware Versions: 2.1.0 and later
Type: .... switch mode (R/W)
Units: .... n/a
Range: .... 0 to 19
Default: 2
EEPROM: .... yes
OPMODES: .... dependent
Drive Status: ...... en/dis

IN3

This variable is used to read the state of the hardware input on user connector C3 Pin 11.

Firmware Versions: V2.1.0 and later
Type: .... switch (R)
Units: .... n/a
Range: .... 0 (off), 1 (on
Default: .... n/a
EEPROM: .... n/a
Opmodes: .... all
Drive Status........ en/dis

IN3MODE

This variable sets the functionality of the IN3 input. The function list is:
IN3MODE=0: No function.
IN3MODE=1: CW limit switch
IN3MODE=2: CCW limit switch
IN3MODE=3: Gear disable input
IN3MODE=4: Gear mask input
IN3MODE=5: Gear A input
IN3MODE=6: Gear B input
IN3MODE=7-16: Reserved
IN3MODE=17: Trigger active disable
IN3MODE=18: Reserved
IN3MODE=19: Hold position

Firmware Versions: 2.1.0 and later
Type: .... switch mode (R/W)
Units: .... n/a
Range: .... 0 to 19
Default: 3
EEPROM: .... yes
OPMODES: ...... dependent
Drive Status: ..... en/dis

INPOS

This variable indicates if the actual position is following the commanded position within the following error set by PEINPOS.

Firmware Versions: 2.1.0 and later
Type: .... switch (R)
Units: .... n/a
Range: .... 0 (not in posn), 1 (in posn)
Default: .... n/a
EEPROM: .... n/a
OPMODES: 4
Drive Status: .... en/dis

ISCALE

Torque Input Scale Factor:
This variable scales the analog input for OPMODE 3 (analog torque mode). The value entered is the desired motor current if 10 volts were applied to the analog input. This variable may be either higher or lower than 100%, but the current loop command is limited by the application current limit (ILIM).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... (% of DIPEAK* 0.1) / 10 Volts
Range: .... 100 to 10,000
Default: 1250
EEPROM: .... Yes
OPMODES: 3
Drive Status: ..... Enabled / Disabled

ISCALE

scales the analog input for OPMODE 3 (analog torque mode). The value entered is the desired motor current if 10 volts were applied to the analog input. This variable may be either higher or lower than 100%, but the current loop command is limited by the application current limit (ILIM).

Firmware Versions: all
Type: .... variable (R/W)
Units: .... (% of DIPEAK*0.1) /10V
Range: .... 100 to 10,000
Default: 1250
EEPROM: .... yes
OPMODES: 3
Drive Status: ...... en/dis

ISTOP

This variable sets the current command for the braking function. This variable is subject to the UNITS command. See also STOPMODE.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units:.................. % of DIPEAK *0.1
Range: .... 0 to IMAX
Default:.............. IMAX
EEPROM: .... yes
Opmodes: .... all
Drive Status: ...... en/dis

IZERO

Applied Phase Current in ZERO Mode:
This variable sets the C-B phase current for ZERO Mode (A=0).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... % of MICONT
Range: .... 1 to 100
Default: 25
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

KV

Proportional Gain, Velocity Loop, PDFF:
This variable is a tuning variable which sets the proportional gain for the Pseudo Derivative Feedback with Feed-Forward Velocity Control Loop (PDFF loop; COMPMODE = 1). This variable is set manually by the user. Executing the TUNE command successfully may change the value of this parameter.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... n/a
Range: .... 0 to 32767
Default: 1000
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

KVFR

Feed-Forward to Feedback Gain Ratio, Velocity Loop, PDFF:
This variable is a tuning variable which sets the feed-forward to feedback gain ratio for the Pseudo Derivative Feedback with Feed-Forward Velocity Control Loop (PDFF loop; COMPMODE = 1). This variable is set manually by the user. Executing the TUNE command successfully may change the value of this parameter.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Percent * 0.1
Range: .... 0 to 1000
Default: 0
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

KVI

Integral Gain, Velocity Loop, PDFF:
This variable is a tuning variable which sets the integral gain for the Pseudo Derivative Feedback with Feed-Forward Velocity Control Loop (PDFF loop; COMPMODE = 1). This variable is set manually by the user. Executing the TUNE command successfully may change the value of this parameter.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... n/a
Range: .... 0 to 32767
Default: 1000
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

LIMDIS

Limit Switch Disable:
This switch variable enables/disables the End Travel Limit function. This function only pertains to units with the limit switch option.
0 = limit switch function enabled
1 = limit switch function disabled; LED decimal point flashes

Firmware Versions: All
Type: .... Switch Variable (Read / Write)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

LMJR

This variable sets the ratio of the estimated Load Moment of Inertia (LMJ) relative to the Motor Moment of Inertia (MJ). The variables LMJR and MJ and the required closed loop bandwidth (BW) are used for the Velocity Control Loop design in the Standard Pole-Placement controller (COMPMODE = 2 or 4). Executing the TUNE command successfully may change the value of this parameter.

Firmware Versions: all
Type:.................. variable (R/W)
Units:................. percent of MJ
Range:................ 0 to 10,000
Default: 0
EEPROM: .... yes
OPMODES: ......all
Drive Status: ...... en/dis

LPFHZ1

This variable sets the cutoff frequency of the first Low Pass Filter (LPF) used in the velocity loop. This variable only affects the system when FILTMODE = 1 or 2.

Firmware Versions: all
Type: ....variable (R/W)
Units: ....Hz
Range: ....20 to 800, steps of 20(20, 40, ..., 800)
Default: 500
EEPROM: ....yes
OPMODES: 0,1
Drive Status: ......en/dis

LPFHZ2

This variable sets the cutoff frequency of the second Low Pass Filter (LPF) used in the velocity loop. This variable only affects the system when FILTMODE = 2.

Firmware Versions: all
Type: ....variable (R/W)
Units: ....Hz
Range: ....20 to 800, steps of 20 (20, 40, ..., 800)
Default: 500
EEPROM: .... yes
OPMODES: 0,1
Drive Status: ......en/dis

MBEMF

This variable displays the motor's back EMF constant. This value is used for current loop controller design. This variable requires a CONFIG command when changed.

Firmware Versions: all
Type: ....variable (R/W)
Units: .... rotary: (Volts RMS) / kRPM
............................linear: (Volts Peak) / (m/sec)
Range: ....1 to 3900
Default:............... motor data
EEPROM: ....yes
OPMODES: .... all
Drive Status: ...... dis

MBEMFCOMP

Sets a back EMF compensation percentage value. This variable affects the amount of back EMF compensation that is applied to the motor command. This variable requires a CONFIG command when changed. Note that for firmware version, 2.0.0, this variable was called BEMFCOMP.

Firmware Versions: V2.0.1 and later
Type: .... variable (R/W)
Units:.................. percent
Range: .... 1 to 100
Default: .... motor data / 50
EEPROM: .... yes
OPMODES: ..... all
Drive Status: ..... dis

MENCOFF

Motor Encoder Feedback Option:
This variable sets the encoder index position (encoder feedback systems only). This variable is expressed in units of encoder counts after quadrature, and the range is from 0 to (4 * encoder resolution - 1) or (4 * MENCRES - 1). This variable can be set automatically using ENCINIT.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Encoder Counts / Motor Revolution
Range: .... 0 to (4 * MENCRES - 1)
Default: .... Motor Data (120 Electrical Degrees if undefined)
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

MENCRES

This variable displays the resolution of the motor encoder (encoder feedback systems only) in number of lines per revolution of the motor. Note that the number of encoder counts per revolution is obtained by multiplying MENCRES by 4. This variable requires a CONFIG command when changed.

Firmware Versions: all
Type: .... variable (R/W)
Units: .... rotary: lines/motor revolution
.... linear: lines/mm
Range: .... 100 to 10,000,000
Default: .... motor data
EEPROM: .... yes
OPMODES: ...... all
Drive Status: ..... dis

MENCTYPE

This variable sets the motor encoder type. When this variable is changed on an encoder-based system, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG). In version 1 firmware prior to 1.2.0, MENCTYPE = 0 is assumed. This variable may take value from 0 to 6.

ENCODER CHARACTERISTICS				
MENCTYPE	A/B Quad	Marker Pulse	Hall Effects	Firmware Versions
0	√	√	√	all
1	√	√		all
2	√	√		all
3	√			all
4	√			all
5			√	reserved
6	√		√	2.1.0 and later

INITIALIZATION METHODS	
MENCTYPE	Method of Initialization
0	Initialization is optional and may be performed using the ENCINIT command.
1	Initialization is required and is triggered by the ENCSTART command. This may optionally be followed by marker pulse location using the ENCINIT command.
2	Initialization is required and is triggered on power up (when the drive is enabled) or by using ENCSTART. This may optionally be followed by marker pulse location using the ENCINIT command.
3	Initialization is required and is triggered by the ENCSTART command.
4	Initialization is required and is triggered on power up (when the drive is enabled) or by using the ENCSTART command.
5	Reserved for later introduction
6	Initialization is required and is triggered by the ENCSTART command.

MENCTYPE For MENCTYPE 0-2, ENCINIT should be performed where MENCOFF is unknown.

Firmware Versions: all
Type: .... switch mode (R/W)
Units: .... n/a
Range: .... 0 to 6 (5 is reserved for future)
Default: .... motor data (0 if undefined)
EEPROM: .... yes
Opmodes: .... all
Drive Status: ...... dis

MHINVA

This is a variable which applies to encoder-based systems which use hall switches to commutate. This variable inverts the hall sensor A feedback, causing the system to read the 'A' hall channel as inverted data.

MHINVA = 0: do not invert hall A
MHINVA = 1: invert hall A

Firmware Versions: All
Type: .... switch (R/W)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... yes
OPMODES: ...... all
Drive Status: ..... en/dis

MHINVB

is a variable which applies to encoder-based systems which use hall switches to commutate. This variable inverts the hall sensor B feedback, causing the system to read the 'B' hall channel as inverted data.

MHINVB = 0: do not invert hall B
MHINVB = 1: invert hall B

Firmware Versions: All
Type: .... switch (R/W)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... yes
OPMODES: ...... all
Drive Status: ..... en/dis

MHINVC

This is a variable which applies to encoder-based systems which use hall switches to commutate. This variable inverts the hall sensor C feedback, causing the system to read the 'C' hall channel as inverted data.

MHINVC = 0: do not invert hall C
MHINVC = 1: invert nall C

Firmware Versions: All
Type: .... switch (R/W)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... yes
OPMODES: ...... all
Drive Status: ..... en/dis

MICONT

Motor's Continuous Rated Current:
This variable sets the motor's continuous rated current. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Amperes RMS*0.1
Range: .... 10 to 1750
Default: .... Motor Data
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Disabled

MIPEAK

Motor's Peak Rated Current:
This variable sets the motor's peak rated current. This variable is used for current limit algorithms. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Amperes RMS*0.1
Range: .... 10 to 3500
Default: .... Motor Data
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Disabled

MJ

This variable sets the motor's rotor inertia (rotary motors) or motor coil mass (linear motors, MOTORTYPE=2). The Motor rotor inertia (MJ) and the Load moment of inertia ratio (LMJR) define the total system moment of inertia. The variables LMJR and MJ and the required closed loop bandwidth (BW) are used for the Velocity Control Loop design in the Standard Pole-Placement controller (COMPMODE = 2 or 4).

Firmware Versions: all
Type: .... variable (R/W)
Units: .... rotary: Kg $*m^2 *10^{-6}$
.... linear: grams
Range: .... 1 to 2,000,000,000
Default: .... motor data
EEPROM: .... yes
OPMODES: .... all
Drive Status:....... dis

MLGAINC

This variable sets the current loop adaptive gain value at continuous motor current (MICONT). MLGAINC, MLGAINP, and MLGAINZ define the adaptive gain algorithm that is based on motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases. The current-based adaptive gain algorithm is set up by defining the gains at peak motor current (MLGAINP), at continuous motor current (MLGAINC), and at zero motor current (MLGAINZ). All other gains between zero, continuous, and peak current are interpolated. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: all
Type: .... variable (R/W)
Units: .... % *10
Range: .... 1 to 100
Default: 7
EEPROM: .... yes
OPMODES: ..... all
Drive Status: ...... dis

MLGAINP

This variable sets the current loop adaptive gain value at peak motor current (MIPEAK). MLGAINC, MLGAINP, and MLGAINZ define the adaptive gain algorithm that is based on motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases. The current-based adaptive gain algorithm is set up by defining the gains at peak motor current (MLGAINP), at continuous motor current (MLGAINC), and at zero motor current (MLGAINZ). All other gains between zero, continuous, and peak current are interpolated. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: all
Type: .... variable (R/W)
Units: .... % *10
Range: .... 1 to 100
Default: 4
EEPROM: .... yes
OPMODES: ..... all
Drive Status: .... dis

MLGAINZ

This variable sets the current loop adaptive gain value at zero motor current. MLGAINC, MLGAINP, and MLGAINZ define the adaptive gain algorithm that is based on motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases. The current-based adaptive gain algorithm is set up by defining the gains at peak motor current (MLGAINP), at continuous motor current (MLGAINC), and at zero motor current (MLGAINZ). All other gains between zero, continuous, and peak current are interpolated. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: all
Type: .... variable (R/W)
Units: .... % * 10
Range: .... 1 to 100
Default: 10
EEPROM: .... yes
OPMODES: ...... all
Drive Status:....... dis

MLMIN

Motor's Minimum Inductance, Line-to-Line :
This variable sets the motor's minimum line-to-line inductance. This variable is used for current loop controller design. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Millihenries * 10^{-2}
Range: .... 1 to 32767
Default: .... Motor Data
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Disabled

MOTOR

Motor Name:
This string variable is the name of the motor connected to the drive. The motor string variable MUST BE PRECEDED BY DOUBLE QUOTES (") when entered.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... n/a
Range: .... 10 Characters
Default: .... Motor Data
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

MOTORTYPE

This variable sets the Drive control algorithms to different motor types as follows:

MOTORTYPE=0: permanent magnet rotary motor
MOTORTYPE=1: reserved; do not use
MOTORTYPE=2: permanent magnet linear motor

Firmware Versions: 2.1.0 and later
Type: .... switch mode (R/W)
Units: .... n/a
Range: 0-2
Default: 0
EEPROM: .... yes
Opmodes: .... all
Drive Status: ...... dis

MPHASE

Motor Phase:
This variable defines the resolver/encoder phase relative to the "standard" commutation table. This variable can be used to compensate for resolver/encoder offset or alignment error and should be set to 0 if there is no resolver offset.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Electrical Degrees
Range: .... 0 to 359
Default: .... Motor Data
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Disabled

MPITCH

This is a variable for use with linear motors (MOTORTYPE = 2). It defines the pole-pitch ((length in millimeters of one electrical cycle - 360 electrical degrees) of the motor and allows the ServoStar to be able to calculate other variables such as velocity. The drive assumes a 'no-comp' state after an entry of this parameter and requires the CONFIG command.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... mm per 360 elec. Degrees
Range: .... 1 to 500
Default: 16
EEPROM: .... yes
OPMODES: ...... all
Drive Status: .... dis

MRESPOLES

This variable sets the number of individual poles in the feedback device. This variable is used for the commutation function, as well as for velocity feedback scaling, and represents the number of individual poles, not pole pairs. When this variable is changed on a resolver system, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

MRESPOLES

Motor Resolver Poles:
This switch mode variable sets the number of individual poles in the motor resolver (most resolvers have 2 poles). This variable is used for the commutation function, as well as for velocity feedback scaling, and represents the number of individual poles, not pole pairs. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: All
Type: .... Switch Mode Variable (Read / Write)
Units: .... Poles
Range: .... 2, 4, 6, 8, 12, 14, 16 (resolver based)
.... 0, 2, 4, 6, 8, 12, 14, 16 (encoder based)
Default: .... Motor Data
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Disabled

MSG

Message Enable:
This switch variable will enable/disable the sending of error messages from the drive to the serial port.
0 = disable messages
1 = enable messages

Note that MSG = 1 is needed for proper operation of MotionLink.

Firmware Versions: All
Type: .... Switch Variable (Read / Write)
Units: .... n/a
Range: 0, 1
Default: 1
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

MSPEED

This variable defines the maximum recommended velocity of the Motor. When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: all
Type: .... variable (R/W)
Units:.................. rotary: RPM
.... linear: mm/sec
Range: .... 10 to 32767
Default: .... motor data
EEPROM:........... yes
OPMODES:........ all
Drive Status:....... dis

MTANGLC

Torque-Related Commutation Angle Advance at Motor Rated Current (MICONT): This variable sets the value of the torque-related commutation angle advance at the motor's continuous current rating (MICONT). This variable helps increase reluctance torque (for use with Goldline motors).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Electrical Degrees
Range: .... 0 to 45
Default: 10
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

MTANGLP

Torque-Related Commutation Angle Advance at Motor Peak Current (MIPEAK): This variable sets the value of the torque-related commutation angle advance at the motor's peak current (MIPEAK). This variable helps increase reluctance torque (for use with Goldline motors).

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Electrical Degrees
Range: .... 0 to 45
Default: 23
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

MVANGLF

Velocity-Related Commutation Angle Advance at the Motor Max Speed (MSPEED): This variable sets the value of the velocity-related commutation angle advance to be used when the motor is operating at motor max speed (MSPEED). Between MSPEED/2 RPM and MSPEED, the angle advance will be linearly interpolated based on MVANGLH and MVANGLF.

When a CLREEPROM command is issued, MVANGLF is set to a value of 10. If a CONFIG command is then issued, MVANGLF is set to a default value based on MSPEED and MPOLES. Once the user enters their own value for MVANGLF, however, MVANGLF will keep that value and will not be changed if a CONFIG is executed.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Electrical Degrees
Range: .... 0 to 90
Default: .... Calculated from motor data
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

MVANGLH

Velocity-Related Commutation Angle Advance at Half of the Motor Max Speed (MSPEED):
This variable sets the value of the velocity-related commutation angle advance to be used when the motor is operating at half of the motor max speed (MSPEED). Between 0 RPM and half of MSPEED, the angle advance will be linearly interpolated based on MVANGLH.

When a CLREEPROM command is issued, MVANGLH is set to a value of 5. If a CONFIG command is then issued, MVANGLH is set to a default value based on MSPEED and MPOLES. Once the user enters their own value for MVANGLH, however, MVANGLH will keep that value and will not be changed if a CONFIG is executed.

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... Electrical Degrees
Range: .... 0 to 90
Default: .... Calculated from motor data
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

O1

This variable is used to read or write the state of the hardware input on user connector C3 Pin 12. Note that writing O1 in certain O1MODE conditions does not stop the drive from resetting the output according to drive conditions. See also O1MODE.

Firmware Versions: 2.1.0 and later
Type: .... switch (R/W)
Units: .... 0, 1 (Off / On)
Range: .... 0 to 1
Default: .... n/a
EEPROM: .... n/a
OPMODES: .... all
Drive Status: ...... en/dis

O1MODE

This is a switch mode variable used to define the function of O1:

0 - Disabled
1 - Output goes on when absolute motor speed is above 01TRIG speed.
2 - Output goes on when the absolute actual output current is above O1TRIG current.
3 - Output Goes On when drive is in FOLDBACK.
4 - Output goes on when absolute motor speed is above O1TRIG but less than O1RST.
5 - Brake Mode: Output is off only when Drive is disabled, or after an active disable time-out and on when the drive is enabled.
6 - Motion Complete Output (tied to STOPPED switch)
7 - Reserved
8 - Zero Speed Detect: Output on if absolute motor speed < O1TRIG.
9 - Beyond Position: Output if Position beyond preset point.

Firmware Versions: 2.1.0 and later
Type: .... switch mode (R/W)
Units: .... n/a
Range: .... 0 to 9
Default: 6
EEPROM: .... yes
OPMODES: ...... all
Drive Status: ..... en/dis

O1RST

This is a variable used to define the reset level for O1MODE. Range is dependent on O1MODE:

0 through 3 - n/a
4 - Absolute: 0 to 15000 RPM (mm/sec for linear motors)
5 - Absolute: 0 to 15000 RPM (mm/sec for linear motors)
6 through 9 - n/a

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... RPM
Range: .... see above
Default: .... VOSPD
EEPROM: .... yes
OPMODES: ..... all
Drive Status: .... en/dis

O1TRIG

This is a variable used to define the trip level for O1MODE. Range is dependent on O1MODE:

0 - n/a
1 - Absolute: 0 to 15000 RPM (mm/sec for linear motors)
2 - Absolute: 0 to 1000 (0.1 percent of DIPEAK)
3 - n/a
4 - Absolute: 0 to 15000 RPM (mm/sec for linear motors)
5 through 7 - n/a
8 - Absolute: 0 to 15000 RPM (mm/sec for linear motors)
9 - n/a

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... see above
Range: .... see above
Default: 1000
EEPROM: .... yes
OPMODES: .... all
Drive Status: ..... en/dis

OPMODE

Operational Mode:
This switch mode variable sets the operational mode for the drive. The drive can be configured as a velocity or torque loop controller.

0 = serial velocity mode
(See also J, COMPMODE, PROFMODE, S, STOP)
1 = analog velocity mode
(See also VSCALE, COMPMODE, PROFMODE, S)
2 = serial torque mode
(See also T, S, STOP)
3 = analog torque mode
(See also ISCALE, S)

Firmware Versions: All
Type: .... Switch Mode Variable (Read / Write)
Units: .... n/a
Range: .... 0 - 3
Default: 1
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Disabled

PE

This variable displays the position following error. If this value is greater than PEMAX, then the drive will be disabled. Position is in counts.

Firmware Versions: 2.1.0 and later
Type: .... variable (R)
Units:................. counts
Range: .... +/- 2,147,483,647
Default: .... n/a
EEPROM:........... n/a
OPMODES:........ 4
Drive Status: ...... en/dis

PEINPOS

This variable sets the threshold position error for the INPOS flag. If PE is less than PEINPOS, the INPOS switch is set, indicating that the drive is in position (see INPOS). If PE is greater than PEINPOS, the INPOS switch is not set. Position is in counts.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... Counts
Range:................ 0 to 32767
Default: 100
EEPROM: .... yes
OPMODES: 4
Drive Status: .... en/dis

PEMAX

This variable sets the maximum allowable following error in OPMODE 4. If the error exceeds this value, then the drive is disabled on fault. PEMAX = 0 disables this function. Position is in counts.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... counts
Range: .... +/- 2,147,483,647
Default: 0
EEPROM: .... yes
OPMODES: 4
Drive Status: ..... en/dis

PEXT

This variable displays the accumulated position feedback from the external encoder. This variable is similar to PFB for the resolver feedback.

Firmware Versions: 2.1.0 and later
Type: .... variable (R)
Units: .... counts
Range:................ +/- 2,147,483,647
Default: .... n/a
EEPROM: .... n/a
OPMODES: .... all
Drive Status: ..... en/dis

PEXTOFF

This variable is an offset that is added to the internal accumulated position feedback from the external encoder to give the value of PEXT.

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units: .... counts
Range:................ +/- 2,147,483,647
Default: .... n/a
EEPROM: .... no
OPMODES: ...... all
Drive Status: .... en/dis

PFB

Hardware Counter Position Feedback:
This variable displays the position feedback directly from the feedback hardware counter. For all systems PFB is a right-justified 32 bit variable.

Firmware Versions: All
Type: .... Standard Variable (Read Only)
Units: .... counts
Range: .... 0 to 2^32
Default: .... n/a
EEPROM: .... No
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

PFBOFF

This is a feedback offset that is added to the internal cumulative position counter to give the value of PFB.

Firmware Versions: 2.0.0 and later
Type: .... variable (R/W)
Units: .... counts
Range: .... +/- 2,147,483,647
Default: .... n/a
EEPROM: .... no
OPMODES: .... all
Drive Status: ...... en/dis

PRD

Position Feedback:
This variable displays the absolute position feedback of the hardware feedback device (for both resolver and encoder based systems). PRD will increment from 0 to 65,535 throughout the course of one mechanical motor shaft revolution (360 degrees). The range of PRD will not change. Its resolution for resolver feedback systems is dependent upon the value of RDRES:

RDRES = 12, resolution of PRD = 16.
RDRES = 14, resolution of PRD = 4.
RDRES = 16, resolution of PRD = 1.

For encoder-based systems, until the encoder has been initialized, PRD will be uninitialized and its value will not be useful or meaningful. For information on encoder initialization requirements according to the type of encoder, see MENCTYPE, ENCINIT, and ENCINITST.

Firmware Versions: All
Type: .... Standard Variable (Read Only)
Units: .... counts
Range: .... 0 to 65,535
Default: .... n/a
EEPROM: .... No
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

PROFMODE

Profile Mode:
This switch variable selects the acceleration and deceleration algorithm used by the drive (profile mode). Note that PROFMODE is associated with ACC and DEC but may not affect ramping depending upon the values of ACTFAULT, STOP, and DECSTOP.

0 = no acceleration and deceleration ramp limits
1 = linear acceleration and deceleration ramp limits

Firmware Versions: All
Type: .... Switch Variable (Read / Write)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... Yes
OPMODES: 0, 1
Drive Status: ..... Enabled / Disabled

PROMPT

Prompt Enable:
This switch variable enables/disables the serial port prompt (-->) output by the SERVO**STAR** after each message.

0 = disable the prompt
1 = enable the prompt

Note that PROMPT = 1 is needed for proper operation of MotionLink.

Firmware Versions: All
Type: .... Switch Variable (Read / Write))
Units: .... n/a
Range: 0, 1
Default: 1
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

R

R Polynomial Vector:
This vector variable defines the R (feed-forward path) polynomial of the Advanced Pole-Placement velocity controller (COMPMODE = 3). This vector includes three integers that represent the polynomial coefficients and three shift parameters, one to scale each polynomial.
Note that if this variable is changed, a REFRESH command is required.

Syntax: R [r0] [r0shift] [r1] [r1shift] [r2] [r2shift]
Example: R 10000 2 30000 4 50000 6
Ranges : vectorN = -2,147,483,647 to +2,147,483,647
scaleN shift = 0 to 32767

Firmware Versions: All
Type: .... Vector Variable (Read / Write)
Units: .... n/a
Range: .... See Above
Default: .... all 0's
EEPROM: .... Yes
OPMODES: 0, 1
Drive Status: ...... Enabled / Disabled

RDRES

Resolver-to-Digital Resolution:
This switch mode variable displays the resolver resolution on resolver-based systems. RDRES is a read-only variable which is automatically calculated in order to achieve maximum resolution. The setting is based on VLIM, which is the maximum application velocity. The relationship between VLIM and RDRES is given below:
if (VLIM ≥ 6101) then RDRES = 12
if (1501 ≤ VLIM ≤ 6100) then RDRES = 14
if (VLIM ≤ 1500) then RDRES = 16

Firmware Versions: All
Type: .... Switch Mode Variable (Read Only)
Units: .... Bits
Range: 12, 14, 16
Default: .... n/a
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

READY

Software Ready State:
This switch variable is a flag indicating the status of the software enable. READY = 1 means that there are no faults (DRIVEOK = 1) and a communication enable request has been commanded (SWEN = 1). An external Remote Enable (REMOTE = 1) and a Dip Switch Enable (DIPEN = 1) are still required to enable the drive (ACTIVE = 1).

0 = faults exist or SWEN = 0
1 = no faults exist and SWEN = 1

Firmware Versions: All
Type: .... Switch Variable (Read Only)
Units: .... n/a
Range: 0, 1
Default: .... n/a
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

RECDONE

Record Done:
This switch variable indicates whether or not the Record command is done and data is available.

0 = recording not finished
1 = recording done; data available

Firmware Versions: All
Type: .... Switch Variable (Read Only)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

RECING

Recording:
This switch variable indicates if data recording is in progress.

0 = recording not in progress
1 = recording in progress

Firmware Versions: All
Type: .... Switch Variable (Read Only)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

RECRDY

Record Ready State:
This switch variable indicates the ready status of the RECORD function. This variable is used to indicate if the system is waiting for RECTRIG.

0 = RECTRIG has been received
1 = recording function waiting for RECTRIG

Firmware Versions: All
Type: .... Switch Variable (Read Only)
Units: .... n/a
Range: 0, 1
Default: 1
EEPROM: .... No
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

RECTRIG

Record Trigger:
This variable sets up the trigger mechanism for the RECORD function. RECORD must be set up before a RECTRIG command is issued. Four parameters are required to set up RECTRIG: Mode, Level, Location, and Direction.

1. MODE is a string variable that specifies the parameter that will be used to trigger recording. Mode can be a variable name or a triggering condition. Mode determines what other parameters must be entered in order to completely set up the trigger. Mode must be preceded by a double-quote when entered, as shown in the following table, which tells what other parameters are required (LEVel, LOCation, and DIRection) depending upon the selected Mode.

2. LEVEL specifies the value that the variable defined by Mode must reach for recording to begin.

3. LOCATION specifies how many data points to save before the trigger in the Recording buffer (see the RECORD command for a description of the 1024 data points that are available). When recorded data is retrieved and displayed, the trigger point's location in the 1024-point buffer will be at the place specified by Location.

4. DIRECTION has two meanings depending upon the type of Mode parameter that is used. For Mode variables (PRD, IA, IC, V, ICMD, VCMD), it defines the direction the variable value must be changing when it crosses Level in order to trigger recording (1 = increasing, 0 = decreasing). For Mode switch inputs (RMT, CW, CCW) it defines the logic level the input must achieve in order to trigger recording (1 = HI, 0 = LOW).

Required RECTRIG Parameters Based on MODE Parameter				
MODE	DESCRIPTION	LEV	LOC	DIR
"PRD	Trigger on PRD	√	√	√
"IA	Trigger on Phase A Current	√	√	√
"IC	Trigger on Phase C Current	√	√	√
"V	Trigger on Velocity	√	√	√
"ICMD	Trigger on Current Command	√	√	√
"VCMD	Trigger on Velocity Command	√	√	√
"RMT	Trigger on REMOTE Input	X	√	√
"CW	Trigger on CWLIM Sw	X	√	√
"CCW	Trigger on CCWLIM Sw	X	√	√
"CMD	Trigger on Next Command	X	√	X
"IMM	Trigger Immediately	X	X	X

√ = Required Parameter
X = Don't care; something must be entered to make the command work, but it doesn't matter what is entered.

Syntax: RECTRIG [mode] [level] [location] [direction]

Range: mode = see table above
level = depends upon the mode variable
(range of PRD levels is 0-65535, all others are -32768 to 32767)
location = 0 - 1023
direction = 0 or 1

Firmware Versions: All
Type: .... Standard Variable (Read / Write)
Units: .... See Above
Range: .... See Above
Default: .... Level=0, Location=0, Direction=1
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

RELAY

Relay State:
This switch variable indicates the status of the Fault / Drive Up Relay.
0 = relay open
1 = relay closed

Firmware Versions: All
Type: .... Switch Variable (Read Only)
Units: .... n/a
Range: 0, 1
Default: .... Hardware Defined
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

RELAYMODE

Status Relay Mode:
This switch variable sets the operation of the Drive Up / Drive Ready Relay.

0 = relay will be closed when no faults exist and drive is ready to be enabled
1 = relay will be closed when ACTIVE equals 1

Firmware Versions: All
Type: .... Switch Variable (Read / Write)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

REMOTE

Remote Enable State:
This switch variable indicates the state of the external hardware enable input line. When REMOTE is set to 1, the software is ready (READY = 1), and Dip Switch 8 is set (DIPEN = 1), the Drive is Enabled (ACTIVE = 1).

0 = remote enable input off
1 = remote enable input on

Firmware Versions: All
Type: .... Switch Variable (Read Only)
Units: .... n/a
Range: 0, 1
Default: .... Hardware Defined
EEPROM: .... No
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

SERIALNO

This variable indicates the serial number of the drive in which the firmware is installed. This variable is password protected. This variable is included in the VER string.

Firmware Versions: All
Type: .... String Variable (Read Only)
Units: .... n/a
Range: .... 10 ASCII characters
.... (0x23 to 0x5A and 0x61 to 0x7A)
Default: .... Blanks
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

STAT

Status Summary:
This variable will output a drive status summary word to the serial port. This summary word is in ASCII-hex format, prefixed by the letter 'H'. See STATUS for information on how to obtain more detailed drive status information. The format of the STAT word is described in the following table.

Bit #	Function	Convention
0 (LSB)	Disable Status	1 = drive is DISabled 0 = drive is ENabled
1	Fault Status	1 = fault exists 0 = no fault exists
2	Safety Status	1 = safety feature triggered/inactive* 0 = drive is safe
3	Special Mode Status	1 = Step, Burnin, or Zero is active 0 = normal
4	Hold Mode Status	1 = drive is in Hold mode 0 = drive is not in Hold mode
5-15	not used	

*CWLIM = 1, CCWLIM = 1, LIMDIS = 1, THERMODE = 1 or 2, or FOLD = 1.

Firmware Versions: All
Type: .... Standard Variable (Read Only)
Units: .... n/a
Range: .... See Above
Default: .... n/a
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

STATUS

Drive Status:
This variable will output the drive status detail words to the serial port. Five words are transferred in ASCII-HEX format, with each word preceded by the letter 'H'. The words are separated by a space.

The following tables break the status words down bit by bit (bit 7 = MSB; bit 0 = LSB). For all bits, 0=false and 1=true.

Word 1: Disable Status Word	
If the drive is disabled (Bit 0 of the STAT word = 1), the process(es) which have caused that disable condition will have their bits set to 1 in this word.	
Bit #	**Description**
0 (LSB)	Remote disable (REMOTE = 0)
1	Software disable (SWEN = 0)
2	DIP switch disable (DIPEN = 0)
3	Fault disable
4	Velocity loop design failure
5	Encoder initialization process active
6-15	not used

Word 2: Fault Status Word	
If a fault exists (Bit 1 of the STAT word = 1), the fault(s) which exist(s) will have the corresponding bits set to 1 in this word.	
Bit #	**Description**
0 (LSB)	Drive over temperature
1	Over voltage condition
2	Over current condition
3	Feedback loss
4	Under voltage condition
5	Motor over temperature
6	Analog supply fault
7	Over speed condition
8	EEPROM fault
9	EEPROM checksum fault
10	No comp (compensation) for the motor
11	Foldback condition
12-15	not used

STATUS Word 3: Safety Status Word	
If safety of the drive is compromised (Bit 2 of the STAT word = 1), the condition which is causing that state has its corresponding bit set to 1 in this word.	
Bit #	**Description**
0 (LSB)	CWLIM = 1 (motor has reached CW travel limit)
1	CCWLIM = 1 (motor has reached CCW travel limit)
2	LIMDIS = 1 (limit switch function disabled by user)
3	THERMODE = 1 or 2 (set to non-zero by the user)
4	FOLD = 1 (drive is in foldback mode)
5*	LIMDIS=0 & CW switch not routed (INxMODE 1)
6*	LIMDIS=0 & CCW switch not routed (INxMODE 2)
7-15	not used

Word 4: Special Mode Status Word	
If the drive is in a special operating mode (Bit 3 of the STAT word = 1), the special mode that the drive is in has its corresponding bit set to 1 in this word.	
Bit #	**Description**
0 (LSB)	Drive is in Step mode (see STEP)
1	Drive is in Burnin mode (factory function)
2	Drive is in Zeroing mode (see ZERO)
3-15	not used

Word 5: Hold Mode Status Word	
If the drive is in Hold mode (Bit 4 of the STAT word = 1), the condition which caused the drive to enter Hold mode has its corresponding bit set to 1 in this word).	
Bit #	**Description**
0 (LSB)	User request (user set HOLD = 1)
1	DIP switch setting (DIP switch 7 = 1)
2	Drive is in Active disable state
3	Limit switch(es) tripped 1. velocity command is in direction of tripped switch in opmode 0 or 1 with drive enabled; or 2. both limit switches are activated.
4-15	not used

Firmware Versions: All
Type: ....Standard Variable (Read Only)
Units: ....n/a
Range: ....See Above
Default: ....n/a
EEPROM: ....No
OPMODES: ....All
Drive Status: ......Enabled / Disabled

STATUS2 will output drive status detail words to the serial port. Four words are transferred in ASCII-HEX format, with each word preceded by the letter 'H.'

The following tables break the status words down bit by bit (bit 7 = MSB; bit 0 = LSB; n/u = not used). For all bits, 0=false and 1=true.

STATUS2 Word 1: Feedback Loss Status Word	
If the drive has experienced a feedback loss fault (Bit 3 of STATUS Word 2 = 1), the condition which caused that fault will have its bit set to 1 in this word.	
Bit #	**Description**
0 (LSB)	Resolver line break
1	Resolver/Digital Converter Error bit (following err)
2	not used
3	Line break of encoder A/B input
4	Line break of encoder index input
5	Illegal halls state
6-15	not used

STATUS2 Word 2: Analog Supply Fault Status Word	
If the drive has experienced an analog supply fault (Bit 6 of STATUS Word 2 = 1), the condition which caused that fault will have its bit set to 1 in this word.	
Bit #	**Description**
0 (LSB)	Positive analog supply fault
1	Negative analog supply fault
2	Both analog supplies failed
3-15	not used

STATUS2 Word 3: Position Deviation and Over Travel Fault Status Word	
This status word is reserved	
Bit #	**Description**
0-15	Reserved

STATUS2 Word 3: Position Deviation and Over Travel Fault Status Word	
This status word is reserved	
Bit #	**Description**
0-15	Reserved

STATUS2 Word 4: Limit Switches Status Word	
If the drive has experienced a limit switch fault (Bit 3 of STATUS Word 5 = 1), the condition which caused that fault will have its bit set to 1 in this word.	
Bit #	**Description**
0 (LSB)	CW Limit Switch tripped (CWLIM=1)
1	CCW Limit Switch tripped (CCWLIM=1)
2-15	not used

Firmware Versions: 2.1.0 and later
Type: .... variable (R)
Units: .... n/a
Range: .... see above
Default:............... n/a
EEPROM: .... no
OPMODES: all
Drive Status: ......en/dis

STOPMODE

This variable sets the mode of dynamic braking operation. See also ISTOP.
0 = no braking operation (default).
1 = brake on fault only.
2 = brake on fault and/or drive disable.
Note: Faults do not include Over Voltage or Power Stage Faults!

Firmware Versions: 2.1.0 and later
Type: ....switch mode (R/W)
Units: ....n/a
Range: 0, 1, 2
Default: 0
EEPROM: ....yes
OPMODES: ....all
Drive Status: ......en/dis

SWEN

Software Enable:
This switch variable is a software enable switch that defines the status of the communication Enable request. If SWEN is set to 1, and there are no faults (DRIVEOK = 1), then switch variable READY is set = 1.
0 = software disabled
1 = software enabled

Firmware Versions: All
Type: ....Switch Variable (Read Only)
Units: ....n/a
Range: 0, 1
Default: 1
EEPROM: ....No
OPMODES: ....All
Drive Status: ......Enabled / Disabled

TF

This variable sets the damping factor for the velocity loop when using COMPMODE 2 or COMPMODE 4 (Standard Pole Placement). A value of 100 is backward compatible to all previous firmware. As TF approaches zero, overshoot is diminished while sacrificing some tracking ability. As TF approaches 200, the system may overshoot more but will have excellent steady-state tracking ability. Successful execution of the TUNE command may result in this parameter being changed.

Firmware Versions: all
Type: .... variable (R/W)
Units: .... n/a
Range: .... 0 to 200
Default: 100
EEPROM: .... yes
OPMODES: 0,1
Drive Status: ...... en/dis

THERM

Thermostat Input State:
This switch variable indicates the state of the motor thermostat input.

0 = thermostat input closed (normal)
1 = thermostat input open (overheat condition)

Firmware Versions: All
Type: ....Switch Variable (Read Only)
Units: ....n/a
Range: 0, 1
Default: ....Hardware Defined
EEPROM: ....No
OPMODES: ....All
Drive Status: ......Enabled / Disabled

THERMODE

Thermostat Input Mode:
This switch mode variable determines the operation of the drive when the Motor Thermostat Input (THERM) opens.

0 = disable drive and open Fault relay immediately
1 = disable drive after 2 minutes; open Fault relay immediately
2 = do not disable drive; open Fault relay immediately

Firmware Versions: All
Type: ....Switch Mode Variable (Read / Write)
Units: ....n/a
Range: ....0 - 2
Default: 0
EEPROM: ....Yes
OPMODES: ....All
Drive Status: ......Enabled / Disabled

TRUN

Run Time:
This variable provides a relative incremental run time counter. Error log stamps include the value of this counter at the time of the error. The clock is a very coarse counter and is incremented every 15 minutes. It is intended for use by Industrial Drives Quality Assurance Program personnel. This clock has a resolution of 15 minutes and is reset only when the CLREEPROM command is used.

Firmware Versions: All
Type: .... Standard Variable (Read Only)
Units: .... Hours:Minutes
Range: .... 0000:00 to 9999.45
Default: .... n/a
EEPROM: .... Yes
OPMODES: .... All
Drive Status: ...... Enabled / Disabled

UNITS

This variable defines whether physical units or internal bits are used. This variable is relevant mainly for Current, Velocity, Acceleration and Analog Input variables, in order to allow more precise definitions while using the internal bits of the Integer variables. It is recommended that most users use the physical units.

0 = use physical units
1 = use internal units

The descriptions in this guide use the physical units. Variables that may be defined using internal units are listed in the following table, along with their internal unit ranges and units.

INTERNAL VARIABLE UNITS		
Variable	**Range**	**Internal Units**
ANDB	0 to 16383 bits	1 bit = 10V / 16384
ANIN	-16383 to 16383 bits	1 bit = 10V / 16384
ANOFF	-16383 to 16383 bits	1 bit = 10V / 16384
I	0 to 65535 bits	32768 bits = DIPEAK * (√2/0.8)
IA	-32767 to 32767 bits	32768 bits = DIPEAK * (√2 / 0.8)
IC	-32767 to 32767 bits	32768 bits = DIPEAK * (√2 / 0.8)
ICMD	-32767 to 32767 bits	32768 bits = DIPEAK * (√2 / 0.8)
ICONT	0 to IMAX bits	32768 bits = DIPEAK * (√2 / 0.8)
ILIM	0 to IMAX bits	32768 bits = DIPEAK * (√2 / 0.8)
IMAX	0 to 32767 bits	32768 bits = DIPEAK * (√2 / 0.8)
J <vel> <time>	vel:-16383 to 16383 time:0 to 32767	vel:1 bit = VLIM / 16384 time:1 bit = 0.5 ms
STEP <period> <speed>	per:0 to 32767 sp:-16363 to +16383	period: milliseconds speed: VLIM / 16384

T	-ILIM to ILIM	32768 bits = DIPEAK * (√2 / 0.8)
V	-32767 to 32767	1 bit = 60000 / 65535 RPM
VCMD	-VLIM to VLIM	1 bit = VLIM / 16384
VE	-16383 to 16383	1 bit = VLIM / 16384

Firmware Versions: all
Type: switch (R/W)
Units:.................. n/a
Range:................ 0, 1
Default: 0
EEPROM: .... yes
Opmodes: .... all
Drive Status:....... en/dis

UVRECOVER

Under Voltage Recovery:
This switch variable defines the way to handle the Fault procedure when a temporary Under Voltage (UV) condition occurs.

0 = recover by toggling drive from disable to enable condition after the UV condition clears
1 = automatically recover when the UV condition clears

Firmware Versions: All
Type: .... Switch Variable (Read / Write)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... Yes
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

V

This variable displays the velocity as calculated from the hardware feedback (resolver or encoder). The velocity that is displayed is subject to averaging by the variable AVGTIME, except when it is recorded for graphical display by MotionLink, in which case it is not averaged.

Firmware Versions: all
Type:................... variable (R)
Units:.................. rotary: RPM
.... inear: mm/sec
Range: .... -15000 to 15000
Default: .... n/a
EEPROM: .... no
OPMODES: ...... all
Drive Status: ..... en/dis

VBUS

Bus Voltage:
This variable sets the drive bus voltage. This variable is used for current controller design. VBUS also affects the value of VMAX (see VMAX). When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command (see CONFIG).

Firmware Versions: All
Type: ....Standard Variable (Read / Write)
Units: ....Volts
Range: ....10 to 500
Default: 325
EEPROM: ....Yes
OPMODES: ....All
Drive Status: ......Disabled

VCMD

This variable displays the Velocity command to the velocity controller. This value is equivalent to the Analog Input (ANIN) in OPMODE 1, to the Jog Command (J) in OPMODE 0, and the output of the position controller in Hold Position mode. This variable is averaged, based on the AVGTIME, when it is requested via the serial port. It is not averaged when it is recorded for graphical display.

Firmware Versions: all
Type: .... variable (R)
Units: .... rotary: RPM
............................linear: mm/sec
Range: .... -VLIM to +VLIM
Default: .... n/a
EEPROM: .... no
OPMODES: 0,1
Drive Status: ......en/dis

VD

This variable sets the D (forward path) polynomial of the Advanced Pole-Placement velocity controller (COMPMODE = 3). The vector defined by this variable includes five integers that represent the polynomial coefficients and a shift parameter that scales the polynomial. If this variable is changed, a REFRESH command is required.

NOTE: prior to firmware version 2.1.0, this command mnemonic was “D”.

SYNTAX: VD [vector1] [vector2] [vector3] ... [vector5] [scale]
EXAMPLE: VD 100 200 300 400 500 1

RANGES: vectorN = -32768 to 32767
scale = 0 to 15

Firmware Versions: 2.1.0 and later (previously D)
Type: ....vector variable (R/W)
Units: ....n/a
Range: ....see above
Default: ....0 (all parameters)
EEPROM: ....yes
OPMODES: 0, 1
Drive Status: ......en/dis

VE

This variable displays the velocity error, which is the difference between the commanded motor velocity (VCMD) and the actual motor velocity (V). This value is an instantaneous reading.

Firmware Versions: all
Type: .... variable (R)
Units: .... rotary: RPM
.... linear: mm/sec
Range: .... -32768 to 32767
Default: .... n/a
EEPROM: .... no
OPMODES: 0,1
Drive Status: ..... en/dis

VER

Firmware Version:
This variable indicates the version of the drive firmware in use. This variable also displays other pertinent information such as the drive name, current ratings, TRUN, SERIALNO, etc. The VER variable has two optional forms: requesting VER 1 will return the feedback type (encoder or resolver), and VER 2 will return the firmware version.

Firmware Versions: All
Type: .... String Variable (Read Only)
Units: .... n/a
Range: .... VER {1 or 2}
Default: .... n/a
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

VEXT

This variable displays the instantaneous velocity feedback as calculated from the external encoder input channel. The command uses XENCRES to calculate velocity. This variable is similar to V for the motor feedback. This variable is subject to AVGTIME

Firmware Versions: 2.1.0 and later
Type: .... variable (R)
Units: .... rotary: RPM
.... linear: mm/sec
Range: .... -32767 to +32767
Default: .... n/a
EEPROM: .... n/a
OPMODES: ..... all
Drive Status: ..... en/dis

VH

This sets a vector of variables that defines the H (feedback path) polynomial of the Advanced Pole-Placement velocity controller (COMPMODE = 3). The vector defined by this variable includes four integers that represent the polynomial coefficients, and four shift parameters, one that scales each polynomial. If this variable is changed, a REFRESH command is required.

NOTE: prior to firmware version 2.1.0, this command mnemonic was "H".

SYNTAX: VH [h0] [h0shift] [h1] [h1shift] ... [h3] [h3shift]
EXAMPLE: VH 10000 2 30000 4 50000 6 70000 8

RANGE: hN = -2,147,483,647 to 2,147,483,647
hNshift = 0 to 32767

Firmware Versions: 2.1.0 and later (previously H)
Type: ....vector variable (R/W)
Units: ....n/a
Range: ....see above
Default: ....0 (all parameters)
EEPROM: ....yes
OPMODES: 0,1
Drive Status: ......en/dis

VLIM

This variable sets the application velocity limit, allowing the user to limit the motor's peak velocity. VLIM limits the velocity command that will be accepted from the user (using the J command in Opmode 0) or issued by the control loops (in Opmode 1). VLIM is an independent variable that is not calculated from hardware parameters and is not tied to any other variables. VLIM is similar to ILIM (used in Opmodes 2 & 3) and can be used to protect delicate load equipment. For rotary motors, VLIM > 6100 only if ENCOUT ≤ 1024.

Firmware Versions: all
Type: .... variable (R/W)
Units: .... rotary: RPM
............................linear: mm/sec
Range:................. 10 to VMAX
Default: ....VMAX
EEPROM: .... yes
OPMODES: 0, 1
Drive Status: .....dis

VMAX

This variable displays the system velocity maximum for a drive and motor combination. This variable is based on drive and motor hardware parameters and is set equal to the MINIMUM of the four following values:

1.) MSPEED
2.) (VBUS * 0.707 / MBEMF) * 1000
3.) 12,000
4.) 61,440,000 / MENCRES (encoder-feedback systems only)
5.) 96000 / MRESPOLES (resolver system with MRESPOLES>8)

Note that 12,000 is the highest value VMAX can take. VMAX is used to limit VLIM and VOSPD.

Firmware Versions: all
Type: .... variable (R)
Units: .... rotary: RPM
linear: mm/sec
Range: .... 10 to 12000
Default: .... see above
EEPROM: .... yes
OPMODES: 0,1
Drive Status: ...... dis

VOSPD

This variable sets the over-speed trip limit for the motor. The drive is disabled with an error condition when the drive velocity exceeds this limit. This default value of this variable is 20% above the system velocity maximum (VMAX) but can be reduced by the user during regular motor operation for protection.

Firmware Versions: all
Type:.................. variable (R/W)
Units: .... rotary: RPM
linear: mm/sec
Range: .... 10 to (1.2 * VMAX)
Default: .... VMAX * 1.2
EEPROM: .... yes
OPMODES: ..... all
Drive Status: ..... dis

VR

This is a vector variable that defines the R (feed-forward path) polynomial of the Advanced Pole-Placement velocity controller (COMPMODE = 3). This vector includes three integers that represent the polynomial coefficients and three shift parameters, one that scales each polynomial. If this variable is changed, a REFRESH command is required.

NOTE: prior to firmware version 2.1.0, this command mnemonic was "R".

SYNTAX: VR [r0] [r0shift] [r1] [r1shift] [r2] [r2shift]
EXAMPLE: VR 10000 2 30000 4 50000 6

RANGES: rNvector = -2,147,483,647 to 2,147,483,647
rNshift = 0 to 32767

Firmware Versions: 2.1.0 and later (previously R)
Type: ....vector variable (R/W)
Units: ....n/a
Range: ....see above
Default: ....all 0's
EEPROM: ....yes
OPMODES: 0,1
Drive Status: ......en/dis

VSCALE

This variable is a velocity input scale factor that scales the analog input for OPMODE 1. The value entered is the desired motor velocity if 10 volts were applied to the analog input. This variable may be either higher or lower than the application velocity limit (VLIM), but the input is limited by VLIM.

Firmware Versions: all
Type: .... variable (R/W)
Units: .... rotary: RPM / 10 volts
...........................linear: mm/sec / 10 volts
Range: .... 10 to 32767
Default: .... VLIM / 0.8
EEPROM: .... yes
OPMODES: 1
Drive Status: .....en/dis

XENCRES

This variable sets the resolution of the external encoder input channel and is used to calculate VEXT.

Firmware Versions: 2.1.0 and later
Type: ....variable (R/W)
Units: ....Lines per revolution
Range: ....100 to 65535
Default: 1024
EEPROM: ....yes
OPMODES: ....all
Drive Status: ......dis

XENCDIR

This variable sets the direction defined as positive rotation for the external encoder input.

0 = normal
1 = inverted

Firmware Versions: 2.1.0 and later
Type: .... variable (R/W)
Units:.................. n/a
Range: 0,1
Default: 0
EEPROM: .... yes
OPMODE: .... all
Drive Status: ...... dis

ZERO

Zero Mode:

This switch variable enables/disables Resolver/Encoder Zeroing Mode. If Zeroing Mode is enabled, SERVOSTAR rotates the motor to an electrical null by placing IZERO current from the motor C terminal to the motor A terminal.

0 = zeroing mode disabled
1 = zeroing mode enabled (places the drive in OPMODE 2)

Firmware Versions: All
Type: .... Switch Variable (Read / Write)
Units: .... n/a
Range: 0, 1
Default: 0
EEPROM: .... No
OPMODES: ...... All
Drive Status: ..... Enabled / Disabled

6.6 VARIABLE DEFAULT VALUES

DEFAULT VALUES ASSIGNED BY "RSTVAR"		
ACC = 399,987	GV = 500	MSG = 1
ACKMODE = 0	GVI = 20	O1MODE = 6
ACTFAULT = 0	IENCSTART = 25	O1RST = VOSPD
ANDB = 0	IFRIC = 0	O1TRIG = 1000
ANOFF = 0	IGRAV = 0	OPMODE = 1
ANOUT = 0	ILIM = IMAX	PROFMODE = 0
AVGTIME = 0	ILSBMODE = 0	PROMPT = 1
BW = 20	IN1MODE = 1	RELAYMODE = 0
COMPMODE = 2	IN2MODE = 2	STOPPED = 0
DEC = 399,987	IN3MODE = 3	TF=100
DECSTOP = 5000	ISCALE = 1250	THERMODE = 0
DIR = 1	ISTOP = IMAX	UNITS = 0
ECHO = 1	IZERO = 25	UVRECOVER = 0
ENCOUT = 1024	KV = 1000	VD = all 0's
FILTMODE = 0	KVI = 1000	VH = all 0's
FOLDMODE = 0	KVFR = 0	VLIM = VMAX
GEAR = 1	LIMDIS = 0	VOSPD = 1.2*VLIM
GEARI = 1	LMJR = 0	VR = all 0's
GEARMODE = 0	LPFHZ1 = 500	VSCALE = 1.25*VLIM
GEARO = 1	LPFHZ2 = 500	XENCRES = 1024
GETMODE = 0	MPHASE = 0	

6.7 COMMANDS & VARIABLES SORTED BY FUNCTION

DRIVE PARAMETERS		
DICONT	DIPEAK	VBUS

MOTOR PARAMETERS AND COMMANDS		
MBEMF	MJ	MPOLES
MBEMFCOMP	MKT*	MRES*
MENCOFF	MKTI*	MRESPOLES
MENCRES	MLGAINP	MSININT
MENCTYPE	MLGAINZ	MSPEED
MHINVA	MLMIN	MTANGLC
MHINVB	MLMAX*	MTANGLP
MHINVC	MOTOR	MVANGLF
MICONT	MOTORTYPE*	MVANGLH
MIPEAK	MPHASE	***MLIST***

- These variables are not documented and are no longer used.

They are listed here because they are included in an MLIST. MOTORTYPE must be set equal to 0!

DRIVE STATUS		
ACTIVE	RELAY	TRUN
CCWLIM	SERIALNO	VER
CWLIM	STAT	***ERR***
DIP	STATUS	***FLTCLR***
DRIVEOK	THERM	***FLTHIST***
READY		

DRIVE CONFIGURATION AND MODES		
ACTFAULT	GEARMODE	STOPMODE
COMPMODE	GEARO	THERMODE
DIP	HOLD	UNITS
DIR	LIMDIS	UVRECOVER
FILTMODE	OPMODE	ZERO
GEAR	PROFMODE	***TESTLED***
GEARI	RELAYMODE	

VARIABLE SETTING AND CLEARING		
CLREEPROM	***LOAD***	***RSTVAR***
DUMP	***MLIST***	***SAVE***
LIST		

DRIVE ENABLING AND DISABLING		
ACTIVE	REMOTE	***K***
DIPEN	SWEN	***S***
DRIVEOK	***EN***	***STOP***
READY	***DIS***	***SWEN***

COMMUNICATIONS		
ACKMODE	ECHO	PROMPT
ADDR	MSG	

ENCODER / RESOLVER-RELATED		
ENCINITST	MENCOFF	PFBOFF
ENCOUT	MHINVA	PRD
HALLS	MHINVB	RDRES
HWPOS	MHINVC	XENCRES
IENCSTART	MPHASE	***ENCINIT***
MENCTYPE	MRESPOLES	***ENCSTART***
MENCRES	PFB	

ANALOG INPUT-RELATED		
ANDB	ANOFF	VSCALE
ANIN	ISCALE	***ANZERO***

CURRENT PARAMETERS AND COMMANDS		
DIPEAK	ICMD	ILIM
DICONT	ICONT	IMAX
FOLD	IENCSTART	ISCALE
FOLDMODE	IFRIC	ISTOP
I	IGRAV	IZERO
IA	MICONT	***CONFIG***
IC	MIPEAK	***T***

VELOCITY PARAMETERS AND COMMANDS		
ACC	V	VMAX
DEC	VCMD	VOSPD
DECSTOP	VE	VSCALE
ILSBMODE	VEXT	***J***
MSPEED	VLIM	***S***
		STOP

MOTION CONTROL PARAMETERS		
ACC	DECSTOP	PROFMODE
CCWLIM	DIR	STOPMODE
CWLIM	LIMDIS	
DEC	OPMODE	

LOOP COMPENSATION AND GAINS		
BW	LPFHZ1	TF
COMPMODE	LPFHZ2	VD
FILTMODE	MJ	VEXT
GV	MLGAINP	VH
GVI	MLGAINZ	VR
KV	MTANGLC	***TUNE***
KVI	MTANGLP	***REFRESH***
KVFR	MVANGLH	
LMJR	MVANGLF	

VARIABLE RECORDING AND PLAYING		
AVGTIME	RECING	***GET***
GETMODE	RECRDY	***RECORD***
RECDONE	RECTRIG	***RECOFF***
		STEP

USER I/O		
ANOUT	IN1MODE	O1MODE
IN1	IN2MODE	O1RST
IN2	IN3MODE	O1TRIG
IN3	O1	

SECTION 7

TROUBLESHOOTING

7.1 INTRODUCTION

The information in this section will enable you to isolate and resolve common system problems. The SERVO**STAR** aids in diagnostic evaluation through its Seven-Segment Display Indicator and the SERVO**STAR** Error Log. Both of these features are explained to assist you in finding solutions. As another part of Industrial Drives' obligation to it's customers, Factory Support and Repair is also defined.

The SERVO**STAR** control system uses high voltage and performs mechanical motion. Potential danger exists. Electrocution and personal injury are possible. Refer all troubleshooting to qualified personnel!

7.2 POWER UP TESTS

The SERVO**STAR** performs a series of power up tests when power is first applied. Upon completion of the tests, the status display will assume a numerical display indicating the OPMODE for which it is configured. If the event of the power up test failure the status display will indicate an error. Many error conditions can be easily diagnosed using the indication.

7.3 LED DISPLAYS

7.3.1 Status Display

The SERVO**STAR** has a Seven-Segment Display Indicator that indicates errors. The Display has four types of states: Power-up, Steady State, Flashing State, and Momentary State.

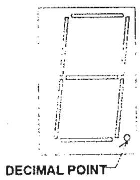
SEVEN-SEGMENT LED DISPLAY	
DRIVE STATE	**DISPLAY APPEARANCE**
Power-up	Momentarily illuminates all display segments (forming the number 8) and the decimal point.
Steady State (No Faults)	Displays the operational mode (opmode) of the drive (0-8).

Flashing State	Used to indicate an abnormal operating state: 1. If the position hold feature is active, the opmode number will flash at a 1 Hz rate. 2. If a fault was detected, a flashing code will be displayed to identify the fault. Some codes consist of a sequence of two or more digits (see below). In general, these faults will cause a latched disable (sometimes controllable through software switches). To clear fault, toggle remote enable (except for Overcurrent). 3. If the encoder initialization function (ENCSTART) is active, the opmode number will flash at a 3 Hz rate.
Momentary Fault	Displays a character momentarily for 500 ms before returning to the steady state. The timer is resetable. C = Communications Error F = Drive is in Foldback mode

7.3.2 Decimal Point

The decimal point directly relates to the global drive enable. The states are defined as follows:

LED DISPLAY DECIMAL POINT	
DECIMAL POINT STATE	**DRIVE STATUS**
Steady OFF	No power to the motor
Steady ON	Drive enabled, power to the motor
Flashing	Drive enabled, power to the motor, but a motor safety feature has been disabled (for example, LIMDIS = 1).



WARNING

A flashing decimal point in the seven segment display indicates the drive is enabled with a safety feature disabled.

7.3.3 Discrete LED's

All SERVOSTAR's power supply sections have discrete LEDs. The 8 Amp Power Supply has a single Bus LED. The 28A Power Supply has an additional Regen Active LED. The 50 and 75 Amp models have a total of 4 LEDs. The purpose of the LEDs are defined as:

D.C. BUS - This green LED turns on when the Power Supply bus voltage is above about 30 Vdc.

REGEN - This yellow LED turns on when the Power Supply Regen circuit is actively dumping power.

OVERLOAD - This LED is only on the 50 and 75 Amp Power Supply Units. This red LED indicates a hardware fault. It turns on when the Regen circuit has absorbed too much energy.

POWER SUPPLY FAULT - This red LED indicates that the power supply soft start circuitry is disabled. The soft start circuit requires that the power supply's internal heat sink not be overtemperature, control voltage is on, and the High Voltage Bus Power is applied.

7.4 ERROR CODES

In addition to the Status Display and LEDs the SERVOSTAR also sends error codes with a text message to the serial port. The same message is saved in non-volatile memory in an error history log. (See ERR.)

7.4.1 Error Levels

The **SERVOSTAR** 's response to an error depends on the error's severity. There are three levels of severity as listed below:

1	Errors which cause warnings. These are called Errors.
2	Errors which disable the system and indicate fault status. These are called Faults.
3	Errors which disable almost all SERVOSTAR functions (including communications). These are called Fatal Errors.

Error Severity Levels and Actions

7.4.2 Fault Log

In addition to the Status Display, LEDs, and the serial notification of errors, the SERVOSTAR also records the error message in non-volatile memory for later interrogation. Each entry into the error log has an associated error code number as defined in the tables below. The SERVOSTAR stores the ten most recent faults in the Fault History Buffer. To display the entire Fault History, type:

FLTHIST

This causes the Fault History to be sent out the serial port to the terminal, with the most recent fault sent first.

To clear the Fault History, type:

FLTCLR

The Fault History Buffer remains intact even through power-down.

To display the last error generated type:

ERR

The SERVOSTAR responds with the last (most recent) fault that is in the Fault History and was not transmitted previously, and the last error.

7.4.3 Fatal and Non-Fatal Error Codes

Table 7-2. Error Codes

Error Code	Fault Message	Fatal	Non-Fatal	LED Display
0	No Error		x	
1	Power stage over temperature	x		t
2	Over voltage	x		o
3	Over current	x		P
4.1	Resolver line break	x		r 1
4.2	RDC error	x		r 2
4.4	A/B line break	x		r 4
4.5	Index line break	x		r 5
4.6	Illegal halls	x		r 6
5	Under voltage	x		u
6	Motor over temperature	x		H
7.1	Positive analog supply fail	x		A 1
7.2	Negative analog supply fail	x		A 2
8	Over speed	x		J
9	EEPROM failure	x		E
10	EEPROM checksum failure	x		e
12	Foldback	x		F
14.1	Positive over travel fault	x		d 5
14.2	Negative over travel fault	x		d 6
15.1	Numeric position deviation	x		d 1
15.2	Excessive position deviation	x		d 2
16	Communication interface	x		c
20	Unknown command		x	
21	Unknown variable		x	
22	Checksum error		x	
23	Drive active		x	
24	Drive inactive		x	
25	Value out of range		x	
26	Value negative		x	
27	Not in proper Opmode		x	
28	Syntax error		x	
33	Auto tune failed		x	
34	Bad bandwidth		x	
35	Bad stability		x	
36	Not programmable		x	
37	Invalid configuration		x	
38	Communication error		x	
39	Not in proper Compmode		x	
40	EXT velocity parameter warning		x	
41	Velocity loop design failed		x	
42	Invalid EEPROM		x	
43	Recording active		x	
44	Record data not available		x	
45	EEPROM is empty		x	
46	Argument must be binary		x	
47	Burn-in is active		x	

48	Burn-in is not active		x	
49	Conflicts with ENCOUT		x	
50	Conflicts with VLIM		x	
51	Not available		x	
52	Drive is in hold mode		x	
53	Limit switch hold		x	
54	Command into limit		x	
55	Drive is in Zero mode		x	
56	Motor is jogging		x	
57	Argument not divisible by 20		x	
58	Encoder initialization process active		x	
60	Tune failed - No rotation		x	
62,66,70,74	Tune failed - Current sat		x	
63,67,71,75	Tune failed - No velocity design		x	
76	Disable during Tune		x	
77	Hold during Tune		x	
78	Low velocity limits		x	
79	Use lower bandwidth		x	
80	Drive is in Dual Feedback mode		x	
81	Drive is in Gear mode		x	
82	Functionality is occupied		x	
83	Warning: A/B line not routed		x	
84	Warning: Limit switch not routed		x	
90	Incorrect password		x	
91	Password protected		x	

7.4.4 Fatal and Non-Fatal LED Display codes: (non-error message faults)

Fault	Fatal	Non-Fatal	LED Display flashing	LED Display steady
Watchdog (DSP)	x		≡	
Watchdog (HPC)	x			≡
No Compensation	x		-	
CW limit switch open		x	L 1	
CCW limit switch open		x	L 2	
CW and CCW limit switches open		x	L 3	
± analog supply failure	x		A 3	
RAM failure (during init)	x			I
EPROM checksum (during init)	x			c
Altera load fail (during init)			e 1 0 1	
Altera DPRAM failure (during init)			e 1 0 2	
DSP load fail (during init)			e 1 0 3	
DSP alive failure (during init)			e 1 0 4	

* These faults can only be cleared by cycling power

7.5 RETURNING TO FACTORY SET STATE

Modifying variables and operational characteristics can create problems because due to too many changes to back-track and trouble-shooting communications impairment. The SERVO**STAR** can easily be returned to the state as it was shipped from the factory. Simply execute the SERVO**STAR** Auto-Setup function in MotionLink by selecting "New Drive 1-2-3" from MotionLink's Config menu on the MotionLink main display screen.

If a SAVE command has not been issued since the drive was shipped from the factory, cycling power will return the SERVO**STAR** to the factory set state.

Refer to Section 3 for complete information on drive setup and configuration.

7.6 FACTORY SUPPORT AND REPAIR POLICIES

Industrial Drives is committed to helping you install, operate, maintain and troubleshoot your SERVO**STAR** servo system. If your SERVO**STAR** did not pass the "Initial Check Out" tests or is not operating properly, then contact the Field Service Department of Industrial Drives. Be prepared to provide the full SERVO**STAR** Amplifier, Power Supply, and Motor model numbers listed on the front of your SERVO**STAR** and Power Supply.

The SERVO**STAR** is a programmable product and is shipped from Industrial Drives at all times with internal settings that are factory default. Any user modified variables will no longer be present in the drive when it is returned to the user from Industrial Drives' repair department. For this reason it is imperative that all users download a working copy of the SERVO**STAR**'s internal variables data to disk with the DUMP command, or by using MotionLink's Backup screen. Do not wait until there is a fatal error to save system variables as, depending on the nature of the problem, it may not be possible to download the variables at that time.

Contact us at:

Kollmorgen Customer Support Network

Telephone: (888) 774-KCSN (5276)
FAX: (540) 633-3100

SECTION 8

SERVOSTAR MOTIONLINK

8.1 INTRODUCTION

The information in this section will instruct you on how to use MotionLink for SERVO**STAR**, the drive configuration program. MotionLink is a Windows-based program that installs on your PC and communicates with your SERVO**STAR** via an RS-232 connection between the PC and the SERVO**STAR.** MotionLink is highly intuitive in nature and has an extensive help file. Since the on-line help file provides detailed instructions, this section of the manual is limited to providing an overview of the SERVO**STAR** MotionLink.

MotionLink provides an extensive, context-sensitive on-line Help facility. The on-line Help describes how to use the program in detail and also serves as a valuable reference for drive commands, drive setup, and troubleshooting. Much of the information found in this manual and the pocket-sized "Quick Reference Guide" has been incorporated into the MotionLink on-line Help.

To access Help, press the F1 key on your keyboard while running MotionLink. This will call up the Help facility with a screen pertaining to your location in the MotionLink program. This is called "context-sensitive" help. To go to a general Contents screen for the Help file, access the Help | Contents menu on the main display in MotionLink. This will present the contents and structure of the Help file to you, so you can get an overview of the information that is accessible in the Help facility.

In order to communicate with a SERVO**STAR**, MotionLink is not required, but it is highly recommended. The other alternative is to use a "dumb terminal" to interface with the drive over an RS-232 communications port, but MotionLink is much faster and easier to use than a terminal interface. Computer requirements, software installation, commands, variables, and disk operations are discussed as well in this section.

8.2 COMPUTER REQUIREMENTS

MotionLink requires an IBM-PC or compatible computer with the following features:

- IBM-PC, XT, AT, 386, 486, PS/2, or compatible computer.
- 512 K RAM.
- Windows Version 3.1 or later.
- 3.5" Floppy Drive.
- Standard Video Adapter (CGA, MDA, EGA, MCGA, and VGA).
- Serial Port (for communication link with SERVO**STAR**). The serial communications port may be COM1, COM2, COM3, or COM4. COM1 is the normal configuration:

COM1: Address 3F8h, Interrupt Request #4
COM2: Address 2F8h, Interrupt Request #3
COM3: Address 3E8h, Interrupt Request #4
COM4: Address 2E8h, Interrupt Request #3

8.3 SOFTWARE INSTALLATION

Before installing or using MotionLink, make a copy of the MotionLink disk(s) that came with the SERVO**STAR**. This way if something happens to the master disk(s), you'll always have a copy. Remember, disks can be damaged by heat, magnets, pressure, and dirt — all extensively found in a manufacturing environment.

A. Installation Instructions for Windows:

1. Insert the SERVO**STAR** MotionLink for Windows diskette #1 into a 3.5" floppy drive.

2.1 Windows 3.1 users: in Program Manager, select File | Run.
2.2 Windows 95 users: from the desktop, select Start | Run.

3. Type "a:setup.exe" (or "b:setup.exe" if the MotionLink diskette is inserted in your b: drive) and hit the Enter key.

4. The SERVO**STAR** Setup screen will be displayed. Click on "OK" to start installation.

5. Select Destination Directory . You may install the SERVO**STAR** MotionLink for Windows in any directory or drive, but the default that is presented during installation is highly recommended.

6. Click on "OK" to begin installation. The SERVO**STAR** MotionLink program for Windows will be installed.

7. Click on "OK" to create a "KMTG Motion Suite" Program Group in Program Manager. You may also select an existing group, or create a new group with a different name, in which you would like the SERVO**STAR** MotionLink program icons to appear, but the remainder of this procedure assumes that you will install the program in a program group named "KMTG Motion Suite."

B. Running the Program:

Windows 3.1:

1. From Program Manager, double click on the "KMTG Motion Suite" program group icon.
2. In the "KMTG Motion Suite" program group, double-click on the "SERVO**STAR** MotionLink" icon.

Windows 95:

1. Select Start → KMTG Motion Suite → SERVO**STAR** MotionLink

8.4 MOTIONLINK DESCRIPTION

8.4.1 Overview

MotionLink is a full-featured communications program written by Kollmorgen and designed especially for the SERVO**STAR**. MotionLink turns an IBM-compatible PC into a smart terminal. The user can select from a variety of graphical methods to control the SERVO**STAR,** or a mock terminal screen can be called up in MotionLink, and commands can be typed in on the computer as if a terminal were being used. MotionLink provides "user-friendly" features such as mode setup block diagrams which facilitate setting the system operating parameters required for your application.

MotionLink's features are selected by clicking on selection boxes, topic tabs, and buttons.

8.4.2 On-Line Help

As described previously, MotionLink provides an extensive, context-sensitive on-line Help facility. The on-line Help describes how to use the program in detail and also serves as a valuable reference for drive commands, drive setup, and troubleshooting. To access Help, press the F1 key on your keyboard while running MotionLink. This will call up the Help facility with a screen pertaining to your location in the MotionLink program. This is called "context-sensitive" help. To go to a general Contents screen for the Help file, access the Help | Contents menu on the main display in MotionLink. This will present the contents and structure of the Help file to you, so you can get an overview of the information that is accessible in the Help facility.

8.5 USING MOTIONLINK TO SET UP THE SERVOSTAR

Experience shows that most setup will be done on-line: that is, under actual production conditions. The easiest way to set up a drive using MotionLink is to execute the "SERVO**STAR** Auto-Setup" function in MotionLink, which executes automatically every time MotionLink is started up (the automatic execution of the Auto-Setup function can be disabled, if desired).

This section describes how to execute MotionLink's Auto-Setup function. The topic of setting up a drive is covered in detail in section 3 of this manual. "Screen captures," or graphic depictions of the MotionLink displays, are used in this section to illustrate the appearance of the MotionLink program. Note that due to the constantly evolving state of software, your MotionLink screens may not appear identical to the screens shown in this section. The basic steps and order of execution should still apply, however. Should you require any information beyond what is presented in this manual, access MotionLink Help by pressing F1 at any time.

8.5.1 Start-up Instructions

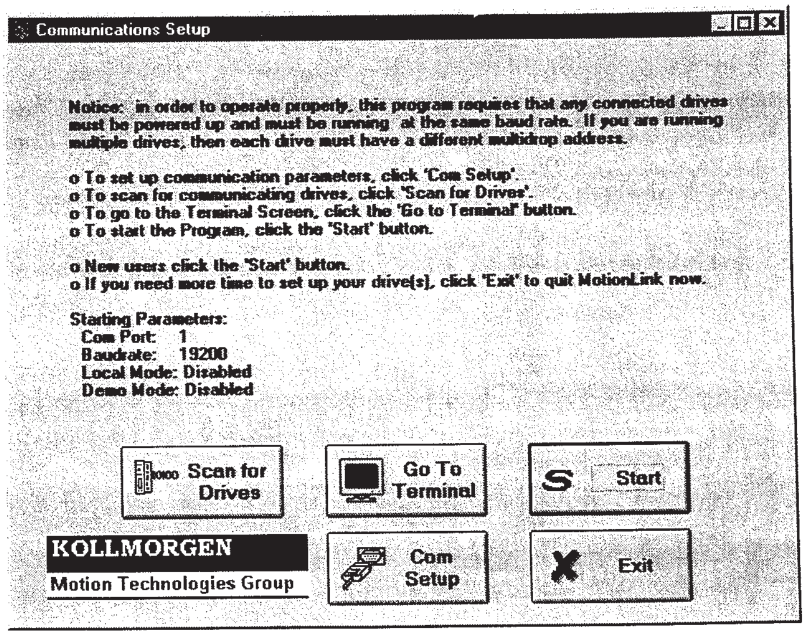
(Note: The SERVO**STAR** must be properly connected and powered up to perform this procedure)
Perform the following steps to run the SERVO**STAR** MotionLink program:

1. a. **Operator Action:** Open the MotionLink program by double-clicking the "ServoStar MotionLink" icon.

   b. **MotionLink Screen Display:** Introduction / About Screen

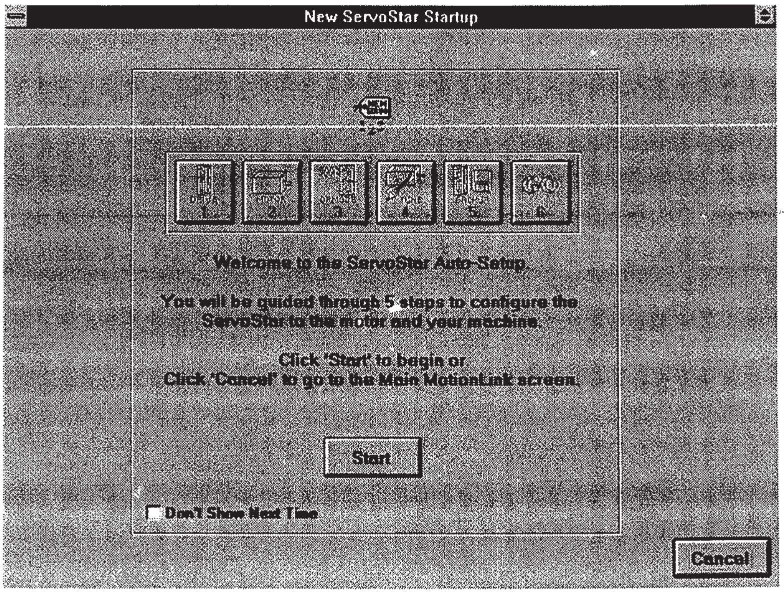
2. a. **Operator Action:** Click on the **OK** button.

   b. **MotionLink Screen Display:** SERVO**STAR** Communications Setup Screen.



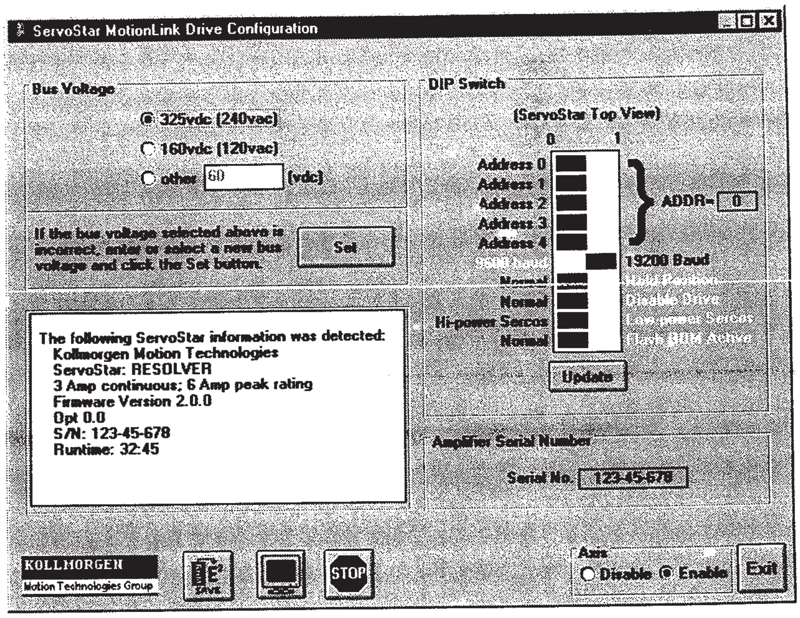
3. a. **Operator Action:** Make sure that your SERVO**STAR** drive is properly connected to your PC via the serial port, and that the SERVO**STAR** is powered up and ready to communicate. Click **Start**. MotionLink will display a drive scan box and will attempt to communicate with your drive by scanning the serial port. If scanning is unsuccessful, you will be presented with several options. If scanning is successful and contact is made, you will see the following screen.

   b. **MotionLink Screen Display:** New SERVO**STAR** Startup Screen.



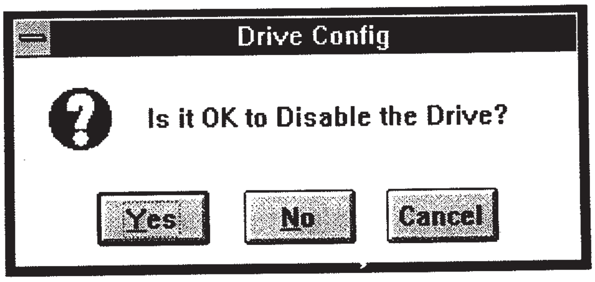
4. a. **Operator Action:** For a first time setup, click on the **Start** button.

   b. **MotionLink Screen Display:** Drive Configuration Screen



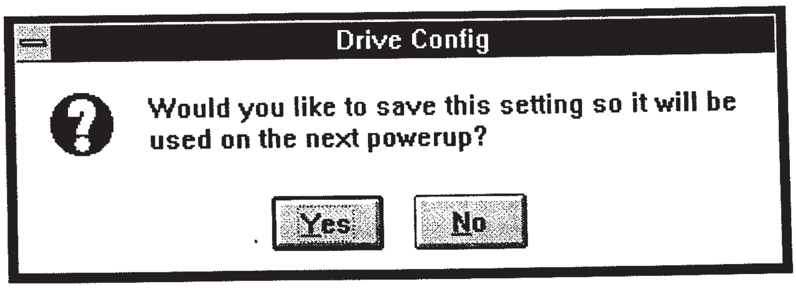
5. a. **Operator Action:** Select appropriate DC Bus Voltage (325 or 160), or click on the **Other** button and enter a value in the box. Click on **Set** button.

   If the drive is enabled when you attempt to set the voltage, the following dialog box will be displayed:



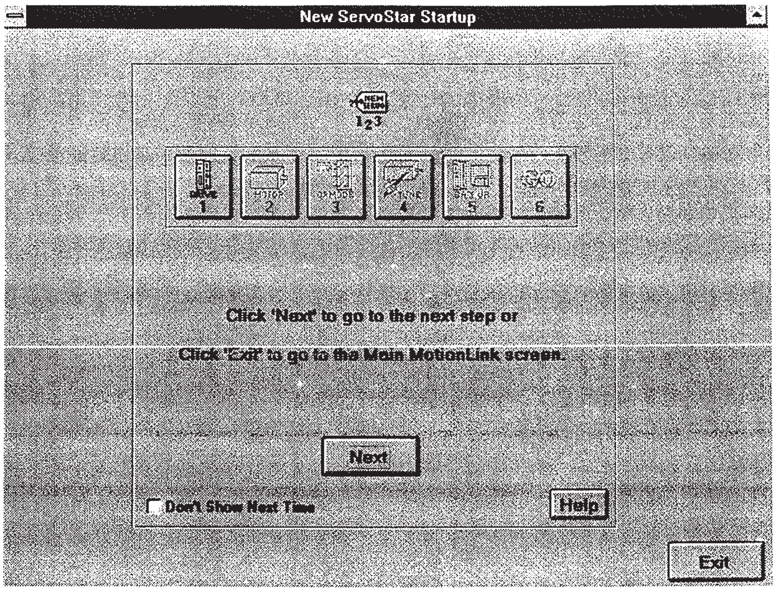
Click on Yes button.

   b. **MotionLink Screen Display:** Drive Config dialog box



6. a. **Operator Action:** Click on Yes button. Click on Exit button.

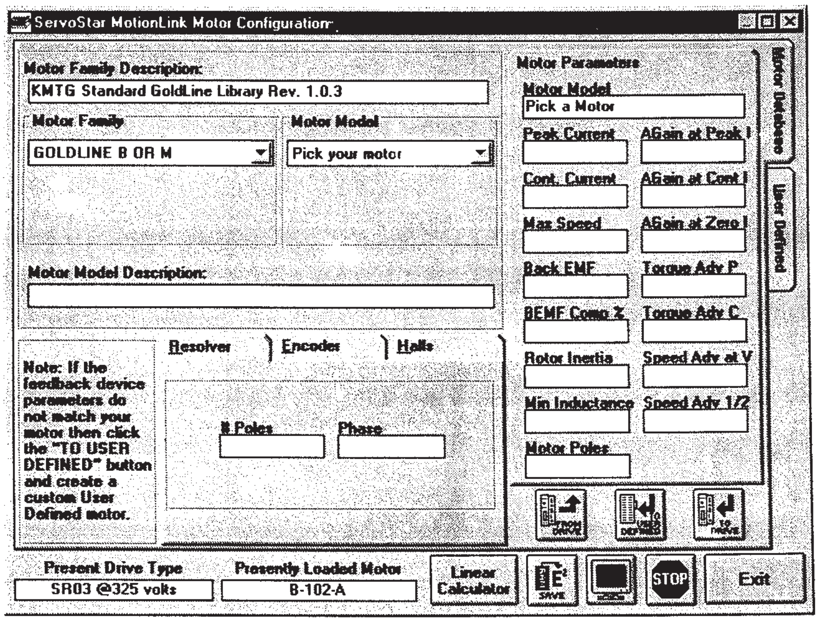
b. **MotionLink Screen Display(s):** New ServoStar Startup Screen



7. a. **Operator Action:** Click on **Next** button.

   b. **MotionLink Screen Display(s):** Motor Configuration Screen(s)

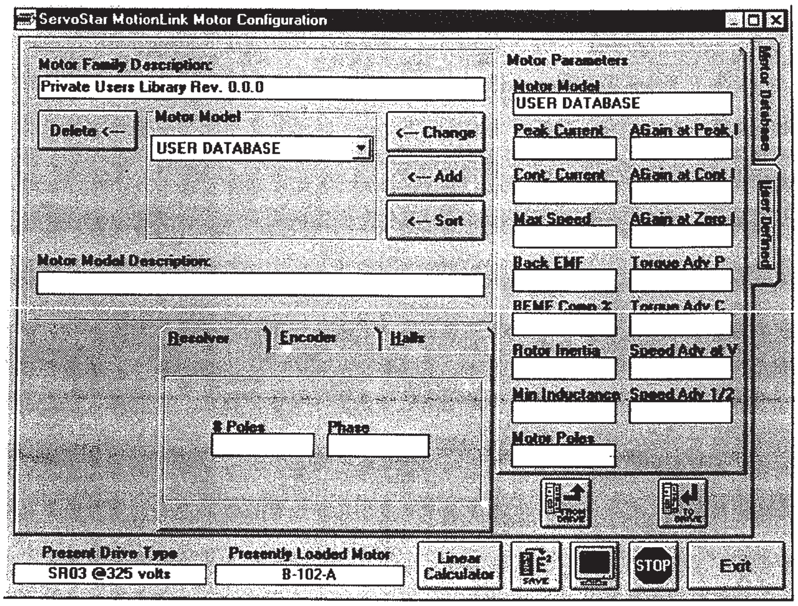
   A. **Motor Database** Tab - Allows the user to select the motor to be used from the Kollmorgen Motion Technologies Group Motor Families.



a. **Operator Action:** Select the Motor Family and Motor Model # you will be using in your application.

b. Proceed to step C.

B. **User Defined** Tab - Allows the user to define motor parameters and create a private motor database. (See Appendix D for more detail)

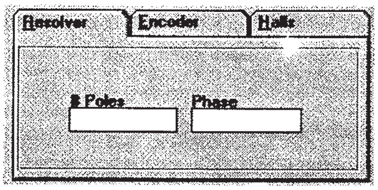


a. **Operator Action:** Define the Motor you will be using in your application by entering values in the fields on the page.

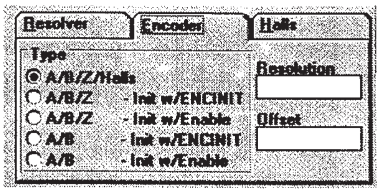
b. Proceed to Step C.

C. Additional **Motor Configuration** Tabs - Allow the user to define parameters and create a motor configuration database for Resolver, Encoder, and Halls.

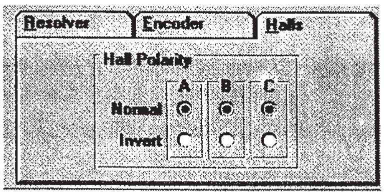
1. **Resolver** Tab



2. **Encoder** Tab

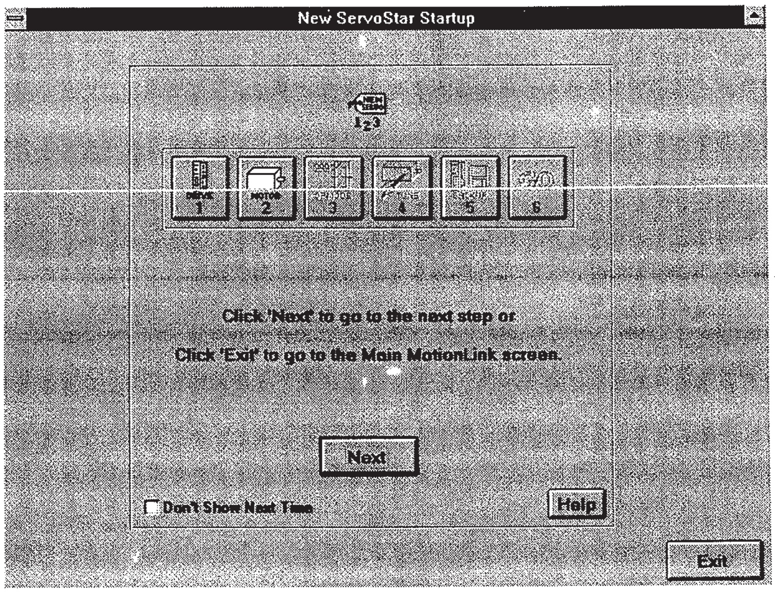


3. **Halls** Tab



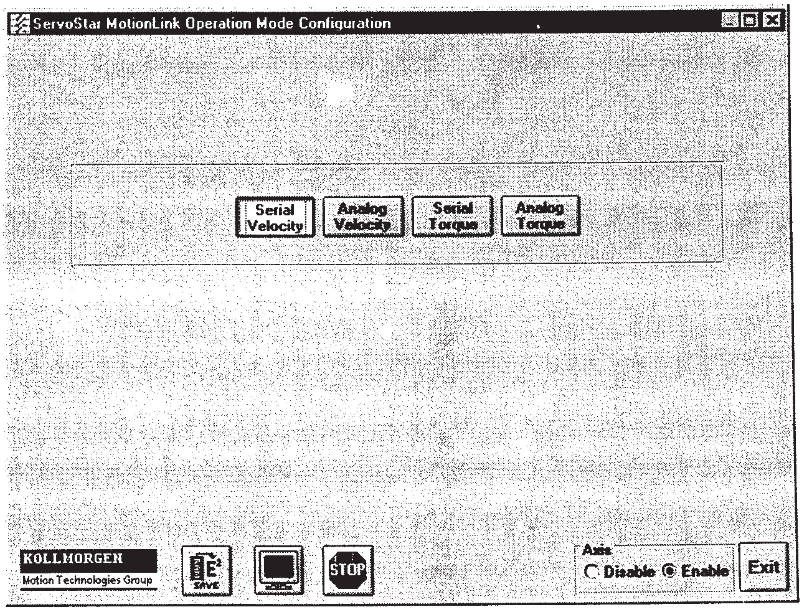
8. a. **Operator Action:** When Motor Configuration Data is entered. Click on the TO DRIVE button. Click on the Exit button.

b. **MotionLink Screen Display(s):** New ServoStar Startup Screen



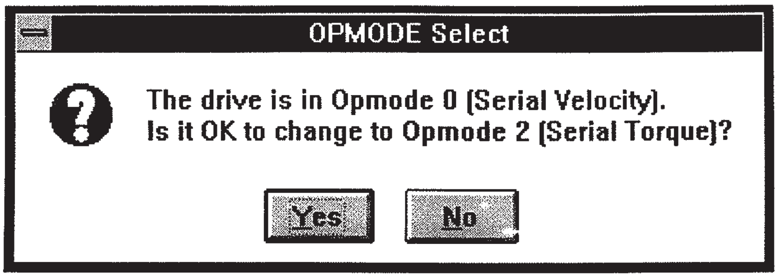
9. a. **Operator Action:** Click on Next button.

b. **MotionLink Screen Display:** Operation Mode Configuration Screen



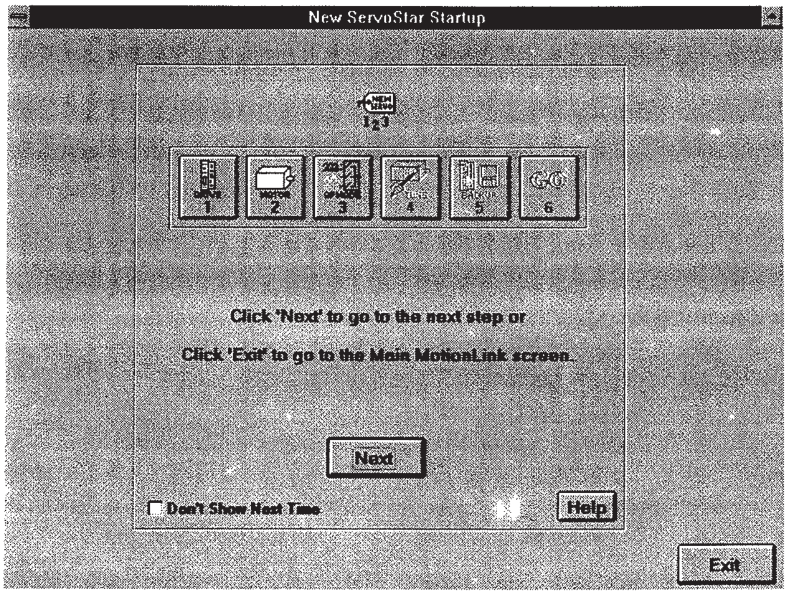
10. a. **Operator Action:** Select an Operation Mode Configuration for the drive.

If you are going to change opmodes, a screen (similar to the following) will be displayed:



11. a. **Operator Action:** Click on Yes button. Click on Exit button.

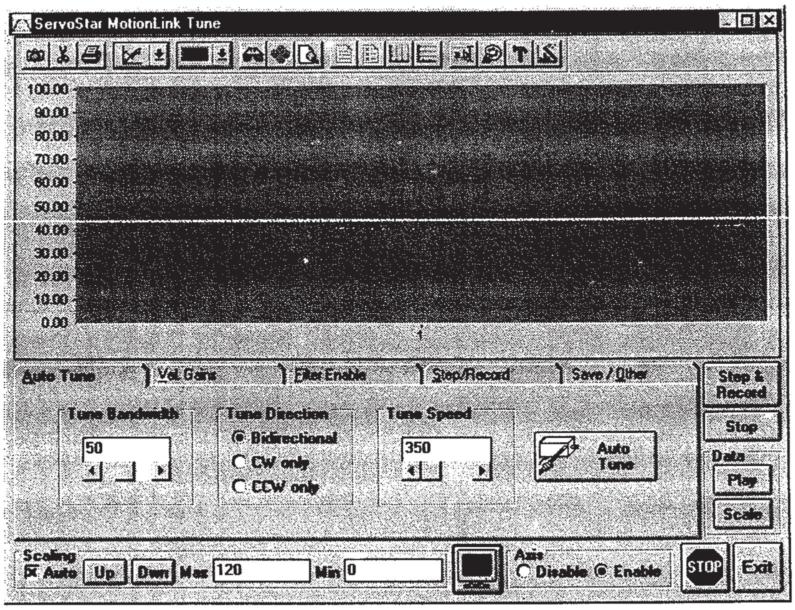
b. **MotionLink Screen Display(s):** New ServoStar Startup Screen



12. a. **Operator Action:** Click on Next button.

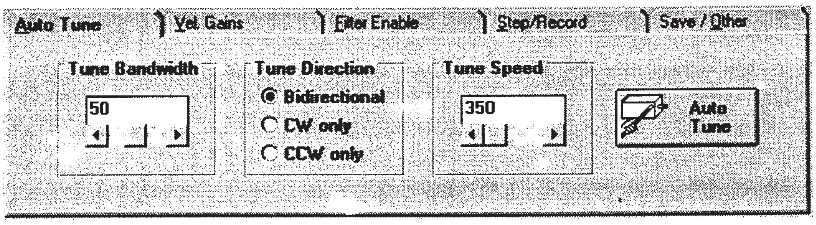
b. **MotionLink Screen Display:** Tune Screen

A. **Tune** Screen

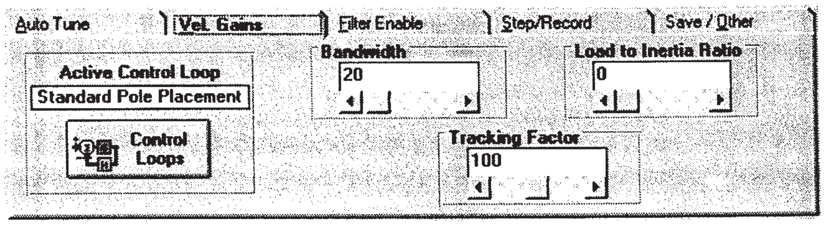


B. **Tune** Screen Tabs - Allow the user to select and define parameters for:

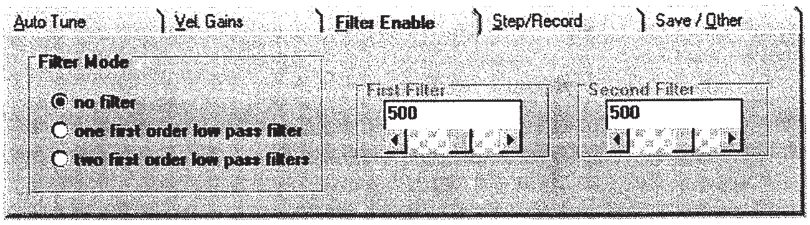
1. **Auto Tune** Tab



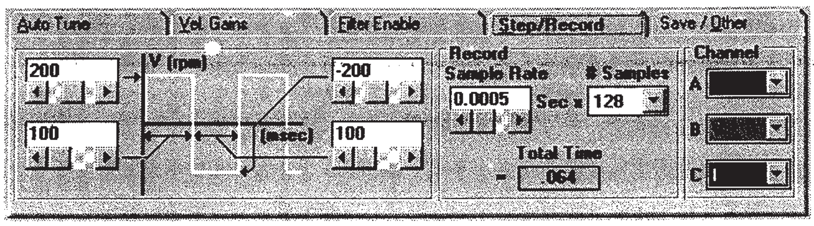
2. **Velocity Gains** Tab



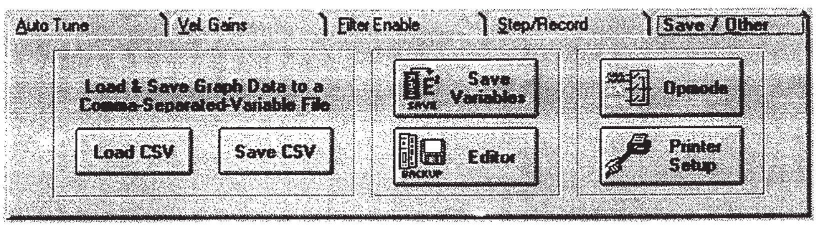
3. **Filter Enable** Tab



4. **Step & Record** Tab

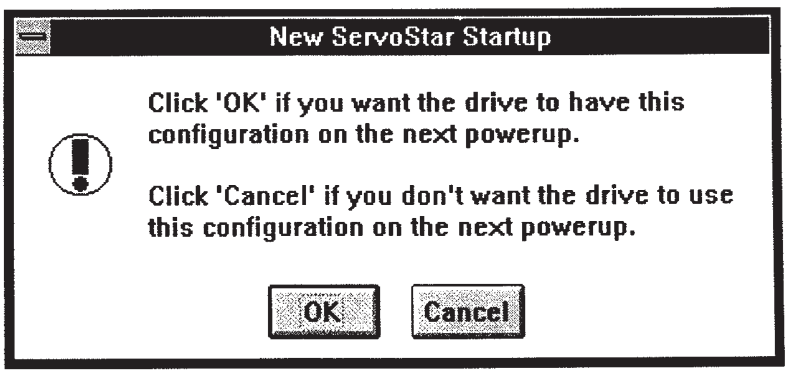


5. **Save/Other** Tab



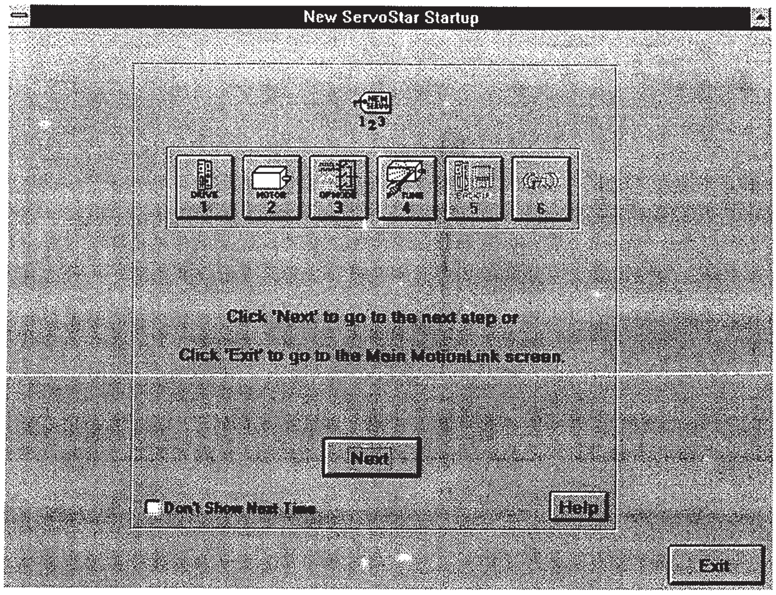
13. a. **Operator Action:** Tune the drive. (Auto Tune or Manual Tune; the MotionLink Help file contains detailed instructions on how to use the Tune Screen to tune the drive). Click on the Exit button.

b. **MotionLink Screen Display:** Save to EEPROM dialog box.



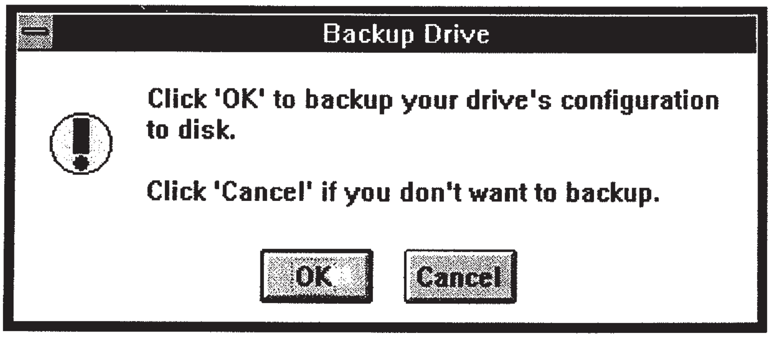
14. a. **Operator Action:** Click on OK button to save this drive configuration to EEPROM

b. **MotionLink Screen Display:** New ServoStar Startup Screen.



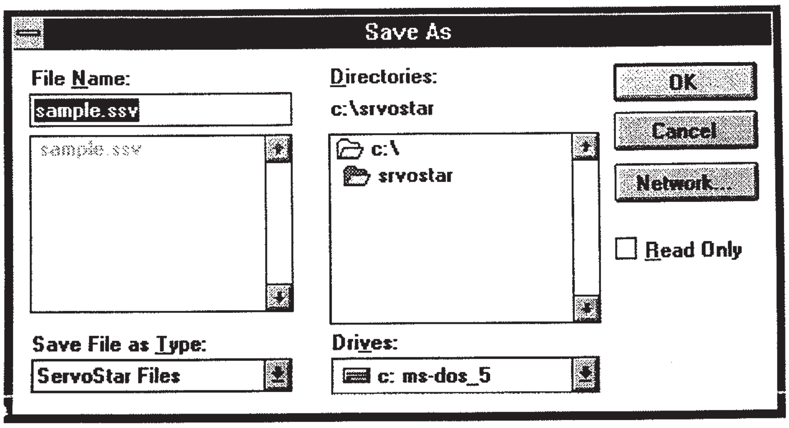
15. a. **Operator Action:** Click on Next button.

    b. **MotionLink Screen Display:** Backup Drive dialog box.



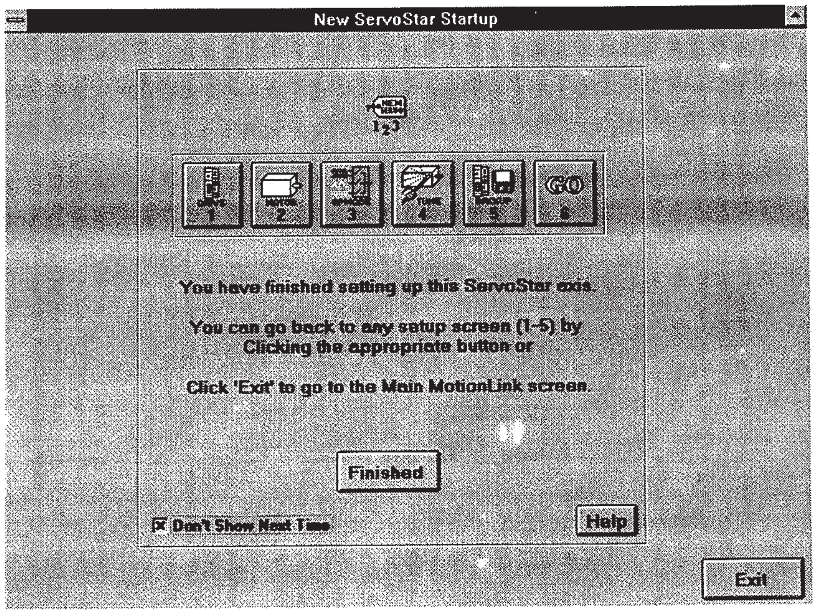
16. a. **Operator Action:** Click on OK button to save the drive configuration to a computer file.

    b. **MotionLink Screen Display:** Save As dialog box.



17. a. **Operator Action:** Assign a Directory and File Name to save the drive setup configuration. Click on the OK button to save this drive configuration to disk.

b. **MotionLink Screen Display:** New ServoStar Startup Screen.

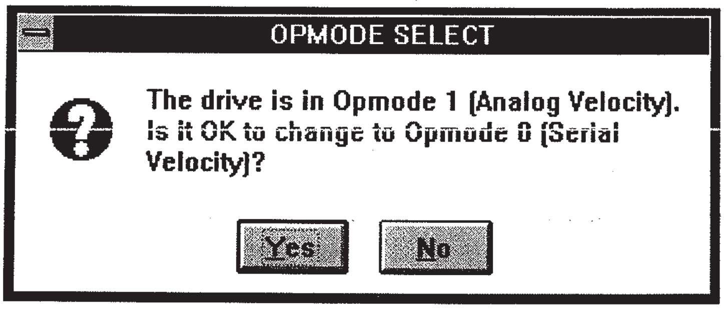


18. a. **Operator Action:** Click on **Don't Show Next Time** so that an **"X"** appears in the box to the left. This will disable the New ServoStar Startup function so that it doesn't automatically appear the next time you run the MotionLink program. Click on the Exit button to enter the MotionLink Main Display Screen (the appearance of the Main Display Screen will vary depending upon what OPMODE the drive is in. The different screens are shown in paragraphs 8.5.2.2 through 8.5.2.5).

8.5.2 Operation Modes

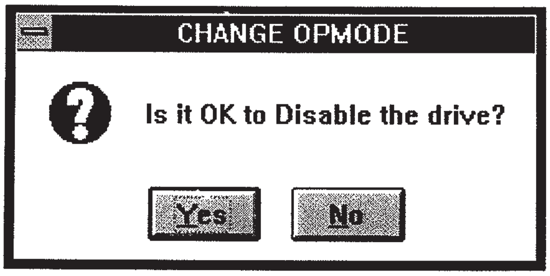
8.5.2.1 General Information

When operating the SERVO**STAR**, the unit must be disabled before an operation mode (OPMODE) can be changed. When an OPMODE change is requested, messages similar to the following will be displayed:



Click on the Yes button to implement the change.

b. If an opmode change request is made with the drive enabled, the following screen will be displayed:



Click on the Yes button to implement the change.

After the MotionLink Start-up procedure has been completed, SERVO**STAR** operation in any of four operation modes can be selected:

OPMODE	DEFINITION
0	Serial Velocity Mode
1	Analog Velocity Mode
2	Serial Current (Torque) Mode
3	Analog Current (Torque) Mode

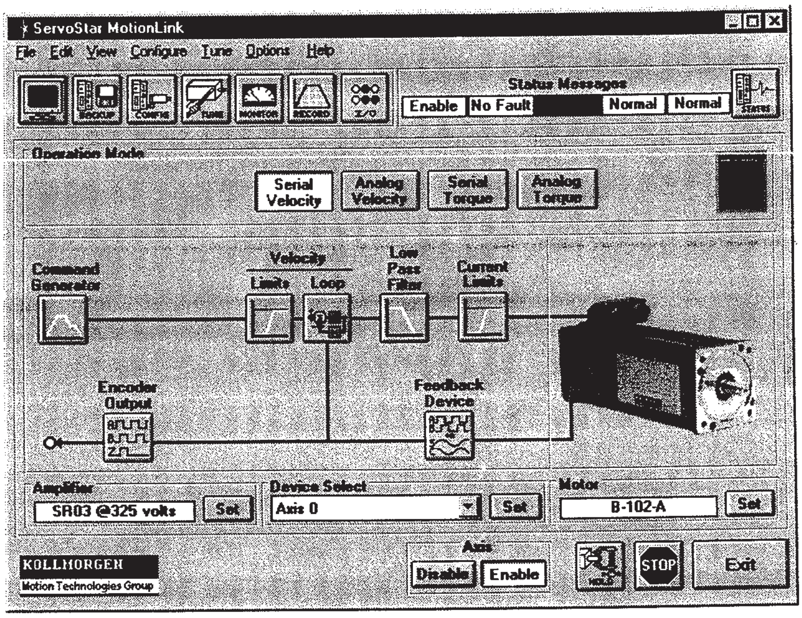
When the SERVO**STAR** is operating, the seven-segment LED Status Display will indicate the OPMODE in which the drive is operating. Refer to section 4 for more information on the LED Status Display.

The following sections introduce the four MotionLink Main Display (Config) Screens that will be displayed, depending upon the selected OPMODE. Each Config Screen displays a control loop graphic consisting of blocks which may be clicked in order to set system control parameters associated with the block (Acceleration Limits, Velocity Limits, etc.).

8.5.2.2 Serial Velocity Mode - OPMODE 0

1. a. **Operator Action:** Click on the **Serial Velocity Operation Mode** button.

   b. **MotionLink Screen Display:** (Similar to the following)



Clicking on the buttons in the upper part of the screen will place the program on another screen. The screens which may be selected in this manner are as follows:

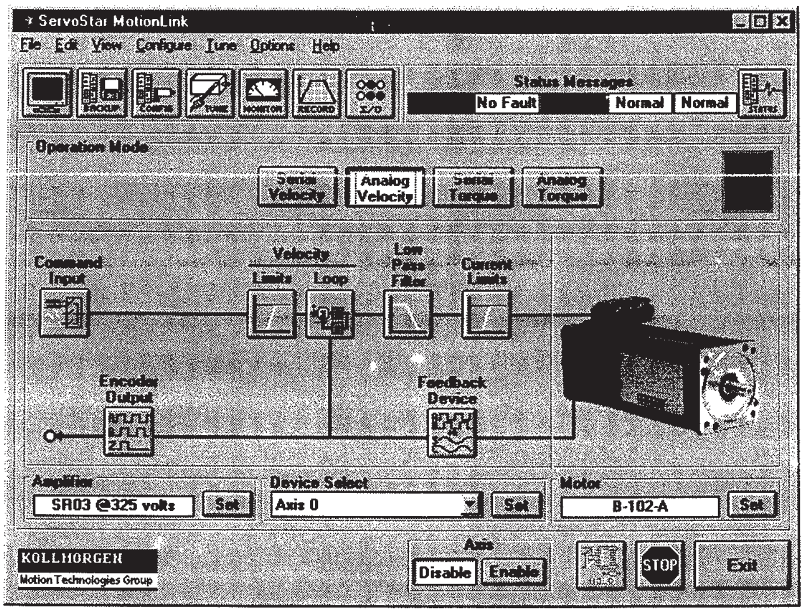
1.) Terminal Screen - emulates a dumb terminal
2.) Backup Screen - backs the drive configuration up to disk.
3.) Config Screen - since this screen is the Config Screen, this button has no function.
4.) Tune Screen - enables the operator to tune the drive.
5.) Monitor Screen - enables the operator to observe system variables "real-time."
6.) Record Screen - enables the operator to record drive data and display it.
7.) I/O Screen - enables the operator to view and set discrete I/O values.
8.) Status Screen - enables the operator to view drive status, fault status, and fault modes.

Clicking on the **DISABLE, ENABLE,** and **HOLD** buttons, respectively, will select the OPMODE 0 status. From the Enabled Mode, each successive click of the **HOLD** button will alternately cause the motor to "Stop Motion and Hold Position" and "Resume Normal Operation".

8.5.2.3 Analog Velocity Mode - OPMODE 1

1. a. **Operator Action:** Click on the **Analog Velocity Operation Mode** button.

   b. **MotionLink Screen Display:** (Similar to the following)



Clicking on the buttons in the upper part of the screen will place the program on another screen. The screens which may be selected in this manner are as follows:

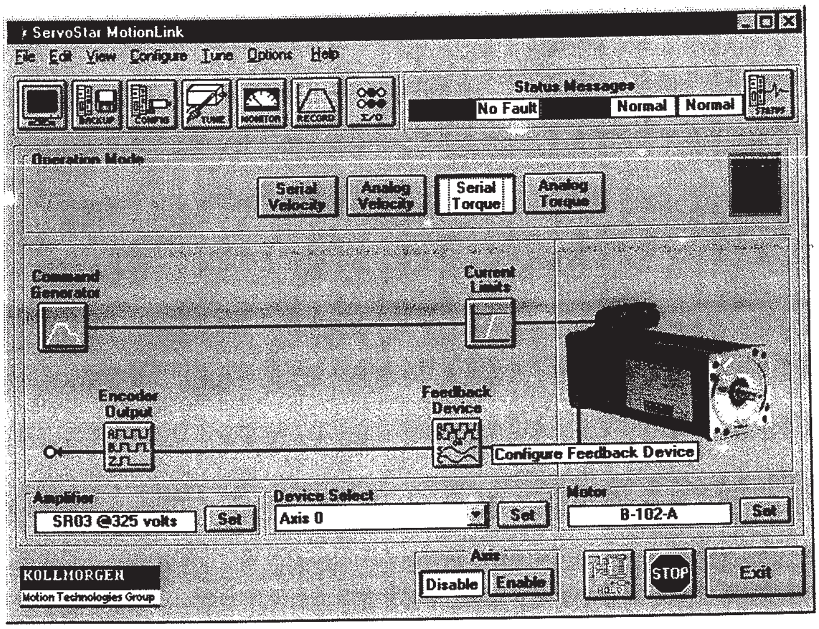
1.) Terminal Screen - emulates a dumb terminal
2.) Backup Screen - backs the drive configuration up to disk.
3.) Config Screen - since this screen is the Config Screen, this button has no function.
4.) Tune Screen - enables the operator to tune the drive.
5.) Monitor Screen - enables the operator to observe system variables "real-time."
6.) Record Screen - enables the operator to record drive data and display it.
7.) I/O Screen - enables the operator to view and set discrete I/O values.
8.) Status Screen - enables the operator to view drive status, fault status, and fault modes.

Clicking on the **DISABLE, ENABLE,** and **HOLD** buttons, respectively, will select the OPMODE 1 status. From the Enabled Mode, each successive click of the **HOLD** button will alternately cause the motor to "Stop Motion and Hold Position" and "Resume Normal Operation".

8.5.2.4 Serial Current Mode - OPMODE 2

1. a. **Operator Action:** Click on the **Serial Current Operation Mode** button.

   b. **MotionLink Screen Display:** (Similar to the following)



Clicking on the buttons in the upper part of the screen will place the program on another screen. The screens which may be selected in this manner are as follows:

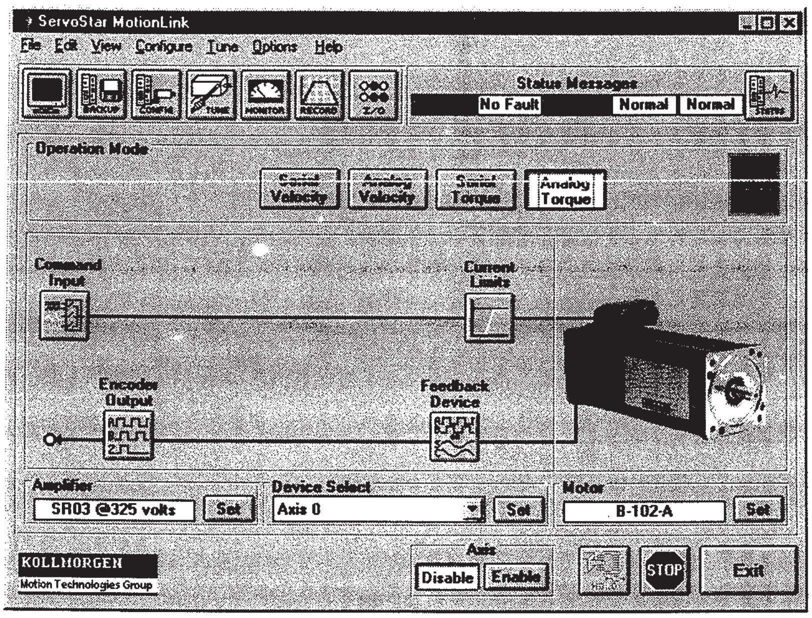
1.) Terminal Screen - emulates a dumb terminal
2.) Backup Screen - backs the drive configuration up to disk.
3.) Config Screen - since this screen is the Config Screen, this button has no function.
4.) Tune Screen - enables the operator to tune the drive.
5.) Monitor Screen - enables the operator to observe system variables "real-time."
6.) Record Screen - enables the operator to record drive data and display it.
7.) I/O Screen - enables the operator to view and set discrete I/O values.
8.) Status Screen - enables the operator to view drive status, fault status, and fault modes.

Clicking on the **DISABLE, ENABLE,** and **HOLD** buttons, respectively, will select the OPMODE 2 status. From the Enabled Mode, each successive click of the **HOLD** button will alternately cause the motor to "Stop Motion and Hold Position" and "Resume Normal Operation".

8.5.2.5 Analog Current Mode - OPMODE 3

1. a. **Operator Action:** Click on the **Serial Current Operation Mode** button.

   b. **MotionLink Screen Display:** (Similar to the following)



Clicking on the buttons in the upper part of the screen will place the program on another screen. The screens which may be selected in this manner are as follows:

1.) Terminal Screen - emulates a dumb terminal
2.) Backup Screen - backs the drive configuration up to disk.
3.) Config Screen - since this screen is the Config Screen, this button has no function.
4.) Tune Screen - enables the operator to tune the drive.
5.) Monitor Screen - enables the operator to observe system variables "real-time."
6.) Record Screen - enables the operator to record drive data and display it.
7.) I/O Screen - enables the operator to view and set discrete I/O values.
8.) Status Screen - enables the operator to view drive status, fault status, and fault modes.

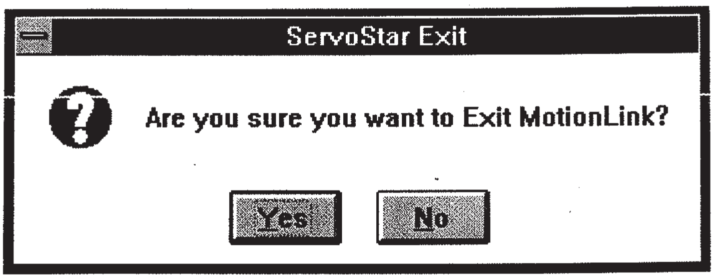
Clicking on the **DISABLE, ENABLE,** and **HOLD** buttons, respectively, will select the OPMODE 3 status. From the Enabled Mode, each successive click of the **HOLD** button will alternately cause the motor to "Stop Motion and Hold Position" and "Resume Normal Operation".

8.5.3 Exiting MotionLink

To exit from the MotionLink program, from any Operation Mode Screen:

1. Click on the Exit button.

   Motion link screen will display



2. Click on the Yes button.

SECTION 9

VELOCITY CONTROLLERS

9.1 INTRODUCTION

The SERVOSTAR offers the user a choice of four velocity controller algorithms (filters):

- **PI** controller (Proportional & Integral) (COMPMODE = 0)
- **PDFF** controller (Pseudo Derivative Feedback + FeedForward) (COMPMODE = 1)
- **PP** controller (Pole-Placement) (COMPMODE = 2)
- **APP** controller (Advanced Pole-Placement) (COMPMODE = 3)

Either controller may be selected for use in the velocity loop (OPMODE = 0 or 1). The position loop has an additional proportional gain controller with feed-forward.

Throughout this section, a discrete system representation approach is taken, since we deal with digital control systems.

9.2 STRUCTURE OF CONTROLLER

The controller structure for the four controller types is shown below:

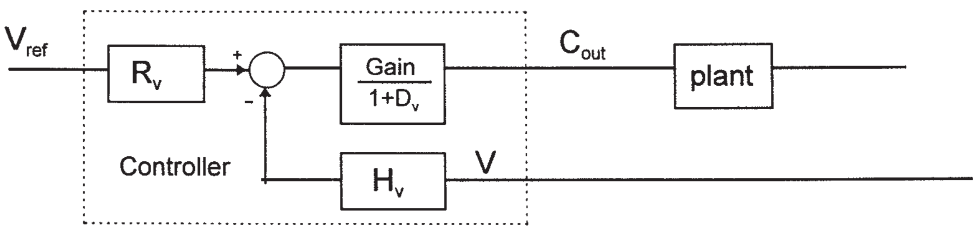


Figure 9-1. Controller Structure

The controller is realized as polynomial in a 32 bit accuracy. It has three main blocks:

- R_v -The pre-filter polynomial.
- H_v - The feedback polynomial.
- $1+ D_v$ - The forward polynomial.

In addition, it has saturation blocks and an anti-windup mechanism which prevents overshoot and oscillation in the case of command saturation.

9.3 TYPES OF CONTROLLERS

9.3.1 PI Controller (COMPMODE = 0)

The PI velocity controller structure is shown below:

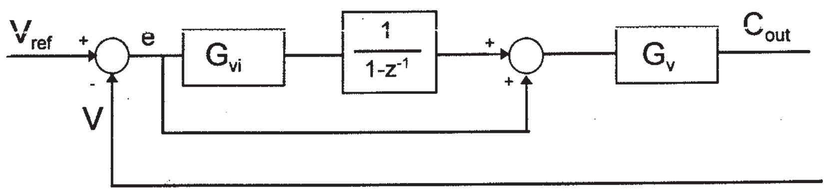


Figure 9-2. PI Controller Configuration

The PI controller has a unity feedback with no pre-filter. The control algorithm is as follows:

$C_{out} = G_v * [\ 1 + G_{vi} /(1\text{-}z^{-1}\ )\ ] * e \qquad e = V_{ref} - V$

The proportional gain term G_v stabilizes the system, while the integral term G_{vi} compensates for the steady state error. The PI controller offers set-point tracking capability, as the algorithm is entirely error driven. It also tends to provide middle range bandwidth which again, permits fair set-point tracking capability.

Both parameters, G_v and G_{vi}, are tuned by trial & error techniques. These terms are represented by the variables GV and GVI respectively, in the SERVO**STAR** system.

9.3.2 PDFF Controller (COMPMODE = 1)

The PDFF controller is illustrated below:

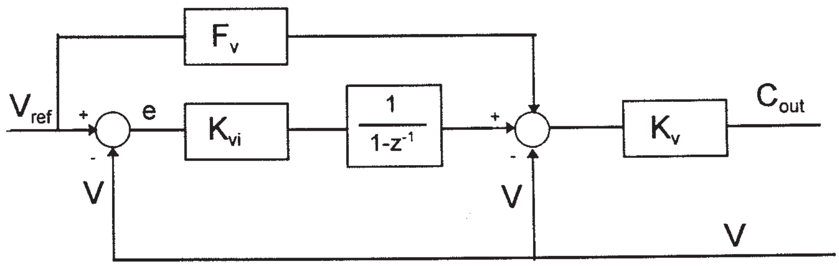


Figure 9-3. PDFF Controller Configuration

The PDFF controller is the familiar PDF controller with a feed-forward term added.

The differences between the PDFF and the PI control scheme are:

- The feedback term that is inserted between the integral and proportional blocks, by which the "pseudo-derivative" effect is achieved. This feedback is superior from load-noise rejection point of view.

- The feed-forward block, which gives the control system the flexibility to change its characteristics from a basic PDF controller (where the feed-forward term $F_v = 0$) to a PI controller ($F_v = 1$). F_v may vary between 0 and 1 and provides this controller structure its advantage of flexibility relative to PDF and PI controllers. Actually, the PDFF is a generalized controller where the PI and PDF controllers are simply two cases of the PDFF controller which are encountered at the limits of F_v.

The controller parameters - K_v , K_{vi} , and F_v are tuned by trial & error techniques. These terms are represented by the variables KV, KVI, and KVFR respectively in the SERVOSTAR system.

9.3.3 Standard Pole-Placement Controller (PP; COMPMODE = 2)

The PP controller parameters are calculated from the system inertia and the desired closed-loop bandwidth by using the Pole-Placement Design Method (PP). The calculation is based on the following equation:

$$\frac{V}{V_{ref}} = \frac{Gain * R_v * B}{(1+A)*(1+D_v) + Gain * B * H_v} = \frac{Gain * B}{(1+C)}$$

where R_v , H_v , and D_v are the controller polynomials and the term B/(1+A) represents the plant model, in which B is proportional to the inverse of the system inertia. (1+C) is the characteristic equation that is constructed from the closed-loop bandwidth (or poles) of the system and other predefined poles automatically calculated by the drives for maximum robustness, and maximum torque disturbances rejection. By specifying the moment of inertia $\cong$ 1/B and the closed-loop bandwidth - [dominant poles of (1+C)], the controller polynomial coefficients can be calculated automatically.

Therefore, only two parameters have to be specified for the controller tuning:

- The **Load Inertia Ratio** parameter (represented by the variable LMJR in the SERVOSTAR)
- The **Closed-loop system Bandwidth** (represented by the variable BW in the SERVOSTAR)

Note: It is not necessary to know the load inertia for the design. A very easy way to tune this parameter is described below and in paragraph 9.6.

The main difference between the PP controller and the PDFF controller is the order of the polynomials:

- The D_v polynomial is not just an integrator as in the PDFF, but a higher order polynomial.
- The structure presents a non unity feedback.
- The R_v and H_v polynomials are of higher order, gives a large number of degree of freedom leading to high performance.

Practically, those differences mean that the controller uses data from previous samplings and therefore can compensate better the delays which occur normally in discrete systems.

In addition, the motor and the load are controlled in the same matter. There is no "disconnection" between the load and the motor. This property of the controller leads to the main advantages of PP controller in relative to the PDFF controller, outlined below:

Advantages of the PP controller over the PI and PDFF controllers

- **Higher closed-loop bandwidth**
  Since the controller polynomials can compensate better for the delay of the system, higher bandwidth is attainable.

- **Better stability & robustness**
  Due to the high order of the D_v polynomial, compared to the integrator of the PI and PDFF controllers, the system stability and robustness are improved.

- **No overshoot design**
  The structure of the PP controller provides a non-overshoot design, whereas the PI controller provides an overshoot by its nature.

- **Separation between bandwidth and gain**
  The PP controller uses the high order polynomials to attain high bandwidth with the controller poles dynamics, compared to the high gain which is required by PI and PDFF to achieve high bandwidth. This improves the noise rejection, robustness and stability of the system.

- **The load and the motor are controlled together**
  The pole placement method for controller design based on the motor position sensor is able to control the motor and the load at the same time, while the PI controller may "disconnect" the load from the motor. When the gain of the PI increases, the resonance "seen" by the motor moves to the anti resonance. On the motor side, the system is more stiff (the resonance and the anti-resonance on the motor side are nearly similar) while on the load side, the resonance of the system can be excited (at the anti- resonance frequency).

- **Easier to tune**
  Since the calculation of the controller parameters is done mathematically, according to the user request of closed-loop bandwidth, the tuning procedure does not involve trial & error and therefore it is more convenient and straight-forward.

The PP structure provides another advantage that makes the controller tuning easier: when an underestimation of the load inertia occurs, the closed velocity loop time response presents an overshoot. An overestimation of the load inertia leads to undershoot and oscillations. The PP type of controller can therefore be tuned simply by adjusting just one parameter - the load inertia ratio. Once the proper load inertia ratio is selected, changing the bandwidth parameter will then just improve or degrade performance.

As mentioned above, the inertia ratio and bandwidth terms are represented by the variables LMJR and BW in the **SERVOSTAR** system.

9.3.4 Advanced Pole-Placement Controller (APP; COMPMODE = 3)

There is an option to use a higher-order polynomials controller, the APP controller, which is preferable for flexible (resonant) systems with considerable high delay. This controller is also referred to as the "Advanced Pole-Placement" controller.

The calculation of the coefficients of the higher order polynomials cannot be done by the drive, but has to be done by external software. For example, Kollmorgen's CONCAD motion analyzer, a control and mechanical diagnostic integration tool, can design this type of controller.

Similar to the PP method, the CONCAD system finds the controller polynomials Rv, Hv and Dv by solving the equation:

$$\frac{Gain*Rv*B}{(1+A)(1+Dv)+Gain*B*Hv}=\frac{Gain*B}{(1+C)}$$

But unlike the PP, here it uses a higher order plant model which includes a pole, an anti-resonance and a resonance, as shown in the following equation:

$$\approx\frac{G(S^2+2\xi_{ar}\omega_{ar}S+\omega_{ar}^2)}{(S+P)(S^2+2\xi_r\omega_rS+\omega_r^2)}$$

Taking the Z transform of the above continuous velocity model equation changes the denominator order and the numerator order to 3 and 5, respectively.

After solving the equation above, and obtaining Rv, Hv and Dv, the coefficients of the polynomials are down-loaded to the drive bv the MotionLink. These terms are represented by the vectcr variables R, H, and D respectively in the SERVO**STAR** system.

9.4 LOW PASS FILTERS

For aid in applications with difficult vibration and/or torsional resonance problems, two independent low pass filters (LPF) are provided in the SERVO**STAR**. When activated, each low pass filter attenuates all the frequency components above its cutoff frequency.

A number of vibration sources exist in servo motor applications. Some of these are: torsional resonance, lateral resonance, power ripple (60/120... Hz), and other structural vibrations. The effects of these undesirable vibrations include audible noise, weakening of structures and mounting failures, reduction in the system bandwidth, excessive motor current, and control loop oscillations. Many of these can be effectively filtered out using one or two low pass filters (however, these filters cannot compensate for a poorly designed mechanical system).

Both low pass filters are implemented digitally in software in the microprocessor. Software filters are easy to tune and are as effective as hardware filters. Each low pass filter is a first order filter. The filter cutoff frequencies (LPFHZ1 and LPFHZ2 for the first and second filters, respectively) are in the range of 20 to 800 Hz, in steps of 20 Hz. A second order filter can be implemented by activating the two filters with the same cutoff frequencies.

The filter is placed in the velocity error signal, before the torque command is clamped by current limits. The filters are available only in the PI, PDFF, and simple pole placement controllers (COMPMODEs 0-2), and are not available in the advanced pole placement controller (COMPMODE 3). To activate a filter, use the FILTMODE command. Setting FILTMODE = 0 turns off both low pass filters. Setting FILTMODE = 1 turns on the first low pass filter, for which the cutoff frequency is LPFHZ1. FILTMODE = 2 turns on both filters (the cutoff frequency for the second filter is LPFHZ2). Autotuning using the TUNE command, sets FILTMODE to 0.

If the filter cutoff is specified at a low frequency, then the bandwidth of the system will be low. Setting the cutoff frequency too low may cause the system to exhibit total instability. If the system is tuned to have high bandwidth and the filter cutoff frequency is low, then the motor may oscillate. In such a case, either the filter cutoff should be increased or the bandwidth of the system decreased. As a general rule of thumb, the filter frequency should not be less than four times the desired system bandwidth.

All filters add phase-lag delays. In applications where the SERVO**STAR** is being controlled by another position loop controller, this will lessen the system's phase margins.

9.5 AUTO-TUNING THE VELOCITY CONTROLLER

9.5.1 What is Auto-Tune?

The Auto-Tune function (using the TUNE command in the SERVOSTAR) is an automatic procedure which does the following:

- Estimates the system load inertia. (LMJR variable).
- Designs a controller based on the estimated load inertia, and the requested bandwidth (BW variable), by using the optimal Pole-Placement Design Method (PP).

Once the PP controller is designed (values are assigned to LMJR and BW), the designs for the PI and PDFF controller types are derived from the PP controller, which results in values being assigned to the variables GV, GVI, KV, KVI, and KVFR.

The TUNE command syntax and usage is described in section 6.

9.5.2 Auto-Tune Concept

The idea of the Auto-Tune procedure is that since the PP controller design is based on the system inertia, the better the load inertia is estimated, then the better the controller will be, regarding the system performance.

One of the PP controller qualities is that if the system model is exact, it will respond with no overshoot to a velocity step command. Usually, the inertia of the load is unknown or cannot be accurately determined. During the Auto-Tune process, the drive estimates the load inertia by assigning an initial value to the load inertia variable (LMJR), issuing a velocity step command, analyzing the system response to the velocity step command, and adjusting the value of LMJR based on that response:

- If the assigned load inertia value is less than the actual load inertia, the system response will have an overshoot.
- If the assigned load inertia value is greater than the actual load inertia, there might be an undershoot in the step response or even oscillations, and the controller might become unstable.
- Once the system response has no or minimum overshoot, you can assume that the load inertia is well estimated. The drive will cease the tuning process at this point and will keep the value it has assigned to LMJR.

From this point, where the load inertia is well estimated, the controller bandwidth may be increased (or decreased) to give better performance.

A well tuned controller is one in which the response to a velocity step command has no overshoot on one hand, and no undershoot or oscillations on the other hand.

9.5.3 Auto-Tune Procedure

The Auto-Tuning procedure is activated by the TUNE command. The system has its default values for the bandwidth, direction and speed, but you may choose other optional values. As described in section 9.3.5, the low pass filters must be turned off by setting FILTMODE = 0 before the TUNE command is used. The following is the description of the Auto-Tune procedure with selection of optional values:

Specify the command: TUNE < optional parameters: bandwidth, direction, and velocity >
A command example: **TUNE 25 0 600**
(Tune the controller with a BW of 25 Hz, bi-directional movement, and maximum step command velocity of 600 RPM)

The optional parameters for the TUNE command are:

- Bandwidth [Hz]:
  Range: [10 to 100] . The default value of the bandwidth is the current value of the variable BW.

- Direction:
  0 - bi-directional rotation during TUNE process (default).
  1 - CW rotation only.
  2 - CCW rotation only.

- Speed [RPM]:
  range: 350 RPM to 0.7 * VLIM
  default: The minimum between (500, 0.7*VLIM, 0.3*VMAX). The minimum should be at least 350 RPM.

The Auto-Tune operation will not be executed if:

- The operation mode in not 'serial velocity' (OPMODE 0).
- The compensation mode is 'advanced pole-placement' (COMPMODE = 3).
- The drive is in 'Hold Position' mode (HOLD = 1).
- The drive is in Zeroing mode (ZERO = 1).
- The Recording process is active.
- The Tune parameters are outside the permitted ranges.

9.6 MANUAL TUNING OF THE VELOCITY CONTROLLER

In the case where the results from the Auto-Tune do not yield satisfactory performance, the manual tuning procedure may be performed. Manual tuning includes the initial steps of load inertia estimation, and the subsequent steps to design the optimum controller.

The following is a manual tuning procedure (it is similar to the procedure that is performed automatically by the drive during the Auto-Tune process). The procedure which is described is based on using a dumb terminal interface utilizing the standard ASCII protocol. The same procedure may be followed, much more conveniently, using the Tune Screen MotionLink (refer to section 8 for information on MotionLink). Go to the Tune Screen in MotionLink and press F1 to access context-sensitive help that will tell you how to tune the drive.

If you tune from a dumb terminal, use the following procedure:

1. Set the operation mode to serial velocity, using **'OPMODE 0'**.

2. Set the velocity compensator to "standard pole-placement", or PP, using **'COMPMODE 2'**.

3. Enable the drive using '**EN**'.

4. Set the desired bandwidth, using '**BW**' command. It is recommended to set low bandwidth (10 Hz), **'BW 10'**.

5. Set the **load inertia ratio** to zero, using **'LMJR 0'**. The load inertia value LMJR is expressed in percent of the rotor inertia.
   Note: the range for LMJR is 0 to 10,000, and it represents a range of load:rotor inertia ratios ranging from 0:1 to 100:1. For example, to set a 10:1 ratio between the load inertia and the motor inertia, use the command LMJR 1000.
   At this point the step response should have the maximum overshoot, but the control loop should be stable (low gains).

6. Activate the recording mechanism to record the **velocity command**, the **current command**, and the **actual (feedback) velocity**, using **'RECORD 1 1024 "VCMD "ICMD "V'**.

7. Jog the motor in constant low speed, using **'J value1'** command. For example start with the command **'J 150'**

8. Activate the recording trigger, **using 'RECTRIG "VCMD value2 20 1'**. Value2 should be greater than value1 in the previous stage. For example **'RECTRIG "VCMD 200 20 1'**

9. Jog the motor to a higher value than specified in the **'RECTRIG'** command. This is actually the step command. The higher the step is, the better are the results. The step should not be below 300 [RPM]. For example **'J 500'**

10. The recording takes about half a second. After this period of time, reduce the motor speed.

11. Verify that the recording process is terminated, using **'RECDONE'** command.

12. Dump the recorded data to the serial port, using **'GET'** command.

13. Analyze the results.
    - If the current command is saturated (gets absolute values of 'ILIM') during the process, reduce the step or reduce the bandwidth, and return to step 6.

    Inspect the feedback velocity:
    - If there is an undershoot or oscillations, reduce 'LMJR' to the mean value of the present 'LMJR' and the last 'LMJR' where an overshoot exists, and return to step 6.
    - If there is an overshoot, increase 'LMJR' by 100 and return to step 6.
    - Repeat this process until the overshoot is minimum or nearly in a "Non overshoot" situation. (The estimated load inertia may be queried by 'LMJR')

9.7 ADVANCED TUNING OF THE VELOCITY CONTROLLER

The automatic and manual tuning procedures, use the Pole-Placement design method based on a simple system model. Of course, this is not always true. When a load is connected to the motor, as long as the gear and the load are very stiff, the simple model is accurate enough - does not change. However, in a case of flexibility in the system (in the gear and in the load), the system model changes to be a resonant one. It includes, in addition, anti-resonance and resonance.

The following figure shows the Bode plot of a amplitude only for simple and resonant plant models (frequency domain):

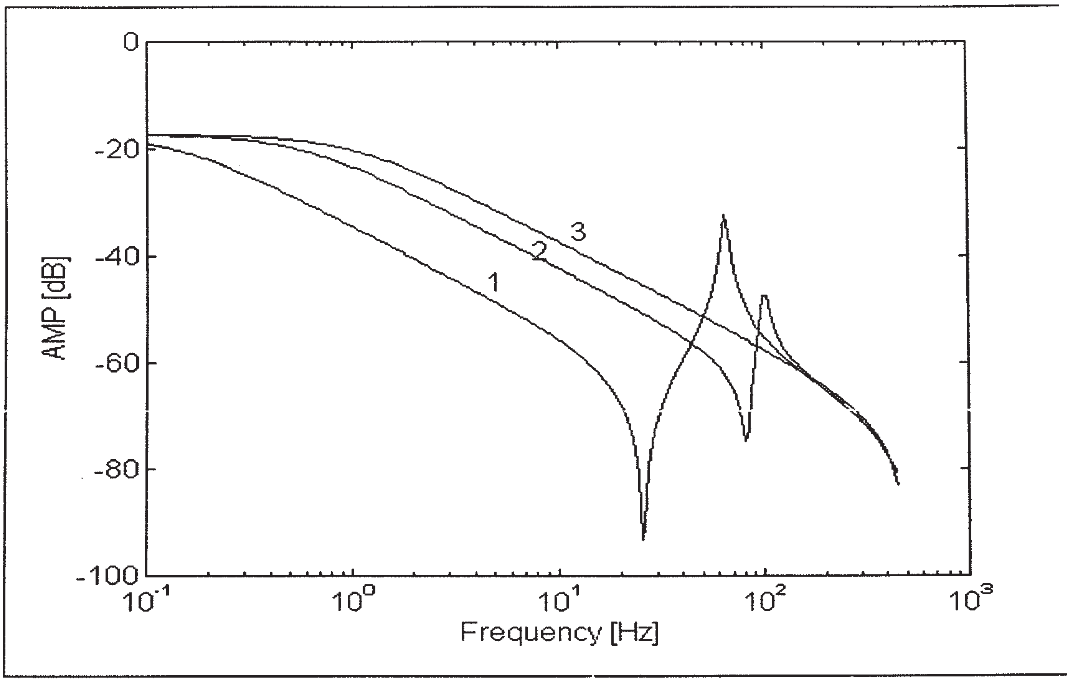


Figure 9-4. Bode Plots for an Electromechanical System

- Plot **3** represents the simple model.
- Plot **1** (high inertia load) and plot **2** (low inertia load) represent the situation where the load is connected to the motor with a non rigid connection (anti-resonance and resonance exists then).

Notice that the three lines meet at the high frequencies range, where the lines represent the inertia of the motor and the part of the load which is rigidly connected to the motor. Usually, the flexibility of a system is within the gear. Therefore, the equivalent model at high frequencies is based on the motor inertia with small effect of the load inertia.

In the case of a resonant model, the closed-loop bandwidth which the Auto-Tune process uses is limited by the anti-resonance frequency. In order to receive higher bandwidth, lower load inertia ratio has to be selected (even though it may cause an overshoot).

Therefore, we can define two cases:

1. The system is very stiff / The load is very small

   The simple model is accurate and the Auto-Tune procedure is suitable for the load inertia estimation and controller parameters.

2. The system is flexible

   - Use the Auto-Tune procedure to analyze the load inertia in the low frequency range. The resulted bandwidth should be defined lower than the anti-resonance frequency (select the minimum of 10 Hz in case of flexible system).
   - Increase the bandwidth by 5 Hz increments until desired performance is achieved or oscillation occurs. Choose the last BW that gives desired performances or stable responses.

- To have higher bandwidth, define zero load inertia and a bandwidth which is higher than the resonance frequency. The resonance frequency may be estimated from the system response oscillations.
- Analyze the system response, regarding the overshoot and the oscillations. If the overshoot is too high, increase the load inertia, until the desired response is received (compromise).

9.8 ADVANCED TOPICS

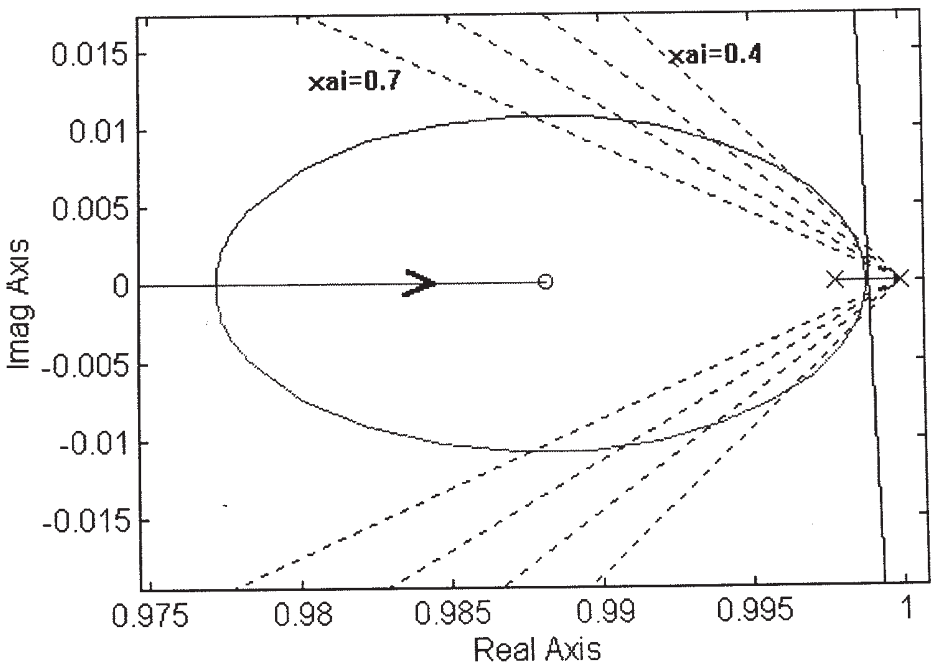
9.8.1 PI Controller

The closed-loop transfer function of the system with plant model B/(1+A) and PI controller is:

$$\frac{V}{V_{ref}} = \frac{BG_v\left[(G_{vi}+1)-z^{-1}\right]}{(1+A)(1-z^{-1})+BG_v(1+G_{vi}-z^{-1})}$$

The equation shows that the closed-loop has a zero due to the integrator of the PI controller. The zero in the system causes an overshoot of the step response.

The following root locus diagram shows the behavior of the closed loop poles when the gain - G_v is changing, for a very stiff system, and the assumption that the frequency of the zero of the system is greater than the mechanical pole of the system.



Only when the gain - G_v is near the infinity the zero is canceled with the closed loop pole. In most cases the closed loop poles will be in the circle which is shown above. The poles are either complex, with damping factor lower than 0.7, or located in higher frequency than the zero. In both cases the effect presents an overshoot in the step response.

9.8.2 PDFF Controller

In order to understand the "derivative" effect of the special feedback in the PDFF controller, the following figure shows an equivalent control loop with a "derivative" term in its main feedback, instead of the feedback signal of the PDFF, which explains the "pseudo derivative" effect of the PDFF controller:

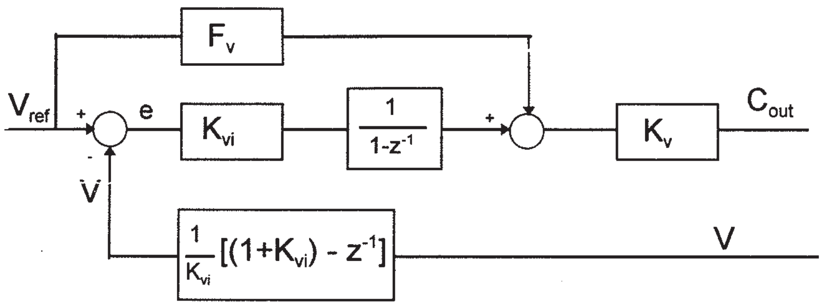


Figure 9-5. Equivalent PDFF Controller configuration

The feed-forward term F_v gives the control system the flexibility to change its characteristics from a simple PDF controller to a PI controller. This may be seen by introducing a pre-filter term - R, equivalent to F_v:

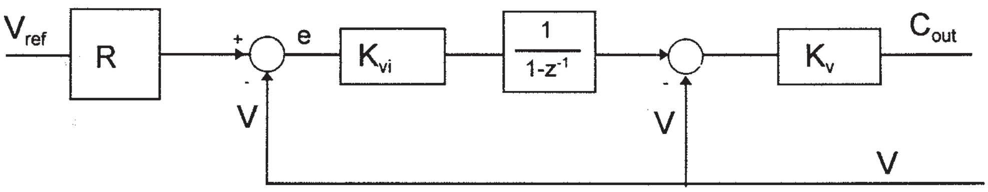


Figure 9-6. PDF Controller configuration with pre-filter, equivalent to PDFF

In general, $R = 1 + F_v / K_{vi} * (1 - z^{-1})$.

It can be seen that when F = 0 then R = 1 , and the control loop is a simple PDF controller.

When F = 1 then $R = 1 + (1 - z^{-1})/ K_{vi} = 1/ K_{vi} [(1 + K_{vi}) - z^{-1}]$ which gives us a simple PI controller.

To understand this, let us introduce the pre-filter block into the control loop of Figure 9-2:

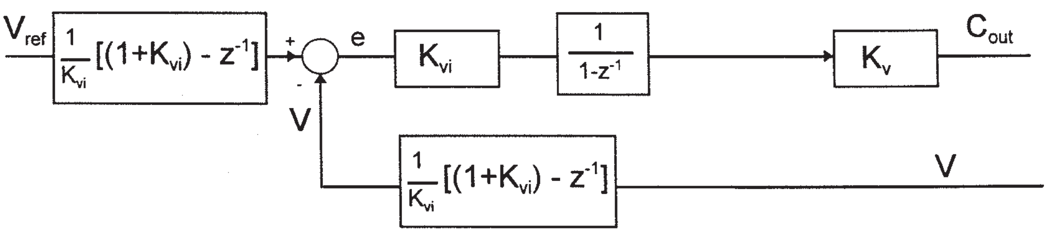


Figure 9-7. PI Controller configuration generated from a PDFF controller

We can see that the terms in the feedback and in the pre-filter are identical. By moving both terms inside the control loop, we get an equivalent control loop scheme as in Figure 9-1, where $K_{vi} = G_{vi}$, which is a PI controller. It means that the PDFF is a generalized controller where the PI controller is simply a particular case of it, for F=1.

There is of course the option to choose the value of F somewhere between 0 and 1, which gives this controller its advantage of flexibility in relative to PDF and PI controllers.

9.8.3 PP Controller

The Pole-Placement Design Method is a direct method for controller design. Its algorithm evaluates the controller parameters based on the plant model, the controller structure and specific requirements of the closed-loop transfer function.

Referring to figure 6, the closed-loop transfer function of the velocity loop is:

$$\frac{V}{V_{ref}} = \frac{Gain * R_v * B}{(1+A)*(1+D_v) + Gain * B * H_v}$$

where the velocity plant is defined as B/(1 + A). A, B, R_v, H_v and D_v are polynomials which are defined as follows:

$$1 + A = 1 + a_1 z^{-1} + \dots\dots + a_n z^{-n}$$
$$B = b_0 + b_1 z^{-1} + \dots\dots + b_m z^{-m}$$
$$1 + D = 1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}$$
$$H_v = h_0 + h_1 z^{-1}$$
$$R_v = r_0 + r_1 z^{-1}$$

In the Pole-Placement method, the desired closed-loop poles and R_v poles are determined, and the polynomial coefficients H_v and D_v are calculated accordingly. In order to receive the desired closed-loop poles, without changing the system zeroes, we define the closed-loop transfer function in the following way:

$$\frac{V}{V_{ref}} = \frac{Gain * R_v * B}{(1+A)*(1+D_v) + Gain * B * H_v} = \frac{Gain * R_v * B}{(1+C)*R_v}$$

where (1 + C) includes the dominant closed-loop poles (user define bandwidth) and other poles automatically calculated in the drives for maximum robustness and maximum torque disturbances rejection. R_v consists the pre-filter poles. Since R_v exists in the numerator and the denominator of the right side of the equation, it is obvious that R_v poles do not affect the closed-loop behavior.

If n is the degree of A and m is the degree of B , the number of poles of R_v and (1 + C) is limited to (n+m). For example, if the degrees of B and A are 5 and 3, respectively, the maximum number of closed-loop and observer poles is 8. (A complex pole is considered as 2 poles).

The following equations are examples of physical continuous system model descriptions:

Systems that are considered Stiff:

$$\approx \frac{K_a K_T / J_T}{S + B_m / J_T}$$

where J_T is the total moment of inertia, of the motor and the load : $J_T = J_{m+} J_l$

Systems that are considered as Flexible system - resonant models:

$$\approx \frac{K_a K_T / J_T * (S^2 + \omega_{ar} \xi_{ar} S + \omega_{ar}{}^2)}{(S + B_m / J_T)(S^2 + \omega_r \xi_r S + \omega_r{}^2)}$$

Where the resonance and anti-resonance are functions of the physical parameters (Moment inertia [load & motor], molecular damping, stiffness, viscosity friction damping [load & motor], etc..)

Since the PP controller design is based on a simple model, a decision has to be made on which plot the controller will be designed - the highest one (load inertia = 0) or the lowest one (maximum load inertia of the system, before the anti-resonance frequency), or somewhere between them. The selection of the load inertia in this range implies also the selection of the system BW. If the highest value of the load inertia is selected, the maximum BW that can be selected is lower than the anti-resonance BW. If the lowest value of the load inertia is selected, the BW that can be selected may be higher than the anti-resonance frequency.

In general, the selected value will be somewhere between these extreme values.

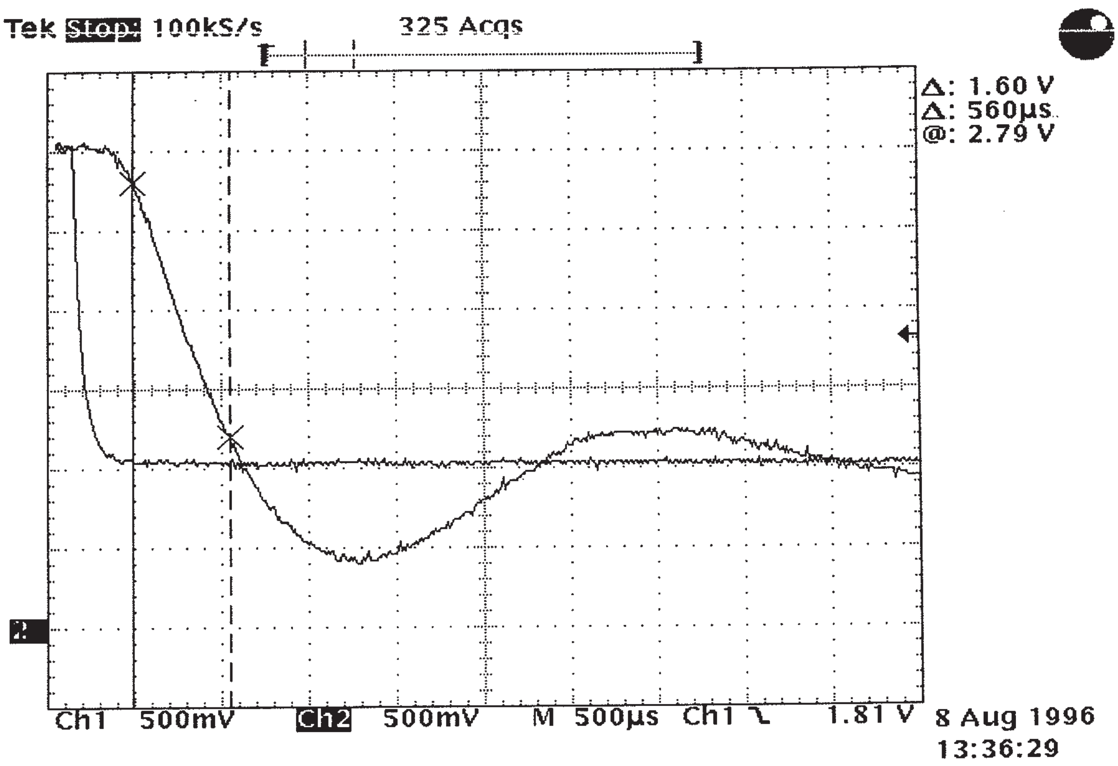
Using CONCAD (Kollmorgen Motion analyzer) and the APP structure, no compromise is needed and the best performance are obtained even for complicated mechanical structures.

9.9 EXAMPLES: COMPARISON BETWEEN PI AND PP / APP

9.9.1 Current Loop Controller

The following graphs show the real behavior (step response) of the phase current loop controller of the SERVOSTAR on a brushless motor (Kollmorgen GoldLine family) using a PP or a PI controller.

PI controller - current loop

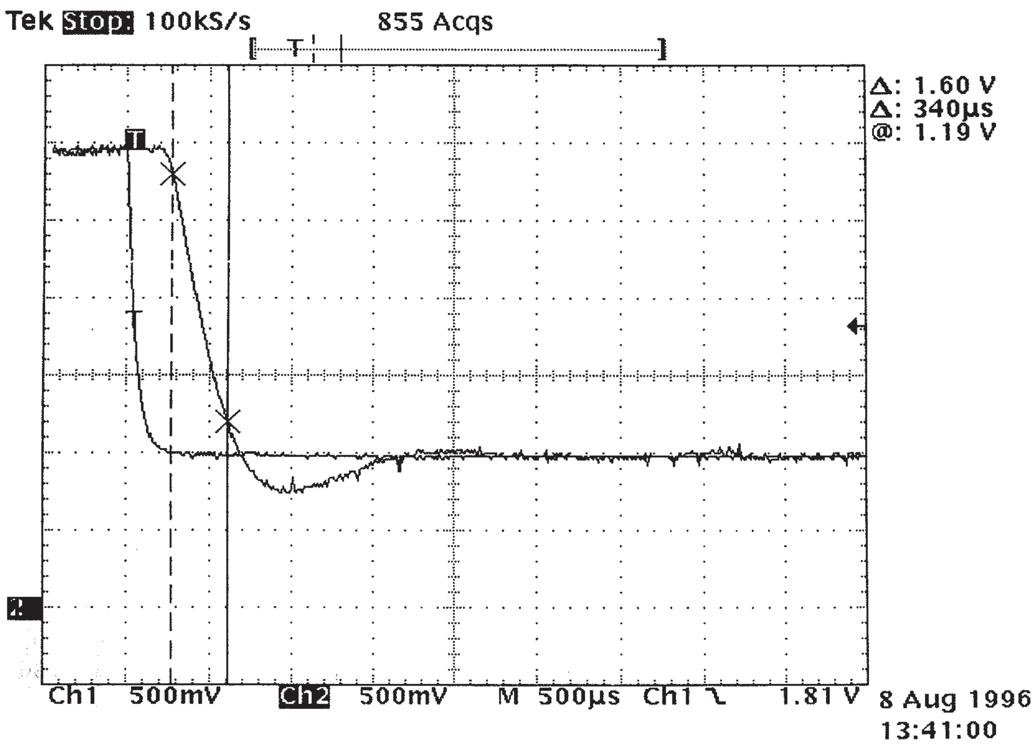


Overshoot : 30 %
Maximum Bandwidth achievable: 850 Hz

where

$$\textbf{Bandwidth} = f_n = \frac{2.5}{2\pi\ tr}\ \textbf{Hz}$$

Pole placement controller - current loop



Overshoot : 10 %
Maximum Bandwidth achievable: 1,404 Hz

where

$$\text{Bandwidth} = f_n = \frac{2.5}{2\pi\, tr}\ \text{Hz}$$

9.9.2 Velocity Controller

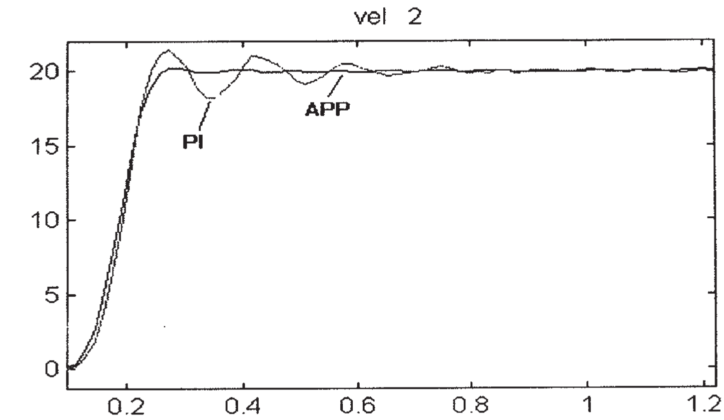
The following graphs show the real behavior (step response) of the velocity of axis 2 of a six degree industrial robot. This electro-mechanical axis presents the following characteristics:

- Low frequency Resonance
- Extreme low damping factor
- Large change in moment of Inertia depending on robot movement
- Large cross-coupling torque disturbances
- Large unbalance torque

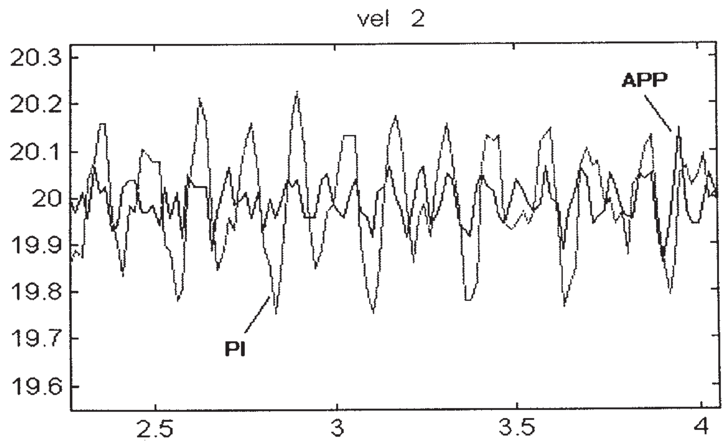
Application requirements by priorities :

1) Rise time
2) Ripple less than 1.5%
3) No overshoot

The results (velocity graph) of two controllers PI and APP



The steady state of the two controllers



	PI	APP
Peak to peak ripple	2.2 %	1.1 %
Overshoot :	15 %	0 %
Oscillations are presented to achieve required rise time.	YES	NO

APPENDIX A

WARRANTY INFORMATION

Industrial Drives, a Kollmorgen Division, warrants that equipment, delivered by it to the Purchaser, will be of the kind and quality described in the sales agreement and/or catalog and that the equipment will be free of defects in design, workmanship, and material.

The terms and conditions of this Warranty are provided with the product at the time of shipping or in advance upon request.

The items described in this manual are offered for sale at prices to be established by Industrial Drives and its authorized dealers.

APPENDIX B

REGIONAL SALES OFFICES

B.1 INTRODUCTION

Industrial Drives, A Kollmorgen Division, is committed to quality customer service. We realize that your time is valuable. Our goal is to provide you with the information and resources you need when you need them. This appendix is a guide to assist you in quickly locating the appropriate Kollmorgen Motion Technologies Group representatives. Our services are divided into four regions across the country and around the world. Please contact the regional office that is closest to you in the U.S., Canada, and Mexico or contact the international office if outside the U.S. The following sections list the regions, their coverage areas, addresses and phone numbers.

B.1.1 Southern Region

Kollmorgen Motion Technologies Group
118 North Avenue
Suite H
Jonesboro, GA. 30236

Telephone: (770) 472-4100
Fax: (770) 472-4342

B.1.2 Eastern Region

Kollmorgen Motion Technologies Group
49 Mall Drive
Commack, New York 11725-5703

Telephone: (516) 864-1000
Fax: (516) 864-2084

B.1.3 Midwest Region

Kollmorgen Motion Technologies Group
1985 W. Big Beaver Road
Suite 212
Troy, MI 48084-3409

Telephone: (810) 816-8740
Fax: (810) 816-9855

B.1.4 Western Region

Kollmorgen Motion Technologies Group
23011 Moulton Parkway, Unit H5
Laguna Hills, California 92653

Telephone: (714) 518-3626
Fax: (714) 581-3628

B.1.5 Northern Europe

Kollmorgen Hightech Ltd.
POB 147
Keighley
Yorkshire BD20 6XB
UK

Telephone: 44-1535-607688
Fax: 44-1535-680520

B.1.6 Southern Europe

Servotronix Ltd.
12 Hachoma St.
POB 5096
Rishon Le Zion 75105
Israel

Telephone: 972-3-9616577
Fax: 972-3-9616579

B.1.7 Far East

Kollmorgen Motion Technologies Group
49 Mall Drive
Commack, NY 11725

Telephone: (516) 864-1000
Fax: (516) 864-2084

B.1.8 Kollmorgen Customer Support

1-888-774-KCSN (5276) Phone

1-540-633-3100 FAX

APPENDIX C

MOTOR AND FEEDBACK PHASING

C.1 SYSTEM PHASING

Three phase servo loop systems require that all signals be correctly polarized. Components purchased from Kollmorgen Motion Technologies Group have much of the phasing requirements already taken care of. When purchasing components, assembling 'kit' motors or integrating a drive with a custom or linear motor may present a challenge. This appendix provides the data to correctly phase the SERVOSTAR Drive.

CAUTION

Incorrectly phased servo systems can cause unpredictable motor motion. Use caution when starting up systems. Make certain that no physical or property damage would result if the motor 'runs away'.

C.2 RESOLVER BASED SYSTEMS

The phasing of resolver based systems tends to be somewhat simpler than that of encoder based systems. There are four important points:

1. The absolute zero of the resolver must be aligned with the electrical phasing of the motors back emf.

2. The resolver must provide an up-count when the motor turns clockwise. (Referenced from viewing the output shaft straight-on).

3. The three phase motor must be sequenced so that an A-C-B signal rotates the motor clockwise.

4. Drive variables must be correct.

These conventions are held as the standard by Kollmorgen Motion Technologies Group.

1.) The first step is to be certain that the MPOLES (Motor Poles) and MRESPOLES (Motor Resolver Poles) variables are set correctly. A typical servomotor has an MPOLES value of 4 or 6 (2 or 3 pole pairs). Typically, a single-cycle resolver has an MRESPOLES value of 2 for two magnetic poles (one pole pair). Be certain that these variables are correctly configured in the SERVOSTAR.

2.) The second step is to be certain that the resolver is wired correctly. With logic power only applied to the SERVOSTAR (connector C5), slowly rotate the motor shaft in a clockwise direction while continuously monitoring the resolver position using the PFB command. For resolvers with one pole-pair (MRESPOLES = 2), the PFB value should count from 0 to 65535 for one complete mechanical revolution of the motor shaft. Resolvers with more than one pole-pair (MRESPOLES > 2) will count from 0 to 65535 once for each pole-pair per mechanical revolution (MRESPOLES = 4 will count from 0 to 65535 twice per mechanical revolution, MRESPOLES = 6 will count from 0 to 65535 three times per mechanical revolution, etc.).

3.) The third step is to be certain that the motor's power is correctly phased. This is assured using an oscilloscope as follows:

1. Ensure that all power is removed.
2.

CAUTION

Be certain that motor wires do not short to each other or to ground through this test.

Disconnect the motor wires from the SERVOSTAR

3. Attach the motor's A-phase wire to the first channel of the oscilloscope.
4. Attach the motor's C-phase wire to the second channel of the oscilloscope.
5. Attach the motor's B-phase wire to the both grounds of the oscilloscope.
6. Spin the motor shaft, by hand, clockwise.
7. Use the oscilloscope's B-channel inversion feature to view the V_{AB} and V_{BC} waveforms together.
8. The A-phase should lead the C-phase by 120 electrical degrees.

4.) The final step is to be certain that the resolver's absolute zero is in alignment with the motor as follows :

1. Ensure that all power is removed.
2. Connect the motor wires to the SERVOSTAR.
3. Turn on the logic power only (connector C5).
4.

CAUTION

The SERVOSTAR variable IZERO sets the amount of current through the motor during this setup procedure. Be certain that it is set to a value that will not damage the motor. This portion of the setup procedure should be performed by qualified personnel only!

Apply bus voltage.

5. Enter the 'ZERO 1' command to the SERVOSTAR. This will place the drive in Zero mode.
6. Carefully rotate the motor shaft to be certain it is locked into a torque detent.
7. Using the PRD command, read the resolver position.
8. PRD should return 0 (± 16) counts for a two pole motor, 0 or 32768 (±16) counts for a four pole motor, or 0, 21845, or 43691 (±16) counts for a six pole motor.
9. Adjust the resolver as necessary to obtain the proper reading.
10. Enter the 'ZERO 0' command.

The system should now be correctly phased. Be certain that the motor's data is correctly configured for all variables in the SERVOSTAR. The system should now be operational as a velocity or torque controller.

C.3 ENCODER BASED SYSTEMS (Finding the Encoder Index Position)

C.3.1 Introduction

The phasing of encoder based systems can be more difficult than that of resolver based systems. This is because additional channels need to be correctly phased :

1. The Hall channels need to be aligned to the motor's back EMF.
2. An offset for the marker must be located.
3. The motor must be correctly phased.

The encoder index is assigned a default position of 120° electrical. Since this may not always be correct, a process is defined with which the index position may be more accurately found. This process is defined as the Encoder Initialization Process.

This process consists of two primary stages:

1. During the first stage, the drive finds the encoder index position (using the Hall switches as a position reference). Since the Hall switches may not be exactly aligned to the motor, the encoder index position thus found may not be accurate enough. This inaccuracy will result in unequal speed in the CW and CCW directions, relative to the input TORQUE command.

2. During the second stage, the user manually tunes the index position until the desired accuracy is obtained. The encoder index position is then stored in EEPROM so that the initialization process will not be required at every power up.

C.3.2 Relevant Commands and Variables

ENCINIT - This command triggers the Encoder Initialization Process

ENCINITST - This variable may be queried by the user and informs the user of the status of the initialization process. This variable assumes one of three values:

0 = The initialization process has not started
1 = The initialization process is in progress
2 = The index position has been determined and the initialization process is complete.

The flag is reset to zero when the user manually sets the index position.

MENCOFF - This command sets or queries the encoder index position. When using this command, the encoder index position is in units of encoder counts after quadrature. The range is from 0 to (Encoder Resolution - 1), where the encoder resolution is in units of encoder counts after quadrature.

MENCTYPE - This is a switch mode variable used to store the encoder interface type.

C.3.3 General Information

The following general information may sound complicated and is all transparent to the user after installation. However, knowing this information can help find initial installation problems. If phased incorrectly, the system can jump or even run away on power-up. Understanding that the SERVOSTAR has three states of commutation helps the user realize what is taking place.

A resolver based system is absolute, which means that when power is applied, the system knows the exact motor position from the resolver feedback. The SERVO**STAR** can therefore commutate the motor. An encoder based system is incremental and does not know where the motor position is at power up, except for information that is imparted by the hall sensors.

The purpose of the hall channels is to provide a rough estimate of the motor shaft position. Hall channel sensors are three encoder outputs which provide a rough estimate of motor shaft position by telling what sector the shaft is in. A "sector" may be as large as 60 electrical degrees. This "rough estimate" is not accurate enough for strong, smooth motor performance, although the system will be able to rotate the motor.

After power-up, the motor shaft must be rotated until the "marker" is encountered. The marker (or index) is an output of the encoder, and it goes high at one specific location in 360 degrees of revolution. Once the marker is located, the SERVO**STAR** will know exactly where the motor shaft is and will be able to rotate the motor smoothly.

As the motor moves (by command, freewheel spinning, or turning by hand) the hall channels may transition to a new state before the marker is encountered. At this point, the SERVO**STAR** has a better indication of where the motor is, but this information is still not considered accurate enough, since typical hall channels within a motor can vary by ± 15 degrees, or greater.

As the motor continues to move, it will eventually cross the marker channel on the encoder, causing the marker output of the encoder to go high. At this point, the SERVO**STAR** sets the value of the calibrated offset variable MENCOFF to set an exact commutation angle for further motion.

C.3.4 Encoder-based SERVOSTAR Phasing Instructions

WARNING

Attempting to run an encoder-based version of the SERVOSTAR product without proper calibration can cause the motor run away in an uncontrolled velocity

The SERVO**STAR** uses the encoders marker signal to align sinusoidal commutation. The SERVO**STAR** must be calibrated to know where to set the commutation with respect to the marker. Failure to perform this initialization can cause improper motor performance and present a danger to the user. This procedure is required only once per system, or after issuing a CONFIG command.

Perform the initialization procedure as follows :

1. Ensure proper system wiring.
2. Apply logic power to the SERVO**STAR** and establish communications.
3. Enter 'OPMODE 0'.
4. Use the PRD variable to verify A and B channel operation. PRD should count up for clockwise motor rotation, as viewing the output shaft.
5. Enter 'ENCINIT'.
6. Rotate the motor shaft two turns clockwise, either by hand or using the Jog (J) command.
7. Verify that the process is complete by entering 'ENCINITST'. The SERVO**STAR** should return '2'. If not, repeat steps 5 and 6.
8. Enter 'SAVE'.
9. Use caution to continue testing the system.
10. The 'MENCOFF ####' variable may be trimmed for best performance.

C.3.5 Phasing Diagrams

If problems exist, use an oscilloscope to verify all waveforms shown in the following phasing diagram(s).

To plot the V_{AB} and V_{BC} waveforms together, connect the oscilloscope as follows :

1. Connect the oscilloscope channel 1 probe to the motor's A-phase, and channel 1 ground to the motor's B-phase.
2. Connect the oscilloscope channel 2 probe to the motor's C-phase, and channel 2 ground to the motor's B-phase.
3. Use the oscilloscope's B-channel inversion feature to view the V_{AB} and V_{BC} waveforms together.

In addition, note :

1. All diagrams are for the motor's output shaft turning clockwise as facing the output shaft.
2. If the motor phasing is determined rotating the motor shaft clockwise (as viewed facing the output shaft) then the PRD counter must count "up" for clockwise rotation.
3. Open collector hall channels will have the A-input tied to +5 volts and will be inactive: Only the B-input is used.
4. To achieve the phase relationship shown in the commutation diagram on the following page, the halls must be connected and the corresponding motor phases must be identified.

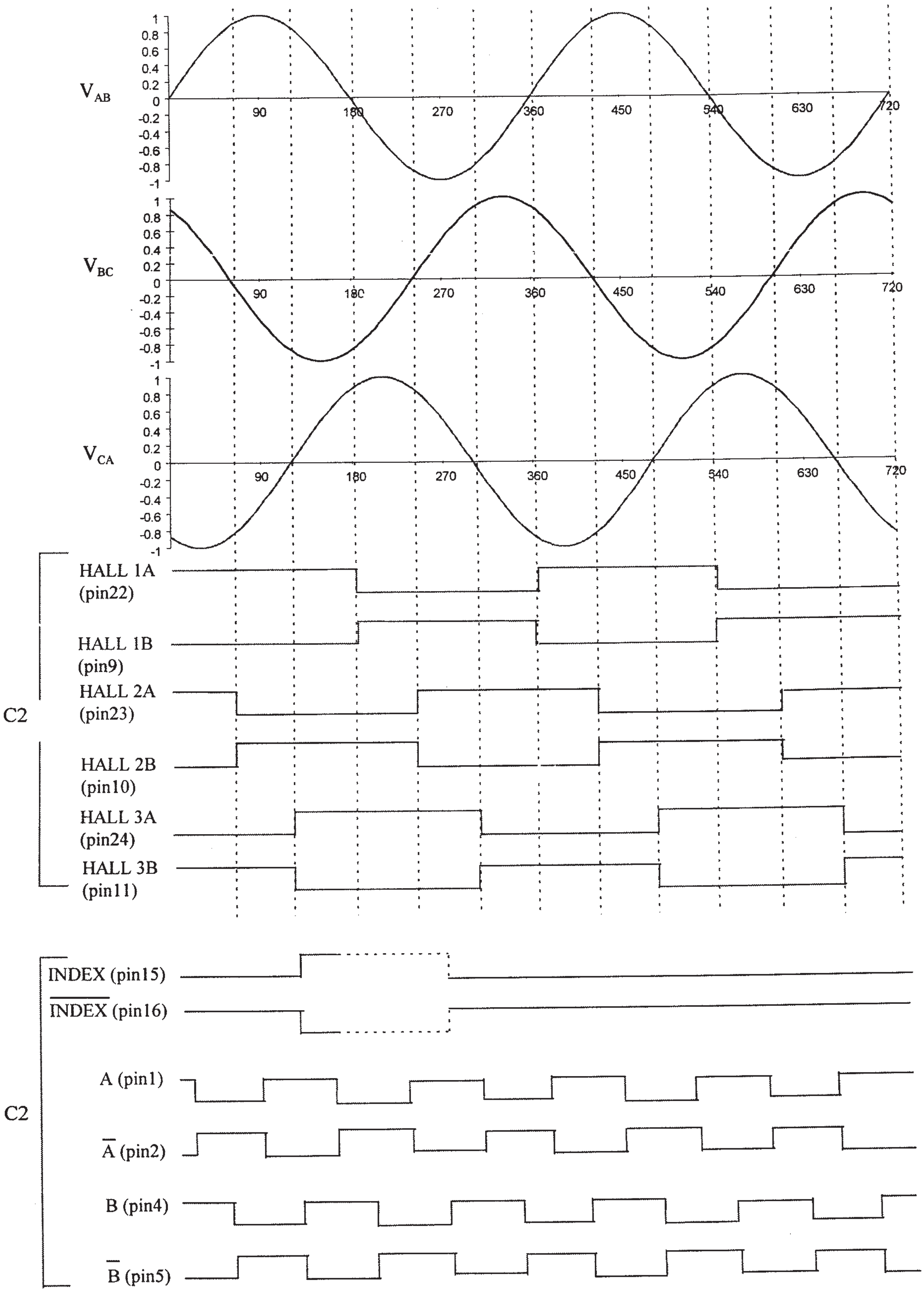
NOTE

The motor's power leads should be disconnected when looking at the motor BEMF waveforms

C.3.6 Encoder Equivalent Outputs

Please note that the encoder equivalent outs for encoder based and resolver based systems are reversed.

SERVOSTAR COMMUTATION DIAGRAM
FOR CLOCKWISE ROTATION VIEWING SHAFT END - PRD COUNTS UP



APPENDIX D

FIELD SETUP OF MOTOR PARAMETERS

D.1 INTRODUCTION

The SERVOSTAR is normally shipped from the factory pre-configured (compensated) to operate with a specific motor. In some cases, the SERVOSTAR may be shipped without any compensation for setup in the field. This Appendix details how to field configure the SERVOSTAR.

A SERVOSTAR that is not configured will power up with a minus (-) sign in the display. This appendix describes the variables in the MotionLink screen used to configure the SERVOSTAR to properly operate almost any permanent magnet motor. The meaning of each variable and its purpose within the drive is described.

MotionLink software contains a series of databases of motor compensation files. A series of standard motors may be chosen from these files and downloaded to the drive under the motor configuration screen. The variables contained in these databases establish proper drive configuration with respect to proper commutation, stable current and velocity loop operation, and system protection limits. For cases where MotionLink does not contain a proper configuration for the motor the user can enter the data for the motor under the "User's Motor" database options.

Note: One other variable within the drive is used to stabilize these control loops that is not part of the motor database but yet is a very important variable. This variable is called VBUS.

D.1.1 Getting Started

Open MotionLink and proceed to the motor configuration screen. (MotionLink operation is described in Section 8) In the motor configuration screen, click on the "User Defined" tab.

D.2 SETTING MOTOR PARAMETERS

The SERVOSTAR uses the motor parameters for basic setup, optimizing system performance, and system protection.

D.2.1 Motor

The first entry on the screen is the MOTOR. Within the drive this variable is known as MOTOR which is a string variable stored in nonvolatile memory and performs no function within the drive itself. It is a user-entered alpha-numeric keeper for identifying this particular setup. The Kollmorgen series of motors are identified by appropriate model numbers here. The user may enter a particular model number of the motor or, use some other identifier for his own purpose. The string variable can be up to ten characters in length.

D.2.2 Motor Data Sheet Parameters

The following entries are typically provided on the motor data sheet provided by the manufacturer. The standards used by the SERVOSTAR for electrical signals assumes sinusoidal RMS units for voltage and ampere entries.

Motion Link-Motor Parameter Entry

MIPEAK
Motor maximum rated current.
Units: 0 .1 amps
Example: enter 1000 for a 100 amp peak rated motor

MICONT
Motor continuous rated current.
Units: 0.1 amps
Example: enter 100 for a 10 amp continuous rated motor

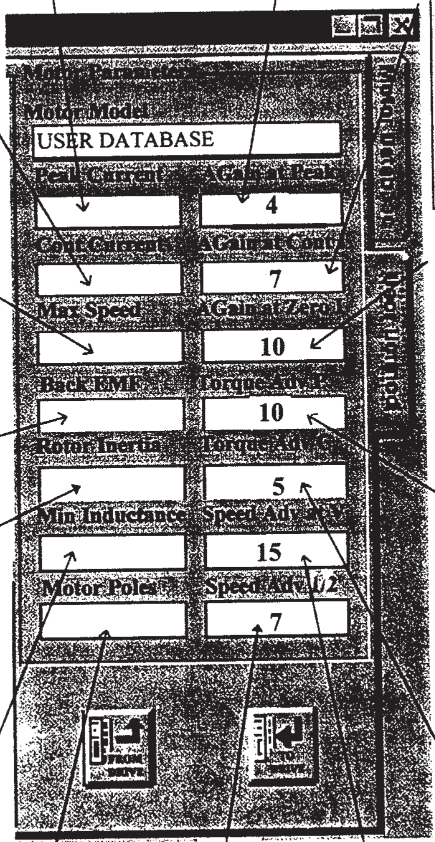
MSPEED
Motor maximum rated speed.
Units: RPM
see Appendix D.2.2.3

MBEMF
Motor Back EMF.
Units: RMS voltage generated by the motor at 1000 RPM

MJ
Motor rotor inertia.
Units: kilograms-meter square times 10^-6

MLMIN
Motor minimum inductance.
Units: mH times 10^-2 .
Example: an 11mH motor would be entered as 1100.
(This is critical to having a good comp.)

MPOLES
Motor poles. **Not pole-pairs**



MLGAINP
Adaptive current loop gain at motor's peak rated current.
Units: 10%
Example: 4 = 40% gain or gain reduction of 60%
Default = 4 (surface magnet motors)
(Note: amplifier may not be able to deliver this current.)

MLGAINC
Adaptive current loop gain at motor's continuous rated current.
Units: 10%
Default = 7 (surface magnet motors)

MLGAINZ
Adaptive current loop gain multiplier at motor's zero output current.
Units: 10%
Default = 10 (surface magnet motors)

MTANGLP
Torque advance at peak motor current.
Default = 10 (surface magnet motors)
Default = 25 (internal magnet motors.)

MTANGLC
Torque advance at continuous motor current.
Default = 5 (surface magnet motors)
Default = 8 (internal magnet motors.)

MVANGLH
Commutation angle advance at velocity of 1/2 MSPEED.
Units: Electrical Degrees
Default = 7

MVANGLF
Commutation angle advance at velocity of MSPEED.
Units: Electrical Degrees
Default =15

See the rest of Appendix D for more detailed information

D.2.2.1 Peak Current

Within the drive this variable is known as MIPEAK. This variable sets a protection or a current clamp limit for the motor that provides a safety feature so the drive will not output excessive current to the motor which could cause demagnetization. The units are in tenths of an amp (a 100 amp peak motor would be entered as 1000).

D.2.2.2 Continuous Current

Within the drive this variable is known as MICONT. Continuous Current is a variable set within the SERVOSTAR, providing current clamping to protect the motor from overheating. The continuous current of the motor is entered in tenths of an amp (a 10 amp continuous motor would be entered as 100).

D.2.2.3 Max Speed

The maximum speed of the motor is entered here. Within the drive this variable is known as MSPEED. The SERVOSTAR uses this entry to determine initial values for over-speed protection, analog input scaling, and velocity limits. The maximum speed is used in conjunction with the BUS voltage (VBUS), the back EMF constant (MBEMF), and the systems limited speed (VLIM) to set a variable called VMAX. VMAX is then used to limit the speed that this motor will ever achieve. In resolver-based systems, VMAX is also used to determine the resolver-to-digital converter resolution. The entry is directly in RPM's.

D.2.2.4 Motor Back EMF

Within the drive this variable is known as MBEMF. This is the voltage (RMS) generated by the motor at 1000 RPM. The drive uses this variable to determine, in conjunction with VBUS, the actual speeds that the system is able to achieve. The relationship is: VBUS - (MBEMF * 1.414 / 1000) * VMAX = 0.

D.2.2.5 Rotor Inertia

Within the drive this variable is known as MJ. The entry for Rotor Inertia is entered with the units of kilograms - meter squared times 10^{-6}. It is used to establish velocity loop compensation parameters.

D.2.2.6 Motor Inductance (Minimum)

Within the drive this variable is known as MLMIN. This is the minimum inductance that the system will ever see and, in conjunction with VBUS, is the basis for establishing the gain of the current loops. Units are entered in millihenries times 10^{-2} (an 11-millihenry motor would be entered as 1100).

D.2.2.7 Motor Poles

Within the drive this variable is known as MPOLES. This variable establishes the number of magnetic poles used within the motor. The number of magnetic poles is entered, but relates to mechanical poles as a two to one factor. The GoldLine series for example: a 40x motor has four magnetic poles which relates to two mechanical poles. This entry is used to establish proper commutation and velocity loop configuration.

D.2.3 Current Loop Adaptive Gain Control

The SERVOSTAR implements adaptive gain control in the current loop controller. It is often the desire to decrease the gain of the current loop as the current through the motor increases to the compensate for magnetic saturation. The saturation effectively lowers the inductance that the SERVOSTAR sees increasing the overall loop gain. The increase in loop gain can exhibit signs of instability in the current loops. Decreasing the current loop gain to compensate for this effect allows for the best overall system performance. The algorithm implemented in the

SERVOSTAR is a single-slope straight line interpolation algorithm to decrease the gain from zero motor current through peak motor current

D.2.3.1 AGain at Zero

Within the drive this variable is known as MLGAINZ. The AGain at Zero entry sets the adaptive gain multiplier at zero output current. An entry of 10 equals 100% gain.

D.2.3.2 AGain at Peak

Within the drive this variable is known as MLGAINP. The AGain at Peak entry sets the adaptive gain multiplier at the motor's peak rated current. (Note that the drive may not actually be able to deliver this current). An entry of 4 equals 40% gain, or a gain reduction of 60%

D.2.3.3 Rule of Thumb For Adaptive Gain

Field compensation of field settings of these parameters can be confusing so the general rule of thumb would be that leaving AGain at Zero (MLGAINZ) at 10 for 100% gain. Set the adaptive gain at peak current (MLGAINP) to 4 for motors that do not have a lot of iron in their construction and peak currents within the boundaries of the drive. If the motor is rated for much more current than what the drive can deliver, and / or if there is a lot of iron in the motor, saturation has less of an effect and there may be an opportunity to increase this variable. The range for this variable would typically be 4 to 7.

D.2.3.4 MBEMFCOMP

A back-emf compensator has been added to the current loop. Its purpose is to adjust the current loop gain for the voltage at the drives output terminals. The more practical results of the compensator is to control the current loop during full speed reversals. The default setting is 50 percent compensation. Some motors may perform better at 80 percent while others perform better at 20 percent. This compensator can have a negative side effect. The compensator is predictive and requires that the motor's BEMF is correctly aligned to the prediction. Misalignment of the feedback to the device can cause current loop instability if this gain is high.

D.2.4 Drive Commutation Control Parameters

This group of entries is used to modify the commutation angle. The commutation angle refers to the alignment of the SERVOSTAR's output voltage with respect to the motor's back EMF waveform. The motor's output torque can often be optimized by advancing the commutation angle proportional to the motor's velocity and current. Torque advance parameters advance the angle proportional to current using a two-slope straight line interpolation algorithm. Speed advance parameters advance the angle proportional to speed, also using a two-slope straight line interpolation algorithm.

D.2.4.1 Torque Adv P

The torque advance at the peak current is known as MLANGLP within the drive. The units are in electrical degrees. (Note that the drive may not actually be able to produce this level of current). The straight line interpolation begins at MLANGLC and extends to this point. See paragraph D.2.4.2

D.2.4.2 Torque Adv C

The torque advance at continuous current is known as MLANGLC within the drive. The units are in electrical degrees. The straight line starts at zero degrees for zero current and extends to this point.

D.2.4.3 Torque Angle Control Rules

As stated, these two variables establish the two-slope line which will advance the commutation angle with respect to the amount of current commanded to the motor. Typically, Torque Angle Advance will produce relatively smaller increases in performance for surface magnet type motors. It is recommended for surface magnet motors that torque advance at continuous current be set to 5, and the torque advance at peak current be set to 10. For internal magnet motor designs the system performance can often enhanced by larger values in these parameters. For these motors, it is recommended that MLANGLC be set to 8 and MLANGLP be set to 25.

D.2.4.4 Speed Adv 1/2

The speed advance at one half the motor's rated velocity is known as MVANGLH within the drive. The units are in electrical degrees. The straight line starts at zero speed and extends to this point.

D.2.4.5 Speed Adv at V

The speed advance at the motor's rated velocity is known as MVANGLF within the drive. The units are in electrical degrees. The straight line starts at one-half the motor's rated speed and extends to this point.

D.2.4.6 Speed Advance Rules

The speed advance at one half speed (MVANGLF) effectively compensates for system loop delays and 20 is a good number for this entry. The speed advance at full speed (MVANGLF) can actually serve two functions. One is to continue to compensate for the system delays. The other is to actually advance the commutation angle relative to the back EMF waveform, which is effectively reducing the back EMF, allowing the drive to deliver more current through the motor, thereby getting horsepower from it outside the limits which would normally be established by the back EMF and the BUS voltage. The DEFAULT variables in all of these torque angle advance constants is a good place to start for systems used in the field.

D.3 Setting Feedback Parameters

The SERVO**STAR** is available in resolver or encoder feedback versions. Select the Resolver Tab for resolver based units or the Encoder Tab for encoder based systems. The Halls Tab is actually a subset of the Encoder version.

D.3.1 Resolver Entries

These entries pertain only to resolver-based SERVO**STAR**'s.

D.3.1.1 # Poles

The entry is called Resolver Poles. Within the drive this variable is known as MRESPOLES. This variable allows the SERVO**STAR** to operate with resolvers with more than two magnetic poles. The normal setting here is two for most systems. However, multi-speed resolvers can be accommodated by adjusting this variable. Again, the units are magnetic poles. This entry is used by the commutation and velocity loops to correctly configure the drive.

D.3.1.2 Phase

The entry is called Motor Phase. Within the drive this variable is known as MPHASE. This entry provides a means to align the resolver position with the motor's electrical position in firmware. It is recommended that this variable be kept at zero and use the ZERO function (described elsewhere in the manual) to align the resolver to the motor's back EMF. In a situation where this is not practical this function may be used. The entry is in mechanical degrees and represents the actual misalignment between the resolver's absolute zero and the motor position where the A-phase back EMF waveform is at its' zero-crossing.

D.3.2 Encoder Entries

These entries pertain only to encoder-based systems.

D.3.2.1 Type

Within the drive this variable is known as MENCTYPE. This allows the **SERVOSTAR** to operate with a wide range of encoder formats. The details of this entry are explained in Section 6 of this manual.

D.3.2.2 Resolution

Within the drive this variable is known as MENCRES. This is the line count of the encoder with respect to one revolution of the rotor. It is the line count, not the quadrature decoded bit count. This entry is used to establish proper commutation and velocity loop setup.

D.3.2.3 Offset

Within the drive this variable is known as MENCOFF. This entry sets a relative location between the marker channel of the encoder and the actual back EMF wave form phasing of the motor . This variable can automatically be set by using the motor encoder initialize command set (associated with the type as indicated within this tab). For systems that have hall channels embedded within the encoder, and the hall channels aligned to the back EMF at the factory, this variable is a repeatable variable. For systems where there is no alignment between the index and the motor's back EMF, this variable will need to be set individually on each system or when the encoder is replaced or otherwise adjusted. Refer to paragraph D.5.1.2. The units that are entered for Encoder Offset are in quadrature decoded bit counts, not encoder lines.

D.3.2.4 Halls Tab

When using the encoder type with hall channels, initial commutation information is acquired by these channels. There are no industrial standards sequencing the hall channels and this tab allows the channels to be interpreted as inverted within the drive to accommodate any needs (See Appendix C for system phasing information). The most common application is for 120-degree sensors and normally, no channels are inverted. The next most common setting is to invert the B channel for systems providing 'sixty-degree' hall sensing. Each inversion has its own variable in the drive: MHINVA, MHINVB, and MHINVC.

D.4 Saving Data

The CONFIG command must be issued anytime one of these basic entries (or drive variables) are changed. It causes the **SERVOSTAR** to recalculate all internal settings. MotionLink does this automatically upon sending the data to the drive.

The parameters should now be saved. They should be written to the drive's EE memory either by using the MotionLink function or by issuing a SAVE command. The user should also backup this data in both a User

Database file and a Backup file. Note that the VBUS variable is not a database variable and is only archived by the Backup function.

D.5 Start Up & Trouble Shooting

An important part of field setup of the SERVOSTAR system is to be certain that the system is phased correctly. The wiring diagrams indicate nominal system connections but do not necessarily cover all possible system configurations. Appendix C is included to help determine if the appropriate electrical signals are present. After determining that the system appears to be wired correctly, a power up sequence followed in the installation section of the manual can be used to start up the system. Caution should be exercised on the initial startup and disconnecting the motor from any load provides the safest condition. Some conditions of having done something wrong through these procedures include instability, motor lock-up, or run away. Safety must be a foremost consideration and all safety precautions within this manual should be reviewed.

D.5.1 Commutation Loop

Problems with commutation can occur on both resolver based systems and encoder based systems and may exhibit themselves in such symptoms as the motor locking up in a torque detent, weak or little torque out of the motor, unequal torque in both directions of rotation, or instability including system run away.

D.5.1.1 Encoder System Operation Explained

An explanation of how encoder-based systems operate with the SERVOSTAR can aid in trouble shooting and start up. The most typical encoder is the Encoder Type zero which supplies Hall Effect channels, incremental A quad B signals, and an index pulse signal. When the SERVOSTAR powers up it reads the hall code. The hall code read by the SERVOSTAR must accurately indicate the motor position (in electrical terms). The drive assumes that the appropriate commutation angle is halfway between the hall code angle. As the motor traverses and crosses the first hall change state, the drive now has a better idea of where the commutation angle should be and adjusts the commutation angle accordingly. Hall channels (especially discrete hall devices) tend to be accurate to about five to fifteen degrees. The system now looks for the marker pulse and will readjust the commutation angle to a more exact location as established by the motor encoder offset variable (MENCOFF).

D.5.1.2 Adjusting the Encoder Offset

A poorly set Offset (MENCOFF) will cause the motor to, in best case, have more torque in one direction than in the other direction. It is possible to determine the correct setting in the field. Command the motor (in velocity mode) to run at about three-quarter speed in one direction and read the I variable (the actual current through the motor). Note that the I variable may jump around if the average time variable (AVGTIME) is set to one. The AVGTIME can be set to sixty-four for a more stable reading. Read the I command several times while the motor is moving in this direction and note the typical value of current required to move the motor in that direction. Now reversing the motor's direction (keeping the speed the same) should result in the same current. An imbalance of the current in two directions indicates that the commutation angle is not accurately set. The Offset entry or the MENCOFF variable may be trimmed to achieve a balance. Once that variable is set it should be written to EE memory using the SAVE command or the write to the EE function from MotionLink. Adjustment of this variable can be made 'on the fly' and the CONFIG command is not required.

D.5.2 Resolver System Phase Adjustment

The adjustment of this parameter is the same as for encoder-based systems. See the above section. In resolver-based systems where different currents are returned for both directions, adjusting the resolver zero mechanically is the preferred way to handle to balance the system. Again, see the ZERO command. In systems where it is less practical to adjust the resolver, the Phase entry or MPHASE variable may be used to adjust the commutation angle.

Again, once this is set correctly it must be written to EE memory. Adjustment of this variable can be made 'on the fly' and the CONFIG command is not required.

D.5.3 System Stability

System instability can manifest itself in a couple of different symptoms. Velocity loop stability is explained in Section 4 of this manual in the paragraphs related to tuning. Current loop instability can cause Power Stage faults (if the gain is too low) or audible noise and excessive motor heating (if gain is too high). If current loop stability problems occur, the easiest way to debug the system is in a Torque Mode. Proper tuning of the current loops at the factory involves a series of complicated steps. The easiest way to compensate for problems in the field is to remember that low gain can cause Power Stage faults (current overshoot) and excessive gain can cause motor heating and audible noise. Increasing the Motor Inductance entry (MLMIN) increases gain. Lowering Bus Voltage (VBUS) also increases gain. The Adaptive Gain entries (MLGAINP and MLGAINZ) can also be used to scale the gains.

Be certain that the VBUS variable is set correctly. On systems using multiple axes, regeneration of one axis may cause the actual BUS voltage of the drive to raise to up to 400 volts DC which is a substantial increase of gain to the current loops from a nominal setting of VBUS of 325. In multiple axis systems where one axis makes an audible noise while another axis is decelerating, the VBUS variable can be raised to 390 - 400 volts to compensate for the increased BUS voltage during that period of time.

D.5.4.1 Notes on Factory Procedures

To correctly configure the current loops, the factory will start by locking the motor shaft into position where the A-Phase current is at maximum output. A step function is then applied to the current loop and the A phase current is monitored for stability and overshoot.

APPENDIX E

LINEAR MOTOR APPLICATIONS

E.1 SERVOSTAR / MOTION LINK - CONVERTING ROTARY PARAMETERS REQUIRED FOR USE WITH LINEAR PARAMETERS FOR LINEAR MOTORS.

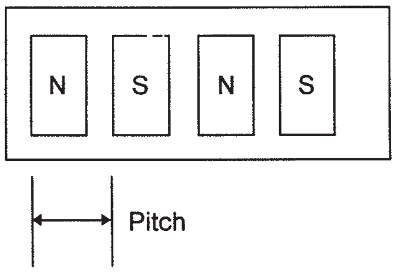
All parameter units for the SERVO**STAR** are designed for rotary motors. In order to use a linear motor with this amplifier, it is necessary to convert some of the linear motor units into numbers that can be used by the SERVO**STAR**. Some numbers can be used directly (peak and continuous current for example). Numbers relating to velocity, acceleration, and encoders must be converted. This appendix will explain the conversion procedure to successfully integrate a SERVO**STAR** and linear motor. For each step of the conversion process, an example of the calculations will be provided. This appendix assumes that the user is familiar with the SERVO**STAR** command set and MotionLink software.

E.2 PARAMETER CONVERSION

Determine the distance of 360 electrical degrees.

Measure the pitch of the magnets. This is the distance from the edge of the first magnet to the edge of the next magnet. This is referred to as the pole pitch.

Multiply the pole pitch by two to get the distance of 360 electrical degrees. Since the magnets are laid out with alternate magnetic fields, (North, South, North, South), two magnets represent 360 degrees.



Ex. The distance between two magnets is the pole pitch = 16 mm.
Find the 360 degree electrical equivalent:

360 degrees = 16 mm x 2 = 32 mm

32 mm = 2 motor poles

32 mm will be equivalent linear distance of 1 revolution.

Set the parameter **MPOLES = 2** (This establishes that 1 revolution equal 2 magnetic poles).

E.3 FIND THE EQUIVALENT ENCODER RESOLUTION FOR ONE ROTARY REVOLUTION.

This is done by calculating the number of encoder lines for 360 degree Electrical cycle.

Convert lines/mm to lines/revolution.

Ex. An encoder has a resolution of 50 lines/mm.

Lines/Rev = Encoder Resolution x Pole Pitch x 360 Degree electrical eq.

$$= \frac{50\text{ lines}}{1\text{mm}} \times \frac{32\text{ mm}}{2\text{ poles}} \times \frac{2\text{ poles}}{1\text{ revolution}}$$

= 1600 lines/revolution

Set the parameter **MENCRES = 1600.**

Ex. An encoder has a resolution of 50 lines/mm with a 10 X multiplier Interpolation device.

Lines/Rev = Encoder Resolution x Multiplier factor x Pole Pitch x 360 Degree electrical eq.

$$= \frac{50\text{ lines}}{1\text{mm}} \times 10 \times \frac{32\text{ mm}}{2\text{ poles}} \times \frac{2\text{ poles}}{1\text{ revolution}}$$

= 16000 lines/revolution

Set the parameter **MENCRES = 16000.**

Note:

The encoder input requirement for the drive is A quad B signals. If you are using an internal or external logic multiplier with your encoder, you must include this in the encoder calculation.

E.4 VELOCITY CONVERSION

Velocity units in the SERVO**STAR** are based on Revolutions per Minute (RPM), while linear stages are specified in units of inches per second (in/sec) or millimeters per second (mm/sec). Since we have already calculated a linear equivalent of one revolution in step E.2, it is easy to convert in/sec to RPM.

Convert in/sec to RPM.

Ex. Calculate the equivalent linear velocity of 1000 RPM
Convert 1000 RPM to revolutions per second (RPS):

$$\frac{1000\text{ Revolutions}}{1\text{ minute}} \times \frac{1\text{ minute}}{60\text{ seconds}} = 16.6667\text{ Revolutions/Second}$$

Multiply by the linear equivalent number from step 1.

$$\frac{16.6667 \text{ Revolutions}}{1 \text{ second}} \times \frac{32 \text{ mm}}{1 \text{ revolution}} = 533.33 \text{ mm/sec}$$

Convert to inches/sec

$$\frac{533.33 \text{ mm}}{1 \text{ second}} \times \frac{1 \text{ inch}}{25.4 \text{ mm}} = 20.99 \text{ inches/sec}$$

Therefore, the linear equivalent of 1000 RPM = 20.99 inches/sec.

E.5 BACK EMF CONVERSION

The SERVOSTAR amplifier requires units Volts per 1000 RPM, the linear stage units are Volts/inches/second, or Volts/mm/second. To convert the linear units to rotary units, the velocity of 1000 RPM must be used.

Convert V/in/sec to Volts/1000 RPM

Ex. The Ke of the linear motor is specified as 1.15 Volts/inches/sec.

$$\frac{1.15 \text{ Volts}}{\text{inches/sec}} \times \frac{20.99 \text{ inches/sec}}{1000 \text{ RPM}} = 24.14 \text{ Volts/Krpm}$$

Set the parameter **MBEMF = 24**

Note:

After setting this parameter, the amplifier will automatically calculate and set the value for the Kt (torque constant), and update the parameters MKT and MKTI.

E.6 SETTING THE VELOCITY LIMITS

The parameters **MSPEED, VLIM, VSCALE, VMAX,** and **VOSPD** are used to define the maximum speed of the system. Linear motors are capable of reaching very high velocities and the limiting factors are usually defined by external devices such as the amplifier's power supply, linear scale, or external controller. Determine the maximum speed that you wish the stage to travel at and use the conversion number from step E.4 to calculate the amplifier parameters.

Ex. The maximum desired speed for the stage is 20 inches/sec

$$20 \text{ inches/sec} \times \frac{1000 \text{ RPM}}{21 \text{ inches/sec}} = 952.38 \text{ RPM}$$

Set the parameters **MSPEED = 952**
VLIM = 952
VSCALE = 952

Note:

The parameters **VMAX** and **VOSPD** are calculated and set automatically.

E.7 ACCELERATION AND DECELERATION DECELERATION PARAMETERS

The parameters **ACC** and **DEC** control the acceleration and deceleration in **OPMODE 0** and **OPMODE 1**, if **PROFMODE = 1**. The units RPM/sec define the rate of change per second. To calculate linear acceleration, use the velocity conversion from step E.4. 20.99 in/sec = 1000 RPM

Calculate the equivalent of 1 G Acceleration.

Ex. Convert 1 G acceleration to RPM/sec.

$$1\text{ G} \times \frac{386.4\text{ in/sec/sec}}{1\text{ G}} \times \frac{1000\text{ RPM}}{20.99\text{ inches/sec}} = 18408\text{ RPM/sec}$$

Therefore, the rotary equivalent of 1 G acceleration = 18408 RPM/sec

Ex. A 0.5 G acceleration is required, convert to RPM/sec:

0.5 G x 18408 = 9204 RPM/sec

Set the parameter **ACC = 9024**

Note:
The acceleration rate will remain constant regardless of the commanded velocity.

E.8 SAMPLE SERVOSTAR COMPENSATION PARAMETERS FOR A LINEAR MOTOR.

Motor Characteristics:

Back EMF =1.15 Volts/inches/sec
Encoder Resolution = 50 lines/mm
Pole Pitch = 30 mm

Continuous Current = 3 Amps
Peak Current = 5.7 Amps

MOTOR="LINEAR
MIPEAK=57
MICONT=30
MSPEED=2000
MBEMF=45
MCRES=3000
MENCOFF=5219
MJ=200
MKT=744
MKTI=549
MLMAX=2500
MLMIN=2000
MPHASE=0
MPOLES=2
MRES=0
MRESPOLES=0
MTANGLC=0
MTANGLP=0
MVANGLF=10
MVANGLH=5
VBUS=160
DIPEAK=200
DICONT=200
ILIM=50
ICONT=150
ISCALE=1000
IZERO=25
VLIM=2000
VOSPD=2400
VSCALE=120
ACC=20000
ACKMODE=0
ACTFAULT=0
ANDB=0
ANOFF=0
AVGTIME=0
BW=50
COMPMODE=2
D=0 0 0 0 0 0
DEC=20000
DECSTOP=5000
DIR=1
ECHO=1
ENCOUNT=3000
FOLDMODE=0
FOLDR=15000
FOLDT=25
GETMODE=0
GV=1
GVI=0
H=0 0 0 0 0 0
KV=0
KVI=0
KVFR=0
LIMDIS=0
LMJR=10
MSG=1
OPMODE=0
PROFMODE=1
PROMPT=1
PWMFRQ=16
R=0 0 0 0 0 0
RELAYMODE=0
THERMODE=0
UNITS=0
UVRECOVER=0
CONFIG

E.9 CONNECTING A LINEAR MOTOR TO THE SERVOSTAR

This procedure assumes :

1. single-ended hall effect commutation.
2. the user is familiar with the Motion Link program.
3. no commutation information is known about the motor.
4. Motor parameters are installed in the drive.

E.9.1 Determining Motor And Hall Effect Connections

1. Connect the encoder and hall effect cables to connector C2. Do not connect the motor leads at this time. Jumper pins 22, 23, 24, to pin 20 (or 19, whichever is free). Connect the three hall signals to pins 9, 10, and 11 .

2. Power up the driver and establish communications using Motion Link.

3. Determine the table direction that provides positive counting on the PRD counter. Enter the terminal mode, and type PRD<LF> to find the value of the PRD counter. Push the stage .5" and again type PRD<LF>. By comparing the two values, you can determine the slide direction for positive counting.

4. Connect channel 1 of a storage scope to C2 pin 9, ref to rtn (pin 8). Connect channel 2 to pin 10, and set both channels to 2 volts/div. Push the stage at a moderate speed, and adjust the sweep time and vertical position to obtain a good image of both hall effect signals (there will be two square waves on the screen).

5. Push the stage in the positive (PRD) direction and store the waveform. Compare the phase shift between the two waveforms. The waveform on channel 2 should lag the waveform on channel 1 by 240°. If the lag is only 120°, swap the wires on pins 10 and 11, and check the phasing again. See figure 1.

6. Connect cha 2 of the scope to any motor lead, ref to any other motor lead. Set to 5v/div (cha 1 still connected to pin 9, ref to rtn, set to 2v/div). Push the stage in the positive direction, and compare the two waveforms.

7. Exchange the motor leads connected to cha 2 until you find that the motor back emf is 180° out of phase with the hall signal (positive direction only!). See Figure 1. Label the motor lead connected to the probe as A, and the lead connected to the ref as B. Label the remaining lead C.

8. Connect cha 1 to pin 10, ref to rtn, and connect cha 2 to motor lead B, ref to lead C. Push the stage in the positive direction and observe the waveform. If the back emf is 180° out of phase with the hall, then the phasing is complete. CAUTION: If the waveforms are not 180° out of phase, then the hall signal logic must be inverted. To invert the logic levels of the hall effect signals, power down the driver and do the following: See figure 2.

   1. Remove the jumpers on pins 22, 23, 24, and 20 (or 19). Install three pull-up resistors, and connect to C5 pin 1.

   2. a. Move the wire from pin 9 to pin 22.
      b. Move the wire from pin 10 to pin 23.
      c. Move the wire from pin 11 to pin 24.

3. Jumper pins 8, 9, 10, and 11.

4. Relabel the A motor lead as B, and the B motor lead as A.

E.9.2 Connecting The Motor And Applying Drive Power

1. Power down the driver and remove all scope leads.

2. Wait a few minutes for the power supply to drain, and connect the motor leads A, B, and C to the driver terminals Ma, Mb, and Mc.

3. Make sure the Enable input is open (this will disable the drive), and apply power to the drive.

4. Set ILIM to 50 to limit the peak current, enable the driver, and push the stage by hand. It should feel smooth, with no dead spots. If it seems to be OK, increase ILIM and give the stage a jog command- use J 40. Check for smoothness of motion.

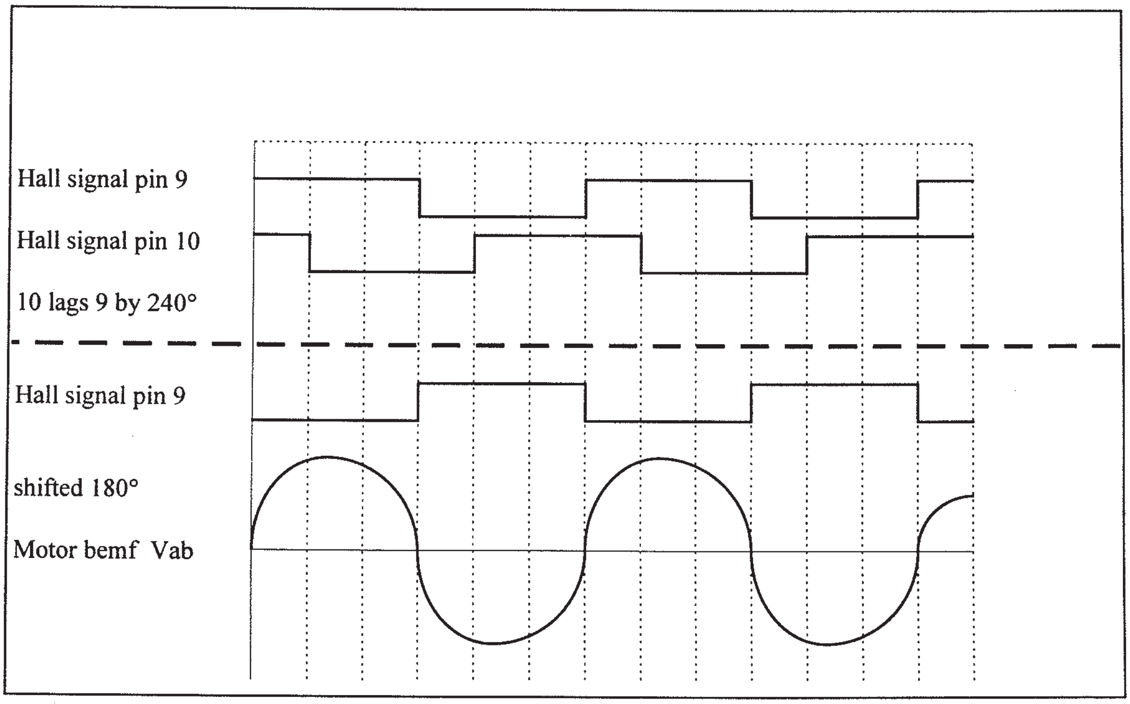


Figure 1. Hall Effect and Motor Phasing

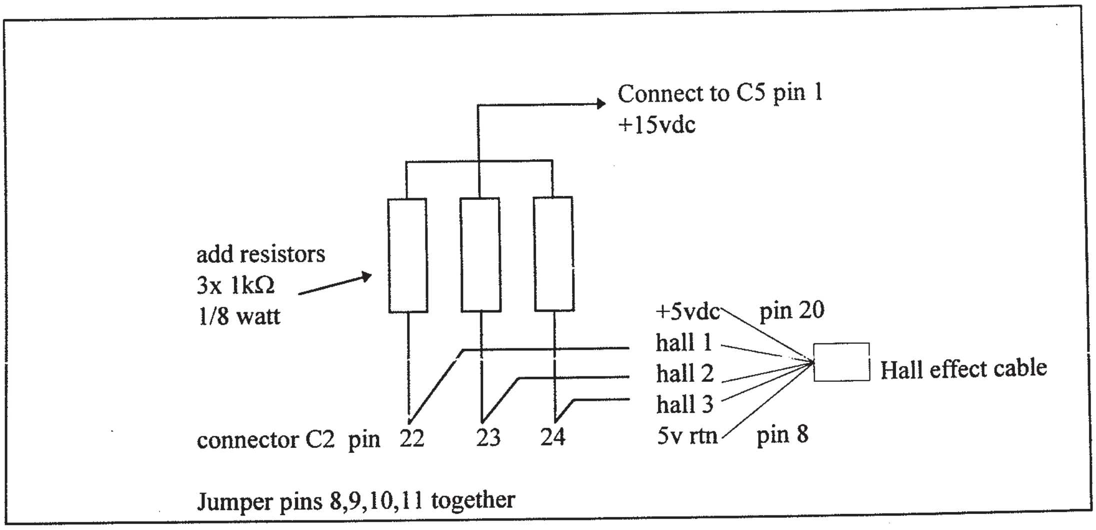


Figure 2. Inverting Hall Effect Logic

APPENDIX F

HOOK-UP DATA

1. GOLDLINE MOTOR WITH ENCODER FEEDBACK

MOTOR RECEPTACLE
SOURIAU 851-02R14-5P50 (CRIMP).
OPTIONAL STRAIGHT MATING PLUG
SOURIAU 851-06EC14-5S50 (SOLDER CUPS).
(5) #16 GUAGE CONTACTS

PHASE A (BROWN LEAD) - PIN "A".
PHASE B (RED LEAD) - PIN "B".
PHASE C (WHITE LEAD) - PIN "C".
CASE GROUND (GREEN w/ YELLOW STRIPE LEAD) - PIN "D".

CAUTION: STATIC SENSITIVE DEVICE

SYSTEM ENCODER, THERMOSTAT, AND OPTIONAL BRAKE RECEPTACLE, SOURIAU 851-02R14-19P50.
OPTIONAL STRAIGHT MATING PLUG SOURIAU 851-06EC14-19SW50.

PIN LETTER	WIRE COLOR (REF ONLY SEE NOTE #9)	FUNCTION	
A	BLUE	A	DATA CHANNELS
M	BLUE / BLACK	$\overline{A}$	DATA CHANNELS
B	GREEN	B	DATA CHANNELS
C	GREEN / BLACK	$\overline{B}$	DATA CHANNELS
D	YELLOW	Z	DATA CHANNELS
E	YELLOW / BLACK	$\overline{Z}$	DATA CHANNELS
F	BROWN	H_{AB}	COMMUTATION CHANNELS NOTE #2
G	BROWN / BLACK	$\overline{H}_{AB}$	COMMUTATION CHANNELS NOTE #2
H	GRAY	H_{BC}	COMMUTATION CHANNELS NOTE #2
J	GRAY / BLACK	$\overline{H}_{BC}$	COMMUTATION CHANNELS NOTE #2
K	WHITE	H_{CA}	COMMUTATION CHANNELS NOTE #2
L	WHITE / BLACK	$\overline{H}_{CA}$	COMMUTATION CHANNELS NOTE #2
N	BLUE - UNJACKETED	OPTIONAL BRAKE	
P	BLUE - UNJACKETED	OPTIONAL BRAKE	
R	RED	+5V	
S	BLACK	COMMON	
T	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
U	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
V	BARE - SHIELD	SHIELD	

NOTES:
1 - REFER TO DOCUMENT ED-27313 FOR DETAILED ENCODER SPECIFICATIONS, SIGNAL PHASING, AND APPLICATION NOTES.
2 - SIMULATED HALL OUTPUTS FOR MOTOR INITIALIZATION ONLY. PROPER MOTOR PERFORMANCE REQUIRES SINEWAVE COMMUTATION OFF DATA CHANNELS.
3 - WITH PHASE SEQUENCE A, B, C MOTOR ROTATION SHALL BE CCW FACING MOUNTING END, (OR A, C, B EQUALS CW DIRECTION).
4 - ENCODER DATA CHANNEL A LEADS CHANNEL B IN CCW DIRECTION.
5 - ALL ENCODER CHANNELS UTILIZE LINE DRIVER AM26LS31 OUTPUTS.
6 - THERMOSTAT PRESET TO OPEN AT 170°C ± 5°C AND CLOSE AT 132°C ± 5°C NORMALLY CLOSED. CONTACTS RATED AT 4 AMPS, 120 V.A.C.
7 - BRAKE DATA: USE CONTINUOUS STATIC HOLDING AND EMERGENCY DYNAMIC STOPPING. TYPE POWER ON, BRAKE OFF, RATING STATIC 2.3 Nm. FOR ELECTRICAL RATING, SEE TABLE BELOW.
8 - USE IM-22 FOR STRIPPING AND CRIMPING INFORMATION.
9 - ENCODER WIRE COLORS ARE FOR REFERENCE ONLY AND ARE SUBJECT TO CHANGE. PIN FUNCTION IS GUARANTEED.

MODEL	BRAKE VOLTAGE	HOLDING CURRENT
B3	24 V.D.C.	0.57 AMPS
B2	90 V.D.C.	0.20 AMPS

HOOK-UP DATA
BEX-10X-X-A2, B2
(SOURCE DATA - KOLLMORGEN DRAWING NUMBER HD-109 ISSUE 1).

MOTOR RECEPTACLE
MS-3102E-18-10P (CRIMP).
OPTIONAL STRAIGHT MATING PLUG
MS-3106E-18-10S (SOLDER CUPS).
(4) #12 GUAGE CONTACTS

PHASE A (BROWN LEAD) - PIN "A".
PHASE B (RED LEAD) - PIN "B".
PHASE C (WHITE LEAD) - PIN "C".
CASE GROUND (GREEN w/ YELLOW STRIPE LEAD) - PIN "D".

CAUTION: STATIC SENSITIVE DEVICE

SYSTEM ENCODER, THERMOSTAT, AND OPTIONAL BRAKE RECEPTACLE, MS-3102E-22-14P-W.
OPTIONAL STRAIGHT MATING PLUG MS-3106E-22-14S-W.

PIN LETTER	WIRE COLOR (REF ONLY SEE NOTE #9)	FUNCTION	
A	BLUE	A	DATA CHANNELS
M	BLUE / BLACK	$\overline{A}$	
B	GREEN	B	
C	GREEN / BLACK	$\overline{B}$	
D	YELLOW	Z	
E	YELLOW / BLACK	$\overline{Z}$	
F	BROWN	H_{AB}	COMMUTATION CHANNELS NOTE #2
G	BROWN / BLACK	$\overline{H}_{AB}$	
H	GRAY	H_{BC}	
J	GRAY / BLACK	$\overline{H}_{BC}$	
K	WHITE	H_{CA}	
L	WHITE / BLACK	$\overline{H}_{CA}$	
N	BLUE - UNJACKETED	OPTIONAL BRAKE	
P	BLUE - UNJACKETED	OPTIONAL BRAKE	
R	RED	+5V	
S	BLACK	COMMON	
T	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
U	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
V	BARE - SHIELD	SHIELD	

NOTES:
1 - REFER TO DOCUMENT ED-27313 FOR DETAILED ENCODER SPECIFICATIONS, SIGNAL PHASING, AND APPLICATION NOTES.
2 - SIMULATED HALL OUTPUTS FOR MOTOR INITIALIZATION ONLY. PROPER MOTOR PERFORMANCE REQUIRES SINEWAVE COMMUTATION OFF DATA CHANNELS.
3 - WITH PHASE SEQUENCE A, B, C MOTOR ROTATION SHALL BE CCW FACING MOUNTING END, (OR A, C, B EQUALS CW DIRECTION).
4 - ENCODER DATA CHANNEL A LEADS CHANNEL B IN CCW DIRECTION.
5 - ALL ENCODER CHANNELS UTILIZE LINE DRIVER AM26LS31 OUTPUTS.
6 - THERMOSTAT PRESET TO OPEN AT 170°C ± 5°C AND CLOSE AT 132°C ± 5°C NORMALLY CLOSED. CONTACTS RATED AT 4 AMPS, 120 V.A.C.
7 - BRAKE DATA: USE CONTINUOUS STATIC HOLDING AND EMERGENCY DYNAMIC STOPPING. TYPE POWER ON, BRAKE OFF, RATING STATIC 6.0 Nm. FOR ELECTRICAL RATING, SEE TABLE BELOW.
8 - USE IM-22 FOR STRIPPING AND CRIMPING INFORMATION.
9 - ENCODER WIRE COLORS ARE FOR REFERENCE ONLY AND ARE SUBJECT TO CHANGE. PIN FUNCTION IS GUARANTEED.

MODEL	BRAKE VOLTAGE	HOLDING CURRENT
B3	24 V.D.C.	0.57 AMPS
B2	90 V.D.C.	0.12 AMPS

HOOK-UP DATA
BEX-20X-X-A2, B2
(SOURCE DATA - KOLLMORGEN DRAWING NUMBER HD-222 ISSUE 1).

MOTOR RECEPTACLE
MS-3102E-18-10P (CRIMP).
OPTIONAL STRAIGHT MATING PLUG
MS-3106E-18-10S (SOLDER CUPS).
(4) #12 GUAGE CONTACTS

PHASE A (BROWN LEAD) - PIN "A".
PHASE B (RED LEAD) - PIN "B".
PHASE C (WHITE LEAD) - PIN "C".
CASE GROUND (GREEN w/
YELLOW STRIPE LEAD) - PIN "D".

CAUTION: STATIC SENSITIVE DEVICE

SYSTEM ENCODER, THERMOSTAT, AND OPTIONAL BRAKE RECEPTACLE, MS-3102E-22-14P-W.
OPTIONAL STRAIGHT MATING PLUG MS-3106E-22-14S-W.

PIN LETTER	WIRE COLOR (REF ONLY SEE NOTE #9)	FUNCTION	
A	BLUE	A	DATA CHANNELS
M	BLUE / BLACK	$\overline{A}$	DATA CHANNELS
B	GREEN	B	DATA CHANNELS
C	GREEN / BLACK	$\overline{B}$	DATA CHANNELS
D	YELLOW	Z	DATA CHANNELS
E	YELLOW / BLACK	$\overline{Z}$	DATA CHANNELS
F	BROWN	H_{AB}	COMMUTATION CHANNELS NOTE #2
G	BROWN / BLACK	$\overline{H}_{AB}$	COMMUTATION CHANNELS NOTE #2
H	GRAY	H_{BC}	COMMUTATION CHANNELS NOTE #2
J	GRAY / BLACK	$\overline{H}_{BC}$	COMMUTATION CHANNELS NOTE #2
K	WHITE	H_{CA}	COMMUTATION CHANNELS NOTE #2
L	WHITE / BLACK	$\overline{H}_{CA}$	COMMUTATION CHANNELS NOTE #2
N	BLUE - UNJACKETED	OPTIONAL BRAKE	
P	BLUE - UNJACKETED	OPTIONAL BRAKE	
R	RED	+5V	
S	BLACK	COMMON	
T	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
U	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
V	BARE - SHIELD	SHIELD	

NOTES:
1 - REFER TO DOCUMENT ED-27313 FOR DETAILED ENCODER SPECIFICATIONS, SIGNAL PHASING, AND APPLICATION NOTES.
2 - SIMULATED HALL OUTPUTS FOR MOTOR INITIALIZATION ONLY. PROPER MOTOR PERFORMANCE REQUIRES SINEWAVE COMMUTATION OFF DATA CHANNELS.
3 - WITH PHASE SEQUENCE A, B, C MOTOR ROTATION SHALL BE CCW FACING MOUNTING END, (OR A, C, B EQUALS CW DIRECTION).
4 - ENCODER DATA CHANNEL A LEADS CHANNEL B IN CCW DIRECTION.
5 - ALL ENCODER CHANNELS UTILIZE LINE DRIVER AM26LS31 OUTPUTS.
6 - THERMOSTAT PRESET TO OPEN AT 170°C ± 5°C AND CLOSE AT 132°C ± 5°C NORMALLY CLOSED. CONTACTS RATED AT 4 AMPS, 120 V.A.C.
7 - BRAKE DATA: USE CONTINUOUS STATIC HOLDING AND EMERGENCY DYNAMIC STOPPING. TYPE POWER ON, BRAKE OFF, RATING STATIC 8.0 Nm. FOR ELECTRICAL RATING, SEE TABLE BELOW.
8 - USE IM-22 FOR STRIPPING AND CRIMPING INFORMATION.
9 - ENCODER WIRE COLORS ARE FOR REFERENCE ONLY AND ARE SUBJECT TO CHANGE. PIN FUNCTION IS GUARANTEED.

MODEL	BRAKE VOLTAGE	HOLDING CURRENT
B3	24 V.D.C.	0.93 AMPS
B2	90 V.D.C.	0.28 AMPS

HOOK-UP DATA
BEX-40X-X-A2, B2
(SOURCE DATA - KOLLMORGEN DRAWING NUMBER HD-428 ISSUE 1).

MOTOR RECEPTACLE
MS-3102E-22-22P (CRIMP).
OPTIONAL STRAIGHT MATING PLUG
MS-3106E-22-22S (SOLDER CUPS).
(4) #8 GUAGE CONTACTS

PHASE A (BROWN LEAD) - PIN "A".
PHASE B (RED LEAD) - PIN "B".
PHASE C (WHITE LEAD) - PIN "C".
CASE GROUND (GREEN w/
YELLOW STRIPE LEAD) - PIN "D".

CAUTION: STATIC SENSITIVE DEVICE

SYSTEM ENCODER, THERMOSTAT, AND OPTIONAL BRAKE RECEPTACLE, MS-3102E-22-14P-W.
OPTIONAL STRAIGHT MATING PLUG MS-3106E-22-14S-W.

PIN LETTER	WIRE COLOR (REF ONLY SEE NOTE #9)	FUNCTION	
A	BLUE	A	DATA CHANNELS
M	BLUE / BLACK	$\overline{A}$	DATA CHANNELS
B	GREEN	B	DATA CHANNELS
C	GREEN / BLACK	$\overline{B}$	DATA CHANNELS
D	YELLOW	Z	DATA CHANNELS
E	YELLOW / BLACK	$\overline{Z}$	DATA CHANNELS
F	BROWN	H_{AB}	COMMUTATION CHANNELS NOTE #2
G	BROWN / BLACK	$\overline{H}_{AB}$	COMMUTATION CHANNELS NOTE #2
H	GRAY	H_{BC}	COMMUTATION CHANNELS NOTE #2
J	GRAY / BLACK	$\overline{H}_{BC}$	COMMUTATION CHANNELS NOTE #2
K	WHITE	H_{CA}	COMMUTATION CHANNELS NOTE #2
L	WHITE / BLACK	$\overline{H}_{CA}$	COMMUTATION CHANNELS NOTE #2
N	BLUE - UNJACKETED	OPTIONAL BRAKE	
P	BLUE - UNJACKETED	OPTIONAL BRAKE	
R	RED	+5V	
S	BLACK	COMMON	
T	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
U	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
V	BARE - SHIELD	SHIELD	

NOTES:
1 - REFER TO DOCUMENT ED-27313 FOR DETAILED ENCODER SPECIFICATIONS, SIGNAL PHASING, AND APPLICATION NOTES.
2 - SIMULATED HALL OUTPUTS FOR MOTOR INITIALIZATION ONLY. PROPER MOTOR PERFORMANCE REQUIRES SINEWAVE COMMUTATION OFF DATA CHANNELS.
3 - WITH PHASE SEQUENCE A, B, C MOTOR ROTATION SHALL BE CCW FACING MOUNTING END, (OR A, C, B EQUALS CW DIRECTION).
4 - ENCODER DATA CHANNEL A LEADS CHANNEL B IN CCW DIRECTION.
5 - ALL ENCODER CHANNELS UTILIZE LINE DRIVER AM26LS31 OUTPUTS.
6 - THERMOSTAT PRESET TO OPEN AT 170°C ± 5°C AND CLOSE AT 132°C ± 5°C NORMALLY CLOSED. CONTACTS RATED AT 4 AMPS, 120 V.A.C.
7 - BRAKE DATA: USE CONTINUOUS STATIC HOLDING AND EMERGENCY DYNAMIC STOPPING. TYPE POWER ON, BRAKE OFF, RATING STATIC 48.0 Nm. FOR ELECTRICAL RATING, SEE TABLE BELOW.
8 - USE IM-22 FOR STRIPPING AND CRIMPING INFORMATION.
9 - ENCODER WIRE COLORS ARE FOR REFERENCE ONLY AND ARE SUBJECT TO CHANGE. PIN FUNCTION IS GUARANTEED.

B3	24 V.D.C.	1.27 AMPS
B2	90 V.D.C.	0.39 AMPS
MODEL	**BRAKE VOLTAGE**	**HOLDING CURRENT**

HOOK-UP DATA
BEX-60X-X-A2, B2

(SOURCE DATA - KOLLMORGEN DRAWING NUMBER HD-621 ISSUE 1).

MOTOR RECEPTACLE
MS-3102E-22-22P (CRIMP).
OPTIONAL STRAIGHT MATING PLUG
MS-3106E-22-22S (SOLDER CUPS).
(4) #8 GUAGE CONTACTS

PHASE A (BROWN LEAD) - PIN "A".
PHASE B (RED LEAD) - PIN "B".
PHASE C (WHITE LEAD) - PIN "C".
CASE GROUND (GREEN w/
YELLOW STRIPE LEAD) - PIN "D".

CAUTION: STATIC SENSITIVE DEVICE

SYSTEM ENCODER, THERMOSTAT, AND OPTIONAL BRAKE RECEPTACLE, MS-3102E-22-14P-W.
OPTIONAL STRAIGHT MATING PLUG MS-3106E-22-14S-W.

PIN LETTER	WIRE COLOR (REF ONLY SEE NOTE #9)	FUNCTION	
A	BLUE	A	DATA CHANNELS
M	BLUE / BLACK	$\overline{A}$	DATA CHANNELS
B	GREEN	B	DATA CHANNELS
C	GREEN / BLACK	$\overline{B}$	DATA CHANNELS
D	YELLOW	Z	DATA CHANNELS
E	YELLOW / BLACK	$\overline{Z}$	DATA CHANNELS
F	BROWN	H_{AB}	COMMUTATION CHANNELS NOTE #2
G	BROWN / BLACK	$\overline{H}_{AB}$	COMMUTATION CHANNELS NOTE #2
H	GRAY	H_{BC}	COMMUTATION CHANNELS NOTE #2
J	GRAY / BLACK	$\overline{H}_{BC}$	COMMUTATION CHANNELS NOTE #2
K	WHITE	H_{CA}	COMMUTATION CHANNELS NOTE #2
L	WHITE / BLACK	$\overline{H}_{CA}$	COMMUTATION CHANNELS NOTE #2
N	BLUE - UNJACKETED	OPTIONAL BRAKE	
P	BLUE - UNJACKETED	OPTIONAL BRAKE	
R	RED	+5V	
S	BLACK	COMMON	
T	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
U	YELLOW W / MARKINGS UNJACKETED	THERMOSTAT	
V	BARE - SHIELD	SHIELD	

NOTES:
1 - REFER TO DOCUMENT ED-27313 FOR DETAILED ENCODER SPECIFICATIONS, SIGNAL PHASING, AND APPLICATION NOTES.
2 - SIMULATED HALL OUTPUTS FOR MOTOR INITIALIZATION ONLY. PROPER MOTOR PERFORMANCE REQUIRES SINEWAVE COMMUTATION OFF DATA CHANNELS.
3 - WITH PHASE SEQUENCE A, B, C MOTOR ROTATION SHALL BE CCW FACING MOUNTING END, (OR A, C, B EQUALS CW DIRECTION).
4 - ENCODER DATA CHANNEL A LEADS CHANNEL B IN CCW DIRECTION.
5 - ALL ENCODER CHANNELS UTILIZE LINE DRIVER AM26LS31 OUTPUTS.
6 - THERMOSTAT PRESET TO OPEN AT 170°C ± 5°C AND CLOSE AT 132°C ± 5°C NORMALLY CLOSED. CONTACTS RATED AT 4 AMPS, 120 V.A.C.
7 - BRAKE DATA: USE CONTINUOUS STATIC HOLDING AND EMERGENCY DYNAMIC STOPPING. TYPE POWER ON, BRAKE OFF, RATING STATIC 48.0 N·m. FOR ELECTRICAL RATING, SEE TABLE BELOW.
8 - USE IM-22 FOR STRIPPING AND CRIMPING INFORMATION.
9 - ENCODER WIRE COLORS ARE FOR REFERENCE ONLY AND ARE SUBJECT TO CHANGE. PIN FUNCTION IS GUARANTEED.

MODEL	BRAKE VOLTAGE	HOLDING CURRENT
B3	24 V.D.C.	1.27 AMPS
B2	90 V.D.C.	0.39 AMPS

HOOK-UP DATA
BEX-80X-X-A2, B2
(SOURCE DATA - KOLLMORGEN DRAWING NUMBER HD-814 ISSUE 1).

APPENDIX G

DYNAMIC BRAKING

G.1 TOPICAL OVERVIEW

Modern machinery has become much safer for the both the operator and the material being processed. Electronic ServoSystems have replaced mechanical linkages, large open gear trains, belts, and hydraulic rams. Added safety often requires that these systems be able to stop quickly upon demand. Most positioning systems do a fine job controlling these stops but have a fair degree of risk in not being able to reliable stop a system under certain failure modes. Many of these systems incorporate permanent magnet motors to provide the motion.

There are three categories for stopping a permanent magnet motor. The first is simply by request. A host positioner system performs this function. The next two categories include more failure modes. There are two categories for dynamically breaking a permanent magnet motor. This first is a fail-safe method where it is extremely important to stop the motor quickly and reliably for safety reasons. This method often carries a liability with it. The second need to stop a motor quickly is simply to prevent it from coasting. This method is applied where personnel safety is less of a issue than machine damage.

Fail-Safe Braking.
There are two accepted methods of braking for fail safe operation. Electronic means are not acceptable at this time due to reliability factors.

1.) The first method is to apply motor brakes. Motor brakes work well but have a limitation on how much stopping force they can apply and a relatively short cycle-life. This is often the best method because it will work over the widest range of failure modes.
2.) The second method is to have a contactor to short the motor windings through a resistor. The resistors are sized so that the motor currents do not exceed the de-mag current of the motor. The contactor should be DC rated which makes it relatively expensive and large. The resistors must be chosen to handle a large instantaneous power surge. The contactor should have an auxiliary contact that is used to disable the drive just before the main contacts short the motor windings.

Electronic Methods for getting the motor to stop
There are many ways to get a motor to stop electronic although the ability of these methods to stop the motor in all failure modes are limited. For example, a failed power device or system over voltage may prevent these means from functioning.

1.) A contactor can be used in conjunction with a current limiting resistor to short the DC bus of a drive system. This method uses the fact that the power output stage has back-diodes across each transistor. If the bus is collapses by the contactor / resistor short then the motors energy regenerates through the diodes where the current is limited by the resistor. The resistor must be sized to limit the current through these diodes so that they are not destroyed. In most cases, the maximum bus voltage and drive peak current can be used to calculate the value of the resistor. Care must be taken to operate the contactor correctly. The main DC bus power from the power supply can be disconnected by the same contactor allowing the resistor to drain only the energy in the motor and the drive's bus capacitor. Here, the contactor should reconnect the bus before the power supply is re energized. Again, the contactor must be DC rated. The contactor is operated by logic which will cause the contactor to drop out in the event of a request to stop or a drive fault. Auxiliary contacts on this contactor can be wired to drop out the contactor for the Power Supply.

2.) The current loops can be reconfigured upon a fault or a drive disable in a number of methods. Consideration should be given to the fact that the drive may not have feedback from the motor due to a feedback device fault. In other words, the drive should not be expected to commutate the motor in this mode.

A.) The transistors can force a regulated amount of current through two motor phases. The DC current will cause the motor to want to lock into a motor-pole detent. This method will cause the motor to bump as it rotates through the torque detents.

B.) The transistors can regulate constant and equal current in all phases of the motor. This method is better than the first because the failure of one transistor bridge can still stop the motor. Since there is no commutation, the motor will again grind to a bumping halt.

Both of these methods have the draw back that the drive will need to pump current out of the motor at times. This regeneration can cause the DC bus to pump up. The problem here is that the drive will limit the amount of voltage rise on the bus causing a drive shut down if the bus gets so high as to cause a catastrophic drive failure. The braking sequence needs to be aborted under these conditions.

C.) The transistors can be used to fire the all three motor leads to the same DC bus rail. The current generated from the motor flows through the transistor or it's back-diode to create a motor short. Regulating the maximum currents in this scenario is nearly impossible as the drive is not able to current though the back diodes. The drive, therefore, would need to know how to switch the transistors between shorting through the top and bottom rails to prevent more than one back diode from conducting at any given time. For example, positive current in two-of-three phases cause the drive to connect the transistors to the bottom rail and negative current in two-of-three phases would cause the drive to connect the transistors to the positive rail. Note that the current still need to be regulated to not allow it to exceed the drive's peak current capability or the motor's de-mag current.

D.) Another electronics means is to switch the transistors on (Top or Bottom only) and PWM the output stage enable. The switching duty of the enable must be such that the Ldi/dt never allows I to reach a level that could destroy the output silicon.

G.2 SERVOSTAR QUICK-STOP FEATURE

Braking a motor has definite legal implications. The only acceptable methods to brake a motor to protect personal safety do not include electronic means. Any method employed in a drive relying on active components should not be promoted as safe. There are times, however, when the objective of braking is not personal safety but other issues such as; protecting machinery, preventing a vertical load from free falling, and simply stopping a spindle quicker than a coast. Electronic braking of a motor using the existing drive electronics has great economic advantage: It is free.

A quick stop feature has been added to the ServoStar product and is available on special order only. The quick stop feature shorts the motor leads together under current control to quickly stop the motor. The feature can not be activated under two fault conditions: Power Stage Fault and Over Voltage Fault.

Note:
The feature also will cause the motor lead voltages to float above ground when this mode is activated. Even standard drive disabling does not guarantee this condition will not occur.

STOPMODE sets the mode of dynamic braking operation.

0 = no braking operation (default).
1 = brake on fault only.
2 = brake on fault and/or drive disable.

Note: Faults do not include Over Voltage or Power Stage Faults!

Type: switch mode (R/W) Range: 0, 1, 2
Units: n/a Default: 0
Opmodes: all Drive Status: en/dis EEPROM: yes

ISTOP sets the current command for the braking function. This variable is subject to the UNITS command.

Type: variable (R/W) Range: 0 to IMAX
Units: % of DIPEAK *0.1 Default: IMAX
Opmodes: all Drive Status: en/dis EEPROM: yes

APPENDIX H

SERVOSTAR APPLICATION NOTES

To receive a copy of the application notes below please contact the Kollmorgen Customer Support Network at any of the following locations:

Phone: 1-888-774-KCSN (5276)

Email: KMTG@Kollmorgen.com

Web Page: http://www.kollmorgen.com

Title	Model Number	Topics Discussed
Regeneration Requirements	A-SU-000-H	Kinetic energy calculations Resistor calculations Average and peak power
Power Supply Sizing	A-SU-001-H	Average power calculations Low Speed applications Logic and Main BUS sizing Transformer sizing
Pulse Following / Electronic Gearing	A-SU-002-H	OPMODE 4 Discussion GEARMODE, GEARI, GEARO Position Loop error and tuning Hookup and wiring information

INDEX

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- X -

- Y -

- Z -