Setting up an XPS with a direct drive stage and sin/cos encoder

SCOPE

This document aims to present a comprehensive and intuitive guide for wiring, connecting and configuring non-Newport direct drive stages with sin/cos encoders for operation with the XPS Motion Controller. All current generation of Newport Stages with ESP technology will configure automatically, however third party stages and Newport stages predating ESP technology will need to be configured for operation.

We present here the configuration process using an obsolete Newport stage that is not currently in the XPS stage database and does not have an EEPROM with ESP technology. We will address the wiring conventions adopted by Newport for use with the XPS and XPS-DRV02 and delve more deeply into electrical inputs that are used to communicate with motors and wired devices. We will then present the XPS Configuration Wizard and corresponding Stage Database. In this presentation, we will work sequentially through the Configuration Wizard as a new user would encounter the utility and provide background on the parameters that are used in control to provide a basic understanding of the steps being performed.

While this document is presented primarily as a how-to guide, it can also serve as a primer on Motion Control with the XPS and can be used for a greater understanding and expertise with operation of your Newport and Non-Newport stages.

BACKGROUND

Direct Drive Stages

Direct drive stages directly translate motor power to motion without any reduction or intervening coupling mechanisms such as a belt, chain or gearbox. Direct drive stages typically offer fast and precise positioning free of backlash or hysteresis caused by mechanical error while simultaneously offering extended lifetimes, increased efficiency and less mechanical noise. The Newport's XM Series of Linear Stages offer Ultra Precision performance ideal for demanding applications that require the use of direct drive linear stages.

Sin/Cos Encoder

A sin/cos encoder uses analog phase shifted signals that can be interpolated for high precision position determination. Typically these encoders use a pitched scale with an optical read head that converts signals to sin/cos electrical signals that are interpolated by the XPS. The industry standard for sin/cos encoders is 1 Vpp (the XPS allows a 20% tolerance), and typically after XPS interpolation results in nanometer or sub-nm resolution.

STAGE DATABASE

Then XPS houses a database of Newport stages with ESP technology (stagedatabase.txt). This extensive database is an excellent resource for quick plug-and-play operation with minimal configuration time. Newport has optimized the settings for each stage with ESP technology for operation in an unloaded state. These stages will automatically be recognized for immediate operation based on these factory settings.

For stages that are not in this database, such as the third party or obsolete Newport stages, it is necessary to configure the stage for basic operation and then later optimization. Each parameter broadly falls into a general category that addresses a fundamental aspect of operation. As such, the XPS configuration wizard utility and configuration file are divided by broad headings with more specific parameter definitions.

Many of the configuration parameters are highly nuanced and involve specialized terminology specific to motion control and in some cases Newport. To help bridge this gap, this document presents the specific settings, relevant values for a test case with a non-ESP Newport stage and goes into greater detail on the terminology used. In some cases, the naming conventions used in the existing stage database and configuration wizard will not be consistent, so a translation table has been provided at the end of this Tech Note to provide a look up table.

Setting up an XPS with a direct drive stage and sin/cos encoder

STAGE DATABASE ENTRY FOR THE XMS100

To better understand the required parameters in stage configuration, we present the Stage Database entry for the XMS100, a modern direct drive stage comparable to our test case stage to be configured. Many of the below parameters will not be immediately recognizable, however once you have completed this utility the entries below will be more readily understood. Preparing a stage configuration that can work with your stage is the ultimate goal of this guide.

[XM@XMS100@XPS-DRV02]

;--- Unit = mm
;--- Configuration_Comment = No load

;--- Smart stage name
SmartStageName = XMS100

;--- Motor driver model parameters
DriverName = XPS-DRV02
DriverMotorResistance = 5.5 ;--- Ohms
DriverMotorInductance = 0.0018 ;--- H
DriverMaximumPeakCurrent = 2.51 ;--- A
DriverMaximumRMSCurrent = 1.14 ;--- A
DriverRMSIntegrationTime = 15 ;--- s
DriverThermistanceThreshold = 1000 ;--- Ohms
DriverCutOffFrequency = 400 ;--- Hz

;--- Driver command interface parameters
MotorDriverInterface = AnalogSin120Acceleration
ScalingAcceleration = 39087 ;--- units / s^2
AccelerationLimit = 17838 ;--- units / s^2
MagneticTrackPeriod = 30 ;--- units
InitializationAccelerationLevel = 5 ;--- Pourcentage

;--- Position encoder interface parameters
EncoderType = AnalogInterpolated
LinearEncoderCorrection = 0 ;--- ppm
EncoderZMPlug = Encoder
EncoderInterpolationFactor = 4000
EncoderScalePitch = 0.004 ;--- units
EncoderSinusOffset = 0 ;--- V
EncoderCosinusOffset = 0 ;--- V
EncoderPhaseCompensation = 0 ;--- deg
EncoderDifferentialGain = 0
Backlash = 0 ;--- units
CurrentVelocityCutOffFrequency = 100 ;--- Hz
CurrentAccelerationCutOffFrequency = 100 ;--- Hz
PositionerMappingFileName =
PositionerMappingLineNumber =
PositionerMappingMaxPositionError = ;--- units
EncoderIndexOffset = 0 ;--- units
EncoderHardInterpolatorErrorCheck = Enabled

;--- Limit sensor input plug parameters
ServitudesType = StandardEOREncoderPlug
MinimumTargetPosition = -50 ;--- units
MaximumTargetPosition = 50 ;--- units
HomePreset = 0 ;--- units
MaximumVelocity = 300 ;--- units / s
MaximumAcceleration = 2500 ;--- units / s^2
EmergencyDecelerationMultiplier = 4
MinimumJerkTime = 0.02 ;--- s
MaximumJerkTime = 0.02 ;--- s
TrackingCutOffFrequency = 25 ;--- Hz

;--- Home search process parameters
HomeSearchSequenceType = MechanicalZeroAndIndexHomeSearch
HomeSearchMaximumVelocity = 100 ;--- units / s
HomeSearchMaximumAcceleration = 500 ;--- units / s^2
HomeSearchTimeOut = 5 ;--- s
HomingSensorOffset = 0 ;--- units

;--- Position servo loop type parameters
CorrectorType = PIDFFAcceleratio
ClosedLoopStatus = Closed
FatalFollowingError = 1 ;--- units
KP = 300000
KI = 10000000
KD = 800
KS = 0.8
GKP = 0
GKD = 0
GKI = 0

KForm = 0 ;--- units
IntegrationTime = 1E+99 ;--- s
DerivativeFilterCutOffFrequency = 5000 ;--- Hz
DeadBandThreshold = 0 ;--- units
KFeedForwardAcceleration = 1
NotchFrequency1 = 0 ;--- Hz
NotchBandwidth1 = 0 ;--- Hz
NotchGain1 = 0
NotchFrequency2 = 0 ;--- Hz
NotchBandwidth2 = 0 ;--- Hz
NotchGain2 = 0
KFeedForwardJerk = 0

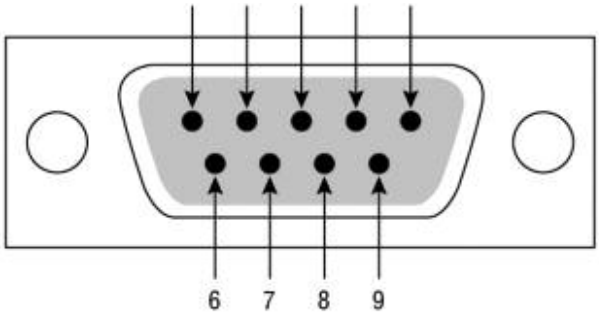
;--- Motion done condition mode parameters
MotionDoneMode = Theoretical

HARDWARE CONNECTIONS

The XPS uses the XPS-DRV02 and XPS-DRV02P driver modules for 3-phrase linear motors typical in direct drive stages. The XP-DRV02 provides 5 A at 44 Vpp and the XPS-DRV02P provides 7 A at 44 Vpp. The XPS-DRV02P can be used for driving stages with greater acceleration and velocity. For higher power applications it is possible to use external drives with XPS-DRV00.

The document presented here will address the standard case with an XPS-DRV02. As such, it is necessary to wire and configure the stages in accordance with Newport wiring conventions for the XPS-DRV02 Driver. Each stage requires three connections, entitled: Motor, Servitudes and Encoder: Each wiring diagram with corresponding connector is listed below, with further details in the following pages on electrical signal inputs:

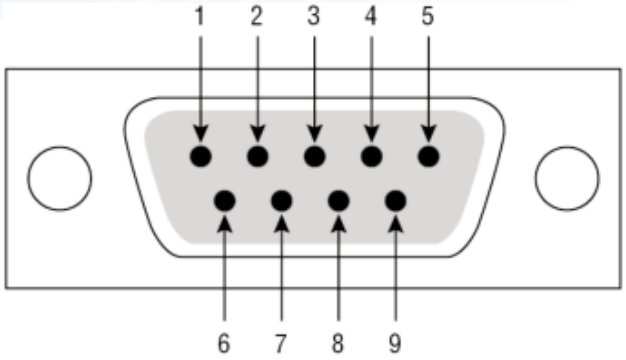
Motor DB9 Wiring Diagram



Motor Connection Wiring Conventions and Pinout

Pin #	Signal
1	Motor U
2	Motor U
3	Motor V
4	Motor V
6	Motor W
7	Motor W
8	GND

Servitudes DB9 Wiring Diagram

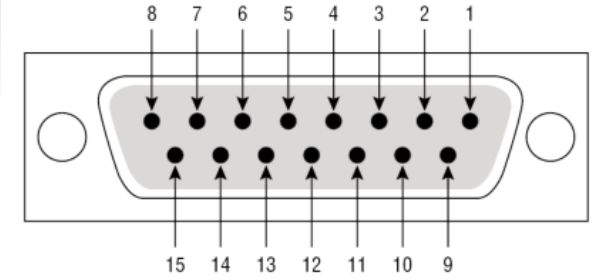


Encoder Connections

Pin #	Signal
1	End of Run Plus [2]
2	End of Run Plus [2]
3	Mechanical Zero [2]
4	+5V
6	Thermistor [1]
7	GND
8	Thermistor [1]

1. Use if motor has an integrated temperature, otherwise pins 6 & 8 must be shorted
2. If Encoder does not have an integrated limit sensor

Analog Encoder Wiring Diagram

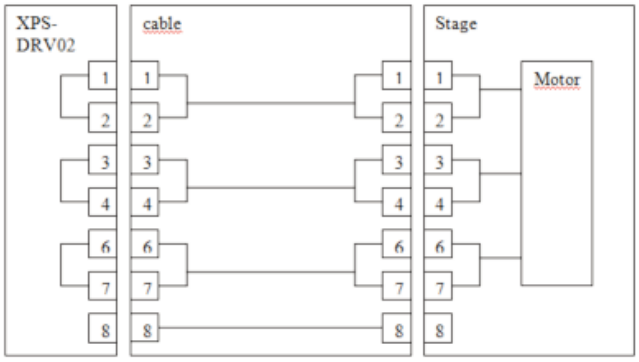


Pin #	Signal
8	Mechanical Zero (1)
4	+5V
2	GND
6	Limit (1)
3	Encode channel A
11	Encode channel /A
1	Encode channel B
9	Encode channel /B
14	Endoder channel I
7	Encoder channel /I

1. If encoder has an integrated limit, otherwise not-connected

MORE ON HARDWARE CONNECTORS

More on Motor Connections



Motor Driver Connections are used to energize phase shifted motor coils in the linear motor.

1. Phase U: This connection will energize phase U motor coil (input 1 and 2 are the same connection), according to the current draw of the motor. This connection will be wired in combination with the common ground. In each case you should be careful to keep your polarity convention consistent across all phase connections and be sure that the chosen polarity produces a magnetic field that drives the stage in the preferred direction. The motor energizing convention should be wired in the appropriate order as well for U, V, W to match each of the 120 degree phase shifted signals.
2. Phase U: Same connection as pin 1.
3. Phase V: energizes phase V (120 degree) shifted from Phase U. Polarity convention must be consistent with chosen convention of Phase U and ordering preserved as required by the motor. .
4. Phase V: same connection as pin 3.
5. Thermistor: The XPS uses Positive Temperature Coefficient (PTC) thermistors to monitor motor heating. If no thermistor is present this connection should be shorted to pin 0, and threshold set to 1000 Ω.
6. Phase W: energizes phase W (240 degree) shifted from Phase U. Polarity convention must be consistent with U and V and ordering preserved as required by the motor
7. Phase W: : same connection as pin 6
8. GND: XPS-DRV02 common ground

More on Servitudes Connections

All connector inputs on Servitudes are TTL compatible open collector type with +5V pull-up resistors referenced to common ground. All servitudes inputs are refreshed synchronously at the servo loop rate of 1 KHz by default. All Inputs are identical.

1. + Travel Limit: This is where the positive polarity of the positive travel limit switch should be wired. The low signal should be wired to common ground.
2. – Travel Limit: This is where the positive polarity of the negative travel limit switch should be wired. The low signal should be wired to the common ground.
3. The origin switch is where the "mechanical zero" high input should be wired. Low should be wired to common ground.
4. NC: This connection is unassigned.
5. Thermistor: This connection is +5 V
6. Thermistor: This connection is reserved for a PTC thermistor
7. GND: Ground
8. Thermistor: This connection is reserved for a PTC thermistor
9. GND: Ground

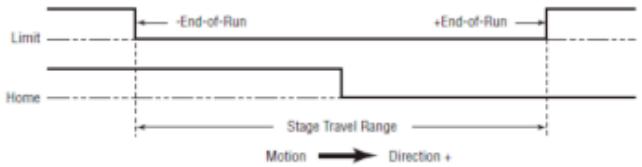
More on Analog Encoder Connections:

The sinusoidal position signals, sine and cosine, must be phase-shifted by 90° and have signal levels of 1 Vpp. Each of these two signals is composed of an analog sinusoidal signal and complement entering in a differential amplifier (Sine =Analog VA - Analog /VA). Newport provides software to examine the Lissajous signal and verify correct signals for operation. See Analog Encoder Calibration. Consult the manufacturer of your encoder for encoder wiring conventions.

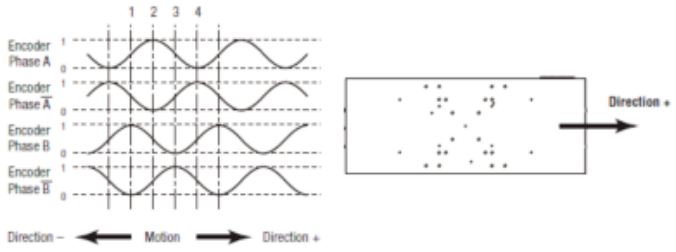
1. Analog VA: used in combination with pin 9 for Analog Sine Signal referenced to glass scale.
2. 0 V
3. Analog VB: used in combination with pin 11 for Analog Cosine Signal referenced to the glass scale.
4. +5 V: Power supply for the encoder read head
5. NC: This connection is unassigned
6. Limit (TTL Compatible Input): Available for limit referencing used with TTL output from encoder electronics referenced on the glass scale.
7. Analog VB: used in combination with pin 11 for Analog Cosine Signal referenced to the glass scale.
8. Home (TTL Compatible Input): Available for homing purposes used with TTL output from encoder electronics referenced to index on glass scale.
9. Analog VA: used in combination with pin 1 for Analog Sine signal referenced to glass scale
10. 0 V
11. Analog VB: used in combination with pin 7 for Analog Cosine signal from the glass scale
12. +5VL: provided for limit and home switch
13. NC: Connection not assigned
14. Analog VB: used in combination with pin 11 for Analog Cosine signal construction referenced to the glass scale.
15. NC: Connection not assigned

XPS reference signal conventions for Motor, Limits and Encoder

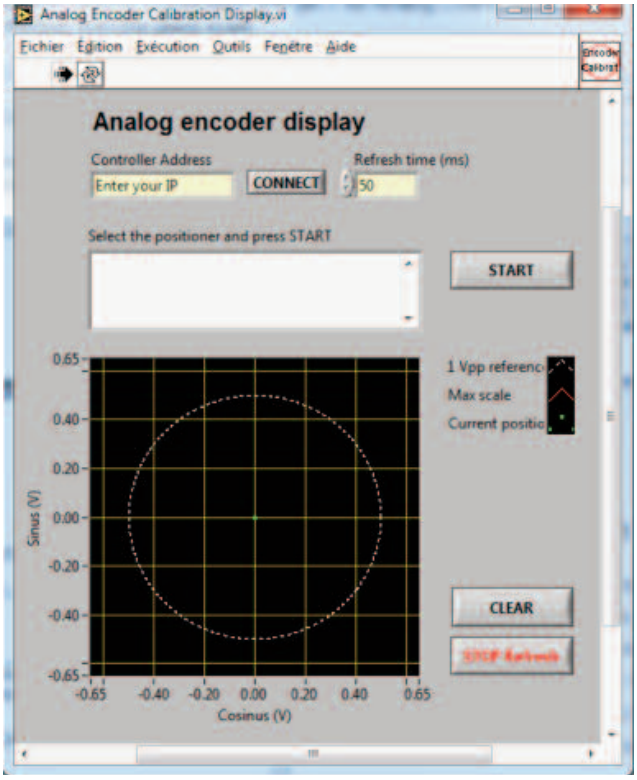
The below figure provides an easily referenced graphical illustrations of XPS conventions for End of Run signals and Home. It is should be noted the home position will coincide with the homing method and sensor location.



The below figure provides an easily referenced graphical depiction of Analog Encoder Signals that are interpolated by the XPS for high resolution position determination.



Optimizing Encoder Signals with Newport Analog Encoder Software



Analog Encoder Calibration software is available on the Newport FTP site to investigate the current Sin/Cos signal of the encoder. The signals should be 1Vpp with a 20% tolerance. The manufacturer of the encoder should provide a method for making the adjustment of Sine/Cosine signals. The XPS also provides a method to correct for this signal with input parameters such as sine and cosine phase offset, etc. More information is provided on this later in the guide.

From Wiring to Configuration

Once the proper wiring is complete and connections are made to the XPS, we can begin the process of building a configuration file. The XPS Configuration Wizard has been developed exclusively for the purpose of providing the user a convenient tool for building stage configurations.

Configuration Wizard Utility

The procedure listed below will guide you through the Configuration Wizard Utility. At successful completion of the utility, you will have created a stage configuration file that can be used for non-ESP stages and loaded in future configurations.

In each step, a brief explanation of the terminology used by the Configuration Wizard Utility and Stage Database will be presented. The utility and database terminology may not always be consistent or intuitive, so be mindful when setting a parameter and referencing the stage database.

Adding a Custom Stage

The Configuration Wizard utility is accessed via the main XPS Web Interface. To navigate to the utility, select "STAGE" and then "Add custom stage". This will prompt a selection process. The process of configuration may require referencing stage and motor configurations, so it is recommended to collect all support documentation for referencing beforehand.

This utility will guide you through a series of top-level settings, each with secondary parameter settings. In this guide we will present the utility as would be encountered by a user. Therefore on each page, we will provide background that refers to that particular menu setting. More detailed parameter settings will be addressed when secondary settings are made.

We will begin with the first entry in the utility: Position Servo Loop Type.

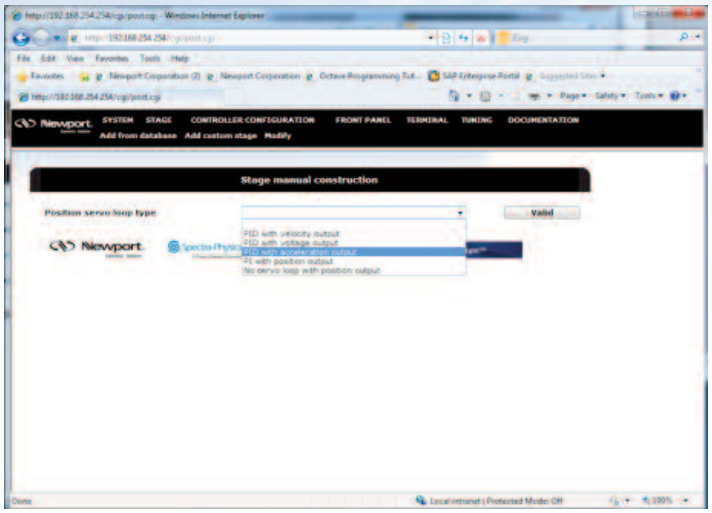
1. Position Servo Loop Type:

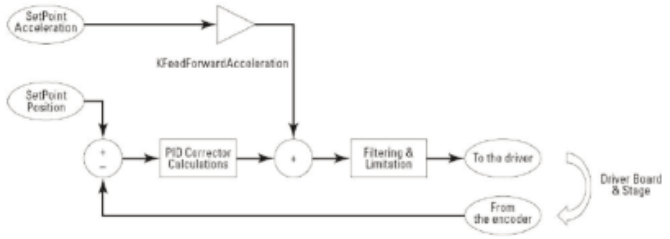
Setting: Set the Position servo loop type to PID with acceleration output

Details:

The position servo loop type will set the feedback loop by which the controller corrects for positioning errors. In the case of direct drive stages, the incorporated motor is typically a brushless linear or rotary motor, which requires setting the position servo loop type to PID with acceleration output.

In this configuration a constant voltage applied to the driver, results in a constant acceleration to the stage. For more information on Servo Loop Corrections, see the Newport Motion tech note on Tuning.





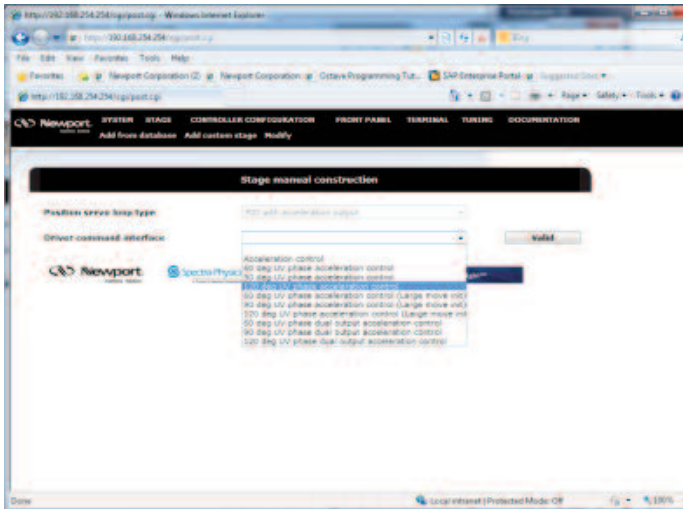
The above illustration provides a graphical representation of a PID loop with acceleration output.

2. Driver Command Interface:

Setting: Set the Driver Command Interface to 120 deg UV phase acceleration control or 120 dual output acceleration control.

Details:

Most modern brushless linear and rotary motors have three phases; as such a 120 deg phase is required for this input. Dual output acceleration control is the specific interface for driving two motors via two drivers in parallel.

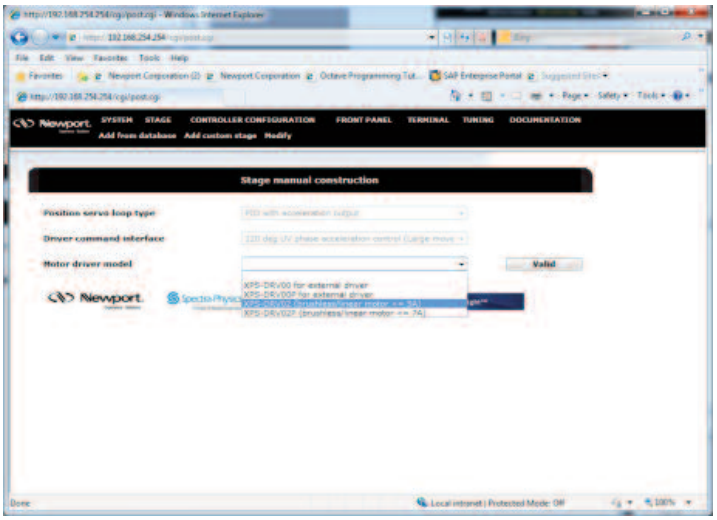


3. Motor Driver Model:

Setting: Set the Motor Driver Model to XPS-DRV02 or XPS-DRV02P

Details:

This type of motor driver model is used for brushless linear motors. The XPS-DRV02 supplies 5 A and 44 Vpp while the XPS-DRV02P supplies 7A and 44 Vpp. Refer to wiring connections for more information on board level connections.

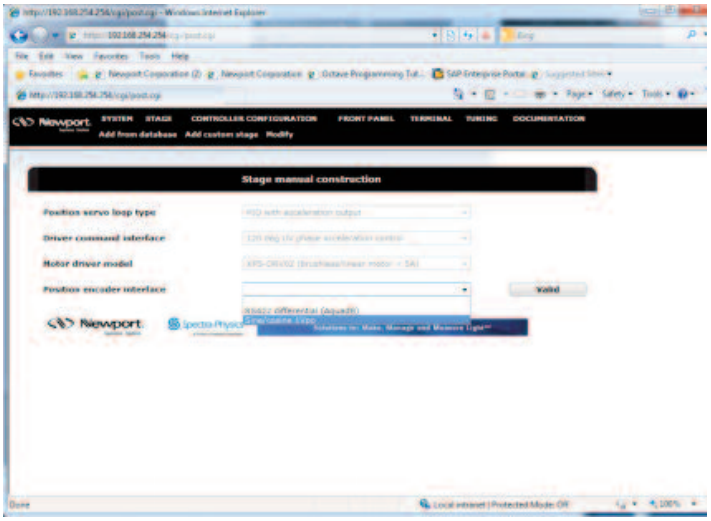


4. Position Encoder Interface:

Setting: Set the position encoder interface to Sine/cosine 1Vpp

Details:

This encoder type is used when the position sensor uses two 1 Voltage peak-to-peak sine/cosine signals that are interpolated by the XPS. It is important to note that the XPS must have an encoder signal 1Vpp (20% maximum allowed tolerance for operation). The XPS will perform an interpolation (up to 32768x) and greatly improve position feedback resolution.

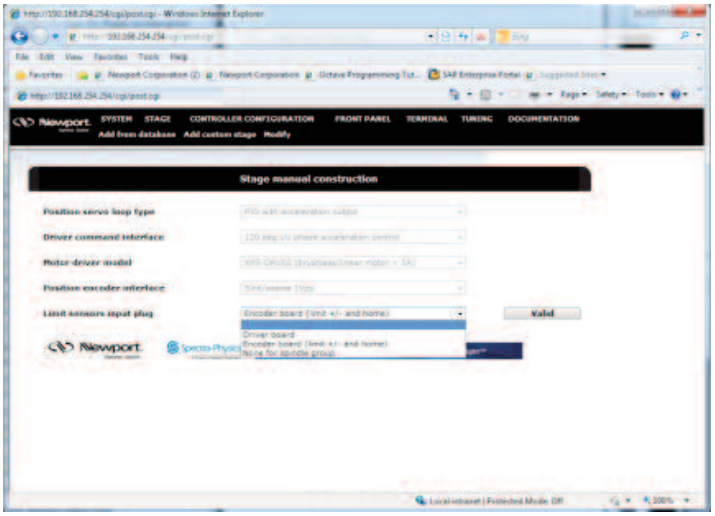


5. Limit Sensors Input Plug:

Setting: Set the Limit sensors input plug according to the wiring used by your stage.

Details:

The XPS supports 3 possible settings for the limit sensors input plug: driver board, encoder board, and none. This choice of the input for limit sensor input plug depends on where the limit sensors are electrically connected to the XPS. For example, if wired through the driver board, this input would be selected as driver board. If this parameter is set incorrectly you will receive an error such as (both end of runs activated), when you try to initialize the stage. Limit sensors wiring at the XPS is TTL compatible.

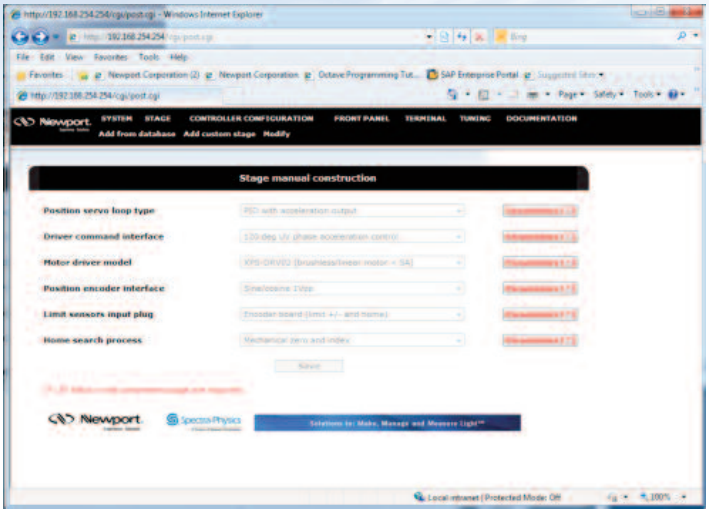


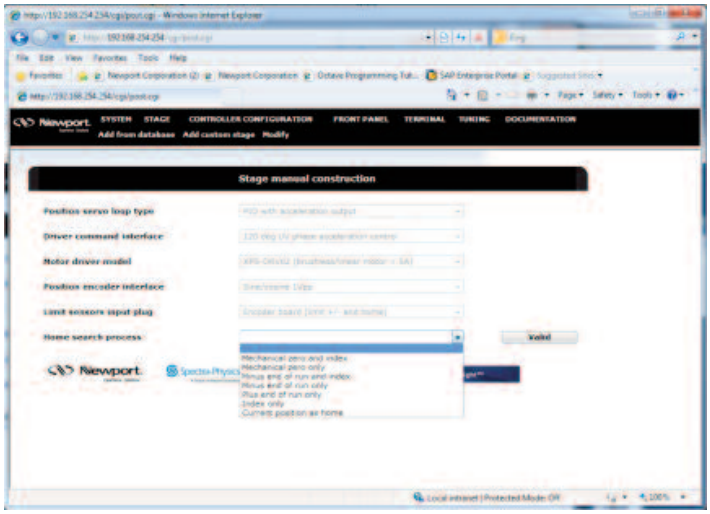
6. Home Search Process:

Setting: Set the Home Search Process according to your preference and the constraints of your stage. For information on Home Search, see the Newport tech note on Home Search Methods.

Details:

There are numerous methods for setting the home search process parameter. The appropriate choice will depend on the availability of reference signals. For example, this setting will depend on whether your stage has a mechanical zero or end of run sensors. Mechanical Zero and Index home search is chosen for our example configuration. The Mechanical Zero will set a change of state when moving from positive domain of travel to negative domain and vice-versa. When referencing Hardware Status in the XPS web Interface a ZM Level high may be observed depending on the region of travel, this is normal.



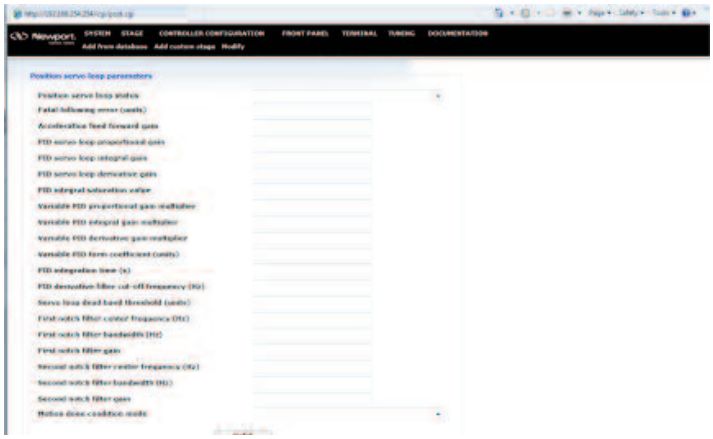


Top-Level Settings:

Once the top-level settings of your direct drive stage configuration have been set, you will be prompted to input a set of parameters based on the selected settings. The parameters buttons text will change color to indicate ready for sub-parameter settings. Do not navigate away from this page when setting parameters, as current settings may be lost. We can now set secondary parameter settings, which have a much greater level of detail. In this guide, we will explain the meanings of these settings as well as provide physical values for direct drive stage and how we came to those values.

Setting Parameters for Top Level Settings:

The parameters button will lead to a new page, which will list the required inputs based on the selected top-level setting. For example, we have set the position servo loop for PID with acceleration output, which will prompt the parameter screen below. Each selection will prompt an alternative set of settings that is specific to that top-level value set above.



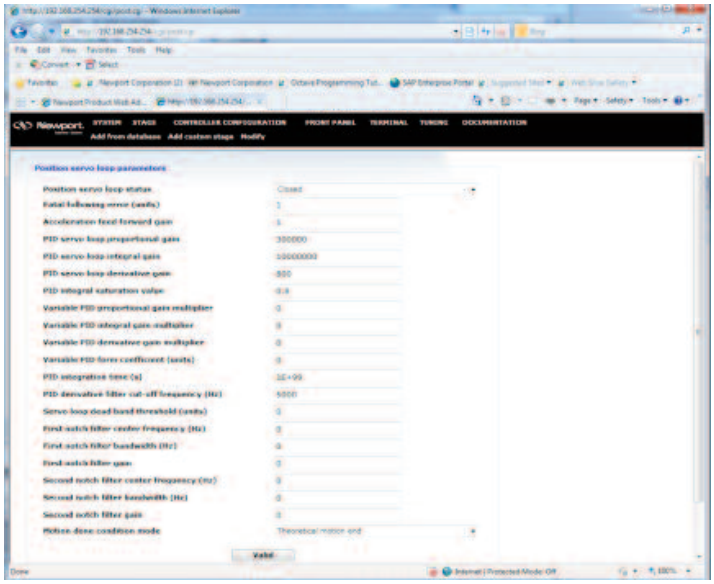
A CLOSER LOOK AT THE SETTINGS IN OUR EXAMPLE:

Our example case is an XM2000 linear stage; which is a predecessor to the current XMS100. Since the two stages share similar characteristics, it is again a worthwhile reference for preparing our own configuration.

We encounter the Position Servo Loop Parameters first, so we will present the best values in this case, and explain their significance. Since we are referencing the stagedatabase.txt entries which do not use the same naming conventions; we have included the stage database naming in a translation chart for reference at the end of this document.

Position Servo Loop Parameters

Position servo loop parameters will impact how we use encoder feedback to make position corrections for following errors. Since we are using closed loop operation with a PID control loop, we must set how the control loop uses these parameters for motion that deviates from ideal profile motion generated by the XPS.



More on Position Servo Loop Parameters

1. **Position Servo Loop:** By setting this value to closed, we have selected to have an active feedback loop correcting position with PID parameters.
2. **Fatal Following Error Threshold:** A following error is the difference between ideal position based on an ideal motion profile generated by the XPS and true position. The Fatal Following Error Threshold sets the maximum allowable position error. In our default case, this value is set to 1 unit. In the case of linear stages, this default unit is set to millimeters in our configuration. A larger setting is generally recommended when configuring a stage for the first time, so as to avoid following error from tuning or other parameters that must be optimized.
3. **Acceleration Feed Forward Gain:** This value is a designation for the electronics control loop, to set the feed forward gain. This value is typically 1. This value must be greater than or equal to zero.
4. **PID Servo Loop Proportional Gain:** The values for Proportional, Integral, and Derivative Gain must be determined empirically based on optimal operating condition preference and load conditions. There is a tuning tech note and XPS utility for optimizing these values. For initial settings, please refer to the description at the end of this section. The proportional gain addresses the current following error of the servo cycle and acts immediately to correct it. It acts as the stiffness of a spring to remove following error.
5. **PID Servo Loop Integral Gain:** See point 4. The Integral term Ki acts as weight factor for the correction of the integrated following error. The effect is typically to address low frequency errors.
6. **PID Servo Loop Derivative Gain:** See point 4. The Derivative term Kd acts as a weight factor for the correction of differential following errors. The effect is typically to address high-frequency errors in the system. Kd acts as a damping coefficient applied on the spring effects of the proportional gain. Kd must be high enough to stabilize the servo loop and low enough to avoid ringing at high frequency.
7. **PID Integral Saturation Value:** The saturation limit factor permits users to limit the maximum correction of Ki that is applied to the total PID correction output. Ks can take a value between 0 and 1. Here we have chosen 0.8.
8. **Variable PID Proportional Gain:** The Variable PID parameters have been implemented to help address nonlinear behavior in stages and apply variable PID corrections; for example stages with non-uniform friction between small and large moves. Here the value for our direct drive stage is 0, as the unloaded stage behavior is fairly uniform relative to commanded motion response.
9. **Variable PID Integral Gain:** Set to 0. See point 8

10. **Variable PID Derivative Gain:** Set to 0. See point 8
11. **Integration Time:** Integration time is used to adjust the duration of integration of following errors for the Integral correction. This helps localize the correction around recent moves. Here we have set essentially an infinite integration time because our system response is uniform. This is set to 1E+99 with default unit of seconds.
12. **PID Derivative Filter Cut-Off Frequency:** This value sets the cut-off frequency for derivative correction of the PID loop, and acts a low pass filter. This is typically set to reduce noise introduced by numerical derivation of the following error. This value must be between 0 and half the servo loop cycle frequency. Default values of servo loop cycle frequency are 10,000 Hz for XPS-Cx and 8000 Hz for XPS-Qx. We have set this to the maximum value of 5000Hz.
13. **Servo Loop Dead Band Threshold:** When applying PID corrections to minimize following errors, a condition may be reached where the system continuously dithers in set-position. This dithering can be avoided by introducing a Dead Band Threshold value. The effect is turn-off the correction within some error threshold near the target reducing settling times and oscillations. The negative aspect of this value is that it allows for residual error from the target position within this threshold. We have set this value to 0.
14. **First Notch Filter Center Frequency:** Mechanical systems have inherent resonances at certain frequencies when operated in a closed-loop. These excitations can be eliminated from the system by introducing notch filters. Notch filter frequency must be less than or equal to half the servo loop cycle frequency. In a default condition we use 0. Should you hear ringing during operation or distinct frequencies, it may be necessary to introduce a notch filter at that frequency. Typically, resonances will result in following errors.
15. **First Notch Filter Bandwidth:** The Notch Filter can be applied over a defined bandwidth to compensate for the resonance spectrum. The value must less than or equal to half the servo loop cycle frequency. The Bandwidth is set to 0, because our first notch filter is deactivated (set to 0).
16. **First Notch Filter Gain:** Sets the gain of the Notch Filter, which must be greater than or equal to 0. Again, we are not using a filter, therefore our gain is set to 0.
17. **Second Notch Filter Center Frequency:** The XPS provides a method to deal with more than one resonance. In this case, we would apply a second filter at a second resonant frequency as outlined above.
18. **Motion Done Condition Mode:** This setting requires one of two selections: Theoretical Motion End or Position and Velocity Checking. A theoretical motion end will use the motion profiler to determine if the move is done. This is strictly based on theoretical position. There will however be some finite following error and potential settling time, which can be considered with Position and Velocity Checking. Please refer to the description at the end of this section, for more information Motion Done Condition Mode.

APPROXIMATING PID SETTINGS FOR FIRST USE

Assuming that Scaling Acceleration has been set appropriately and there is minimal friction in the system as is typical with direct drive stages. It is possible to calculate typical PID parameters based on the following equations that provide PID parameters as a function of Servo Loop Bandwidth (Frq):

$$Kd := Frq \cdot 2 \cdot \pi \qquad Kp := \left(\frac{Frq \cdot 2 \cdot \pi}{2}\right)^2 \qquad Ki := \left(\frac{Frq \cdot 2 \cdot \pi}{3}\right)^3$$

Typical examples are listed below:

$$Frq = \begin{pmatrix} 25 \\ 50 \\ 100 \\ 150 \end{pmatrix} \frac{1}{s} \qquad Kd = \begin{pmatrix} 157.08 \\ 314.159 \\ 628.319 \\ 942.478 \end{pmatrix} \frac{1}{s} \qquad Kp = \begin{pmatrix} 6.169 \times 10^3 \\ 2.467 \times 10^4 \\ 9.87 \times 10^4 \\ 2.221 \times 10^5 \end{pmatrix} \frac{1}{s^2} \qquad Ki = \begin{pmatrix} 1.435 \times 10^5 \\ 1.148 \times 10^6 \\ 9.187 \times 10^6 \\ 3.101 \times 10^7 \end{pmatrix} \frac{1}{s^3}$$

The maximum possible value for a given stage will be dependent on the natural mechanical frequency and driver bandwidth. Non-negligible friction will typically require the Ki value to be set higher respectively.

A good starting point for direct drive stages is to set the following PID parameters:

K_D=300, K_P=1.5*10^4, KI=1.0*10^6

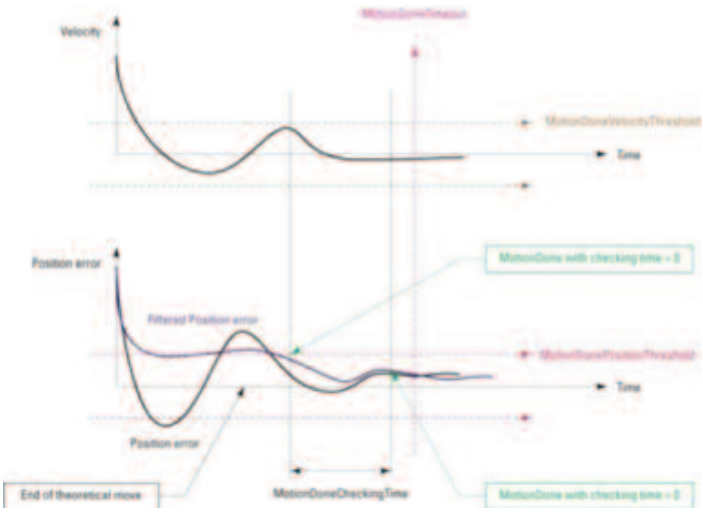
Motion Condition Mode and how it is used

The XPS controller supports two methods to define when a motion is considered done by the controller. These methods are specified by the MotionDoneConditionMode. There is a Theoretical Motion Done which uses the profiler motion end as the reference for determining a move has been completed. A second method is based on empirical data collected from the controller and referenced to user-defined settings (VelocityAndPositionWindowMotionDone). When programming sequential moves, it is important to consider when the controller considers the move complete for process flow.

With the Theoretical setting, a move is considered completed when the controller generated theoretical profile reaches target position. However this does not take into account the settling of the stage at the end of the move. So depending on precision and stability requirements for a motion to be considered "done" and at a suitable target, it is important to consider this setting.

VelocityAndPositionWindowMotionDone is the alternative method that uses empirical data referenced against user-defined settings for velocity and position to determine if the criterion has been met to consider the move done. This Motion End is validated when the following inequalities are satisfied for velocity and position.

1. | PositionErrorMeanValue | < | MotionDonePositionThreshold |
2. |VelocityMeanValue | < | MotionDoneVelocityThreshold |



The above figures provide a graphical representation for satisfying motion condition done for position and velocity.

The parameters to be set for VelocityAndPositionWindowMotionDone are as follows:

MotionDonePositionThreshold: This parameter defines the position error window. The position error has to be within ± of this value for a time duration defined by MotionDoneCheckingTime to validate the motion done condition.

- **MotionDoneVelocityThreshold:** This parameter defines the velocity window. The velocity at the end of the motion has to be within ± of this value for a time duration defined by MotionDoneCheckingTime to validate motion done condition.

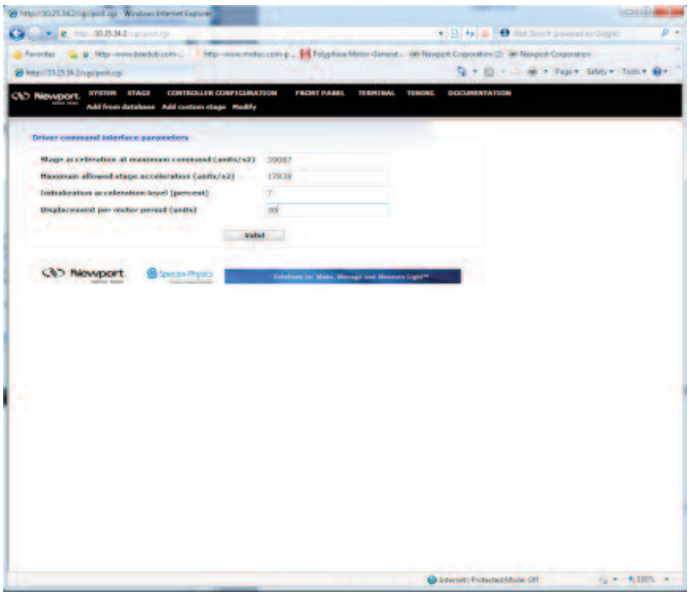
- **MotionDoneCheckingTime:** This parameter defines the period during which the conditions for the MotionDonePositionThreshold and the MotionDoneVelocityThreshold must be true before setting the motion done condition.

- **MotionDoneMeanPeriod:** This parameter is a moving average filter used to attenuate the noise for the position and velocity parameters. The MotionDoneMeanPeriod defines the duration for calculating the moving average position and velocity. The mean position and velocity values are compared to the threshold values as defined above. This parameter is not illustrated on the graph.

- **MotionDoneTimeout:** This parameter defines the maximum time the controller will wait from the end of the theoretical move for the MotionDone condition, before sending a MotionDone time-out error.

Driver Command Interface Parameters

The Driver command interface parameters set important parameters for generating an optimal motion profile for the configured stage. This profile will depend on the driver selected, which is accounted for in the configuration (in our case XPS-DRV02). Here we have our theoretical maximum acceleration value, the maximum acceleration possible with the selected motor and stage, initialization acceleration level and magnetic track period. These values can be determined through several methods however, it will be necessary to find a value analytically before optimizing empirically.



More on Driver Command Interface Parameters

1. Stage Acceleration at Maximum Command: This is the theoretical maximum acceleration the stage can be driven with a full 10V input to the driver. The Scaling Acceleration value is an important parameter as this value is used in generating a motion profile. The Scaling Acceleration must be calculated initially with the following equation:

- **ScalingAcceleration**:

$$\text{ScalingAcceleration} = \frac{\text{MotorForceConstant} \bullet 10\ [\text{Volts}] \bullet \text{TransImpedenceDriver}}{\sqrt{2}\ (\text{Mcar} + \text{Load})}$$

This equation assumes the no-friction case of a direct drive stage. For stages with significant static friction, this equation will not be applicable, and will not produce a proper scaling acceleration value for basic operation.

As an example calculation we will use the values in our case:

– MotorForceConstant: The motor force constant is a specification that must be provided by the manufacturer of your motor. It is specified in units of Newton/Amp (where amp is RMS current). In the case of the motor used on our test stage, this value is 19.9 N/Amp.

– TransImpedanceDriver: The TransImpedanceDriver value is a characteristic of the chosen driver. In the case presented here, this value is 0.5 Amp/Volt. This is again, specific to the XPS-DRV02.

– Mcar: This is the mass of the carriage in units of kg. In our case, we have a carriage that is 1.8 kg. It is important to note that the Mcar and Load strictly refer to the dynamic parts of the stage and applied load (i.e. moving).

– Load: This is the applied load to the stage: We initially begin in a no load condition, so this value is set to zero in our initial calculation. If ScalingAcceleration value is incorrect you will generate "error 50" or following errors.

Considering our individual case, we now calculate a ScalingAcceleration (or more intuitively "Theoretical Maximum Acceleration at maximum driver output") by using the above equation and find:

Scaling Acceleration = 39087 mm/s²

It should be noted that we are working in MKS units until the calculation is done, at which time we adopt the native linear units of the XPS for our configuration, which is mm. Be careful to keep units consistent when setting parameters in the XPS. Also be mindful that this value should be physically realistic.

Please see the Tech Note on Scaling Acceleration for more information on the optimization of this value.

2. Maximum Allowed Stage Acceleration: The maximum allowed stage acceleration is defined by safe operating parameters of the stage. This value is an acceptable acceleration limit that will prevent overheating or other dynamic errors as well as to prevent current limit at the driver also. This value is defined by the following equation for direct drive stages:

LimitAcceleration:

$$\text{LimitAcceleration} = \text{ScalingAcceleration} \bullet \frac{\text{DriverMaximumPeakCurrent}}{\text{MaxDriverCurrent}} \bullet \frac{1}{1.1}$$

In the equation above, we are provided with three variables, including:

1. ScalingAcceleration,
2. DriverMaximumPeakCurrent
3. MaxDriverCurrent.

We have previously found ScalingAcceleration, and know the MaxDriverCurrent (5A for XPS-DRV02 and 7A for XPS-DRV02P), which leaves DriverMaximumPeakCurrent, given by:

$$\text{DriverMaximumPeakCurrent} = \min\left(\frac{\text{XM\_PeakForce}}{\text{MotorForceConstant}}\sqrt{2} \bullet 1.1; \text{MaxDriverCurrent}\right)$$

The MotorForce Constant is provided by the manufacturer of the motor, which we know from before to be 19.9 N/Amp. This leaves XM_PeakForce, which is allowed to be defined as two times the RMS force, given by:

$$\text{XM\_RmsForce} = \sqrt{\frac{T \bullet \text{MotorK}}{\text{MotorRth}}}$$

In this equation, there are three values we must consider:

1. T (motor heating temperature)
2. MotorK (Motor Constant)
3. MotorRth. (Motor Thermal Resistance)

To understand how to apply this equation, it is best to quickly look at these values and where they can be found.

Motor Constant (MotorK or Km): The Motor Constant is a measure of the force output relative to heat generation of the motor. This constant is defined by construction design of the linear motor and magnetic field strength. It is defined in units of Newon2/Watt. There are several Motor Constants that may be referenced, so be sure that your Motor Constant is referenced according to these units. Consult the Manufacturer of your motor for this value. In our case this value is: 24 Newton2/Watt

Motor Thermal Resistance (MotorRth): Motor thermal resistance constant is a measure of the heat dissipated by a motor to the external environment. The units for this constant are ºC /Watt. This value is provided by the manufacturer of your motor. In our case this value is:

1.8 ºC/Watt

XM_RmsForce: is defined by the above equation, which incorporates these two values along with motor heating temperature. As a general practice, we set the motor RmsForce at a value that prevents motor heating beyond 20 °C. We will therefore use this value for T in our equation (in Celsius).

Using the above we find XM_RmsForce to be 16 N. and correspondingly by definition we take our XM_PeakForce to be two times this value, which is 32 N

Now we can return to our original equation to find the correct value for the Acceleration Limit. With a little substitution we find the Acceleration Limit to be: 17838 mm/s2

3. Initialization Acceleration Level:

This value will set the initialization acceleration level. This value must be large enough to overcome static friction and must be less than the DriverRMSCurrent to avoid I2T errors. This value is recommended as 20% of the ScalingAcceleration value. Here in our case we use 5% as a default condition. An incorrect value here could result in a failure of auto-scaling or initialization errors.

Note:

InitializationAccelerationLevel * 5 A < DriverMaximumRMSCurrent (for DRV02)

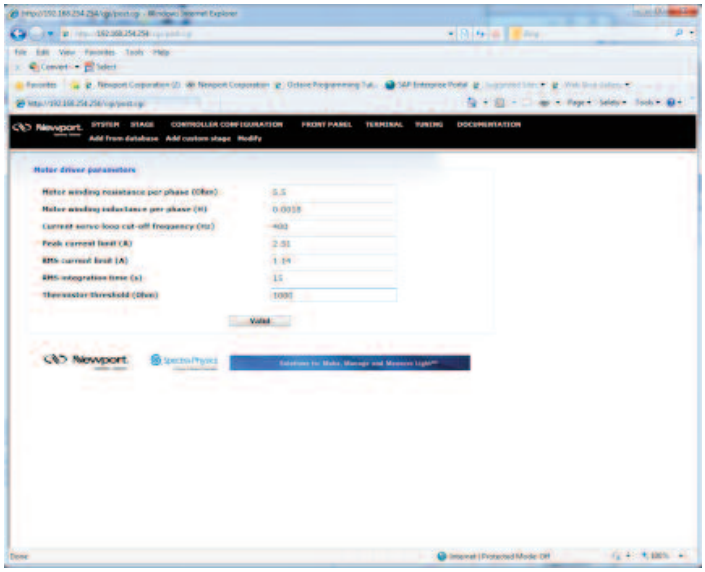
InitializationAccelerationLevel * 7 A < DriverMaximumRMSCurrent (for DRV02P)

4. Displacement per Motor Period:

This value is the distance between successive north poles in the magnetic track. This should be provided by the manufacturer of the linear motor. In our case, this value is 30 mm. When inputting this value be sure that the selected value is physically reasonable.

Motor Driver Parameters

The screen below lists motor driver parameters driver. These values will depend on the motor used. Please refer to documentation of the motor manufacturer to select appropriate values. Listed below are typical values for the XMS100.

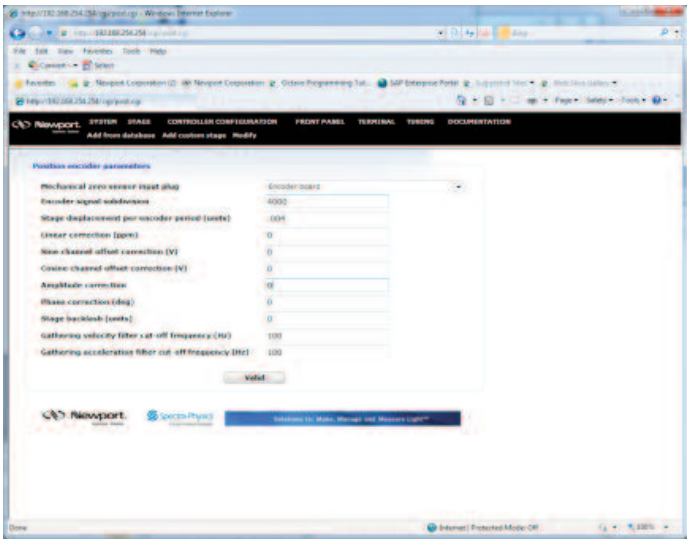


More on the Motor Driver Parameters

1. **Motor Winding Resistance per Phase:** Each phase of the brushless linear motor is driven by a separate winding that is electronically shifted in phase (in our case 120°). This is the motor winding resistance per phase. This value must be less than 65.535 Ohm.
2. **Motor Winding Inductance per Phase:** The linear motor winding inductance per phase. Provided by the manufacturer and measured in Henry. This must be less than or equal to 65.535 mH.
3. **Current Servo Loop Cut-Off Frequency:** This value sets the bandwidth of the driver internal current servo loop. This value should be greater than 0 and less than or equal to 3kHz. This value should typically be higher then 5X the chosen value for position servo loop cutoff frequency.
4. **Peak Current Limit:** This entry specifies the maximum allowed motor current. If this value is eclipsed a driver fault is generated. This value must be greater than 0 and less than 5A (for XPS-DRV02). This is an effective safety feature to prevent motor damage.
5. **RMS Current Limit:** This entry is the same as DriverMaximumRMSCurrent which is previously defined in this document.
6. **RMS Current Integration Time:** The RMS Integration time defines the integration time for the RMS Current. Typically, Newport recommends a small value (< 3 sec). This is because a large ratio between Peak current and RMS current will result in motor damage if the integration time is long. This is because over-current detection could be too late as a large Peak Current is applied a long time before being detected by the DriverMaximumCurrent threshold. This value is measured in seconds. The allowed integration time is greater than zero and less than 480 s.
7. **Thermistor Threshold:** The Thermistor threshold sets the resistance value for which the thermistor exhibits a resistive transition. This can take values greater than 100 Ohm but less than 9 kOhm.

POSITION ENCODER PARAMETERS

The screen below lists position encoder parameters. Encoder signal subdivision is the controller interpolation of the encoder scale. The Stage displacement per encoder period is the encoder scale pitch. Here the mechanical zero sensor input plug is on the encoder board. Position encoder corrections can be applied on this screen, if necessary. Listed below are typical values for the XMS100 linear stage.



More on the Position Encoder Parameters

1. **Mechanical Zero Sensor Input Plug:** The mechanical zero input can be wired to either the driver board or the encoder board plug. Depending on which wiring convention you choose, you must select the choice that corresponds to your wiring. Both inputs are TTL compatible.
2. **Encoder Signal Subdivision:** This setting provides the interpolation factor between adjacent measurements of the scale pitch. This setting must be an integer value greater than or equal to 1 but less than or equal to 32768. Here we do a 4000X interpolation.
3. **Stage Displacement per Encoder Period:** This parameter is the scale pitch of your selected encoder. This value is critical as all position units for moves will be derived from this setting (position, speed, acceleration…etc). In this case we are using a linear scale with a pitch of

4 microns. The manufacturer of your linear scale should specify this parameter. This setting in combination with our Encoder Signal Subdivision will provide ultimate position feedback resolution. Taking the above settings, we find Resolution= .004 mm / 4000 -> 1 nm.

4. **Linear Correction:** The XPS provides a method to compensate for linear errors from the encoder feedback loop. This linear encoder error correction value is measured in ppm and can take a value between ± 0.5*106. As a default condition, this value is set to zero.

5. **Sine Channel Offset Correction:** The electrical encoder signal must be within acceptable tolerances of 1Vpp with minimal sin/cos phase deviation. As can be observed in the parametric Lissajous plot presented previously in this document, we need centered 1 Vpp signal. This sine channel offset correction will introduce a DC offset to the Sine Channel. This parameter can take values of between ± 0.1 V.

   As a default condition, this value is 0, meaning no correction. In many cases channel offsets for sin/cos can be corrected at the encoder electronics level with manufacturer tools.

   Note: GroupInitializeAndEncoderCalibration API on the XPS measures and returns these parameters.

6. **Cosine Channel Offset Correction:** This will introduce a DC offset for the Cosine Channel, which can be observed in the Lissajous. Acceptable values are ± 0.1 V. As a default condition this value is 0. Again, the best method to determine whether this value is optimal is by running the analog encoder calibration and checking the corresponding Lissajous.

7. **Amplitude Correction:** Amplitude correction will adjust the Amplitude of the electrical encoder Cosine Signal. This parameter can take values between ± 0.1 and is applied according to the following equation, essentially making a percentage correction on existing amplitude. A value of 0 disables this correction.

$$CorrectedCosine = (1 + EncoderDifferentialGain) \times Cosine$$

8. **Phase Correction**: This parameter sets the phase correction of the electrical encoder Cosine Signal. This correction is applied after the amplitude correction of the previous entry. The correction is applied as presented in the following equation. A zero value disables this correction.

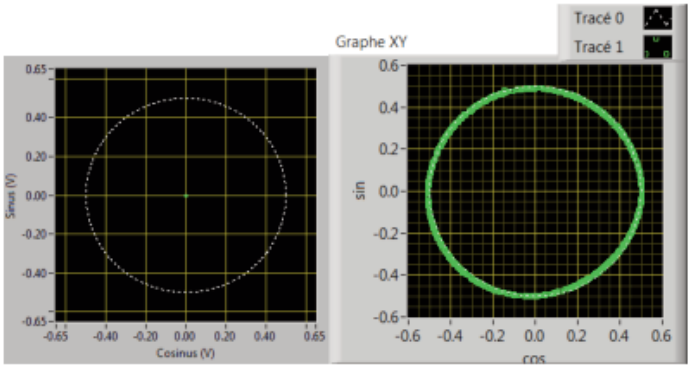
9. **Stage Backlash:** This parameter sets the backlash compensation value applied to the target position for a move. Our Direct Drive stage does not exhibit backlash, so it is simply disabled with a 0 setting.

10. **Gathering Velocity Filter Cut-Off Frequency:** The Gathering Cut-Off Frequencies act as low pass filters for CurrentVelocity and Current Acceleration acquired by the gathering feature of the XPS. These values reduce derivative noise. Here we have set the frequency to 100 (Hz).

$$PhaseCorrectedCosine = \frac{Sine \times \sin\left(PhaseCompensation \times \frac{\pi}{180}\right) + CorrectedCosine}{\cos\left(PhaseCompensation \times \frac{\pi}{180}\right)}$$

11. **Gathering Acceleration Filter Cut-Off Frequency:** As with velocity we set a low-pass filter on acceleration data taken in the Gathering feature of the XPS. Our default value is 100 Hz. It is worth noting here, that this setting is strictly applied to gather and is not critical for operation.

Sin/Cos Encoder Lissajous and Encoder Parameter Offsets



The Lissajous (or Bowditch) Curve above is a parametric plot of the Analog Sine and Cosine outputs from the encoder electronics. An Ideal Lissajous will overlap the dotted line above. These Lissajous curve is of the form:

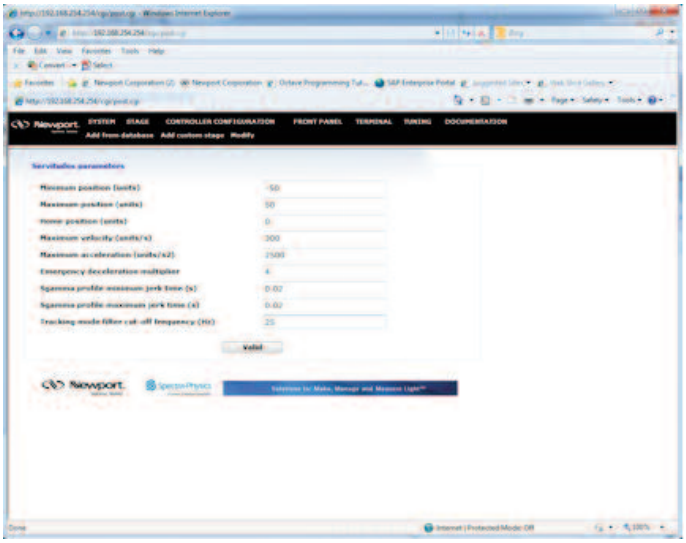
$X = A \sin(\omega t + \delta) + C1$ and $Y = B \cos(t) + C2$

In the context of these equations, we can observe the impact of XPS corrections:

1. Sine Channel Offset Correction (C1): This value will shift the graph along the X-Axis from the origin. The correction can be made to counteract this shift.
2. Cosine Channel Offseet Correctin (C2): This value will shift the graph along the Y-Axis from the origin. The correction can be made to counteract this shift back to the origin.
3. Amplitude Correction: The Cosine Amplitude (B) correction will adjust the eccentricity to correct for an elliptical graph (bringing A=B)
4. Phase Correction: The Phase Correction will adjust δ. An incorrect relative phase will shift the axes of symmetry from y=0 to y=mx (m≠0)

Servitudes Parameters

The screen below lists servitude parameters. Listed below are typical values for the XMS100 linear stage. On this page, fundamental parameters such travel range, maximum velocity and maximum acceleration are defined.



More on the Servitudes Parameters

1. **Minimum Position:** The minimum position defines the lower software bound for commanding stage travel. Here the total travel of our stage is 100mm centered on zero, so we use -50. This software limit can act as a safety feature to avoid unsafe conditions.

   Note: from before, we have zero as our center position because of the following:

   - HomeSearchSequenceType is MechanicalAndIndexHomeSearch
   - The index is on the middle of the stage travel
   - The Mechanical sensor is on the negative part of the stage travel
   - The Home Preset is 0

2. **Maximum Position:** The Maximum position defines the upper software bound for commanding stage travel. In our case, our stage has 100 mm of travel centered on zero, so we set 50 as our value.

3. **Home Position:** The home preset value sets the default home position reference. This value can take values between the minimum lower and maximum upper bound for position defined in the preceding steps.

4. **Maximum Velocity:** The Maximum Velocity sets the maximum command velocity for which the stage can be commanded to move. This value must be greater than zero, but less than stage velocity commanded. Maximum velocity will be a function of applied stage load. In our unloaded default condition, this value is set to 300 (mm/s). Refer to the following pages for information on how this setting impacts motion profiles.

5. **Maximum Acceleration:** The Maximum Acceleration sets the maximum commanded acceleration for which the stage can be commanded to move. This value must be greater than zero and less than the AccelerationLimit we had calculated earlier. The recommended value for smooth motion is 4x the velocity. In our case, we take this value to be 2500. The frictionless nature and highly responsive direct drive stage can achieve smooth motion at higher accelerations. Again this is unloaded value. It is important to remember the carriage of your stage is considered in the total load.

6. **Emergency Deceleration Multiplier:** In the event of an emergency stop or kill, this Deceleration Multiplier will set the stage deceleration for safely returning to a static position. Otherwise, inertia may carry the stage into a position that could cause harm or damage. This is an essential safety feature. This multiplier sets the deceleration between a typical braking and an emergency brake. The default setting for most stages is 4.

7. **SGamma Profile Minimum Jerk Time:** The XPS uses a sophisticated algorithm to create an ideal motion profile based on parameter settings. The Jerk is the time derivative of Acceleration, and Jerk Time is the time that acceleration is given to reach max acceleration. By setting Maximum and Minimum Jerk Time we allow the XPS a value range for generating a motion profile for our stage. Since stage behavior is typically not the same for small and large moves (static friction, etc.), there is a max value and a min value. With small displacements there is little energy required by the profiler and the MinimumJerkTime is used, whereas the opposite is the case with large moves and thus MaximumJerkTime is used. Since our stage is direct drive and there is minimal stiction. The Motion profiler will take this into account based on the move and these inputs.

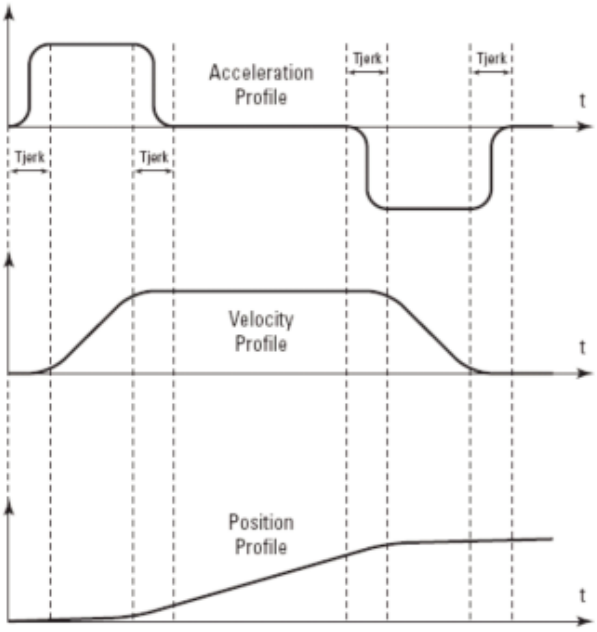
   Note: The minimum Jerk Time must be greater than two times the ISRProfileGeneratorPeriod (which is 0.004 sec for XPS-Cx and 0.005 sec for XPS-Qx).

   The impact of the Jerk Time on motion profile is described in greater detail in the following pages.

8. **SGamma Profile Maximum Jerk Time:** This sets the maximum allowable Jerk Time for building the SGamma profile. Again the XPS will determine the best Jerk time for the profile based on this setting. The profile construction takes into account other settings as well. Our value here is: 0.4 (so each profile will use this Jerk time). Recommend value for smooth motion is Min Jerk Time of 0.005 and Max Jerk Time is 0.05 s.

9. **Tracking Mode Filter Cut-Off Frequency:** The XPS provides a Tracking Mode which can be used to have the stage follow an Analog Input. This setting provides a low-pass filter to avoid errors with high-frequency following. The tracking Mode Filter Cut-Off Frequency should be greater than zero and less than half the profile generator frequency (2500Hz) and less than ten times the servo loop bandwitdth. The Default setting is 25Hz, and a value of 0 disables the filter.

SGAMMA MOTION PROFILE

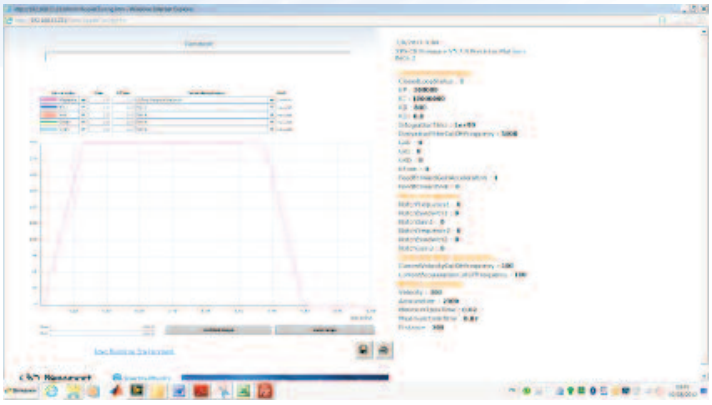


The Illustration above provides a graphical representation of a SGamma position profile and the corresponding Velocity and Acceleration profiles. The top graph provides an intuitive illustration of Jerk Time in the context of the acceleration profile. True position is referenced against profiled position to obtain following error.

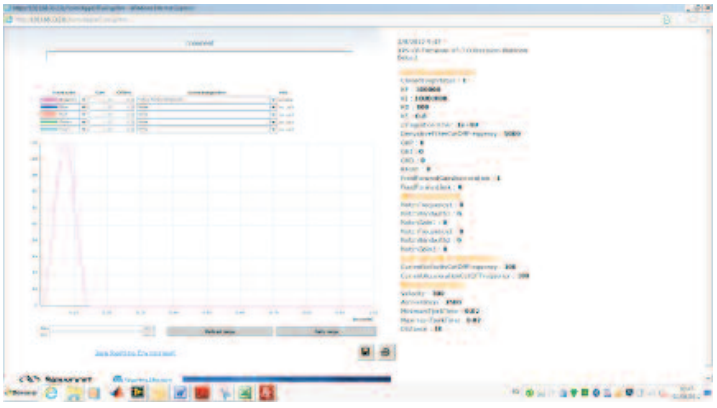
THE MAXIMUM VELOCITY

The XPS generates ideal motion profiles based in part on the Maximum Velocity setting. It is important to understand how performance is affected by this value:

For example, we consider our current situation with maximum velocity of 300 mm/s



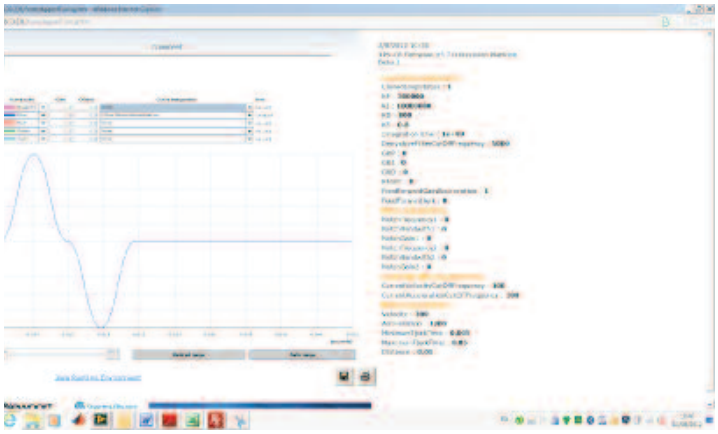
We use SGamma profile velocity of 300 mm/s and command a move of 200 mm. We can see the SGamma profile reaches the maximum velocity of 300 mm/s for the move.



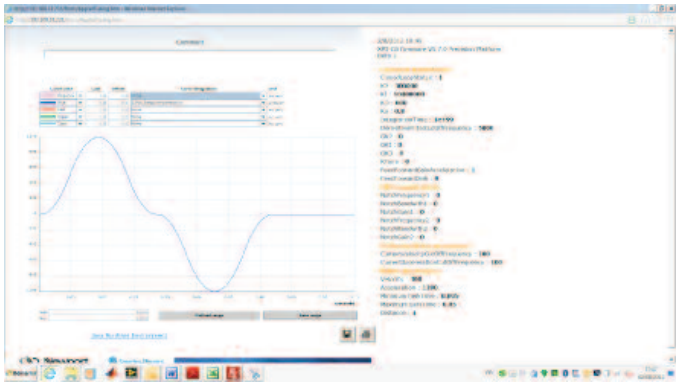
Here the user defines SGamma profile velocity equal to 300 and distance to 10. On the graph, we can observe the maximum velocity achieved by the profile is 135 mm/s. This move was too small to reach the 300 mm/s velocity.

Here the user defines a velocity of 200 mm/s for the profiler. The maximum velocity achieved is 200 mm/s for a 100 mm move. If the user were to define a 400 mm/s velocity, the XPS would not allow it as the maximum velocity is "out of range" and greater than 300 mm/s

THE JERK TIME

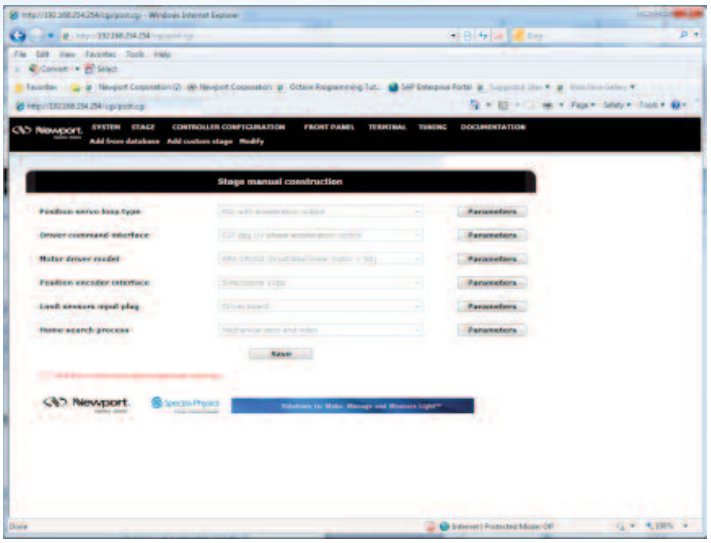


The Jerk time (in this case 0.005 sec) is the time for the profiler to reach maximum acceleration. If the jerk time is small, the travel time is small but the profiler excites the stage like a hammer, resulting in a long settling time.



If we increase the Jerk time (0.02 s), the travel time is longer but the generated profile is smoother and the excitation is not as abrupt.

HOME SEARCH PROCESS PARAMETERS



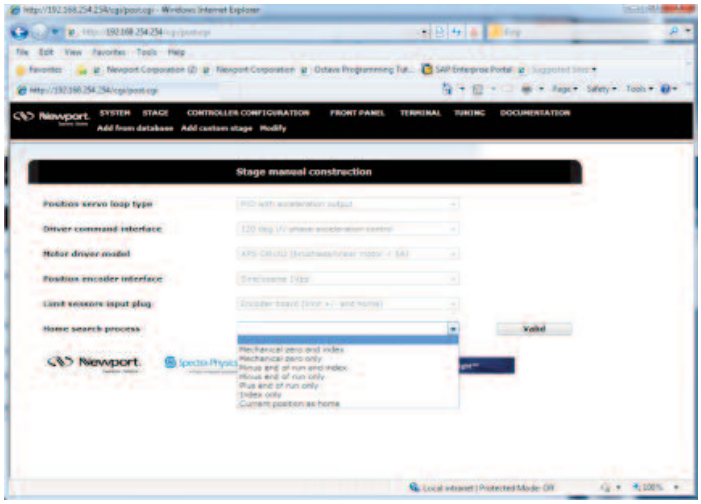
The Home Search Process Parameters are used to set dynamic variables for the Home Search process. These include Maximum Velocity, Acceleration and a Homing Timeout.

More on the Home Search Process Parameters:

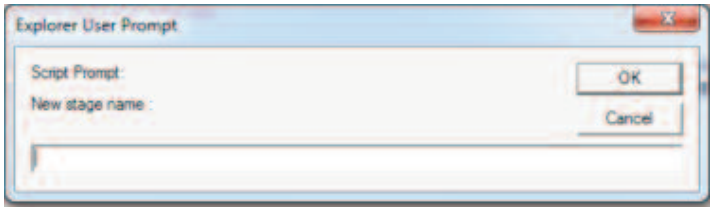
1. **Maximum Velocity:** This parameter sets the maximum velocity for the profile generator when homing the stage. This value must be greater than zero and less than the maximum allowed velocity. It should be noted, when first configuring a stage it is worthwhile to be conservative with this value. In our case we use 100 mm/s, which is about a third of our maximum velocity during regular moves.
2. **Maximum Acceleration:** This parameter sets the maximum acceleration for the profile generator when homing the stage. This value must be greater than zero and less than the maximum allowed acceleration. In our case, we have reduced the acceleration from 2500 to 500 for homing (and we are close to the recommended ratio of 4, to achieve smooth motion in many situations.
3. **Time out:** This parameter sets a time limit for the home search process. If the home search process it not completed by this time threshold, the XPS will generate an emergency stop and abort the home search process. This value is measured in seconds and is set to 5 seconds in our case. The Time Out value must be at least 400 µs at a minimum.

Saving the Configuration

We have now completed the Configuration Wizard Utility. The parameters buttons will now appear with black text indicating the settings are complete. We must now save our settings by clicking Save.

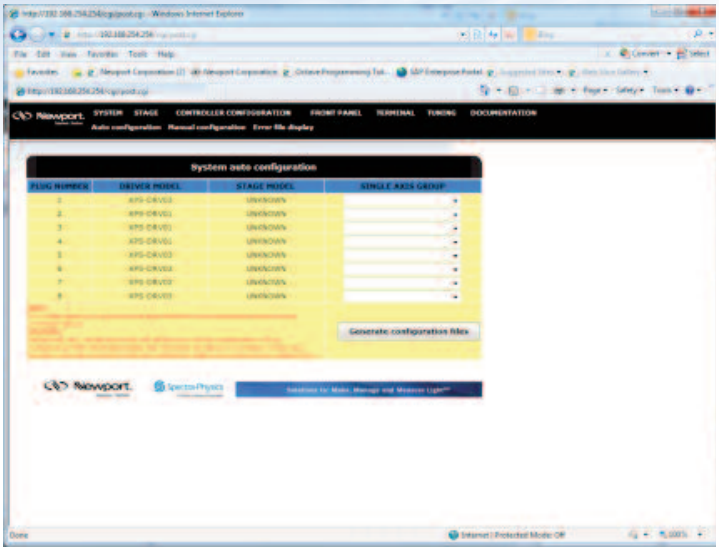


This will prompt a pop-up window requesting the user to provide a stage name. Once you have inputted the stage name, click OK. To continue our example using a brushless linear motor stage with a sin/cos encoder, we use XM2000 and then click OK.

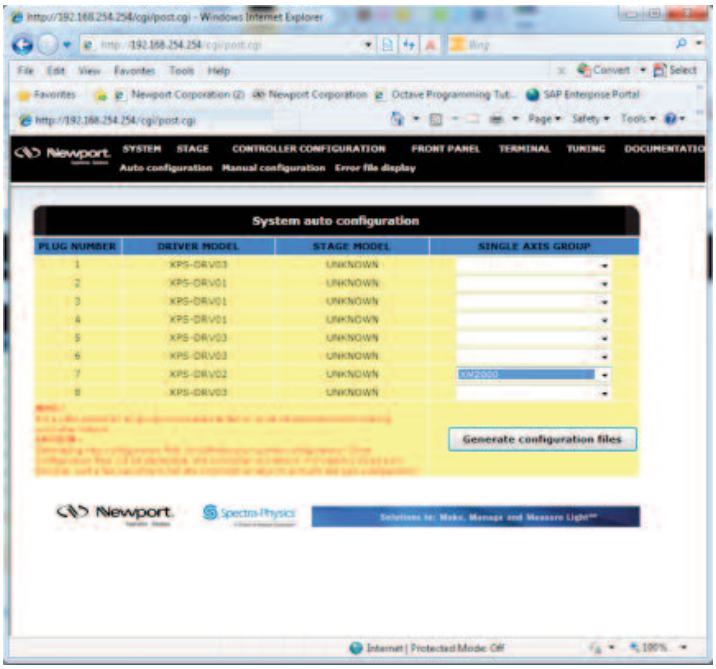


Configuring the XPS

Our stage configuration has now been saved, and can be loaded for immediate operation. To do this, go to the Auto or Manual configuration screen in the XPS GUI. For illustration, we will use Auto-Configuration. Our stage is naturally not recognized immediately (since it is not ESP compatible), so we must select the stage name from the drop down list corresponding to the plug number where our stage is attached.



In this case, we now select the stage name we had saved previously (XM2000), and Click Generate configuration files. This will load the configuration for our stage (that we have carefully set in the preceding directions). It is important to note our stage does not have a smart stage name (since it is not ESP compatible), so the XPS will simply load the configuration we have assigned in this step. It will automatically load this configuration at future reboots, unless we change it.



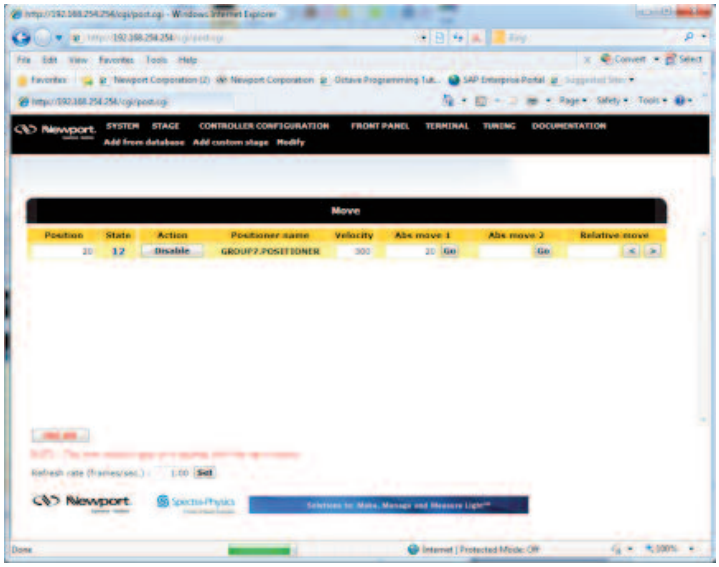
Manually Testing the Configuration

It is important at this step to verify that the configuration manually as a first test. We can easy do this by moving the stage by hand and checking the position feedback loop from encoder reading and mechanical zero and limit sensors.

Once the controller has rebooted, you can now go to the Front Panel and verify proper operation. Here we have done an auto-configuration with only one non-ESP stage attached. It is attached on driver card slot 7, so it is assigned the name GROUP7.POSITIONER. As you can see the velocity is what we had configured previously and our travel limits are defined by maximum travel (other settings are now in effect as well).

At this point is useful to move the stage and verify the following:

- The encoder reading is physically reasonable
- A manual move in the positive direction results in an increasing encoder position that is also positive.
- Move the stage to the negative limit, note the value
- Move the stage to the positive limit note the value
- Calculate the positive and negative limit travel and determine the travel range. Compare to real travel to verify proper position feedback.



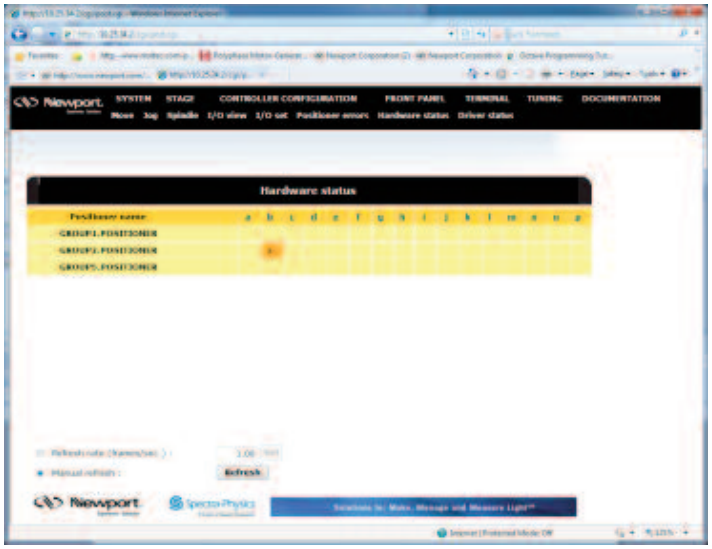
At this point we can also check to be sure our end of run sensors and mechanical zero are working properly. Navigate to the XPS web interface and select Hardware Status. This page along with Positioner Errors and Driver Status will provide a good overview of the current status of our configuration. More detail on verifying the sensors is listed below.

Verify End of Run Sensors and Mechanical Zero.

Make sure the stage is disabled (free to move manually) and move the stage carefully to the positive travel limit. This should indicate a positive end of run on Hardware status indicator d. This box should be checked when the stage is at the positive end of run and empty elsewhere. Repeat the process with the negative travel limit. In this case, Hardware status indicator c will be activated at the negative travel limit.

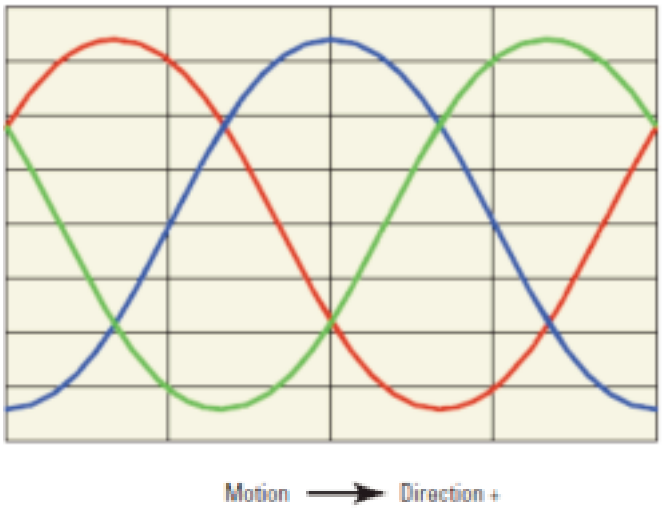
If both indicators are checked simultaneously there is an error in the wiring or functionality of the end limit sensors.

To be sure that the Mechanical Zero is working properly you can move the stage across the travel range from minus travel limit to positive travel limit. The mechanical zero indicator box should change status from low to high as the transition is made from negative to positive travel domains (and remain checked until the mechanical zero is crossed again in the reverse direction). This indicates the switch is working normally. If this change of state does not happen, check the mechanical zero wiring and functioning.



Verify Motor Wiring

It is important at this point to verify motor wiring is consistent with XPS controller convention (i.e. UVW). If the wiring convention is correct, a commanded positive move will result in motor coils being energized in proper order and the stage moving toward positive limit. The XPS provides an input for each phase, labeled schematically as U, V and W (see motor connections).

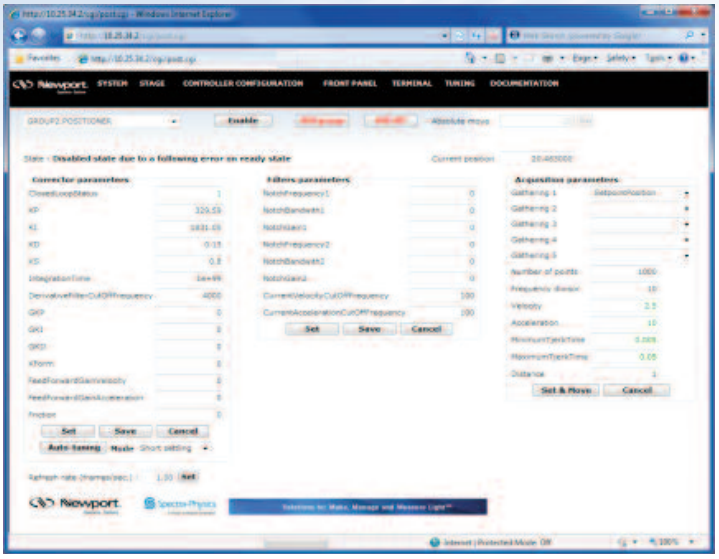


A three phase motor uses three electronically 120 degree shifted signals to produce a magnetic field that drives the stage in preferred directions.

To verify each phase is energized in the proper order and results in a positive move when commanded (and also negative, when commanded), we must set the stage for open loop operation and command positive and negative moves. Of course, we will only run in open loop briefly and once motor wiring convention is correct, we return to closed loop operation.

To set the stage to run in open loop, do the following:

1. Navigate to Tuning Window in the XPS web interface. Here you will be presented with a list of stages via a drop down menu in the upper hand left. The stage you have just configured will be here as well (in our case, the XM2000). Select that stage and change the ClosedLoopStatus value from 1 to 0. Then push the set and save buttons. Our stage is now running in open loop.
2. We can now command a positive move via the Absolute move window in the upper right hand corner. Select some arbitrary safe target position and select go. You should expect to see the stage move in the positive direction and encoder position rise accordingly.

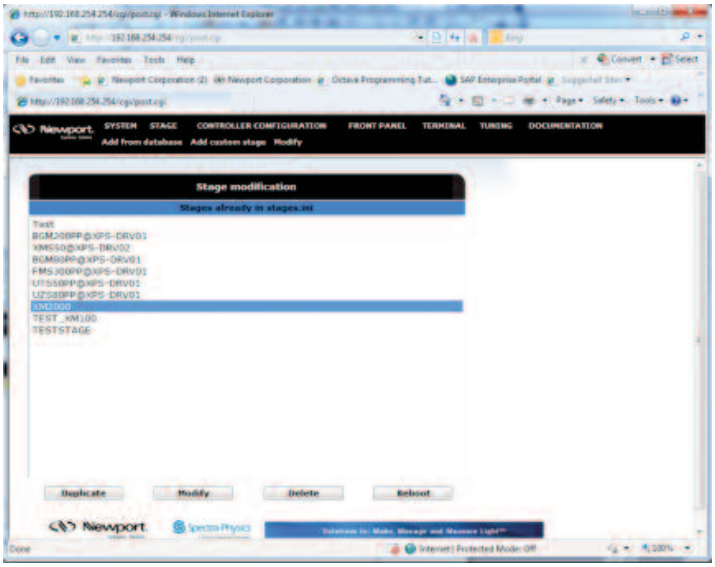


3. If the stage response is to move as expected in both positive and negative directions, the wiring is correct and we may skip to step 6.
4. If the Stage moves in the negative direction on a commanded positive move it will be necessary to reverse the wiring of two of the motor phases on the XPS. For example pins 1-2 of the motor connection can be swapped with pins 6-7. Turn the power to the controller off, disconnect and rewire the stage connector with these two motor connections reversed.
5. Once rewired, perform the same series of steps 1-3 to verify motor wiring is correct and the stage is working normally.
6. Once the stage is moving in the direction as commanded and motor wiring is verified, we may set the ClosedLoopStatus back to 1 in the tuning window. Click Set and Save. Our stage will now operate in closed loop status again.

Modifying Stage Parameters after Initial Configuration

In the previous step, we were able to modify our stage configuration quickly and conveniently, thanks to the XPS web interface. Should you need to modify the other parameter settings of the configured stage, we need not go back through the entire configuration wizard utility. We can simply modify our preferred parameters by going to STAGE in the XPS GUI, and select Modify.

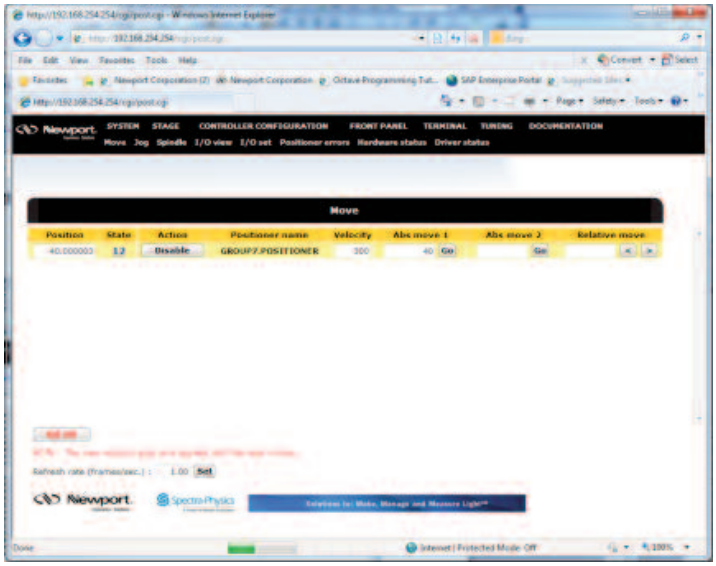
This action will prompt a window which will allow stage configuration parameters to be edited. Once you have modified and saved the revised parameters, you can reboot the controller to load new parameters. This avoids the necessity of having to perform multiple configurations for proper operation and will allow minor adjustments of individual parameters.



As you can see, the parameters we configured over the course of this document are now used to populate a stage configuration file. Instead of revisiting the whole configuration utility we can modify our stage parameters here for adjustments.

What Comes Next

Once the stage is working normally, the performance can be optimized with scaling acceleration and tuning adjustments. Please refer to Newport Tech notes on Scaling Acceleration and Tuning for more information. These notes will help you optimize the performance, after you have achieved basic functioning of your non-ESP stage. It is important to note that selected payload will impact both scaling acceleration and PID parameters.



Newport
Experience | Solutions

TERMINOLOGY USED BY STAGE DATABASE AND CONFIGURATION WIZARD

Configuration Wizard Utility	Stage Database Entry
Motor Driver Parameters	;--- Motor driver model parameters
Motor driver model	DriverName = XPS-DRV02
Motor winding resistance per phase	DriverMotorResistance = 5.5 ;--- Ohms
motor winding inductance per phase	DriverMotorInductance = 0.0018 ;--- H
Peak current limit	DriverMaximumPeakCurrent = 2.51 ;--- A
RMS current limit	DriverMaximumRMSCurrent = 1.14 ;--- A
RMS integration time	DriverRMSIntegrationTime = 15 ;--- s
Thermistor Threshold	DriverThermistanceThreshold = 1000 ;--- Ohms
Current servo loop cut-off frequency	DriverCutOffFrequency = 400 ;--- Hz
Driver Command Interface Parameters	**;--- Driver command interface parameters**
Driver Command Interface	MotorDriverInterface = AnalogSin120Acceleration
Stage acceleration at maximum command	ScalingAcceleration = 39087 ;--- units / s^2
Maximum allowed stage acceleration	AccelerationLimit = 17838 ;--- units / s^2
Displacement per motor period	MagneticTrackPeriod = 30 ;--- units
Initializaiton acceleration level	InitializationAccelerationLevel = 5 ;--- Pourcentage
Position Encoder Interface	**;--- Position encoder interface parameters**
Position Encoder Interface	EncoderType = AnalogInterpolated
Linear correction	LinearEncoderCorrection = 0 ;--- ppm
Mechanical zero sensor input plug	EncoderZMPlug = Encoder
Encoder signal subdivision	EncoderInterpolationFactor = 4000
Stage displacement per encoder period	EncoderScalePitch = 0.004 ;--- units
Sine channel offset correction	EncoderSinusOffset = 0 ;--- V
Cosine channel offset correction	EncoderCosinusOffset = 0 ;--- V
Phase correction	EncoderPhaseCompensation = 0 ;--- deg
Amplitude crrection	EncoderDifferentialGain = 0
Stage backlash	Backlash = 0 ;--- units
Gathering velocity filter cut-off frequency	CurrentVelocityCutOffFrequency = 100 ;--- Hz
Gathering acceleration filter cut-off frequency	CurrentAccelerationCutOffFrequency = 100 ;--- Hz
Not in utility	PositionerMappingFileName =
Not in utility	PositionerMappingLineNumber =

Terminology used by Stage Database and Configuration Wizard (cont'd)

Not in utility	PositionerMappingMaxPositionError = ;--- units
Not in utility	EncoderIndexOffset = 0 ;--- units
Not in utility	EncoderHardInterpolatorErrorCheck = Enabled
Limit Sensors Input Plug	**;--- Limit sensor input plug parameters**
Limit Sensors Input Plug	ServitudesType = StandardEOREncoderPlug
Minimum position	MinimumTargetPosition = -50 ;--- units
Maximum position	MaximumTargetPosition = 50 ;--- units
Home position	HomePreset = 0 ;--- units
Maximum velocity	MaximumVelocity = 300 ;--- units / s
Maximum acceleration	MaximumAcceleration = 2500 ;--- units / s^2
Emergency deceleration multiplier	EmergencyDecelerationMultiplier = 4
Sgamma profile minimum jerk time	MinimumJerkTime = 0.02 ;--- s
Sgamma profile maximum jerk time	MaximumJerkTime = 0.02 ;--- s
Tracking mode filter cut-off frequency	TrackingCutOffFrequency = 25 ;--- Hz
Home search process	**;--- Home search process parameters**
Home search process	HomeSearchSequenceType = MechanicalZeroAndIndexHomeSearch
Maximum velocity	HomeSearchMaximumVelocity = 100 ;--- units / s
Maximum acceleration	HomeSearchMaximumAcceleration = 500 ;--- units / s^2
Time out	HomeSearchTimeOut = 5 ;--- s
Not in utility	HomingSensorOffset = 0 ;--- units
Position servo loop Parameters	**;--- Position servo loop type parameters**
Position Servo Loop Type	CorrectorType = PIDFFAcceleration
Position servo loop status	ClosedLoopStatus = Closed
Fatal following error	FatalFollowingError = 1 ;--- units
PID servo loop proportional gain	KP = 300000
PID servo loop integral gain	KI = 10000000
PID servo loop derivative gain	KD = 800
PID integral saturation value	KS = 0.8
Variable PID proportional gain multiplier	GKP = 0
Variable PID derivative gain multiplier	GKD = 0
Variable PID integral gain multiplier	GKI = 0

Terminology used by Stage Database and Configuration Wizard (cont'd)

Variable PID form coefficient	KForm = 0 ;--- units
PID Integration time	IntegrationTime = 1E+99 ;--- s
PID Derivative filter cut-off frequency	DerivativeFilterCutOffFrequency = 5000 ;--- Hz
Servo loop deadband threshold	DeadBandThreshold = 0 ;--- units
Acceleration feed forward	KFeedForwardAcceleration = 1
First notch filter center frequency	NotchFrequency1 = 0 ;--- Hz
First notch filter bandwidth	NotchBandwidth1 = 0 ;--- Hz
First notch filter gain	NotchGain1 = 0
Second notch filter center frequency	NotchFrequency2 = 0 ;--- Hz
Second notch filter bandwidth	NotchBandwidth2 = 0 ;--- Hz
Second notch filter gain	NotchGain2 = 0
Not in utility	KFeedForwardJerk = 0
Motion done condition mode	MotionDoneMode = Theoretical