
Version 8.0

VisSim/Motion User's Guide

By Visual Solutions, Inc.

Visual Solutions, Inc.

VisSim/Motion User's Guide Version 8.0

Copyright	© 2010 Visual Solutions, Inc. All rights reserved.	Visual Solutions, Inc. 487 Groton Road Westford, MA 01886
Trademarks	VisSim, VisSim/Analyze, VisSim/CAN, VisSim/C-Code, VisSim/C-Code Support Library source, VisSim/Comm, VisSim/Comm C-Code, VisSim/Comm Red Rapids, VisSim/Comm Turbo Codes, VisSim/Comm Wireless LAN, VisSim/Fixed-Point, VisSim/Knobs & Gauges, VisSim/Model-Wizard, VisSim/Motion, VisSim/Neural-Net, VisSim/OPC, VisSim/OptimizePRO, VisSim/Real-TimePRO, VisSim/State Charts, VisSim/Serial, VisSim/UDP, VisSim Viewer, and flexWires are trademarks of Visual Solutions. All other products mentioned in this manual are trademarks or registered trademarks of their respective manufacturers.	
Copyright and use restrictions	The information in this manual is subject to change without notice and does not represent a commitment by Visual Solutions. Visual Solutions does not assume any responsibility for errors that may appear in this document. No part of this manual may be reprinted or reproduced or utilized in any form or by any electronic, mechanical, or other means without permission in writing from Visual Solutions. The Software may not be copied or reproduced in any form, except as stated in the terms of the Software License Agreement.	
Acknowledgements	The following engineers contributed significantly to preparation of this manual: Mike Borrello, Allan Corbeil, and Richard Kolk.	

Contents

Introduction	1
What is VisSim/Motion.....	1
The VisSim product family	2
Resources for learning VisSim/Motion	4
Interactive webinars	4
Sample diagrams	4
Training	4
Using the Tutorials	5
AC induction motor tutorial: speed control of a machine tool lathe.....	5
Setting up the motor, load, and encoder	6
Designing the volts/frequency controller for the motor	8
Customizing the Volts/Hz Controller block	9
Configuring the PID compensator.....	10
Wiring Volts/Hz Controller to the overall simulation	10
AC induction motor application: power train drive system.....	11
Brushless DC (BLDC/PMSM) motor tutorial: target tracking system	12
Motor Specifications	12
Simulation development.....	13
Setting up the plot blocks	19
Setting the simulation properties	21
Final configuration requirements	22
Simulation results.....	23
Other applications	24
Other things you can do with VisSim/Motion	24
Block Reference	25
AC Induction Motor (DQ).....	25
AC Induction Motor Current Model (FOC)	27
AC Induction Motor (Machine Reference).....	29
Basic (Permanent Magnet) DC Motor	31
Brushless DC (BLDC/PMSM) Motor	33
Clarke Transform	35
Commutator (Six Step).....	36
Discrete Integrator	37
Field Orientated Controller (FOC)	38
Frequency Demodulator	41
Hall Sensor	42
Inverse Clarke Transform	43
Inverse Park Transform	44
Linear Encoder	45
Low Pass Filter	46
LVDT	47

Microstep Controller	49
Park Transform.....	51
Park Transform (Stationary Frame).....	52
Permanent Magnet Stepper Motor (2 Phase).....	53
PID Controller (Digital)	54
PID Controller (Ideal).....	56
PWM Brush Servo Amplifier	58
PWM Brush Motor Amplifier (2Q).....	60
PWM Brush Motor Amplifier (4Q).....	61
PWM Brush Motor Amplifier (2Q, Current Feedback)	63
PWM Brush Motor Amplifier (4Q, Current Feedback)	65
PWM Brushless Servo Amplifier	67
PWM (Dual Phase).....	69
PWM (Single Phase)	70
PWM (Space Vector)	71
Rate Estimation (Rooftop) Filter	72
Rotary Encoder.....	73
Rotary Position Sensor	74
Rotary Servo Potentiometer.....	75
Rotary Tachometer	76
Rotational Load	77
Stepper Motor Controller	80
Three Phase AC Source.....	81
Three Phase Square Wave Inverter	82
Translational Load.....	83
Triangle Wave Generator	86
VCO	87

Index

89

Introduction

This section contains...

What is VisSim/Motion

VisSim/Motion is an easy-to-use, yet powerful solution for accurately modeling and simulating motion and motor control systems. VisSim/Motion consists of over 40 high fidelity motion and motor control blocks including:

- Five high-fidelity motor models
 - AC induction (machine reference and DQ)
 - Brushless DC (PMSM)
 - DC Brush
 - Stepper motor models
- Rotational Load and Translational Load blocks
- Brush and Brushless PWM amplifiers
- Eight sensors including Frequency demodulator, Hall, Linear and Rotary Encoder
- Simple dialog box configuration
- Editable Amplifier, Controller, Filter, and Discrete Integrator blocks to simplify the creation of custom blocks

The VisSim product family

The VisSim product family includes several base products and product suites, as well as a comprehensive set of targeted add-on modules that address specific problems in areas such as data communications, data acquisition, linearization and analysis, and digital signal processing.

Base products and product suites

Product	Function
Professional VisSim	<p>Model-based design, simulation, testing, and validation of dynamic systems.</p> <p>A personal version, VisSim PE, is also available. VisSim PE limits diagram size to 100 blocks.</p>
VisSim/Comm Suite	<p>Simulates end-to-end communication systems at the signal level using 200+ communications, signal processing, and RF blocks.</p> <p>Includes Professional VisSim and VisSim/Comm blockset.</p> <p>A personal version, VisSim/Comm Suite PE, is also available. VisSim/Comm PE limits diagram size to 100 blocks and limits the Communication blockset. See the VisSim/Comm datasheet for details.</p> <p>VisSim/Comm Suite add-on modules are available for real-time data acquisition (Red Rapids digital tuner card); modeling PCCC turbo codes, including UMTS specification; and for support of Bluetooth, 802.11 a/b/g (Wi-Fi), and ultrawideband wireless designs.</p>
VisSim/Embedded Controls Developer Suite	<p>Rapidly prototypes and creates embedded controls for DSPs, DSCs, and MSP430 microcontrollers. You can simulate and generate scaled, fixed-point ANSI C code, as well as code for on-chip peripherals.</p> <p>Includes Professional VisSim, VisSim/C-Code, VisSim/Fixed-Point, and one user-specified target support.</p> <p>A personal version, VisSim/Embedded Controls Developer PE, is also available. VisSim/Embedded Controls Developer PE limits diagram size to 100.</p>
VisSim Viewer (free)	<p>Lets you share VisSim models with colleagues and clients not licensed to use VisSim.</p>

Add-on modules

Add-On Module	Function
VisSim/Analyze	Performs frequency domain analysis of a linearized nonlinear subsystem.
VisSim/CAN	Interfaces with a USB CAN device to read and write CAN messages on the CAN bus.
VisSim/C-Code	Generates highly-optimized, ANSI C code that can be compiled and run on any platform that supports an ANSI C compiler.
VisSim/C-Code Support Library Source	Provides source code for the Support Library.
VisSim/Comm blockset	<p>Simulates end-to-end communication systems at the signal level using 200+ communications, signal processing, and RF blocks.</p> <p>A personal version, VisSim/Comm PE, is also available. VisSim/Comm PE is a subset of the Communication blockset. See the VisSim/Comm datasheet for details</p> <p>You can purchase VisSim/Comm add-on modules for real-time data acquisition (Red Rapids digital tuner cards); for modeling PCCC turbo codes, including UMTS specification; for support of Bluetooth, 802.11 a/b/g (Wi-Fi), and ultrawideband wireless designs.</p>
VisSim/Fixed-Point	Simulates the behavior of fixed-point algorithms prior to code generation and implementation of the algorithm on the fixed-point target.
VisSim/Knobs and Gauges	Provides dynamic gauges, meters, and knobs for process control, and measurement and validation systems.
VisSim/Model-Wizard	Generates transfer function model from historic or real-time data.
VisSim/Motion	Simulates motor control systems with customizable amplifiers, controllers, filters, motors, sensors, sources, tools, and transforms.
VisSim/Neural-Networks	Performs nonlinear system identification, problem diagnosis, decision-making prediction, and other problems where pattern recognition is important.
VisSim/OPC	Connects to any OPC server and log data or run a virtual plant in VisSim for offline tuning.
VisSim/OptimizePRO	Performs generalized reduced gradient method of parameter optimization.
VisSim/Real-TimePRO	Performs real-time data acquisition and signal generation using I/O cards, PLCs, and DCSs.
VisSim/Serial	Performs serial I/O with other computers.
VisSim/State Charts	Creates, edits, and executes event-based systems.
VisSim/UDP	Performs data exchange over the internet using UDP.
VisSim Viewer (free)	Lets you share VisSim models with colleagues and clients not licensed to use VisSim.

Resources for learning VisSim/Motion

For those of you that are new to VisSim, we have provided several free services to make your transition to VisSim fast, smooth, and easy:

Interactive webinars

Interactive webinars offer you the opportunity to meet with Visual Solutions product specialists who will introduce and demonstrate our software products live on your computer and answer any questions you have. Each webinar is approximately 45 minutes long. To learn more about our interactive webinars, go to

<http://www.vissim.com/webinars/webinars.html>.

Sample diagrams

VisSim 8.0 includes a directory of fully documented sample diagrams. These diagrams illustrate both simple and complex models spanning a broad range of engineering disciplines, including aerospace, biophysics, chemical engineering, control design, dynamic systems, electromechanical systems, environmental systems, HVAC, motion control, process control, and signal processing.

To access sample diagrams

Click on the **Diagrams** menu in VisSim.

Click on **Examples > Applications**.

Training

Visual Solutions offers training sessions for learning and gaining expertise in VisSim and the VisSim family of add-on products. Training sessions are conducted at Visual Solutions training facility in Westford, MA, as well as at customer sites and as online webinars.

For information on setting up a training session, contact sales@vissol.com.

Using the Tutorials

This section contains...

AC induction motor tutorial: speed control of a machine tool lathe

Tutorial Model: ...\\MOTION\\TUTORIAL\\MACHINE_TOOL_TUTORIAL.VSM

The typical machine tool lathe is operated from a single-speed motor drive, together with multiple gear selection to vary chuck speed. Here a simpler design is considered: one with a single 10:1 gear reducer and a variable speed control drive for a 3-phase AC induction motor.

The lathe is required to operate with the following specifications:

Maximum work piece load: 1 meter by 0.1 meter diameter aluminum bar stock

Chuck speed control range: 30 to 400 RPM

Speed control accuracy: ± 5 RPM from set point steady state

Maximum load torque: not to exceed 0.3 N-m, introduced by cutting tool

The motor specifications are given as:

Motor parameter	Value	Units
Stator resistance (per phase)	9.4	Ohms
Stator self inductance (per phase)	0.402	Henries
Stator leakage inductance	0.032	Henries
Rotor resistance	7.1	Ohms
Rotor leakage inductance	0.032	Henries
Number of poles	2	
Rotor inertia	0.001	Kg-m ²
Rotor viscous damping constant	0.0001	Kg-m ² - s

The moment of inertia of the chuck and moving drive assembly is given as 0.1 kg-m². The moment of inertia of the work piece is calculated as:

$$I = \frac{1}{2}Mr^2 = \left(\frac{1}{2}\right)(21.14\text{kg})\left(\frac{0.1\text{m}}{2}\right)^2 = 0.026\text{kg m}^2$$

Since the axes of the chuck and work piece are coincident, they add to total 0.126 kg m².

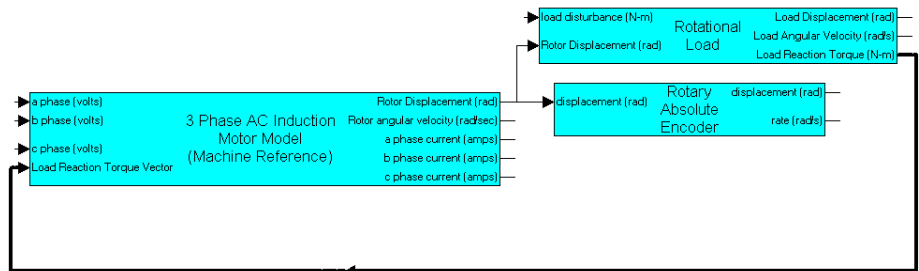
One very effective way of controlling speed by an induction motor is to control the stator field frequency. Since stator flux is inversely proportional to frequency below the base frequency, it is necessary to adjust voltage proportional to frequency to maintain constant flux. For frequency above the base frequency (power supply limitation), the voltage is kept constant. This method is the basis of the design, with one minor improvement. The constant volts to frequency control mentioned above are used as a feed forward leg of a feed forward – proportional integral controller (PI). The PI component of the control is used to adjust any error that may occur due to motor slip and loading from the cutting tool. Motor speed is estimated from motor shaft position measured by an incremental encoder. To drive the motor, an inverter is used with six-step logic to switch polyphase-rectified voltage producing a balanced 3-phase signal.

Setting up the motor, load, and encoder

The first step is to place the following Motion blocks in your diagram:

- Rotational Load (under Loads)
- AC Induction Motor (under Machine Reference)
- Rotary Encoder (under Sensors)

Wire the blocks together and use wirePositioner blocks to clearly represent the feedback of the load reaction torque to the motor model.



The rotational load model is used to simulate the lathe chuck and work piece. The rotary encoder model input is connected to the motor’s rotor shaft displacement output connector. The motor displacement output is also connected to the rotational load model. To complete the dynamic interaction between the motor and load, the load reaction torque output connector must be connected to the load reaction torque vector input of the motor model. Note that this wire is thicker than the other wired connections, indicating that it transmits a vector quantity. The vector contains load dynamic parameters that are reflected back to the motor dynamics through the coupling (linkage) mechanism: in this case, a 10:1 gear reduction.

Setting parameter values

The next step is to enter the parameters for the motor, load and encoder. The parameter values can be changed later to see what affect they may have on the final control solution.

AC Induction Motor block

Click the right mouse button over the AC Induction Motor block to display its Properties dialog box.

Parameter	Value
Stator Inductance (per phase) [H]:	2
Stator Resistance (per phase) [Ohms]:	.402
Stator Leakage Inductance [H]:	.032
Rotor Resistance [Ohms]:	7.1
Rotor Leakage Inductance [H]:	.032
Rotor Moment of Inertia (Kg-m ²):	.001
Rotor Shaft Coulomb Friction Magnitude (N-m):	0
Rotor Shaft Stiction Factor (N-m):	0
Rotor Shaft Viscous Damping Factor (Kg-m ² /s):	.0001

Enter the motor parameters shown above. These parameter values are taken from the motor specifications table shown on page xxxx.

Rotational Load block

Click the right mouse button over the Rotational Load block to display its Properties dialog box.

Parameter	Value
Enable Hard Stops	<input type="checkbox"/>
Linkage Backlash (rad):	0
Lower Stop Limit (rad):	-1
Upper Stop Limit (rad):	1
Linkage Ratio, Rotor Shaft/Load:	.1
Load Moment of Inertia (Kg-m ²):	.126
Load Viscous Damping Factor (Kg-m ² /s):	.1
Load Coulomb Friction Magnitude (N-m):	0
Load Spring Constant (N-m):	0
Load Spring Preload Torque (N-m):	0

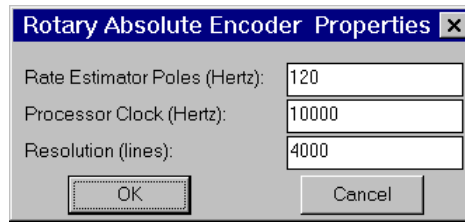
Enter the values shown above.

Note the following:

- The value for the Load Viscous Damping Factor is a rough guess.
- For the linkage ratio (gear ratio for this application), follow this rule: a factor less than 1.0 multiplies torque, and a factor greater than 1.0 multiplies speed; entering 1.0 produces a direct connection between motor and load.
- Default values are shown for the upper and lower stop limits, but since the Enable Hard Stops checkbox is not activated, hard stop limits are not used in the model. Hard stops are useful in position control system applications.

Rotary Encoder block

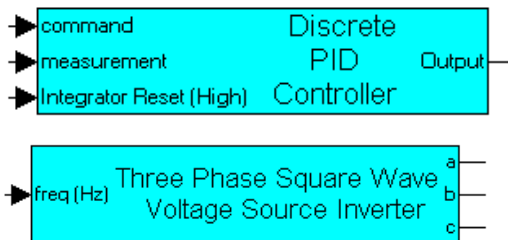
Click the right mouse button over the Rotary Encoder block to display its Properties dialog box.



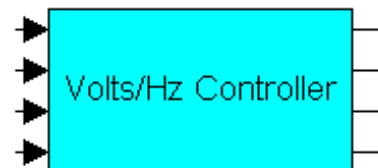
Enter the values shown above.

Designing the volts/frequency controller for the motor

In this step, you use the PID Controller (Digital) block (from the Controllers category), and the Three Phase Square Wave Inverter (from the Sources category) to design the volts/frequency controller for the motor.

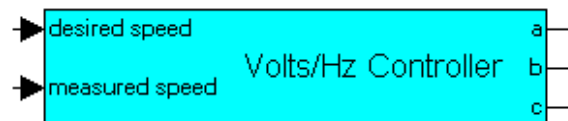


After placing the blocks in your diagram, encapsulate them in a compound block using the Edit > Create Compound Block command. Use the name Volts/Hz Controller for the compound block name.



This block design requires only 2 inputs and 3 outputs. By default, when you create a compound block, VisSim creates and outputs for all the blocks contained in the compound, which may not be appropriate. In this case, you must remove 2 inputs and one output using the Edit > Remove Connector command.

Now label the wire tab connectors by double clicking each tab connector with the left mouse button and entering the names as shown below:



Next, click the right mouse button over the compound block to drill into it. As mentioned before, the process of creating the compound block creates wire connections that may not be needed. Remove all wire connections within the

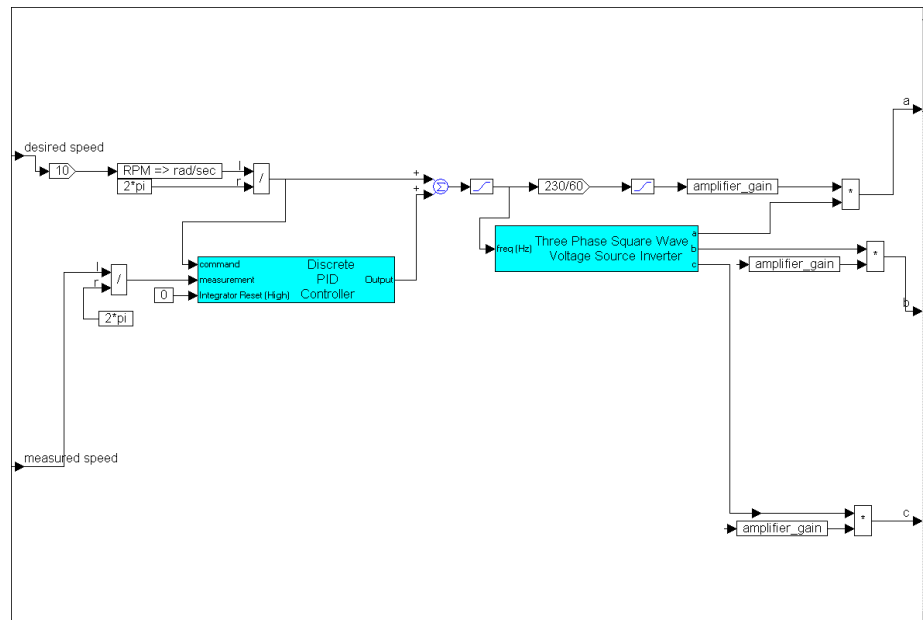
compound by clicking, holding and dragging the wires with the left mouse button to an open space of the desktop and releasing the button.

Customizing the Volts/Hz Controller block

Within the Volts/Hz Controller compound block, insert the following blocks:

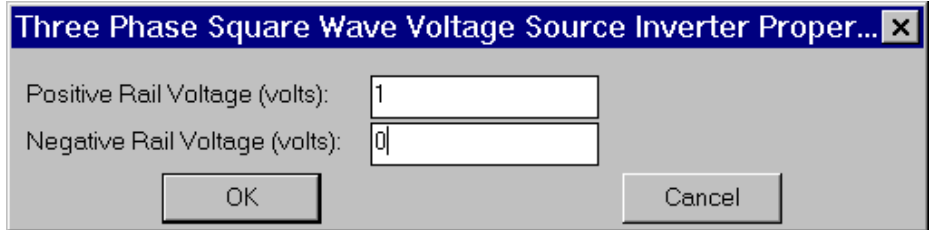
- From the Blocks > Arithmetic category, add a summingJunction, gain, unitConversion, and / block.
- From the Blocks > Signal Producer category, add a const block.
- From the Blocks > Nonlinear category, add a limit block.

Wire the blocks together, as shown below.



The input speed for this block is assumed to be the speed of the chuck; therefore, the gain block is needed to scale this speed up by a gear ratio of 10 since this controller affects the speed on the motor side. RPM is then converted to hertz by using a unitConversion block set to RPM⇒rad/sec and then dividing the output by 2π . The value 2π is produced by using a const block set to $2*\pi$.

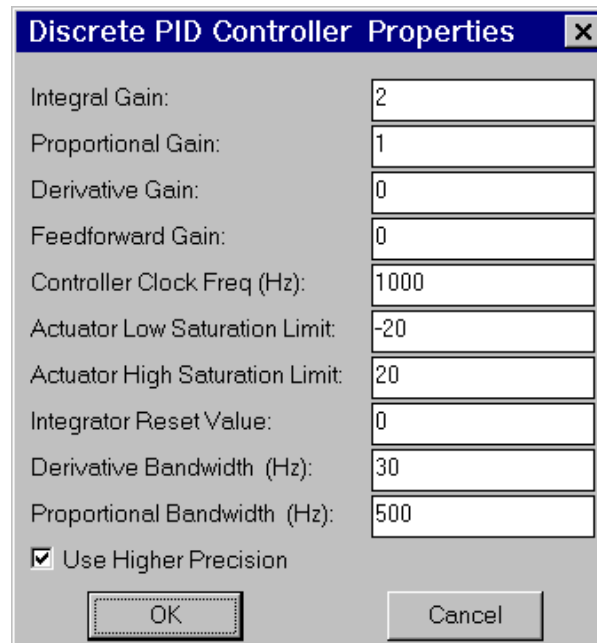
The measured speed comes from the Rotary Encoder and is in radians per second. This measurement is converted to hertz simply by dividing by $2*\pi$. The desired speed in hertz is fed into a summingJunction block, as well as the command input of the PID Controller (Digital) block. The desired speed directly feeds the inverter/amplifier as the feed forward component of the control. PID Controller (Digital) block output is used to correct for minor errors in the feed forward component. The sum of these two components is fed to the inverter/amplifier, the sum is limited to 70 hertz to prevent running the motor into its unstable region of control. The output of the limit block feeds the Three Phase Square wave Inverter block. The Three Phase Square wave Inverter block rail voltages must be set to 0 and 1, as shown below, to effectively provide logic control rather than bus level voltages:



The output of the control summing junction block is scaled inversely proportional to frequency by using a gain block with the factor 230/60. The output is then limited between 0 and 230 volts, and defined to be a variable block with the user-defined name amplifier gain.

Configuring the PID compensator

To configure the PID compensator, enter the following values into the PID Controller (Digital) block:



Since the feed forward and derivative gain are set to 0, the block is actually configured to operate as a PI controller. Saturation is set to limit the influence of the integral correction to ± 20 Hz. Proportional bandwidth is set at Nyquist frequency ($\frac{1}{2}$ the sampling frequency); derivative bandwidth does not matter in this controller. The Use Higher Precision option is turned ON to allow trapezoidal integration to be used.

Integral reset is not used on this controller, so a `const` block with a value of 0 is fed into PID Controller (Digital) to prevent integral reset. The actual values for the proportional and integral gain were determined experimentally in the final configuration to obtain minor overshoot and settling in the control.

This completes the construction of the Volts/Hz Controller compound block.

Wiring Volts/Hz Controller to the overall simulation

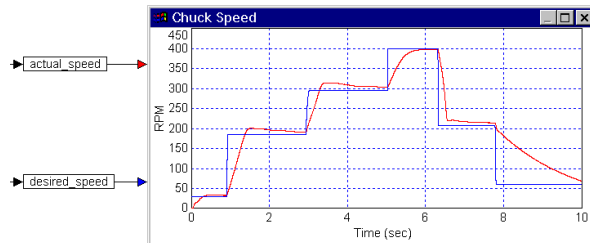
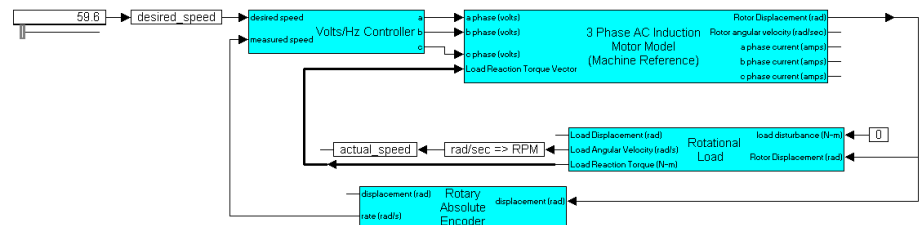
The three outputs for Volts/Hz Controller are connected to the corresponding inputs of the induction motor block. Measured speed from the Rotary Encoder block is

connected to the measured speed input of the Volts/Hz Controller block. A slider block, scaled between 30 and 400, is connected to the desired speed input of the Volts/Hz Controller block as RPM speed input. A plot block is wired to compare the desired and actual speeds. The actual speed is determined by converting load angular velocity to RPM. A const block set to 0 is connected to the load disturbance input of the rotational load model. WirePositioner and variable blocks are used to make the diagram legible.

Before simulating the model, click on the Simulate > Simulation Properties command and make the following selections:

- In the Start Time box, enter 0
- In the Step Size box, enter 0.0001
- In the End Time box, enter 10

Through minor exploration, the motor drive is found to have sufficient torque at all speeds to overcome maximum tool exertion.



Now with a working simulation, you have met the design requirements and can now begin playing simulation games to optimize performance and reduce cost. For example, a fairly high-resolution encoder was used for estimating rate. How coarse can the resolution become before performance is degraded? Also the motor may be oversized for the particular application. Surveys show that over 50% of the motors selected in the US are oversized for their application. Simulation provides a lower cost alternative to performing extensive analysis or purchasing a variety of motors to empirically determine which is best suited for an application. This is true for any motion control application; not just limited to machine tools.

AC induction motor application: power train drive system

The process of building the simulation for a machine tool application can be similarly applied to a power train drive system for a passenger electric vehicle. Here the design could start with a 3-phase AC induction motor model, which provides the necessary horsepower and speed to drive the vehicle. Based on the vehicle mass and wheel size, a rotational load model can be configured. Losses due to wheel bearing friction can also be included as a combination of stiction, Coulomb, and viscous

friction within the **Rotational Load** block parameters. Using the wheel size, output angular velocity for this model can be converted to vehicle linear speed.

Additional static blocks can be configured to model drag forces on the vehicle that can be fed back into the load disturbance of the rotational load block. Selecting a linkage ratio of 1, and using additional blocks to connect between the motor and load displacement connections can simulate a clutch and automatic transmission. Gravity induced loading can be simulated as a function of road angle and superimposed with other load disturbance inputs. The purpose of such a simulation could possibly focus on optimization of motor parameters or the development of a new method for automatic gear shifting.

Brushless DC (BLDC/PMSM) motor tutorial: target tracking system

Tutorial Model:

...\MOTION\TUTORIALS\TARGET_TRACKING_TUTORIAL.VSM

This example simulates a servo-controlled positioning system that maintains focal plane line of sight coincident with target angle. The permanent magnet synchronous motor model is selected as an actuator to provide fast response.

Motor Specifications

Automatically acquiring and maintaining the line of sight of a video camera or focal plane sensor is often required in various aerospace, defense, and security system applications. One way to mechanize such a system is to reflect the field of view through two independently-controlled mirrors that each rotate in axes orthogonal to one another. The object of the control system is to acquire the target, and by controlling rotation of each mirror, move the line of sight coincident with the target angle. This places the virtual image of the target in the center of the focal plane. Once the image of the target is acquired on the focal plane, an error in azimuth and elevation can be determined by a variety of image processing techniques, such as contrasting, differencing, and area parameter calculations.

For this simulation, such a mechanism is assumed, with a pipeline image processor providing direct angular azimuth and elevation measurements. The following design decisions are also assumed:

Motor type: Permanent magnet DC synchronous motor with Hall sensors for commutation sensing and control.

Motor parameter	Value	Units
Operating voltage	28	Volts
Magnetizing inductance	0.0009	Henries
Stator inductance (per phase)	0.001	Henries
Stator resistance (per phase)	0.5	Ohms
Torque constant	0.1035	N_m/A
Number of poles	2	
Rotor moment of inertia	8.5 E-06	kg-m ²
Rotor shaft viscous damping factor	5.695 E-06	kg m ² /s

For the simulation, the Brushless DC (BLDC/PMSM) Motor block is used, along with the Hall Sensor block for commutation.

Power Electronics: Brushless PWM servo amplifier with speed and current Control. The base frequency of the PWM is 9000 Hz.

For the simulation, the `PWM Brushless Servo Amplifier` block is used

Precision current sense resistors produce voltage that is fed into a processor. An encoder provides motor shaft position and velocity. Encoder angle measurement and phase current measurements are used to obtain direct and quadrature current estimates through Clarke and Park transforms. Current and speed loops are used to set stiff inner loop performance.

Mechanical Load: Precision $\lambda/4$ flat oval mirrors mounted on a gear reducer shaft with rotation center coincident with reflecting surface represent the main load moment of inertia. A torsional spring with preload tension is used to help minimize backlash hysteresis. An optical encoder is provided with 16000 lines to measure mirror angle. PI compensation is used for controlling line of sight. Load parameters are:

Gear reduction	20:1
Backlash	0.0005 radians
Load moment of inertia	0.001 kg – m ²
Load viscous damping	0.01 kg – m ² /s
Load spring constant	0.01 N-m/rad
Load spring preload	0.1 N-m

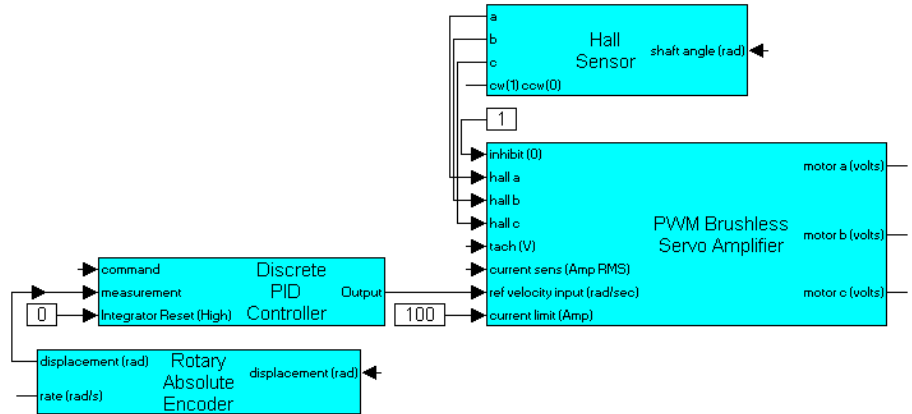
Pipeline Image Processor: Provides 60 Hz frame rate acquisition of target from focal plane array. Pixel resolution is sufficiently higher than expected control requirement of less than ± 3 degrees between target angle and line of sight in both axes. Hierarchical classification and size discrimination of blobs with subsequent calculation of the target centroid determine target position.

Simulation development

From the Motion block set, place the following blocks in your diagram:

- PID Controller (Digital)
- Hall Sensor
- PWM Brushless Servo Amplifier
- Rotary Encoder

Flip the Rotary Encoder and Hall Sensor blocks using the Edit > Flip Horizontal command. Then arrange the blocks and wire them together, as shown below.



Connect the Hall Sensor block outputs to the corresponding inputs of the Brushless PWM Servo Amplifier block. Then, wire the output of the PID Controller (Digital) block to the reference velocity input of the Brushless PWM Servo Amplifier block. Finally, connect the displacement output of the Rotary Encoder block to the measurement input of the PID Controller (Digital) block.

Two const blocks are fed into the Brushless PWM Servo Amplifier block; another const block is fed into the PID Controller (Digital) block:

Set the value of the const block, wired into inhibit, to 1. This prevents inhibit.

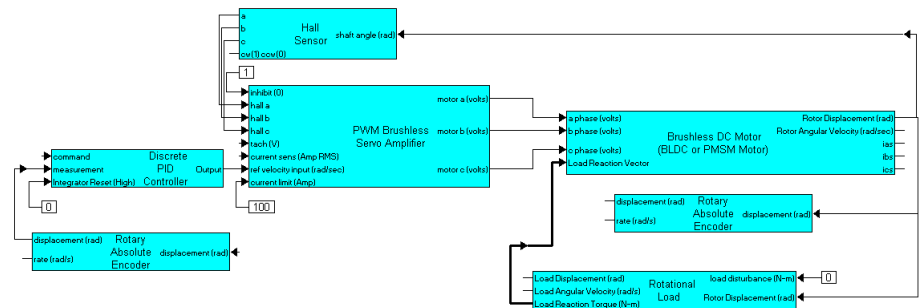
In this particular application there is no reason to reset the integration of the PID Controller (Digital), so a 0 const is wired to Integrator Reset (High) to disable. In other applications repetitive control may be used, and Integrator Reset (High) may be required to re-initialize the control between repetitions.

A value of 100 amps is chosen for this example to make certain saturation does not occur. Later on, you might possibly measure currents encountered in this simulation under highest load conditions and set a more appropriate current limit for the final design.

Next, place the following Motion blocks in the diagram:

- Brushless DC (BLDC/PMSM) Motor
- Rotary Encoder
- Rotational Load

Flip the Rotary Encoder and Rotational Load blocks using the Edit > Flip Horizontal command. Then arrange the blocks to the right of the previous construction, as shown in the following diagram:



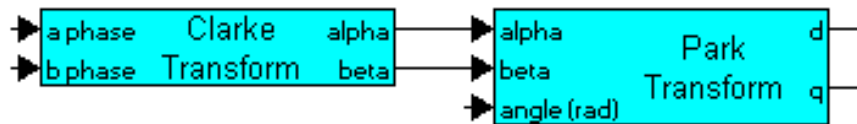
The rotor output displacement of the Brushless DC (BLDC/PMSM) Motor block connects to three other block connections: the displacement input of the Hall Sensor

block, the input of the Rotary Encoder, and the rotary displacement input of the Rotational Load block.

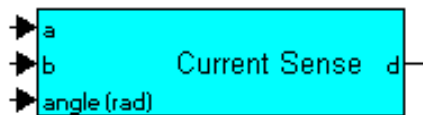
Connect the outputs of the PWM Brushless Servo Amplifier block to the corresponding inputs of the Brushless DC (BLDC/PMSM) Motor block. Connect the load reaction torque vector output connector on the Rotational Load block to the load reaction vector input of the Brushless DC (BLDC/PMSM) Motor block.

Lastly connect a const block with 0 set value to the load disturbance input connector on the Rotational Load block. If there were other torques related to influences that could not be directly represented by the set parameters of the rotary load model, the load disturbance input provides a method for introducing such torques. For the target tracker, it might be conceivable to introduce torque noise induced by structural vibrations of the tracker mount. If the mount were part of a satellite payload, such vibrations could arise from solar array positioning systems. Noise profiles with specific power spectral densities can be generated in VisSim using the Random Generator blocks and transferFunction block. Coefficients of the transfer function are determined by applying spectral factorization techniques to the known PSD.

Next, insert a Park Transform and a Clark Transform block into the diagram and connect them as shown below:

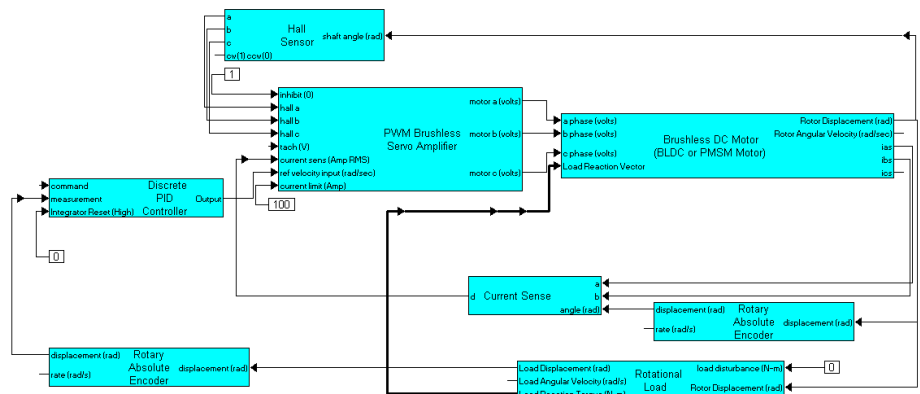


Encapsulate the blocks in a compound block and name it Current Sense. Then label the input and output connector tabs as shown below:



Flip the block 180° and connect the ias and ibs output connectors of the Brushless DC (BLDC/PMSM) Motor block to the corresponding a and b inputs of the Current Sense compound block. Connect the displacement output of the Rotary Encoder block to the angle input of the Current Sense block. Connect the load displacement output of the Rotational Load block to the displacement input of the other Rotary Encoder block.

Complete the wiring by connecting the output of the Current Sense block to the current sense input of the PWM Brushless Servo Amplifier block and the rate output of the Rotary Encoder block to the tach input of the PWM Brushless Servo Amplifier block, as shown in the following diagram:



This block construction represents a cascade control loop. The inner loop senses and controls current; the middle loop senses and controls velocity; and the outermost loop senses and controls position.

Now the entire block construction must be captured within a single compound block. Give this block the name X Axis Servo. Reduce the number of inputs and outputs on the compound block to one of each. Label the input as commanded LOS and output as actual LOS.

Next, drill into X Axis Servo and make certain that the commanded LOS is connected to the command input of the PID compensator block and the displacement output of the rotational load model is connected to the actual LOS output of the compound block.

While still in the X Axis Servo compound block, open the dialog boxes of each Motion block and enter the following parameter values as specified by the design input:

PID Controller (Digital) block

The screenshot shows the 'Discrete PID Controller Properties' dialog box. It contains the following parameters and values:

Integral Gain:	10000
Proportional Gain:	1000
Derivative Gain:	0
Feedforward Gain:	0
Controller Clock Freq (Hz):	1000
Actuator Low Saturation Limit:	-400
Actuator High Saturation Limit:	400
Integrator Reset Value:	0
Derivative Bandwidth (Hz):	30
Proportional Bandwidth (Hz):	500
<input checked="" type="checkbox"/> Use Higher Precision	

Buttons: OK, Cancel

PWM Brushless Servo Amplifier block

The screenshot shows the 'PWM Brushless Servo Amplifier Properties' dialog box. It contains the following parameters and values:

Supply Voltage (V):	28
PWM frequency (Hz):	9000
Velocity Loop Integral Gain (A/rad):	100
Velocity Loop Proportional Gain (A-s/rad):	3
Current Loop Integral Gain (%mod/A-s):	0
Current Loop Proportional Gain (%Mod/Amp):	2.5
Transconductance Gain (Amp/Volt):	.4
Tachometer Sensitivity (V/rad/s):	1

Buttons: OK, Cancel

Rotary Encoder block that feeds back to PID Controller (Digital) block

Rotary Absolute Encoder Properties [X]

Rate Estimator Poles (Hertz):

Processor Clock (Hertz):

Resolution (lines):

Rotary Encoder block that feeds back into the PWM Brushless Servo Amplifier block

Rotary Absolute Encoder Properties [X]

Rate Estimator Poles (Hertz):

Processor Clock (Hertz):

Resolution (lines):

Brushless DC (BLDC/PMSM) Motor block

Brushless DC Motor (BLDC or PMSM Motor) Properties [X]

Number of Poles:

Stator Inductance (per phase) (H):

Stator Resistance (per phase) (ohms):

Stator Magnetizing Inductance (H):

Rotor Moment of Inertia (Kg-m²):

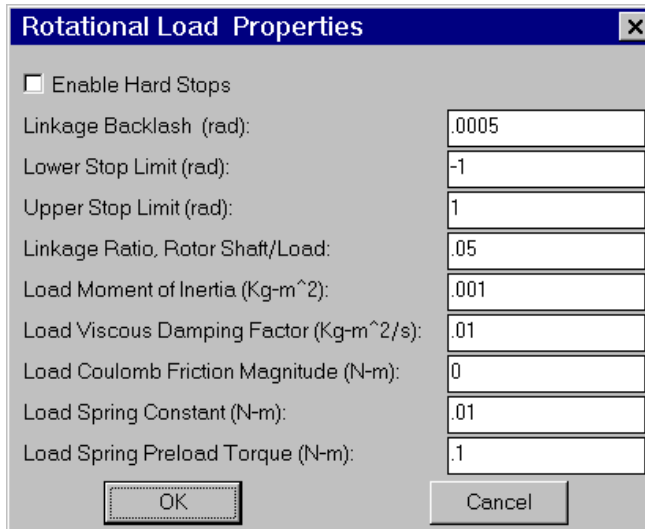
Rotor Shaft Coulomb Friction Magnitude (N-m):

Rotor Shaft Stiction Factor (N-m):

Rotor Shaft Viscous Damping Factor (Kg-m²/s):

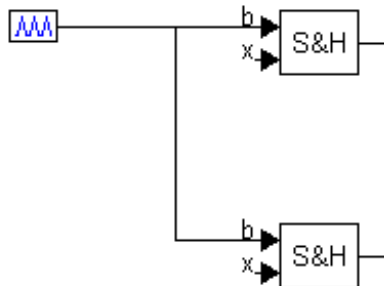
Torque Constant (N-m/A):

Rotational Load block

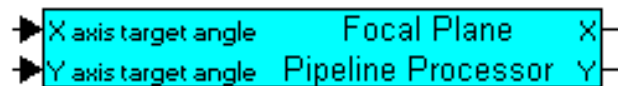


This completes the X-axis of the servo controller. Completing the Y-axis takes only a couple of keystrokes, as all dynamics for this axis are assumed to be equal. Make a duplicate copy of the X Axis Servo block using the Edit > Copy command. In the dialog box for the newly-created X Axis Servo, change the name of the block to Y Axis Servo. At this point, there are two servo controllers in your diagram: an X-axis and a Y-axis servo controller.

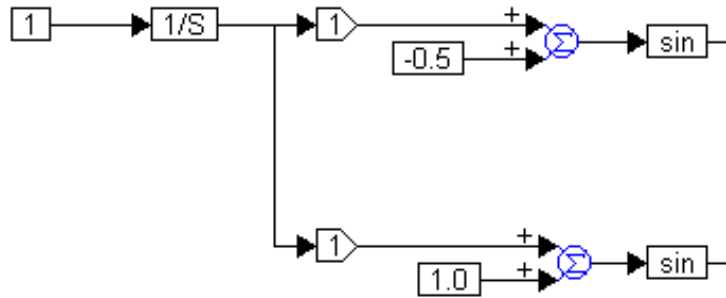
Next, create a simulation of the pipeline image processor. For this processor, the dominant feature is the sample frame rate of 60 hertz. Place two sampleHold blocks (located under Blocks > Nonlinear) and a pulseTrain block (located under Blocks > Signal Producer) in your diagram. Arrange these three blocks as shown below:



In the pulseTrain block, set the time between pulses to 1/60 (0.0167). Then encapsulate the three blocks in a compound block named Focal Plane Pipeline Processor.



Next, create the following block configuration:



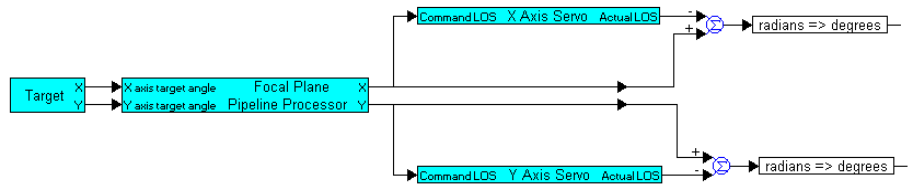
This construction is used to create an elliptical motion for the target in the X-Y plane. Frequency for each axis is the same (1 rad/sec); however, phase differs.

Note that the integrator (1/S) and sin blocks are located under Blocks > Integration and Blocks > Transcendentals, respectively.

Enclose this construction in a compound block and name it Target:



Connect the compound blocks as shown in the following diagram:

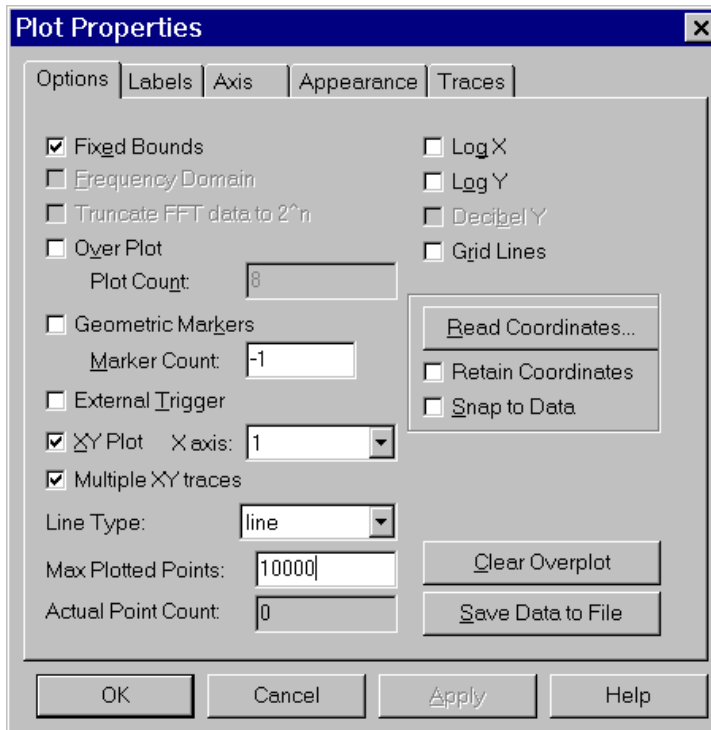


In this construction, command line of sight (LOS) is set to the target angle, which is determined by the pipeline processor. The difference between the target angle and actual line of sight is calculated using summingJunction blocks, which provide focal plane error. The error is converted into degrees by unitConversion blocks set to convert radians to degrees.

Setting up the plot blocks

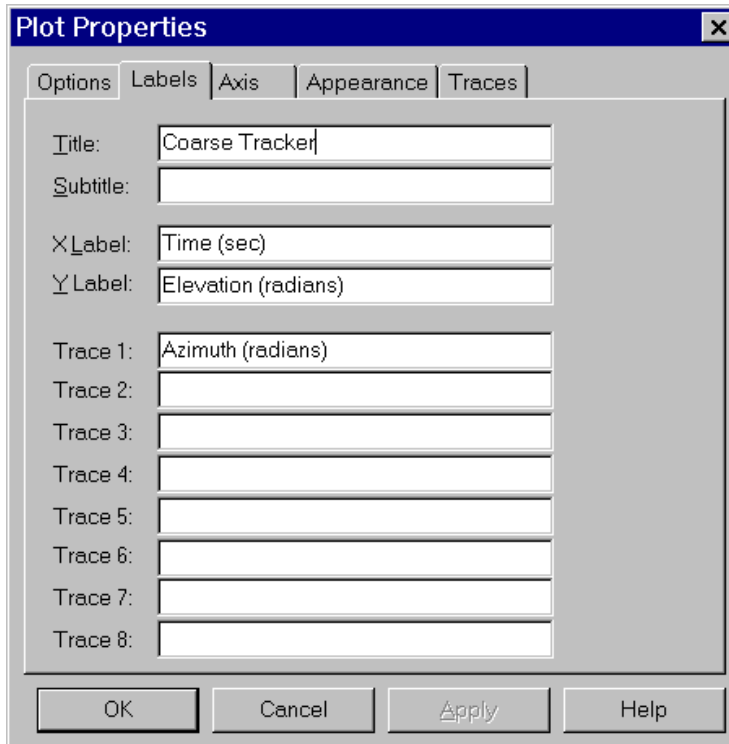
Next, plot blocks are prepared to display simulation output.

Place a plot block (located under Blocks > Signal Consumer) in the diagram; then open its dialog box and under the Options tab enter the following settings:

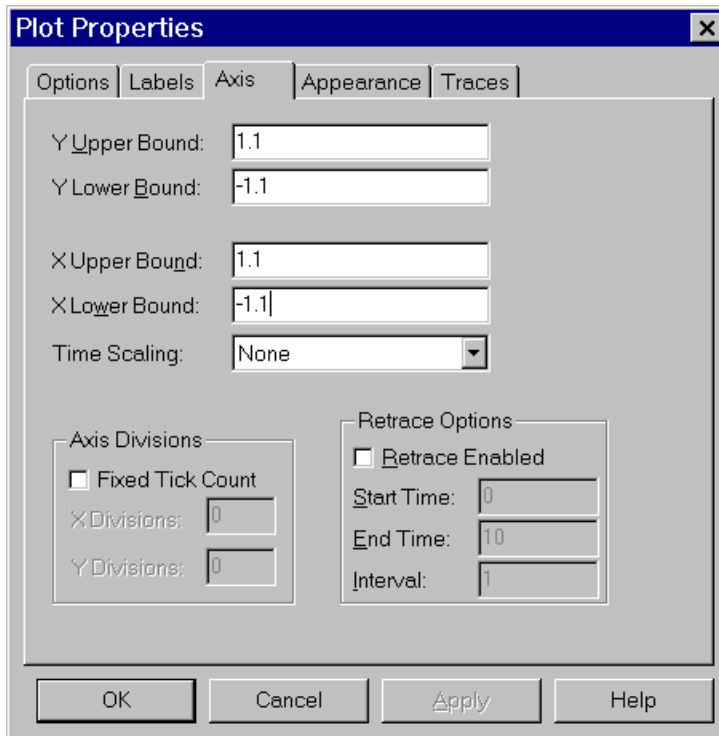


Note that the Multiple XY Traces option is activated. This feature allows the display of the target motion independently from the servo line of sight.

Under the Labels tab, enter the information shown below.



And under the Axis tab, enter the following information:



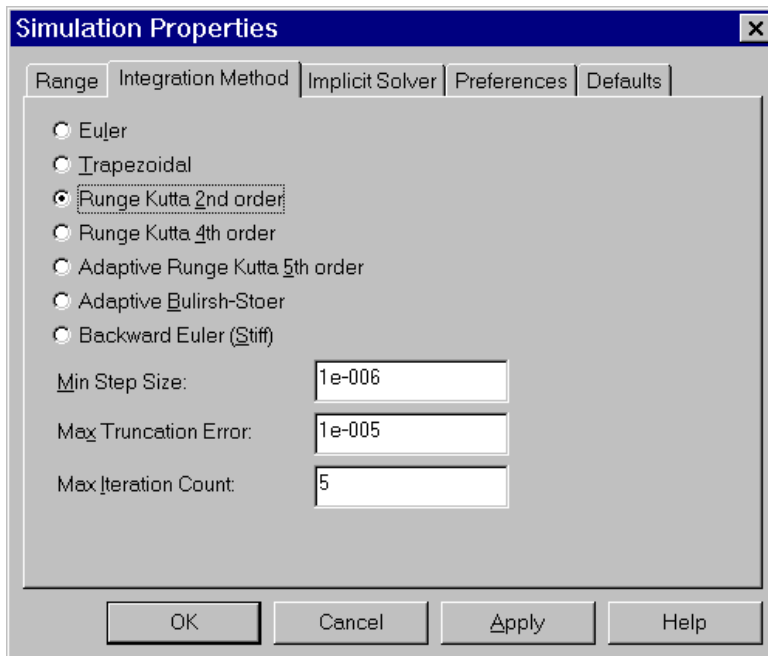
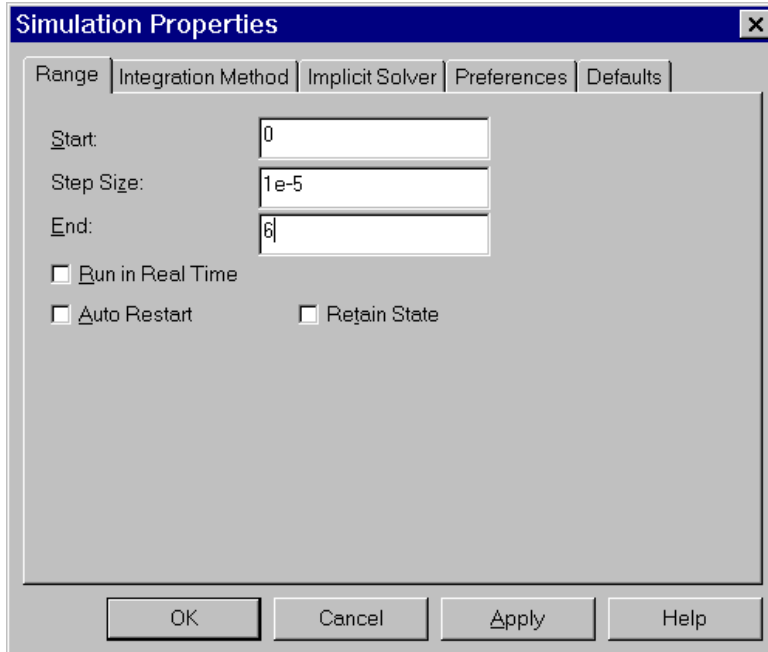
Place another plot block on the diagram. Enter the same parameters as in the previous plot with these exceptions:

Under the Labels tab, make these changes:

- In the Title box, enter Focal Plane
- Enter degrees as units instead of radians
- Under the Options tab, make these changes:
 - Activate the Fixed Bounds option
 - De-activate the Multiple XY Traces option
- Under the Axis tab, make these changes:
 - In the X Upper Bound and Y Upper Bound boxes, enter 5
 - In the X Lower Bound and Y Lower Bound boxes, enter -5.

Setting the simulation properties

Simulation properties are set through the Simulate > Simulation Properties command. Enter the following information to the Simulation Properties dialog box.



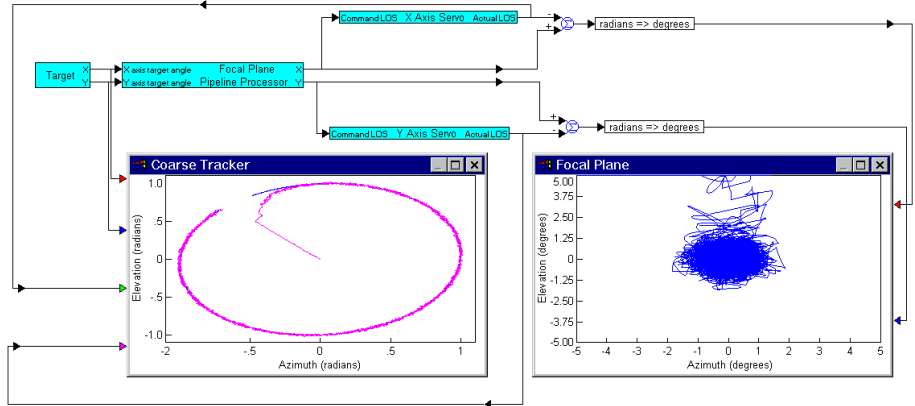
For this simulation, a very small step size is necessary because pulse width modulation is being simulated at 9000 hertz.

Final configuration requirements

Connect the X and Y outputs of the Target compound block to the first two input tabs of the Coarse Tracker plot block and the output X and Y servo compound blocks actual line of sights to the next two input tabs of the same plot block.

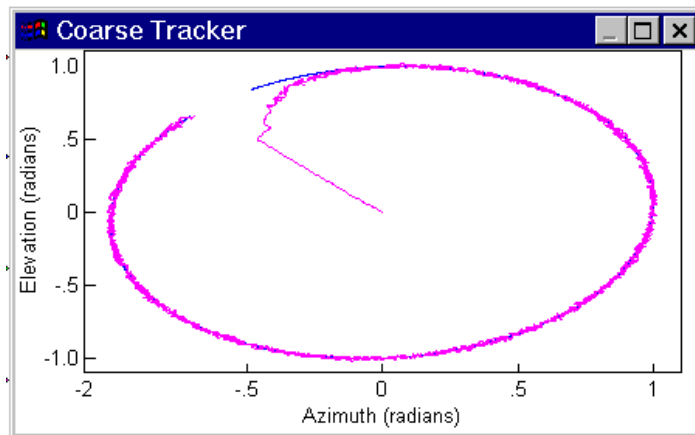
Connect the output of the two unitConversion blocks to the first two input connectors of the Focal Plane plot block.

You are now ready to run the simulation with the Simulate > Go command.

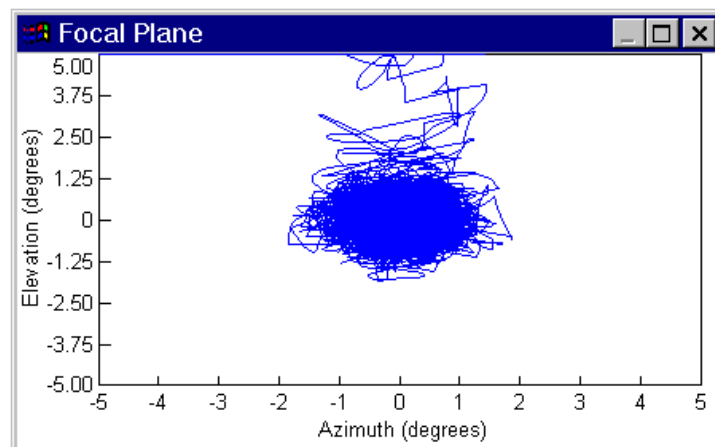


Simulation results

The following plot shows the acquisition and tracking of the actual target's elliptical motion with the servo line of sight:



To better illustrate accuracy, the following plot shows the focal plane error. The darkened circular area represents the time after the servo has acquired the target and begins tracking. These results show errors to be on the order of 1° , exceeding the requirement.



It should be noted that to get to this level of control required tuning of each of the control loops with multiple iterations before an acceptable control was achieved.

Other applications

The process of building the simulation for a tracking system can similarly be applied to a tape drive speed and tension control system. Here the design could start with a permanent magnet DC synchronous motor model providing necessary torque to rapidly accelerate a tape spindle to a desired speed. A model could be developed that provides simulated track positioning information for indexing tape position and controlling start and stop profiling.

Other things you can do with VisSim/Motion

The Motion block set, together with the basic block set in VisSim, provide all the necessary elements to simulate motion control applications in a variety of disciplines, including:

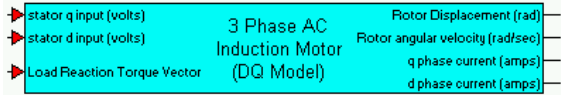
- Aerospace
- Automotive
- Defense
- Factory automation
- Industrial robotics
- Medical instrumentation and surgical tools
- Office automation and computer peripheral drives
- Optics
- Semiconductor pick-and-place machines

Block Reference

The Motion menu lists the blocks provided by VisSim/Motion. When you click on the Motion menu, most of the items that appear have a filled triangle (▶) next to them. These items are block categories. Click on a block category and a cascading menu appears listing the additional blocks.

To make it easier to find blocks in this chapter, they are presented in alphabetical order, regardless of their block category.

AC Induction Motor (DQ)



Example Model: ... \MOTION\EXAMPLES\ACIM_DQ_MODEL_EXMPL.VSM

Block Category: Motors

The AC Induction Motor (DQ) block is derived from the system of nonlinear differential equations describing electromechanical motion of a 3-phase AC induction motor written in the arbitrary reference frame and reduced to the stationary frame by setting the frame angular velocity equal to 0. Input requires direct-quadrature (DQ) voltages and a load reaction torque vector if external mechanical load is used.

Mechanical dynamics include typical parameters, such as rotor shaft inertia and viscous friction. In addition, nonlinear dissipative factors, including Coulomb friction and stiction models are provided. The block can operate stand-alone to produce output displacement or velocity of the motor alone, or when a Rotation Load block is connected, combined dynamic response. When connected to the Rotation Load block, dynamic parameters are reflected back and combined with the motor dynamics by the linkage ratio. The linkage ratio is specified in the Rotation Load block. This creates proper dynamic motion of the combined motor-load connection. To connect the AC Induction Motor (DQ) block to the Rotation Load block, the motor displacement and load reaction vector connections from each block must be wired together.

Rotor shaft position, velocity and stator DQ phase currents are provided for sensor connections in monitoring and feedback applications.

3 Phase AC Induction Motor (DQ Model) Properties	
Number of Motor Poles:	2
Stator Resistance (per phase) (Ohms):	4.495
Stator Inductance (per phase) (H):	.165
Stator Leakage Inductance (H):	.016
Rotor Resistance (Ohms):	5.365
Rotor Leakage Inductance (H):	.013
Rotor Moment of Inertia (Kg-m ²):	.001
Rotor Shaft Coulomb Friction Magnitude (N-m):	0
Rotor Shaft Stiction Factor (N-m):	0
Rotor Shaft Viscous Damping Factor (Kg-m ² /s):	.001

Number of Motor Poles: Requires the number of motor pole pairs.

Stator Resistance (per phase): Requires the stator per phase coil resistance in ohms.

Stator Inductance (per phase): Requires the stator per phase coil inductance in henries.

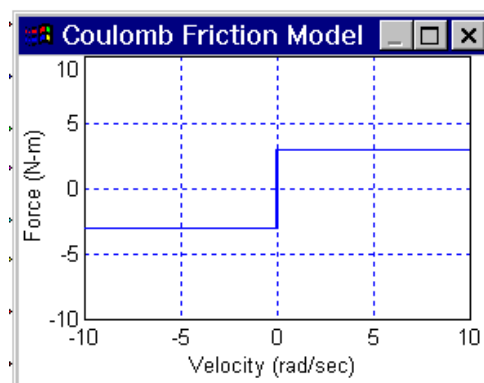
Stator Leakage Inductance: Requires the specified stator leakage inductance in henries.

Rotor Resistance: Requires the rotor winding resistance in ohms.

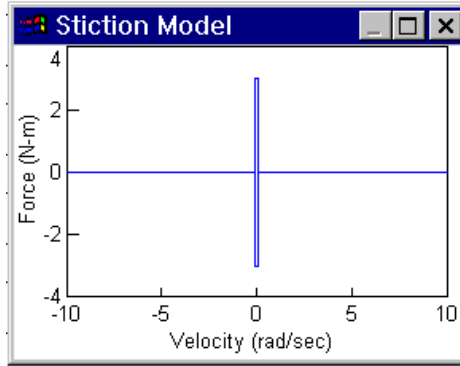
Rotor Leakage Inductance: Requires the specified rotor leakage inductance in henries.

Rotor Moment of Inertia: Requires the moment of inertia of the rotor with respect to the axis of rotation in kg-m².

Rotor Shaft Coulomb Friction Magnitude: Allows specification of constant directional dissipative force (Coulomb model) in units of N-m.



Rotor Shaft Stiction Factor: Allows specification of a stiction force value or break-away torque. This parameter is normally not specified by the motor manufacturer, but can be obtained experimentally. Units are in N-m.

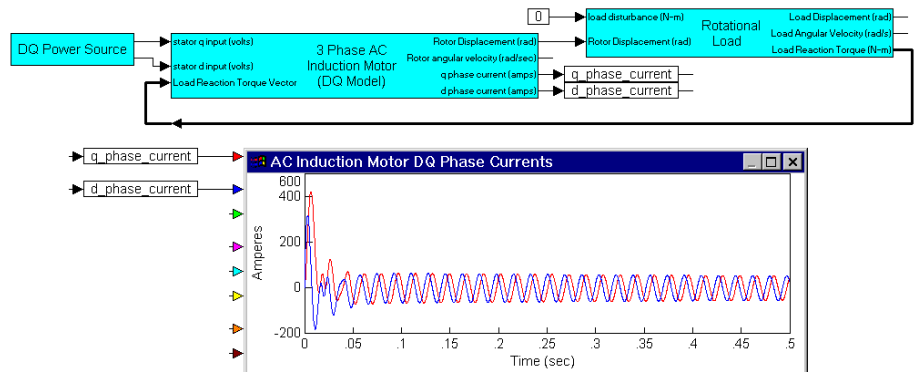


Rotor Shaft Viscous Damping Factor: Requires the factor that linearly relates viscous damping force to angular velocity. This parameter is normally not specified by the motor manufacturer, but can be determined experimentally. Units are $\text{kg}\cdot\text{m}^2/\text{s}$.

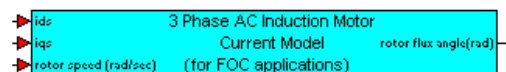
Example

Display of DQ reference frame stator currents at start up of AC induction motor (... \MOTION\EXAMPLES\ACIM_DQ_MODEL_EXMPL.VSM)

A DQ frame voltage source is created using a 3-phase source, and Clarke and Park's transforms (see example under Park Transform). Rotor displacement is connected to the Rotational Load block. Reaction torques are fed back to the motor model. DQ transient currents are observed by plotting signals from the output wires provided on the motor model.



AC Induction Motor Current Model (FOC)

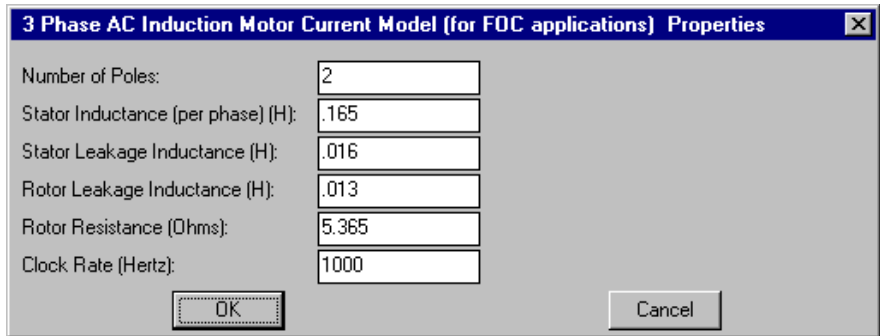


Example Model: ... \MOTION\EXAMPLES\ACIM_FOC_MODEL_EXMPL.VSM

Block Category: Controllers

The AC Induction Motor Current Model (FOC) block is used for field-oriented control applications.

The AC Induction Motor Current Model (FOC) block uses sensed stator currents of an AC induction motor model (converted to DQ frame currents) and speed measurements to estimate the angular position of the stator magnetic field (direct axis) based on known motor parameters.



Number of Poles: Requires the 3-phase AC induction motor number of pole pairs.

Stator Inductance (per phase): Requires the 3-phase AC induction motor stator per phase inductance in henries.

Stator Leakage Inductance: Requires the 3-phase AC induction motor specified stator leakage inductance in henries.

Rotor Leakage Inductance: Requires the specified 3-phase AC induction motor rotor leakage inductance in henries.

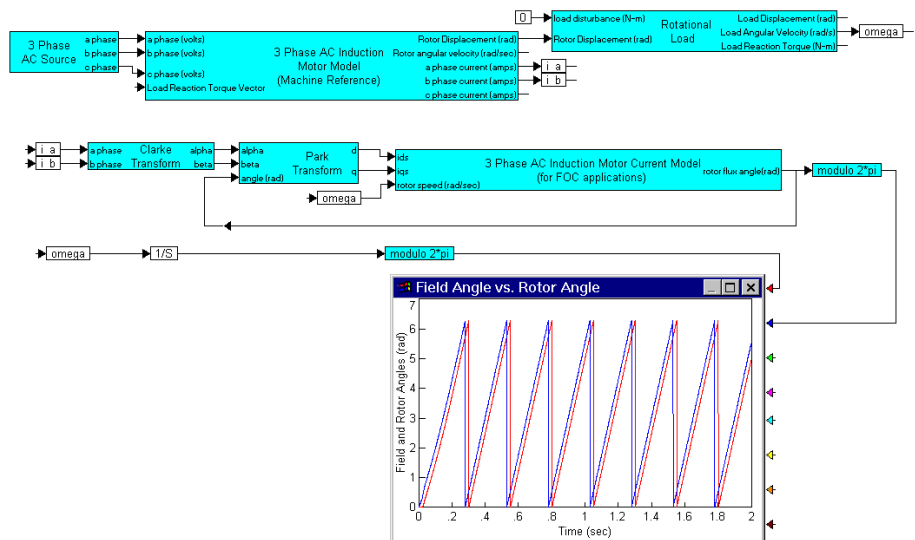
Rotor Resistance: Requires the 3-phase AC induction motor rotor winding resistance in ohms.

Example

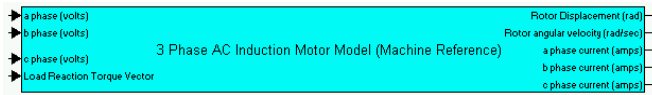
Comparison of rotor (shaft) angle and stator (field) angle estimated from current model

(..\MOTION\EXAMPLES\ACIM_FOC_MODEL_EXMPL.VSM)

The following simulation demonstrates how the field angle varies over time with respect to the rotor angle while operating the motor open loop at steady state velocity.



AC Induction Motor (Machine Reference)



Example Model: ... \MOTION\EXAMPLES\ACIM_MACH_REF_EXMPL.VSM

Block Category: Motors

The AC Induction Motor (Machine Reference) block is derived from the system of nonlinear differential equations describing electromechanical motion of a 3-phase AC induction motor written in the arbitrary reference frame, and reduced to the stationary frame by setting the frame angular velocity equal to 0. Park and Inverse Park transforms are then used to transform 3-phase voltage input to DQ voltages and output DQ currents to the 3-phase (abc) reference frame. Input requires 3-phase voltages, and a load reaction torque vector if external mechanical load is used.

Mechanical dynamics include typical parameters such as, rotor shaft inertia and viscous friction; in addition, nonlinear dissipative factors including, Coulomb friction and stiction models are provided. The AC Induction Motor (Machine Reference) block can operate stand-alone to produce output displacement or velocity of the motor alone, or when a Rotational Load block is connected, combined dynamic response. When connected to the Rotational Load block, dynamic parameters are reflected back and combined with the motor dynamics by the linkage ratio. The linkage ratio is specified in the Rotational Load block. This creates proper dynamic motion of the combined motor-load connection. To connect the AC Induction Motor (Machine Reference) block to the mechanical Rotational Load block, the rotor displacement and load reaction vector connections from each block must be wired together.

Rotor shaft position, velocity and stator 3-phase currents are provided for sensor connections in monitoring and feedback applications.

Number of Motor Poles: Requires the number of motor pole pairs.

Stator Inductance (per phase): Requires the stator per phase inductance in henries.

Stator Resistance (per phase): Requires the stator per phase resistance in ohms.

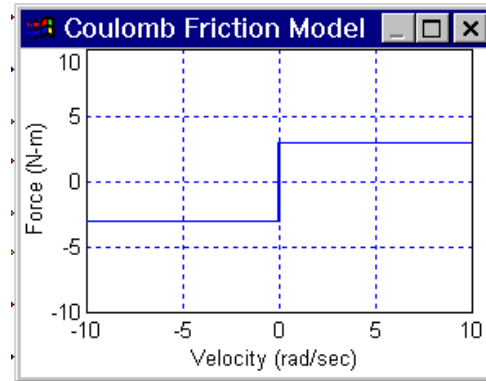
Stator Leakage Inductance: Requires the specified stator leakage inductance in henries.

Rotor Resistance: Requires the rotor winding resistance in ohms.

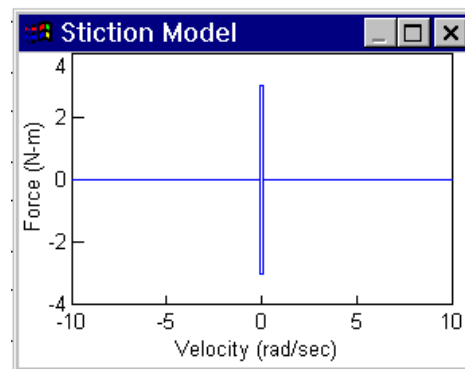
Rotor Leakage Inductance: Requires the specified rotor leakage inductance in henries.

Rotor Moment of Inertia: Requires the moment of inertia of the rotor with respect to the axis of rotation in $\text{kg}\cdot\text{m}^2$.

Rotor Shaft Coulomb Friction Magnitude: Allows specification of constant directional dissipative force (Coulomb model) in units of N-m.



Rotor Shaft Stiction Factor: Allows specification of a stiction force value or break-away torque. This parameter is normally not specified by the motor manufacturer, but can be obtained experimentally. Units are N-m.

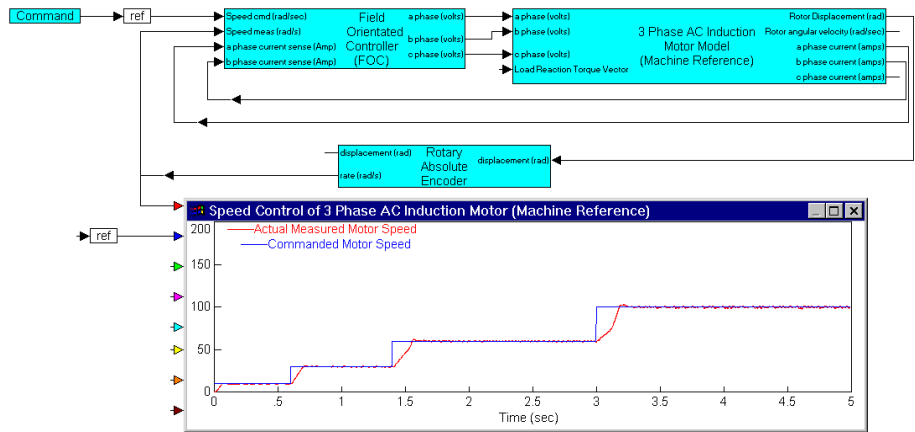


Rotor Shaft Viscous Damping Factor: Requires the factor that linearly relates viscous damping force to angular velocity. This parameter is normally not specified by the motor manufacturer, but can be determined experimentally. Units are $\text{kg}\cdot\text{m}^2/\text{s}$.

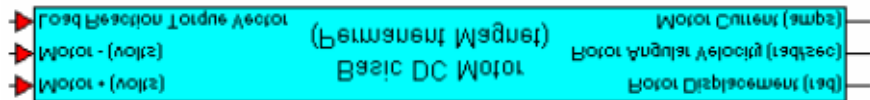
Example

Field-oriented speed control of a 3-phase AC induction motor (...\MOTION\EXAMPLES\ACIM_MACH_REF_EXMPL.VSM)

The following simulation example illustrates the application of the Field Orientated Controller (FOC) block in controlling speed of a 3-phase AC induction motor using a rotary encoder for feedback. Current is sensed using the motor model output currents. The command profile shows stable and accurate control of speed over a wide range of speeds.



Basic (Permanent Magnet) DC Motor



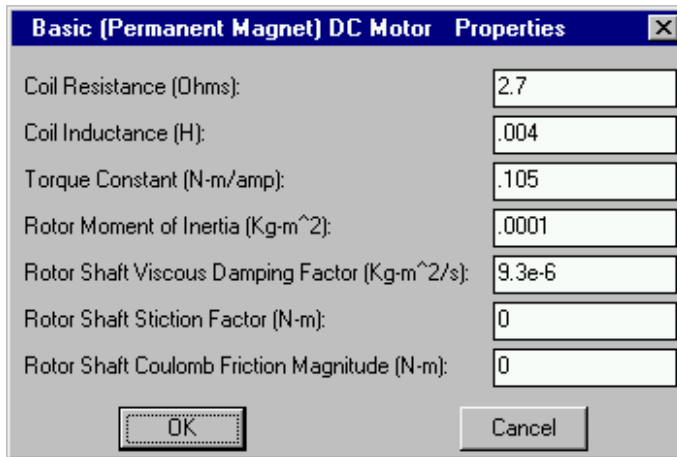
Example Model: ... \MOTION\EXAMPLES\BASIC_DC_MOTOR_EXMPL.VSM

Block Category: Motors

The Basic (Permanent Magnet) DC Motor block is derived from the linear system model of a simple DC permanent magnet machine. It is assumed that the back EMF is equal to the torque constant, and that the torque current relation is indeed constant over the rotation of the motor. Input requires a DC voltage and a load reaction torque vector if external mechanical load is used.

Mechanical dynamics include typical parameters, such as rotor shaft inertia and viscous friction. In addition, nonlinear dissipative factors, including Coulomb friction and stiction models are provided. The Basic (Permanent Magnet) DC Motor block can operate stand-alone to produce output displacement or velocity of the motor alone, or when a Rotational Load block is connected, combined dynamic response. When connected to the Rotational Load block, dynamic parameters are reflected back and combined with the motor dynamics by the linkage ratio. The linkage ratio is specified in the Rotational Load block. This creates proper dynamic motion of the combined motor-load connection. To connect Basic (Permanent Magnet) DC Motor block to the Rotational Load block, the rotor displacement and load reaction vector connections from each block must be wired together.

Rotor shaft position, velocity and motor current are provided for sensor connections in monitoring and feedback applications.



Coil Resistance: Indicates the line-to-line coil resistance in ohms.

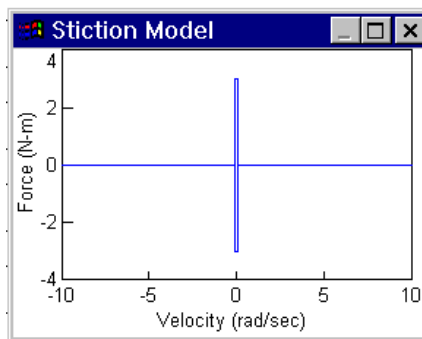
Coil Inductance: Indicates the line-to-line coil inductance in henries.

Torque Constant: Indicates the specified motor torque constant in N-m/Amp.

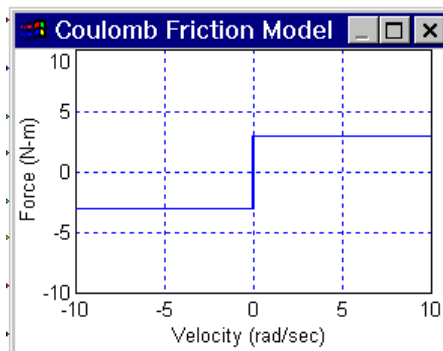
Rotor Moment of Inertia: Indicates the moment of inertia of the rotor with respect to the axis of rotation in $\text{kg}\cdot\text{m}^2$.

Rotor Shaft Viscous Damping Factor: Indicates the factor that linearly relates viscous damping force to angular velocity. This option is normally not specified by the motor manufacturer, but can be determined experimentally. Units are in $\text{kg}\cdot\text{m}^2/\text{s}$.

Rotor Shaft Stiction Factor: Allows specification of a stiction force value or break-away torque. This option is normally not specified by the motor manufacturer, but can be obtained experimentally. Units are N-m.



Rotor Shaft Coulomb Friction Magnitude: Allows specification of constant directional dissipative force (Coulomb model) in units of N-m.

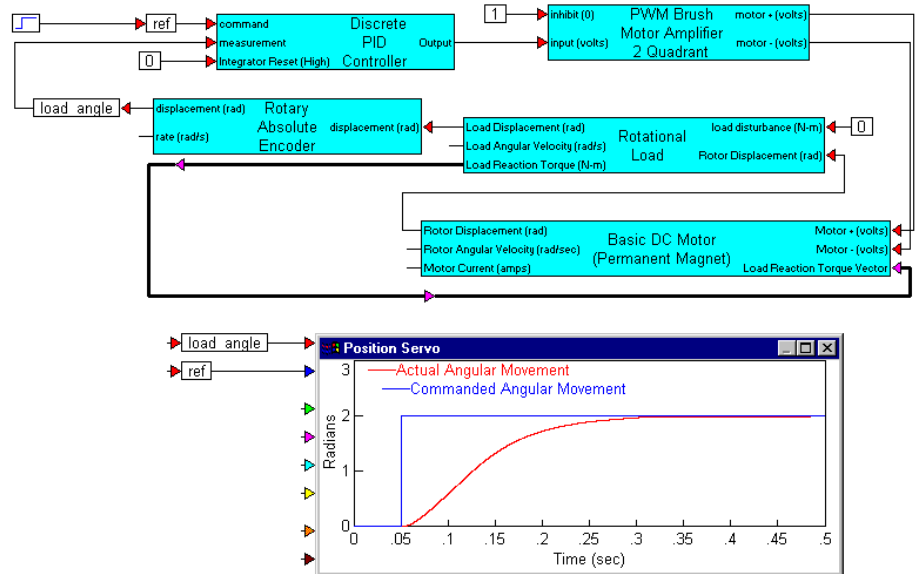


Example

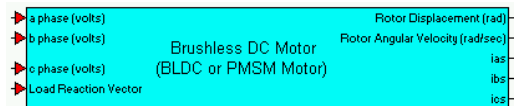
Angular positioning servo

(...\MOTION\EXAMPLES\BASIC_DC_MOTOR_EXMPL.VSM)

The following simulation example demonstrates a voltage-controlled servo loop for positioning a geared load (0.5 ratio) to multiply torque. The load is spring loaded by setting a torsional spring constant in the Rotational Load block. A Rotary Encoder block is coupled to the load and the measurement is fed back to a PID compensator for voltage control of the motor. A power amplifier of unity voltage gain with sufficient power gain to drive the motor is assumed. The motor is commanded to move 2 rad in the positive direction.



Brushless DC (BLDC/PMSM) Motor



Example Model: ... \MOTION\BRUSHLESS_PWM_SERVO_AMP_EXMPL.VSM

Block Category: Motors

The Brushless DC (BLDC/PMSM) Motor block simulates the motion of a 3-phase permanent magnet synchronous motor in machine variables. This block is realized from the system of nonlinear differential equations describing the electromagnetics for a permanent magnetic field and 3-phase Y connected load together with rotor mechanical dynamics.

Mechanical dynamics include typical parameters, such as rotor shaft inertia and viscous friction. In addition, nonlinear dissipative factors, including Coulomb friction and stiction models are provided. The Brushless DC (BLDC/PMSM) Motor block can operate stand-alone to produce output displacement or velocity of the motor alone, or when a Rotary Load block is connected, combined dynamic response. When connected to the Rotary Load block, dynamic parameters are reflected back and combined with the motor dynamics by the linkage ratio. The linkage ratio is specified in the Rotary Load block. This creates proper dynamic motion of the combined motor-load connection. To connect the Brushless DC

(BLDC/PMSM) Motor block to the Rotary Load block, the rotor displacement and load reaction vector connections from each block must be wired together.

Rotor shaft position, velocity and stator 3-phase currents are provided for sensor connections in monitoring and feedback applications.

Parameter	Value
Number of Poles:	2
Stator Inductance (per phase) (H):	.001
Stator Resistance (per phase) (ohms):	.5
Stator Magnetizing Inductance (H):	.0009
Rotor Moment of Inertia (Kg-m ²):	1e-5
Rotor Shaft Coulomb Friction Magnitude (N-m):	0
Rotor Shaft Stiction Factor (N-m):	0
Rotor Shaft Viscous Damping Factor (Kg-m ² /s):	.001
Torque Constant (N-m/A):	.07

Number of Poles: Indicates the number of rotor poles.

Stator Inductance (per phase): Indicates the per phase inductance of the stator winding in henries.

Stator Resistance (per phase): Indicates the per phase winding resistance of the stator in ohms.

Stator Magnetizing Inductance: Indicates the difference between the stator per phase inductance and the stator leakage inductance in henries.

Rotor Moment of Inertia: Indicates the rotor moment of inertia along the axis of rotation in kg-m².

Rotor Shaft Coulomb Friction magnitude: Indicates the constant dissipative force imposed by the rotor bearings/bushings in N.

Rotor Shaft Stiction factor: Indicates the static friction (break-away) force imposed by the rotor bearings/bushings in N.

Rotor Shaft Viscous Damping Factor: Indicates the viscous damping factor imposed by the rotor bearings/bushings in kg-m²/sec.

Torque Constant: Indicates the average torque constant of the motor in Newton meters/amp.

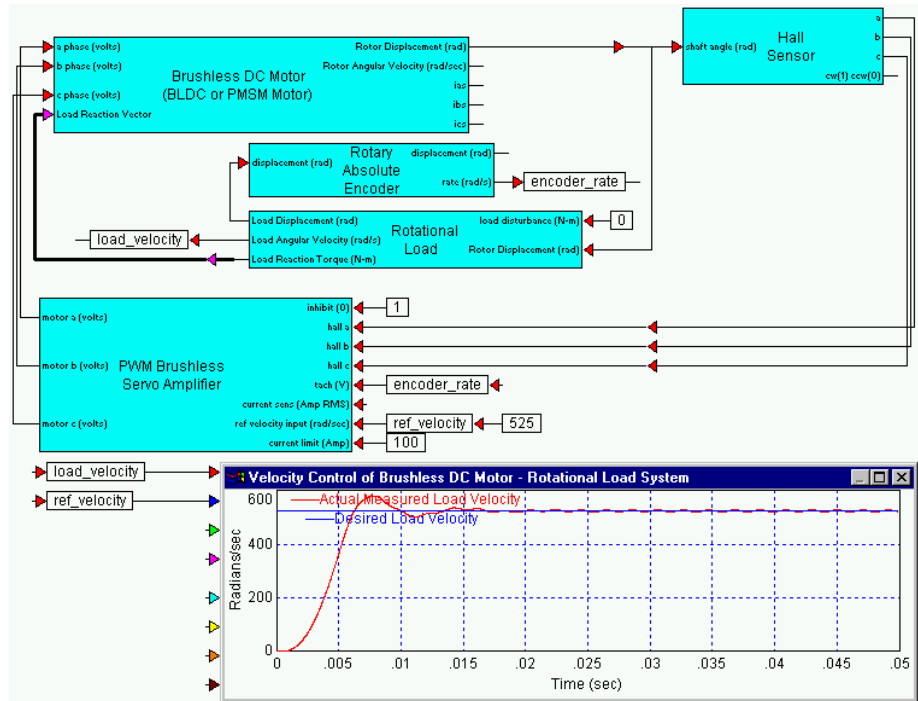
Load Reaction Vector (input): Indicates the vector input used to feedback dynamic parameters and linkage ratio imposed on motor by load model.

Example

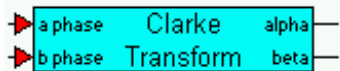
Speed control of a brushless DC motor

(..\MOTION\BRUSHLESS_PWM_SERVO_AMP_EXMPL.VSM)

The following example illustrates speed control of a DC brushless motor using a PWM Brushless Servo Amplifier block in feedback with a rotary tachometer generator. Note that Hall sensors used to commutate the motor are connected through the PWM amplifier.



Clarke Transform



Example Model: ...MOTION\EXAMPLES\CLARKE_TRANSFORM_EXMPL.VSM

Block Category: Transforms

The Clarke Transform block performs a Clarke forward transform (2 X 2).

The Clarke Transform block provides a magnitude invariant transformation of 3-phase signals/systems, components [a b] to orthogonal reference frame components [alpha beta]. This Clarke transformation is specific to the case where the primary axis of the 3-phase coordinate system is aligned with the primary axis of the orthogonal coordinate system.

The transform is given by:

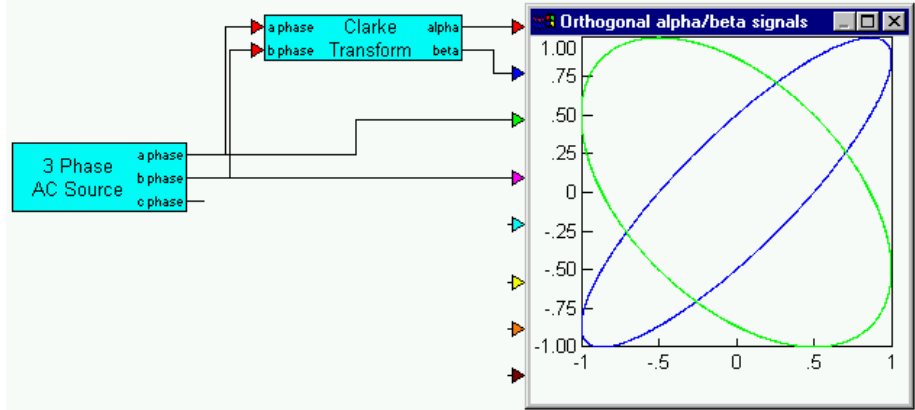
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

Example

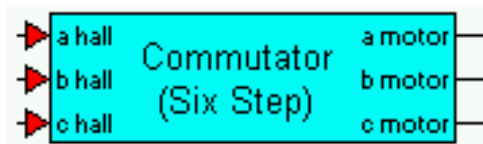
Transformation of a and b phase signals of 3-phase source to orthogonal alpha/beta signals

(...MOTION\EXAMPLES\CLARKE_TRANSFORM_EXMPL.VSM)

The following simulation shows the transformation and compares the phase differences as a Lissajous figure.



Commutator (Six Step)



Example Model: ... \MOTION\EXAMPLES\COMMUTATOR_SIX_STEP.VSM

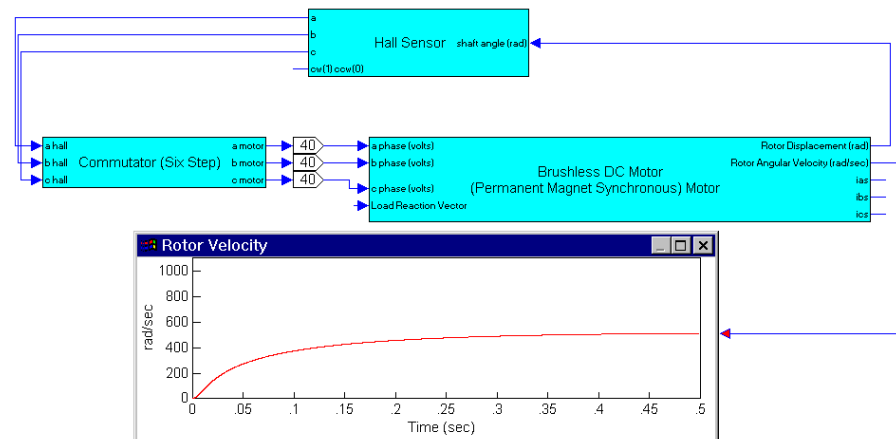
Block Category: Controllers

The Commutator (Six Step) block provides six step motor commutation logic signals from Hall sensor input.

Example

Simple open loop control of a permanent magnet synchronous motor (... \MOTION\EXAMPLES\COMMUTATOR_SIX_STEP.VSM)

In this example, Hall sensors measure shaft displacement. Hall sensor output is fed directly to the Commutator (Six Step) block. The output of the Commutator (Six Step) block is scaled to the drive voltage and fed to the motor phases. The voltage level can control speed; however, for large transients, current feedback together with PWM improves control performance.



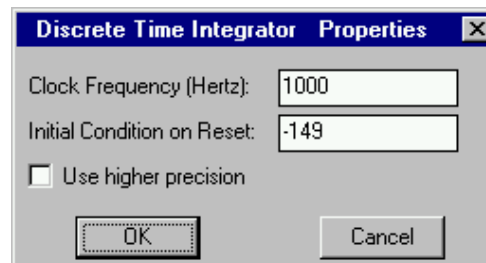
Discrete Integrator



Example Model: ... \MOTION\EXAMPLES\DISCRETE_INTEGRATOR_EXMPL.VSM

Block Category: Tools

The Discrete Integrator block provides discrete time integration of signals.



Clock Frequency: Establishes the integrator step time = $1/(\text{clock frequency})$

Initial Condition on Reset: Defines the initial value of the integrator output upon reset.

IN: Indicates integrator input.

RST: Indicates Boolean input. When RST is high, integrator output is latched to initial condition; otherwise, integration runs.

LOLIM: Defines the lower limit of the integrator output. Limiting disables further integration in this direction, thus preventing windup.

HILIM: Defines the higher limit of the integrator output. Limiting disables further integration in this direction, thus preventing windup.

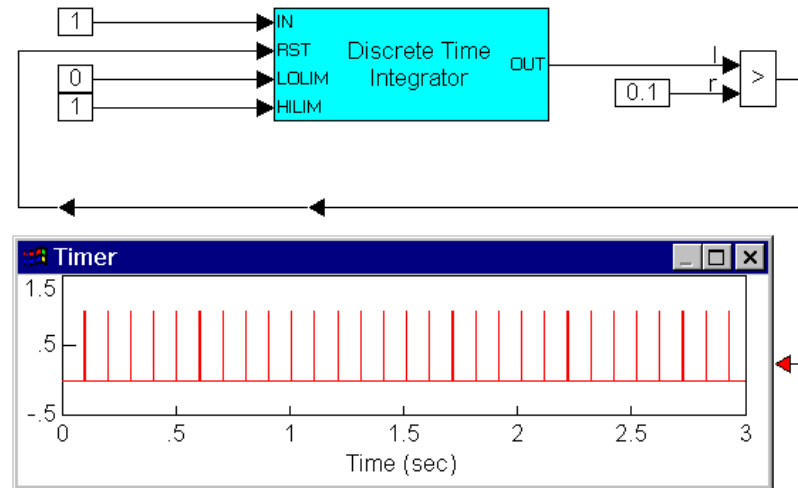
Use Higher Precision: When activated, the Discrete Integrator block uses trapezoidal integration; when de-activated, it uses backward rectangular integration.

Example

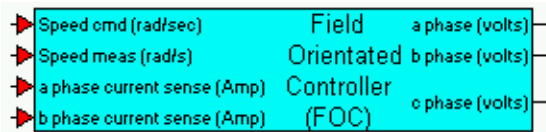
Pulse generator/timer

(... \MOTION\EXAMPLES\DISCRETE_INTEGRATOR_EXMPL.VSM)

The Discrete Integrator block can be used to generate timing pulses. Set IN of the integrator to 1; and set the initial condition to 0. This causes the output of the Discrete Integrator block to ramp with the output value equal to simulation time. Compare this output to the desired interval between timing pulses, and feedback the comparator output to the Discrete Integrator block's RST. After completing a ramp cycle, the Discrete Integrator block is reset to 0. HILIM and LOLIM are arbitrary; however, they must be within the limits of the timing period.



Field Orientated Controller (FOC)



Example Model: ... \MOTION\EXAMPLES\FOC_CONTROLLER_EXMPL.VSM

Block Category: Controllers

The Field Orientated Controller (FOC) block attempts to maximize torque output by controlling the stator flux position in the direct (d) axis and current in the orthogonal quadrature (q) axis. The Field Orientated Controller (FOC) block includes a number of interconnected subsystems to accomplish this goal. Clarke's and Park's transformations are used to convert the 3-phase current measurement into DQ frame currents, which are applied to PI direct axis current control and PI quadrature axis current control, respectively. Each of these controllers is commanded by an outer loop PI speed controller that measures rotor shaft velocity, usually from a tachometer or encoder. DQ frame controls are converted back to 3-phase using the inverse Park's transform and space vector pulse width modulation (SVPWM). SVPWM is typically used to lower harmonic content over typical PWM methods and increase DC link voltage. A current model estimates the rotor flux position from measured direct and quadrature currents, and known motor parameters. The flux (stator field) position estimate is used in the Park's forward and inverse transformations.

Field Orientated Controller (FOC) Properties	
Number of Poles:	2
Sampling Rate (Hertz):	10000
Inverter DC Bus Voltage (volts):	310
Stator Inductance (per phase) (H):	.165
Stator Leakage Inductance (H):	.016
Rotor Leakage Inductance (H):	.013
Rotor Resistance (Ohms):	5.365
Direct Axis Current Control Integral Gain:	.000625
Direct Axis Current Control Proportional Gain:	.8
Direct Axis Current Control Upper Sat limit (Amp):	220
Direct Axis Current Control Lower Sat Limit (Amp):	-220
Quadrature Axis Current Control Integral Gain:	.000625
Quadrature Axis Current Control Proportional Gain:	.8
Quadrature Axis Current Control Upper Sat limit (Amp):	220
Quadrature Axis Current Control Lower Sat Limit (Amp):	-220
Speed Control Integral Gain (Amp/rad):	5e-5
Speed Control Proportional Gain (Amp-s/rad):	45.1
Speed Control Upper Sat limit (Amp):	50
Speed Control Lower Sat Limit (Amp):	-50

Current Model (Field Angle Estimator) Parameters

Number of Motor Poles: Requires the number of motor pole pairs.

Sampling Rate (Hertz): Sets the clock rate for all the operations in the FOC. This rate must be set sufficiently large to provide modulation depth for the SVPWM. A value of 10,000 Hz or greater are typically set. Note that the simulation step size must at least meet this frequency.

Inverter DC Bus Voltage: Sets the voltage level of the PWM pulses for each phase voltage. Units are in volts.

Stator Self Inductance (per phase): Requires the 3-phase AC induction motor stator per phase inductance in henries.

Stator Leakage Inductance: Requires the 3-phase AC induction motor specified stator leakage inductance in henries.

Rotor Leakage Inductance: Requires the specified 3-phase AC induction motor rotor leakage inductance in henries.

Rotor Resistance: Requires the 3-phase AC induction motor rotor winding resistance in ohms.

PI Direct Axis Current Controller

PI direct axis current control receives a direct axis current command from a field model and direct axis current measurement. The error between these quantities is then fed to an anti-windup PI compensator.

Direct Axis Current Control Integral Gain: Typically sets the speed (bandwidth) and tracking error of the control loop. Units are 1/sec.

Direct Axis Current Control Proportional Gain: Typically sets the damping of the control loop.

Direct Axis Current Control Upper Sat Limit: Sets the upper output limit of the PI direct axis current controller. Hard limiting is designed to prevent compensator integral wind-up. Units are in amps.

Direct Axis Current Control Lower Sat Limit: Sets the lower output limit of the PI direct axis current controller. Hard limiting is designed to prevent compensator integral wind-up. Units are in amps.

PI Quadrature Axis Current Controller

PI quadrature axis current control receives a quadrature axis current command from the speed controller and quadrature axis current measurement. The error between these quantities is then fed to an anti-windup PI compensator.

Quadrature Axis Current Control Integral Gain: Typically sets the speed (bandwidth) and tracking error of the control loop. Units are 1/sec.

Quadrature Axis Current Control Proportional Gain: Typically sets the damping of the control loop.

Quadrature Axis Current Control Upper Sat Limit: Sets the upper output limit of the PI quadrature axis current controller. Hard limiting is designed to prevent compensator integral wind-up. Units are in amps.

Quadrature Axis Current Control Lower Sat Limit: Sets the lower output limit of the PI quadrature axis current controller. Hard limiting is designed to prevent compensator integral wind-up. Units are in amps.

PI Speed Controller

PI speed control receives a target speed command, and actual speed measurement. The error between these quantities is then fed to an anti-windup PI compensator.

Speed Control Integral Gain: Typically sets the speed (bandwidth) and tracking error of the speed control loop. Units are amp/rad.

Speed Control Proportional Gain: Typically sets the damping of the speed.

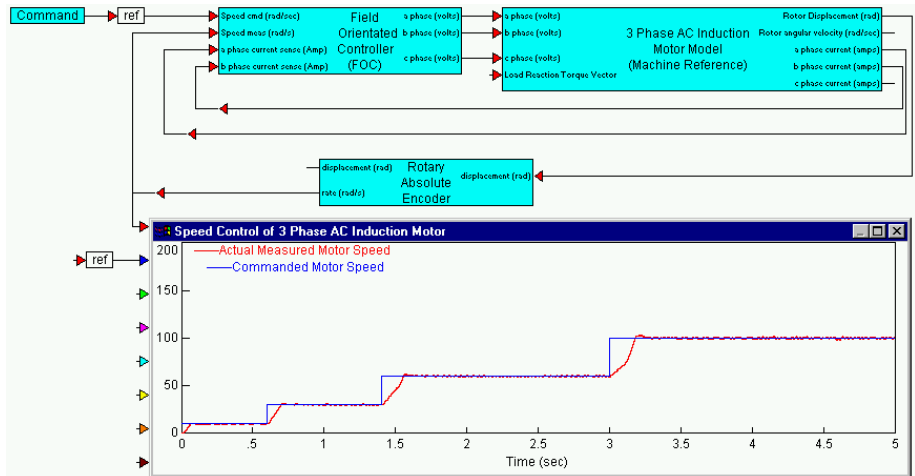
Speed Control Upper Sat Limit: Sets the upper output limit of the PI speed controller. Hard limiting is designed to prevent compensator integral wind-up. Units are in amps.

Speed Control Lower Sat Limit: Sets the lower output limit of the PI speed controller. Hard limiting is designed to prevent compensator integral wind-up. Units are in amps.

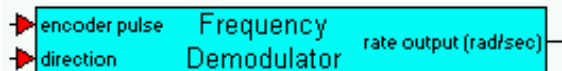
Example

Field-oriented speed control of a 3-phase AC induction motor (..\MOTION\EXAMPLES\FOC_CONTROLLER_EXMPL.VSM)

The following simulation example illustrates the application of the Field Orientated Controller (FOC) block in controlling speed of a 3-phase AC induction motor using a rotary encoder for feedback. Current is sensed using the motor model output currents. The command profile shows stable and accurate control of speed over a wide range of speeds.



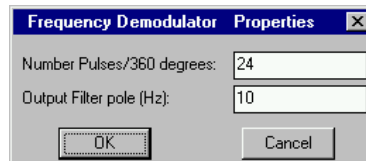
Frequency Demodulator



Example Model: ...MOTION\EXAMPLES\FREQUENCY_DEMODULATOR_EXMPL.VSM

Block Category: Sensors

The Frequency Demodulator block demodulates rate signal information from a frequency-modulated signal. Some types of sensors (interrupter type) produce pulses that vary in frequency proportional to the cyclical rate of the sensor. One way to extract the cyclical rate from the carrier frequency is through demodulation.



Number of Pulses/ 360 degrees: Indicates the number of pulses that occur over one complete rotation.

Output Filter pole: Defines the roll-off frequency in hertz of the demodulator filter. The choice of the filter pole is usually a trade-off between noise rejection and accuracy of the demodulated rate.

Encoder Pulse: Indicates the input of the demodulator. Acceptable values are 0 and 1.

Direction: Indicates input of the direction of shaft rotation to discriminate between positive and negative velocity. Input is 1 for positive velocity, -1 for negative velocity.

Rate Output: Indicates the demodulated output.

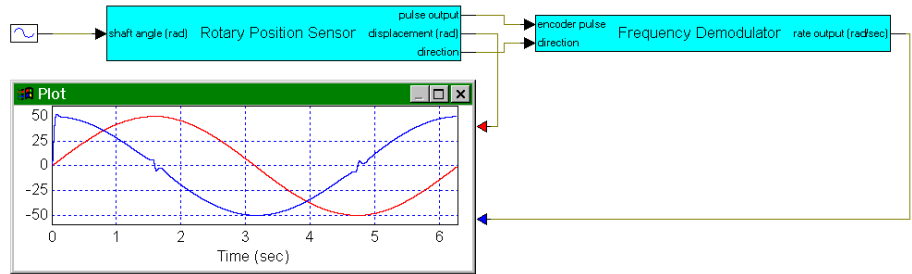
Example

Rate estimation from vane operated Hall sensor
 (...MOTION\EXAMPLES\FREQUENCY_DEMODULATOR_EXMPL.VSM)

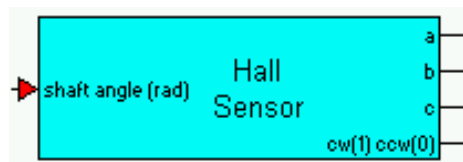
The following simulation diagram demonstrates how to obtain rate measurement from a Hall vane position sensor. The vane sensor produces pulses that are frequency

modulated, and so a frequency demodulator is designed to extract the rate signal. Note that direction has to be wired out from the Hall sensor block to provide phase sensing for the demodulator.

For this example, the interrupter in the Hall sensing device contained 256 vanes, producing 256 pulses per rotation. A 1 rad/sec shaft angular velocity is used. At 1 radian second, displacement and rate signals have the same magnitude.



Hall Sensor



Example Model: ...MOTION\EXAMPLES\HALL_SENSOR_EXMPL.VSM

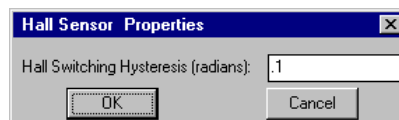
Block Category: Sensors

The Hall Sensor block is a Hall sensor triad with commutation and direction sensing logic.

The Hall Sensor block models 3 digital output (ON/OFF) Hall sensors arranged in a circular pattern 120° apart for sensing the rotational position and direction of the rotor shaft in a DC permanent magnet synchronous motor (brushless motor). The following table summarizes this relationship:

angle/phase	a	b	c
0-60deg	1	0	0
60-120	1	1	0
120-180	0	1	0
180-240	0	1	1
240-300	0	0	1
300-360	1	0	1

Direction is resolved by discriminating sequencing between phases.



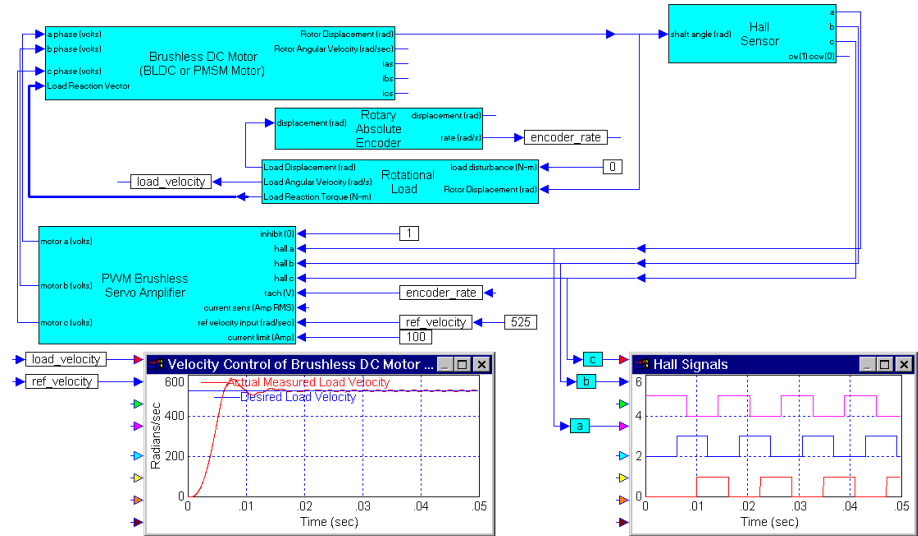
Hall Switching Hysteresis: Sets the Hall device switching hysteresis characteristics. Units are in radians.

Example

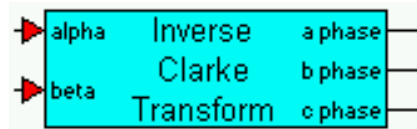
Commutation of a brushless DC motor

(...\MOTION\EXAMPLES\HALL_SENSOR_EXMPL.VSM)

The simulation on the following page shows a speed control loop for a DC brushless motor using the Hall sensor triad for electronic commutation.



Inverse Clarke Transform



Example Model: ...\MOTION\EXAMPLES\INVERSE_CLARKE_TRANSFORM_EXMPL.VSM

Block Category: Transforms

The Inverse Clarke Transform block provides a magnitude invariant transformation of orthogonal reference frame components [alpha beta] to 3-phase signals/systems [a b c]. This Clarke transformation is specific to the case where the primary axis of the 3-phase coordinate system is aligned with the primary axis of the orthogonal coordinate system.

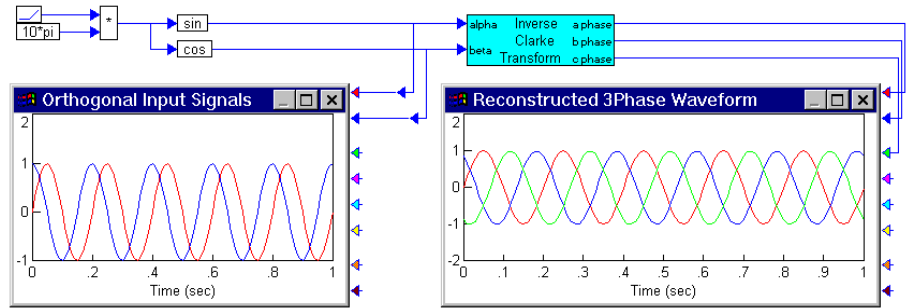
The transform is given by:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

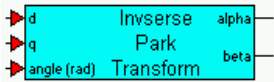
Example

Reconstructing 3-phase waveform from 2-phase orthogonal waveform

(...\MOTION\EXAMPLES\INVERSE_CLARKE_TRANSFORM_EXMPL.VSM)



Inverse Park Transform



Example Model: ...MOTION\EXAMPLES\INVERSE_PARK_3P_MODULATION_EXMPL.VSM

Block Category: Transforms

The Inverse Park Transform block transforms direct and quadrature signals [d q] to orthogonal components [alpha beta] given a particular rotation angle, θ .

Park's Inverse Transform is defined as:

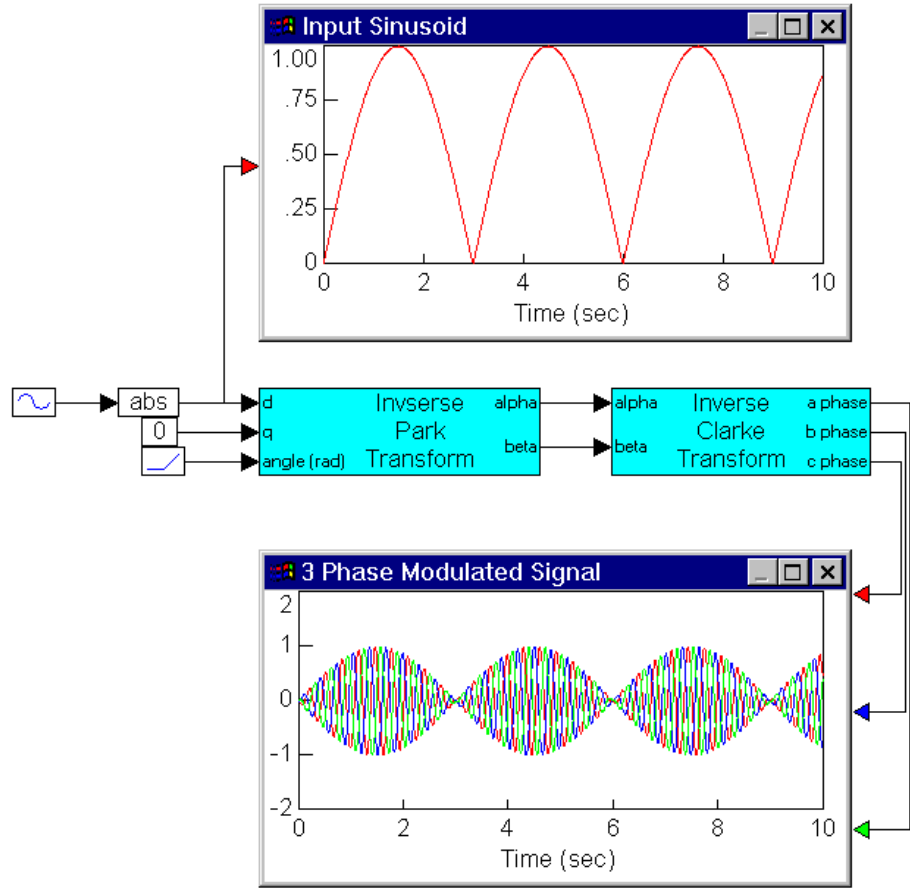
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} d \\ q \end{bmatrix}$$

Example

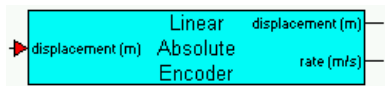
Three-phase modulation of a signal

(...MOTION\EXAMPLES\INVERSE_PARK_3P_MODULATION_EXMPL.VSM)

The following example illustrates the use of Park's and Clarke's inverse transforms to create a 3-phase modulated signal at a specified rotation frequency (20 rad/sec). The input signal defines the envelope of the modulated signal and the angle rate defines the 3-phase frequency.



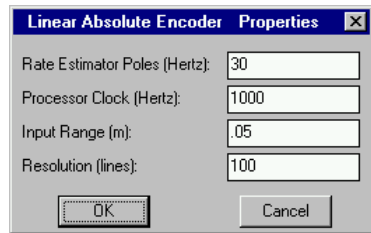
Linear Encoder



Example Model: ...MOTION\EXAMPLES\LINEAR_ENCODER_EXMPL.VSM

Block Category: Sensors

The Linear Encoder block models the quantized displacement measurements obtained from a generic encoding device such as a linear optical encoder or any device that creates incremental counts from linear motion. The block provides the quantized displacement and simulation of processor rate estimation from the quantized signal.



Rate Estimator Poles: Defines the bandwidth of rate estimation. The selection typically requires a tradeoff between rate estimation accuracy and noise. Units are in hertz.

Processor Clock: Defines the processor clock rate for rate estimation.

Input Range: Specifies the end-to-end stroke of the encoder in meters.

Resolution: Indicates the total number of lines over the stroke of the encoder.

Displacement (input): Indicates where the physical displacement is connected. The units are assumed to be meters.

Displacement (output): Indicates where the quantized displacement measurement is read. The units are in meters.

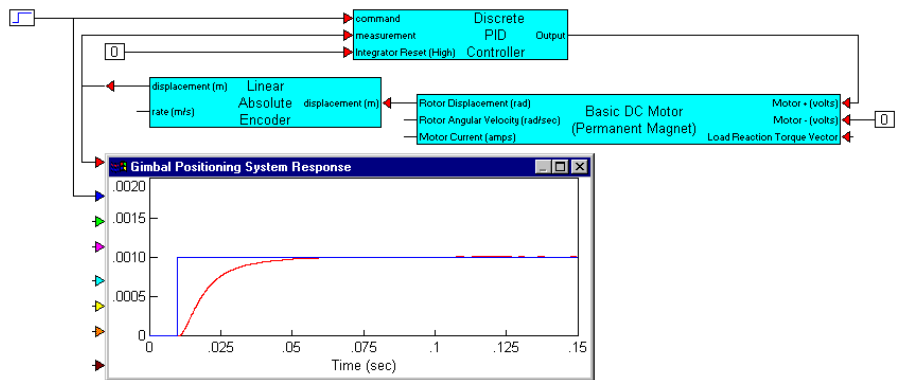
Rate: Indicates where the estimated rate is read. The units are in m/sec.

Example

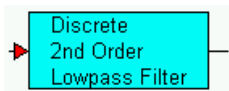
Gimbal position controller

(..\MOTION\EXAMPLES\LINEAR_ENCODER_EXMPL.VSM)

A voice coil actuator is driven by a gimbaled mount for fine angular positioning of a steering mirror. Since the angular stroke is small, a small linear optical encoder is used at the gimbal periphery for estimated angular position and rate feedback measurements. The requirement calls for a step input response rise time of under 10 ms with minimal overshoot and settling. Steady state accuracy must be within 2 arc sec. To test initial feasibility for the proposed control, the following simulation diagram is used with the Linear Encoder block to assess a rough design. The number of lines is varied to determine the effects on the dynamic response and steady state accuracy of the feedback loop.



Low Pass Filter

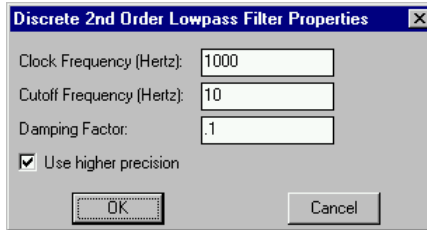


Example Model: ...\MOTION\EXAMPLES\LOW_PASS_FILTER_EXMPL.VSM

Block Category: Filters

The Low Pass Filter block is a discrete time 2nd Order, low-pass filter

The Low Pass Filter block provides passage of signal frequency components below the cutoff frequency and attenuation of components above the cutoff frequency.



Clock Frequency: Defines the processor clock frequency for filtering.

Cutoff Frequency: -3 dB attenuation frequency of filter.

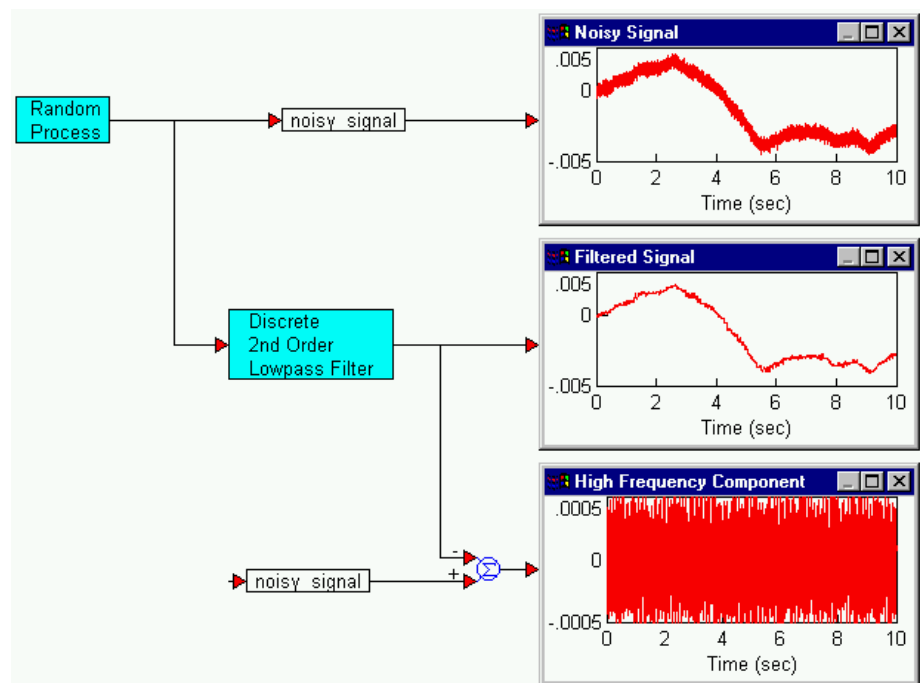
Damping Factor: Controls filter peaking.

Use Higher Precision: When this option is activated, the Low Pass Filter block, uses trapezoidal integration; when this option is de-activated, it, uses backward rectangular integration.

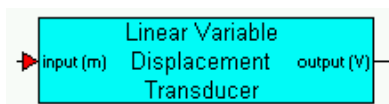
Example

Removing high frequency noise from a signal

(...\MOTION\EXAMPLES\LOW_PASS_FILTER_EXMPL.VSM)



LVDT



Example Model:

...\MOTION\EXAMPLES\LINEAR_VARIABLE_DISPLACEMENT_TRANSDUCER_EXMPL.VSM

Block Category: Sensors

The LVDT block is a linear variable displacement transducer. This block models a LVDT sensor with the aspects of carrier demodulator noise, filtering, linearity, null offset, and range limitations.

Property	Value
Linearity (% input range):	.01
Sensitivity (V/m):	1000
Range (m):	.005
Null Offset (V):	.005
Filter Cutoff Frequency (Hz):	600
Carrier Frequency (Hz):	1500
Output Ripple Amplitude (V):	.02

Linearity: Simulates symmetric curvilinear input-output characteristic of a LVDT as a percent of the transducer input range.

Sensitivity: Defines the linear gain of the transducer in volts output per meter displacement.

Range: Defines the measurable stroke of the transducer in meters. Inputs that exceed the input range cause output saturation.

Null Offset: Defines the offset voltage at 0 displacement in volts.

Filter Cutoff Frequency: Indicates the bandwidth of a two-pole analog filter used to remove the carrier. Units are in hertz.

Carrier Frequency: Defines the LVDT modulator carrier frequency in hertz.

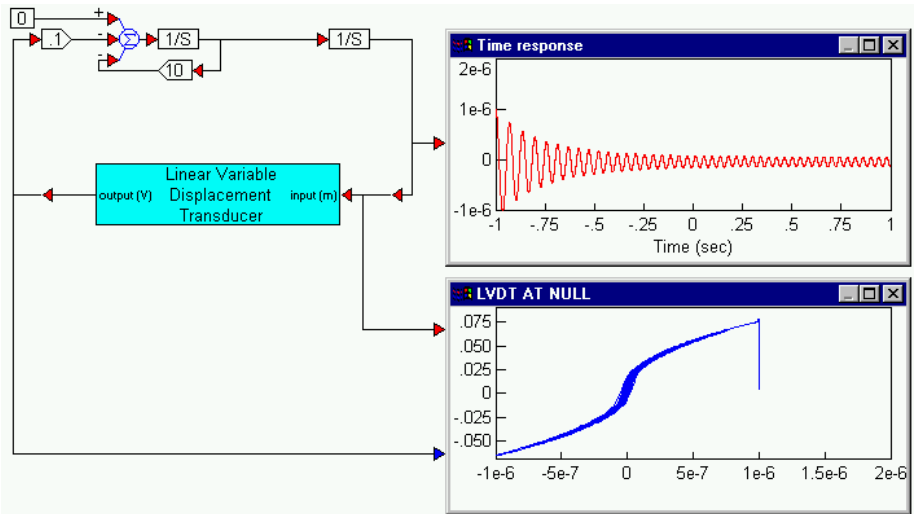
Output Ripple Amplitude: Defines the influence of the carrier on the output signal as the amplitude of a sinusoidal voltage signal in volts.

Example

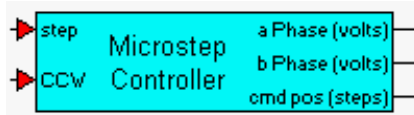
Limit cycle induced by nonlinearity in feedback

(...\MOTION\EXAMPLES\LINEAR_VARIABLE_DISPLACEMENT_TRANSDUCER_EXMPL.VSM)

The following simplified simulation shows a dynamic system in which an LVDT block is used to feedback measured position. The LVDT block is adjusted to include a fairly severe nonlinearity. The high gain resulting at null from this nonlinearity creates a sustained oscillation, or limit cycle, at steady state.



Microstep Controller

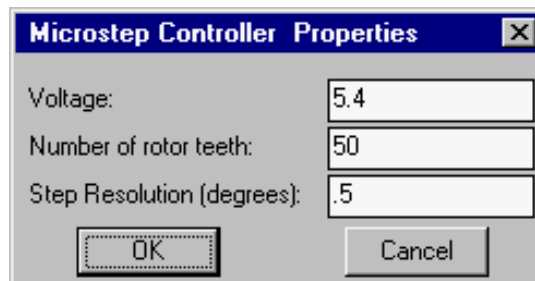


Example Model: ...MOTION\EXAMPLES\MICRO_STEP_CONTROLLER_EXMPL.VSM

Block Category: Controllers

The Microstep Controller block simulates the operation of microstep control. Microstep control is typically implemented in a microprocessor, commercial IC, or ASIC for precision incremental motion of 2-phase permanent magnet stepper motors. The controller can be used for open-loop positioning or speed (slew) control applications, or used as a component in a step motor closed-loop control system. Logic inputs provide step activation and directional control. A counter output is provided to track the current commanded (accumulated) position in steps.

Note that the slew rate (speed) of the motor in steps/sec depends on the input pulse rate. Actual shaft (angular rate) depends on this pulse rate. The main feature that differentiates the Microstep Controller block from the full/half step controller is that the Microstep Controller can provide much smaller fractions of a full step. (One full step of a step motor in degrees is $360/(\text{number of rotor teeth})$). The Microstep Controller provides phase output voltages that are proportional to the sine and cosine of the commanded step angle, and based on the known number of rotor teeth. For an actual microstep controller, the ultimate positioning resolution depends on the LSB voltage resolution of the digital to analog converters or the power amplifier/system's noise floor. This model assumes no limitations introduced by these components.



Supply Voltage: Defines the RMS voltage level of the phase voltages through one cycle of motor rotation. Units are in volts.

Number of Rotor Teeth: Indicates the physical number of teeth in rotor.

Step Resolution: Defines the control resolution. Units are in degrees.

Step: Specifies logic transition input (0 to 1). The rising edge of this input causes the motor to move by one microstep (one microstep is equivalent to the specified step resolution).

CCW: Specifies logic level input (0 or 1). When CCW is true (1), the motor steps in a counter-clockwise (CCW) direction. When CCW is false (0), the motor steps in a clockwise (CW) direction.

a Phase: Connects to “a” phase input of stepper motor block. Units are in volts.

b Phase: Connects to “b” phase input of stepper motor block. Units are in volts.

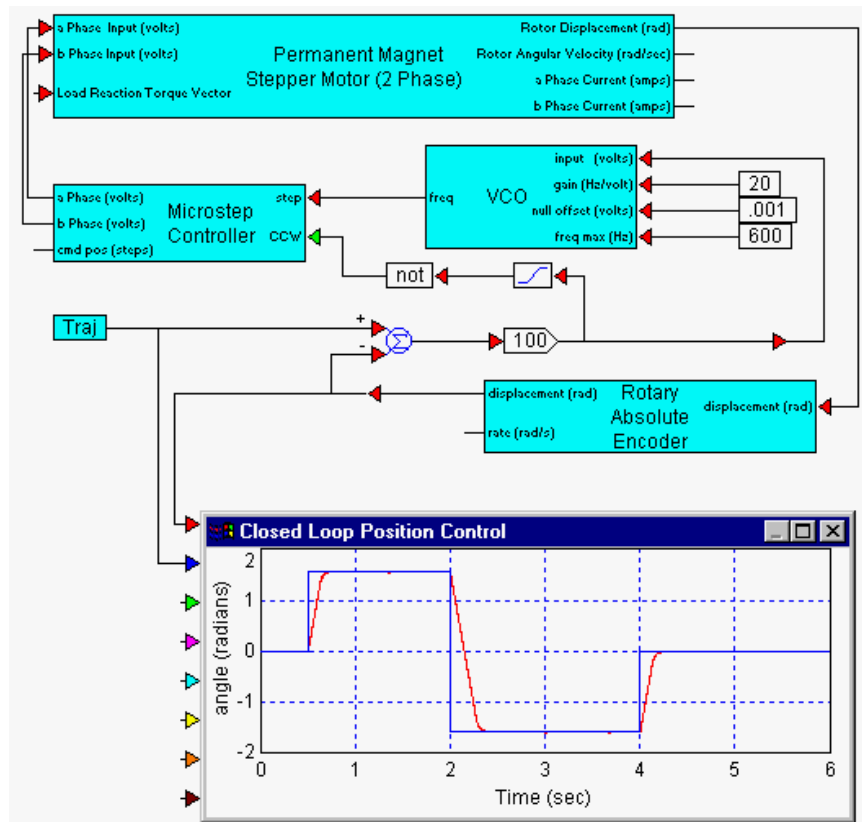
cmd pos: Provides the current commanded step position in steps (actually microsteps).

Example

Closed loop positioning control of a stepper motor

(...MOTIONEXAMPLES\MICRO_STEP_CONTROLLER_EXMPL.VSM)

This example illustrates a typical application where a stepper motor is used in feedback with an optical encoder.



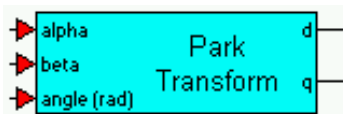
A Rotary Encoder block is used to directly measure shaft position of the stepper motor. Angular displacement is compared against the trajectory. The error is amplified with a high gain to saturate the VCO (Voltage Controlled Oscillator) input and drive the motor to the maximum specified slew rate (600 steps/sec). Since only

proportional control is used, the VCO quickly desaturates as the motor closely approaches the target trajectory. The sign of the comparator provides direction control as a 0 or 1 in the Microstep Controller.

In this example, microstep resolution is set at 0.1°. Closer examination of the rotor position shows a limit cycle once the motor reaches its trajectory. This hunting could be eliminated using deadband or other logic mechanisms that could easily be implemented using other standard VisSim blocks.

The disadvantage of this type of controller is the expense and complexity of using an encoder for feedback and having to use additional control logic. The advantage is that if the motor inadvertently loses step, it will recover and eventually reach the targeted command. Stepper motors may sometimes lose step if external disturbance torques are encountered that exceed design torque, or if the motor is slewed too quickly.

Park Transform



Example Model: ... \MOTION\EXAMPLES\PARK_TRANSFORM_EXMPL.VSM

Block Category: Transforms

The Park Transform block transforms orthogonal signals and systems components [alpha beta] to direct and quadrature components [d q] given a particular rotation angle, θ .

Transformation is given by:

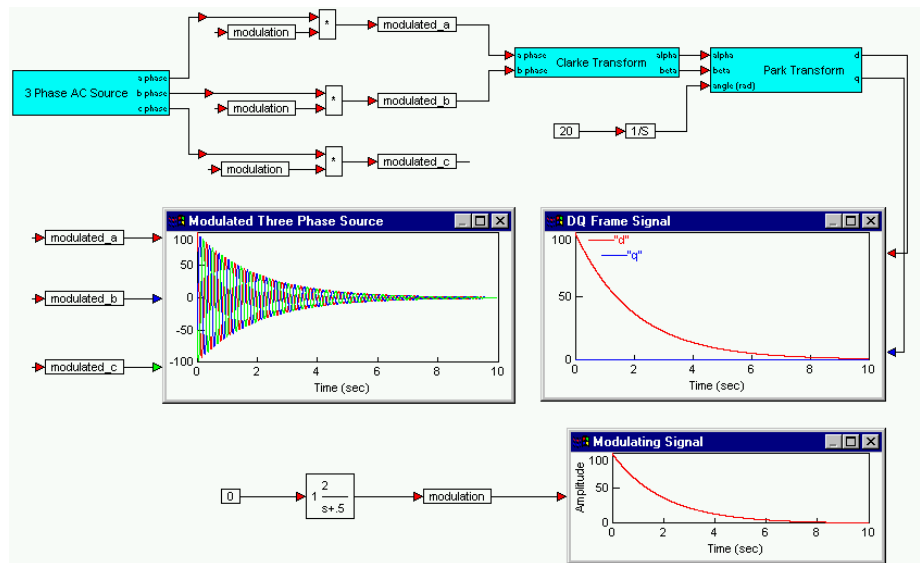
$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Example

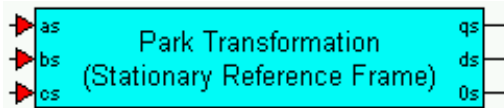
Transformation of 3-phase currents or voltages to rotating frame direct and quadrature components

(... \MOTION\EXAMPLES\PARK_TRANSFORM_EXMPL.VSM)

This example shows the transformation of a 3-phase source modulated by a converging exponential envelope, first transformed to orthogonal components alpha, beta by Clarke's transform, then to the rotating DQ frame by Park's transformation. Here the input angle to Park's transformation is the angle of rotation according to the 3-phase signal's frequency. The result is the direct component tracing the signal envelope, and the quadrature signal at 0.



Park Transform (Stationary Frame)



Example Model:

...\MOTION\EXAMPLES\PARK_TRANSFORM_STATIONARY_FRAME_EXMPL.VSM

Block Category: Transforms

The Park Transform (Stationary Frame) block performs a Park's forward transformation for stator voltages in the stationary reference frame (3 X 3).

The Park Transform (Stationary Frame) block transforms 3-phase signals/systems [a b c] to quadrature, direct and homopolar components [q d o] in the stationary reference frame (rotation angle = 0).

This transformation is given by:

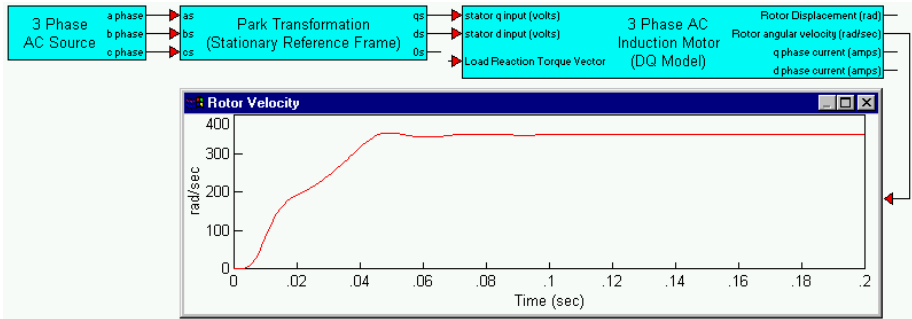
$$\begin{bmatrix} q \\ d \\ o \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Example

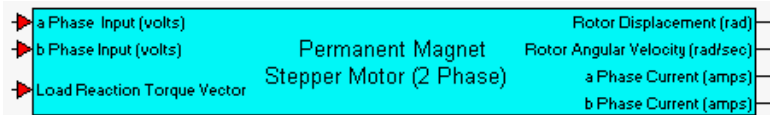
Open loop control of 3-phase AC induction motor model (DQ Frame) with 3-phase voltage source

(...\MOTION\EXAMPLES\PARK_TRANSFORM_STATIONARY_FRAME_EXMPL.VSM)

Park's transformation is used in this example to transform a 3-phase voltage source to DQ frame voltages for use on the DQ frame model.



Permanent Magnet Stepper Motor (2 Phase)



Example Model: ...MOTION\EXAMPLES\MICRO_STEP_CONTROLLER_EXMPL.VSM;
 ...MOTION\EXAMPLES\STEPPER_MOTOR_CONTROLLER_EXMPL.VSM

Block Category: Motors

The Permanent Magnet Stepper Motor (2 Phase) block uses the set of coupled nonlinear differential equations that describe motion of a 2-phase permanent magnet stepper motor with bipolar windings (4 wire input). Typical linear electrical and mechanical properties are provided, as well as nonlinear properties, including bearing stiction and Coulomb friction for a high-fidelity simulation of stepper motor dynamics. The number of rotor teeth determines the full step angle (resolution) of the motor. The step angle in degrees is equal to $360/(\text{number of rotor teeth})$.

Permanent Magnet Stepper Motor (2 Phase) Properties	
Supply Voltage (V):	30
Number of Rotor Teeth:	25
Holding Torque (N-m):	.2
Stator Resistance (per phase) (Ohms):	2.2
Stator Inductance (per phase) (H):	.003
Rotor Moment of Inertia (Kg-m ²):	2.5e-5
Rotor Shaft Coulomb Friction Magnitude (N-m):	0
Rotor Shaft Stiction Factor (N-m):	0
Rotor Shaft Viscous Damping Factor (Kg-m ² /s):	7e-5
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Number of Rotor Teeth: Indicates the physical number of teeth in the rotor.

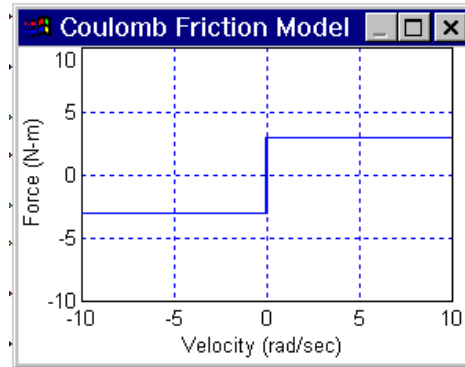
Holding Torque: Represents the motor holding torque. Units are in N-m/amp

Stator Resistance (per phase): Represents the resistance of the stator coil windings (per phase). Units are in ohms.

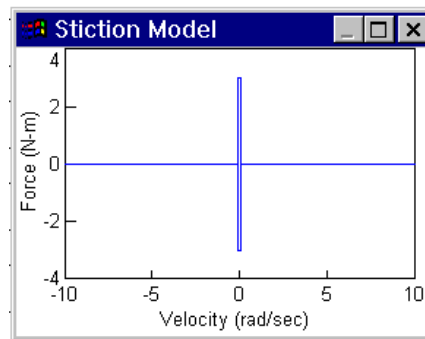
Stator Inductance (per phase): Represents the inductance of the stator coil windings (per phase). Units are in henries.

Rotor Moment of Inertia: Represents the rotor moment of inertia with respect to the axis of rotation. Units are in Kg-m².

Rotor Shaft Coulomb Friction Magnitude: Allows specification of constant directional dissipative force (Coulomb model) in units of N-m.



Rotor Shaft Stiction Factor: Allows specification of a stiction force value or break-away torque. This parameter is normally not specified by the motor manufacturer, but can be obtained experimentally. Units are N-m.

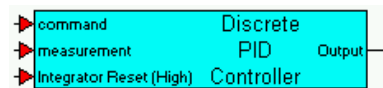


Rotor Shaft Viscous Damping Factor: Requires the factor that linearly relates viscous damping force to angular velocity. This parameter is normally not specified by the motor manufacturer, but can be determined experimentally. Units are kg-m²/s.

Example

See the examples included with the Stepper Motor Controller and Microstep Controller blocks for application of the Permanent Magnet Stepper Motor (2 Phase) block.

PID Controller (Digital)



Example Model: ...MOTION\EXAMPLES\DISCRETE_PID_CONTROLLER_EXMPL.VSM

Block Category: Controllers

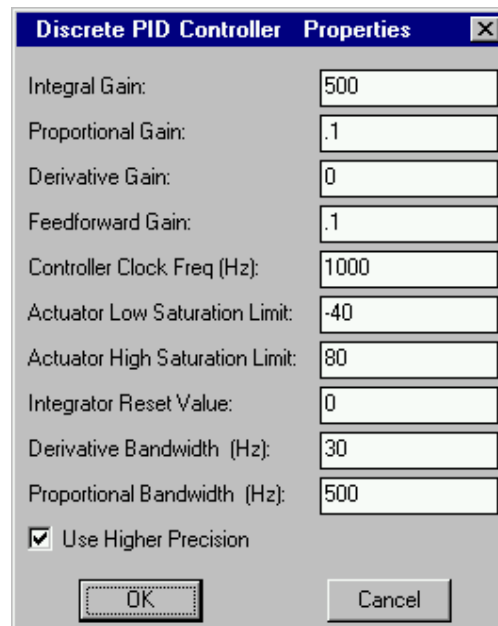
The PID Controller (Digital) block is discrete time PID, feed forward compensator.

The PID Controller (Digital) block models a PID compensator as though implemented on a processor using either backward-rectangular approximation or the higher precision trapezoidal method for integration and filtering. This block provides

anti-windup of the integrator component by separately calculating the PDF components, comparing this to the actuator saturation limits, and setting integration limits at each time step. PIDF, IF, PD, I, or P control can be implemented by choosing 0 gains for the unused components.

An input is provided to reset the integrator. This input, if held high, latches the integrator to the specified initial condition. Momentary reset can be accomplished by using an external one shot.

Band limiting filters are provided for the proportional and derivative components. Derivative bandwidth specification guarantees no more than 10° phase difference at that frequency as compared to true derivative. A rooftop filter is used to provide roll-off at high frequency. A simple second order low pass filter is used to limit proportional component bandwidth.



Integral Gain: Indicates the multiplying factor for the integral component of control.

Proportional Gain: Indicates the multiplying factor for the proportional component of control.

Derivative Gain: Indicates the multiplying factor for the derivative component of control.

Feed forward Gain: Indicates the multiplying factor for the control component that feeds directly from the input to the output of the PID controller.

Controller Clock Frequency: Sets the update rate of the PID compensator in hertz.

Actuator Low Saturation Limit: Defines the lower operating limit of the device connected to the PID output.

Actuator High Saturation Limit: Defines the upper operating limit of the device connected to the PID output.

Integrator Reset Value: Defines the value at which the integrator is initialized upon setting the Integrator Reset input high.

Derivative Bandwidth: Indicates the limiting frequency of the derivative component of control in hertz.

Proportional Bandwidth: Indicates the limiting frequency of the proportional component of control in hertz.

Use Higher Precision: When this option is activated, Tustin's method of integration is used for higher precision in integration and filtering. When this option is deactivated, backward rectangular integration is selected.

Command: Indicates the input signal to PID controller.

Measurement: Indicates the controlled variable input measurement for PID controller.

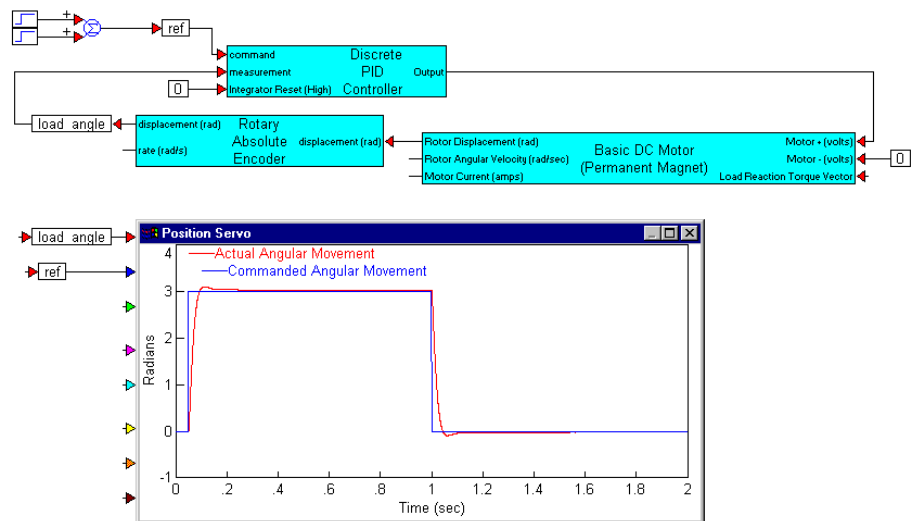
Integrator Reset: When this option is set high, it latches integrator to Integrator Reset Value. When low, it allows integration to resume.

Output: Indicates the PID controller output signal.

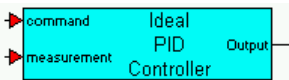
Example

Position control of a DC torque motor (...MOTION\EXAMPLES\DISCRETE_PID_CONTROLLER_EXMPL.VSM)

The following example illustrates the use of the PID compensator in shaping the dynamic response of a position control loop in feedback with a DC torque motor and rotary optical encoder. A profiler issues a command to move from 0 to 3 rad, and back to 0.



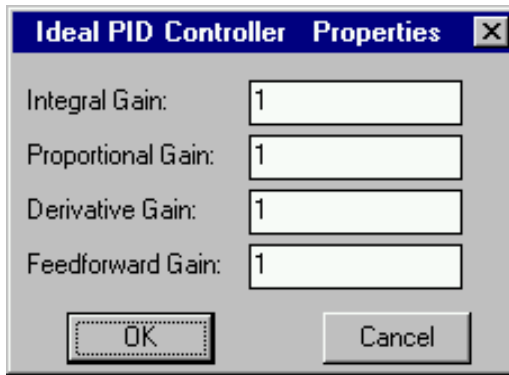
PID Controller (Ideal)



Example Model: ...MOTION\EXAMPLES\IDEAL_PID_CONTROLLER_EXMPL.VSM

Block Category: Controllers

The PID Controller block implements an ideal proportional, integral, derivative (PID) compensator for feedback controls with feed forward. This block assumes an ideal linear plant and does not provide for actuator saturation. Furthermore, the compensator provides only a simple derivative and therefore is unable to properly cope with measurement noise.



Integral Gain: Indicates the multiplying factor for the integral component of control.

Proportional Gain: Indicates the multiplying factor for the proportional component of control.

Derivative Gain: Indicates the multiplying factor for the derivative component of control.

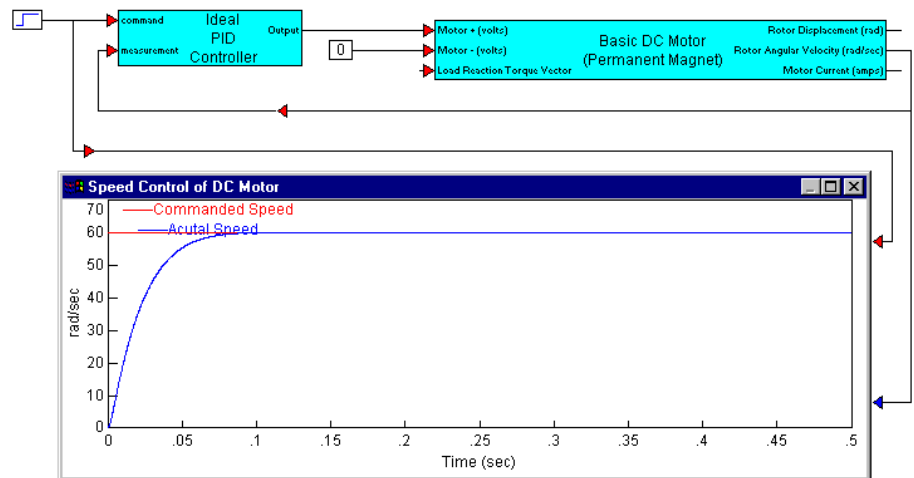
Feed forward Gain: Indicates the multiplying factor for the control component that feeds directly from the input to the output of the PID controller.

Example

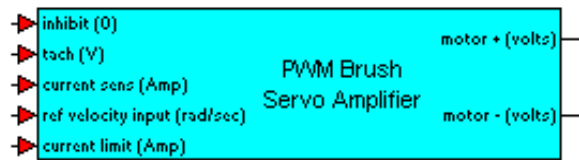
Conceptual draft of a DC motor speed control

(...\MOTION\EXAMPLES\IDEAL_PID_CONTROLLER_EXMPL.VSM)

This example illustrates how easy it is to determine a rough design for the speed control of a DC motor. The PID is configured as a PD compensator by choosing 0 integral and feed forward gains. A basic brush DC motor is used. Proportional and derivative gains are adjusted until adequate response time and overshoot are achieved. You can proceed from this simple feasibility model to include a discrete time controller, and sensor feedback model in simulation, and eventually to hardware in the loop simulation.



PWM Brush Servo Amplifier



Example Model: ...\\MOTION\\EXAMPLES\\BRUSH_PWM_SERVO_AMP_EXMPL.VSM

Block Category: Amplifier

The PWM Brush Servo Amplifier block simulates a brush DC motor bipolar PWM servo amplifier (4 quadrant) with current and velocity feedback that can be used for bi-directional motor control. The control logic simulates an H bridge inverter that allows PWM voltage swing between plus and minus the specified supply voltage over a 0 to 100% modulation range. The amplifier is biased at 50% modulation, which represents an average 0 output voltage. This allows bipolar input current control. An inhibit control is provided to disable amplifier output for a control voltage less than 0.3 V (logic low).

Control of the PWM is accomplished through inner loop current feedback control. Motor current is fed back through a current sense (resistor). The controller for the inner cascade current loop is PI, the integral and proportional gains can be specified by you. By selecting appropriate values of PI proportional and integral gain, current loop dynamics can be adjusted for any particular motor. Logic is provided that clamps the output to limit load current according to the specified current limit. Anti-windup control is provided in the integrator.

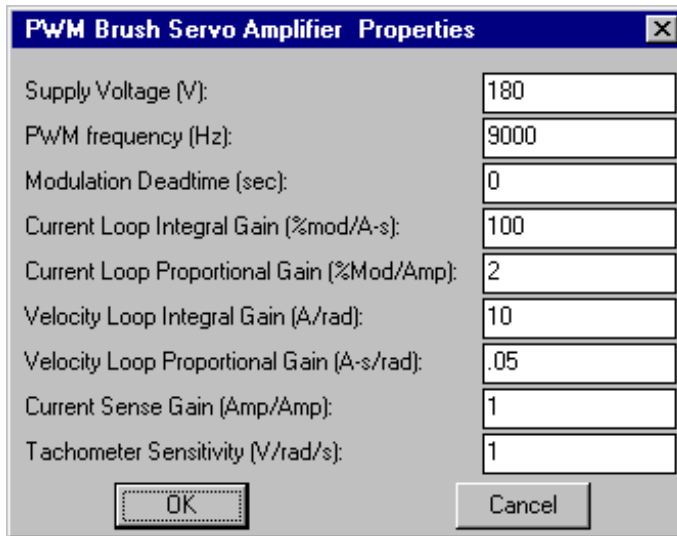
The outer cascade loop is velocity feedback control. A tachometer input (V) provides the measured shaft velocity that is converted to engineering units (rad/sec) by the tach sensitivity and compared against the reference (rad/sec). The controller is PI, the integral and proportional gains can be specified by you. The PI output is clamped by the current limit control that limits the current loop reference.

Deadtime that may occur between pairs of switching devices in the H bridge can be simulated by selecting an appropriate deadtime, or it can be set to 0 to disregard this affect.

Motor (+) and Motor (-) output provide the PWM control signal. To connect a single terminal device, the net modulation effect can be simulated by using a summingJunction block to create a difference between the bipolar output ((+) - (-)).

To provide adequate simulation of PWM behavior, the simulation step size should be chosen to be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Direct access to the PWM can be accomplished by setting both the current and tachometer sense gain and current loop and velocity loop integral gain equal to 0, and the current loop and velocity loop proportional gain equal to 1. In this situation current limiting is still available as long as motor current output is connected to the current sense input. By selecting appropriate gain combinations in the two PI controls, current feedback only, and velocity feedback control schemes can be configured.



Supply Voltage: Sets the voltage level output of the PWM in volts.

PWM Frequency: Sets the carrier frequency for the PWM in hertz. Note that the reciprocal of the simulation step size (simulation frequency) must be equal to or greater than the carrier frequency to perform a proper simulation.

Modulation Deadtime (sec): Sets the switching deadtime between H-bridge device pairs. Note that the value of deadtime should be significantly smaller than the PWM frequency, but larger than the specified simulation frequency rate. Units are in seconds..

Current Loop Integral Gain: Typically sets the speed and tracking accuracy (bandwidth) of the current control loop. Units are 1/sec.

Current Loop Proportional Gain: Typically sets the damping of the current control loop.

Velocity Loop Integral Gain: Typically sets the speed and tracking accuracy of the speed control loop. Units are in amps.

Velocity Loop Proportional Gain: Typically sets the damping of the speed control loop. Units are in amp-sec.

Current Sense Gain: Sets the current sensing device. Units are in amps/amp.

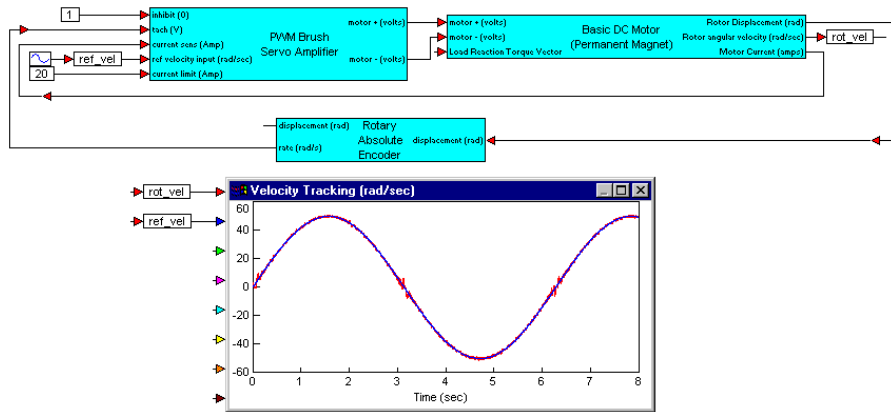
Tachometer sensitivity: Allows you to rescale tachometer readings from volts to rad/sec. For example if the known tachometer sensitivity is 200 volts/radian/sec, enter this number into the tachometer sensitivity setting of the PWM Brush Servo Amplifier block to scale the reading back to radians/sec.

Null Offset: Sets the minimum PWM frequency. This value must be greater than 0.

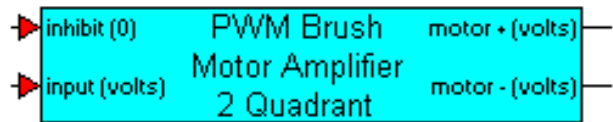
Example

Speed tracking control of a brush DC motor with current feedback (..\MOTION\EXAMPLES\BRUSH_PWM_SERVO_AMP_EXMPL.VSM)

This example illustrates the single controller package solution for servo tracking control of a DC brush motor with current feedback. A rotary encoder provides the velocity feedback measurement.



PWM Brush Motor Amplifier (2Q)



Example Model: ...MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_2Q_EXMPL.VSM

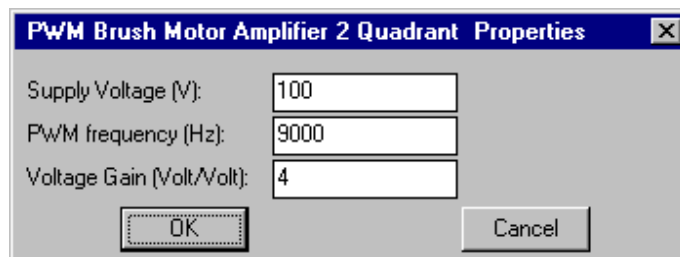
Block Category: Amplifier

The PWM Brush Motor Amplifier (2Q) block simulates a brush DC motor unipolar PWM amplifier (2 quadrant) that can be used for unidirectional motor control. The control logic simulates a two-device inverter that allows PWM swing from zero to the specified supply voltage over a 0 to 100% modulation range. An inhibit control is provided to disable amplifier output for a control voltage less than 0.3 volts (logic low). Control of the PWM is by direct voltage gain. You specify voltage gain.

Motor (+) output provides the PWM control signal, Motor (-) is referenced at a constant 0 volts.

Input voltage is limited between 0 and 10,000 volts.

To provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.



Supply Voltage: Sets the voltage level output of the PWM in volts.

PWM Frequency: Sets the carrier frequency for the PWM in hertz. Note that to provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Voltage Gain: Sets the voltage gain of the PWM amplifier.

Input: The amplifier input must be a positive voltage. The amplifier saturates beyond the range of 0 to 100% modulation. To avoid operation beyond saturation, the input should be limited between 0 volts and the supply voltage divided by the amplifier gain.

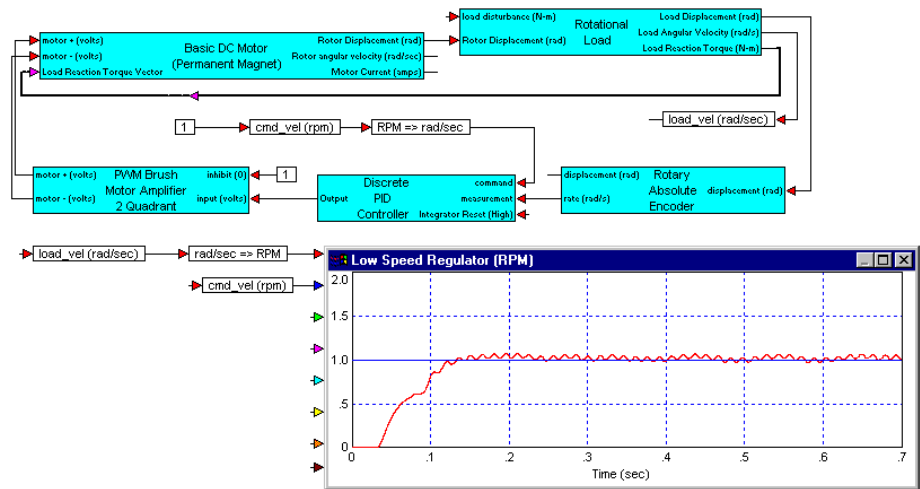
Inhibit: Provides immediate inhibiting of the output signal and can be used with limit switches for out-of-range control. Active when input is less than 0.3 volts (Active low).

Example

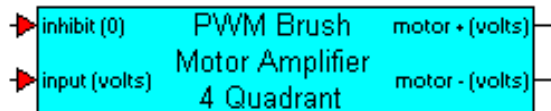
Low speed regulation

(...\MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_2Q_EXMPL.VSM)

In this example, a high-speed brush servomotor is used to regulate a very low speed on a high inertia load through a 10:1 gear reducer. Supply voltage for the motor is 8 volts and PWM frequency is 9000 Hz. Voltage gain of the PWM amplifier is set at 20. A PI controller is used to obtain a nearly 0 steady-state error on the speed of the load by using pure integral control at a gain of 100. Load rotation is measured using a 4000 line optical encoder, and rate is estimated using processor filtering. The result shows that the load can be accelerated to the desired rate very rapidly by using the gear reducer. In this example, gear backlash was set to 0. The fine ripple at 1 RPM is due to the noise generated by the encoder and rate estimation.



PWM Brush Motor Amplifier (4Q)



Example Model: ...\MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_4Q_EXMPL.VSM

Block Category: Amplifier

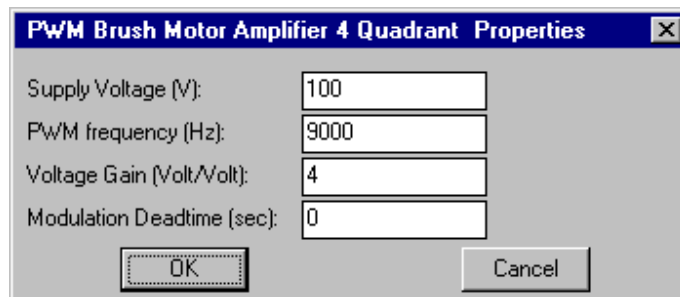
The PWM Brush Motor Amplifier (4Q) block simulates a brush DC motor bipolar PWM amplifier (4 quadrant) that can be used for bi-directional motor control. The control logic simulates an H bridge inverter that allows PWM voltage swing between plus and minus the specified supply voltage over a 0 to 100% modulation range. The amplifier is biased at 50% modulation, which represents an average 0 output voltage.

This allows bipolar input voltage control. An inhibit control is provided to disable amplifier output for a control voltage less than 0.3 volts (logic low). Control of the PWM is accomplished through voltage gain of the input signal. You can specify voltage gain.

Deadtime that may occur between pairs of switching devices in the H bridge can be simulated by selecting an appropriate deadtime, or set to 0 to disregard this affect.

Motor (+) and Motor (-) output provide the PWM control signal. To connect a single terminal device, the net modulation effect can be simulated by using a summingJunction block to create a difference between the bipolar output ((+) - (-)).

To provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.



Supply Voltage: Sets the voltage level output of the PWM in volts.

PWM Frequency: Sets the carrier frequency for the PWM in Hertz. Note that to provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Voltage Gain: Sets the voltage gain of the PWM amplifier.

Modulation Deadtime: Sets the switching deadtime between H – bridge device pairs. Note that the value of deadtime should be significantly smaller than the PWM frequency, but larger than the specified simulation frequency rate. Units are in seconds.

Input: The amplifier input can be a positive or negative voltage. The amplifier saturates beyond the range of 0 to 100% modulation. To avoid operation beyond saturation, the input should be limited plus or minus the supply voltage divided by the amplifier gain.

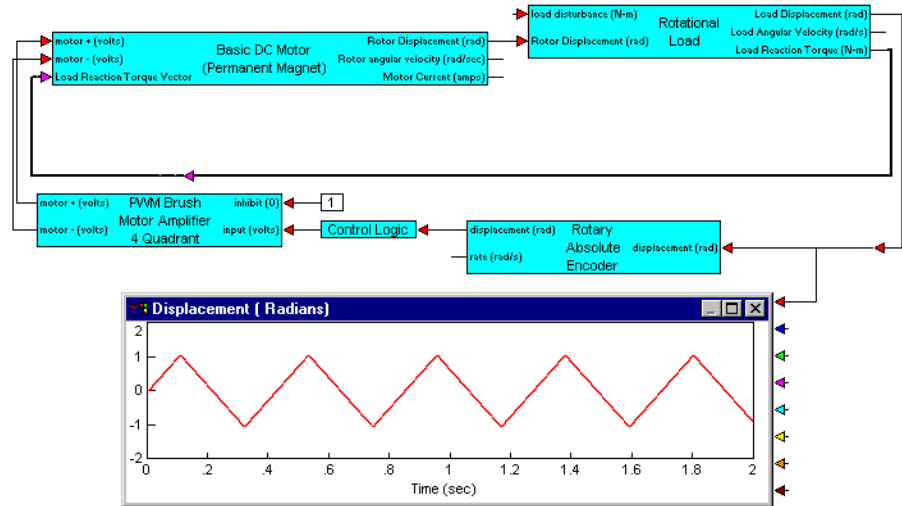
Inhibit: Provides immediate inhibiting of the output signal and can be used with limit switches for out-of-range control. Active when input is less than 0.3 volts (Active low).

Example

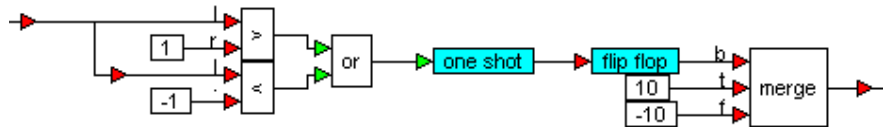
Reciprocating motion controller (...\MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_4Q_EXMPL.VSM)

This example shows reciprocating motion control of a load through a DC brush motor under PWM control. Rather than using a PID controller, control logic is designed that uses the absolute measurement of load position from an optical encoder. Control signals are flipped each time the motor reaches a determined end position in the positive and negative direction, in this case 1 radian of displacement. The supply voltage to the amplifier was chosen as 100 V with a PWM frequency of 9000 Hz. In this case, a command of plus or minus 10 V is switched through an

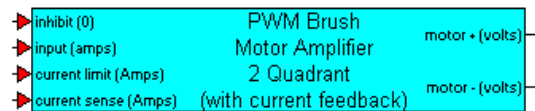
amplifier gain of 10 so that the PWM is saturated in both directions of motion. A gear 10:1 reducer is used to amplify torque output from the motor.



The following diagram details the contents of the Control Logic compound block. Each time the desired endpoints of plus or minus 1 radian is sensed by the encoder, a one shot fires and switches the flip-flop to reverse motor direction.



PWM Brush Motor Amplifier (2Q, Current Feedback)



Example Model: ...MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_2Q_WCF_EXMPL.VSM

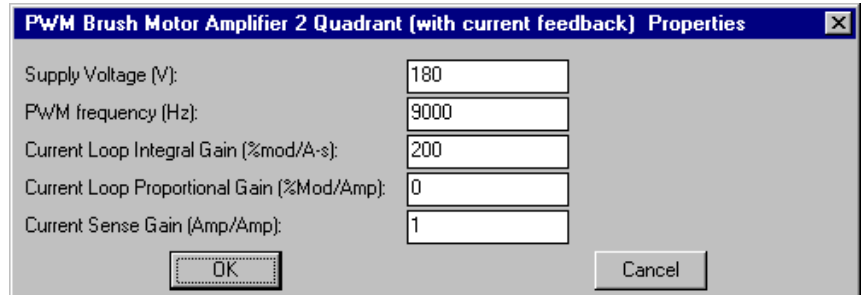
Block Category: Amplifier

The PWM Brush Motor Amplifier (2Q, Current Feedback) block simulates a brush DC motor unipolar PWM amplifier (2 quadrant) with current feedback that can be used for unidirectional motor control. The control logic simulates a two-device inverter that allows PWM swing from 0 to the specified supply voltage over a 0 to 100% modulation range. An inhibit control is provided to disable amplifier output for a control voltage less than 0.3 V (logic low). Control of the PWM is accomplished through current feedback control. Motor current is fed back through a current sense (resistor). The controller for the current loop is PI; the integral and proportional gains can be specified by you. By selecting appropriate values of PI proportional and integral gain, current loop dynamics can be adjusted for any particular motor. Logic is provided that clamps the output to limit load current according to the specified current limit. Anti-windup control is provided in the integrator.

Motor (+) output provides the PWM control signal; Motor (-) is referenced at a constant 0 V.

To provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Direct access to the PWM can be accomplished by setting the current sense gain and current loop integral gain equal to 0, and the current loop proportional gain equal to 1. In this situation current limiting is still available as long as motor current output is connected to the current sense input.



Supply Voltage: sets the voltage level output of the PWM in volts.

PWM Frequency: sets the carrier frequency for the PWM in hertz. Note that to provide adequate simulation of PWM behavior, the simulation step size should be chosen to be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Current Loop Integral Gain: Typically sets the speed and tracking accuracy (bandwidth) of the current control loop. Units are 1/sec.

Current Loop Proportional Gain: Typically sets the damping of the current control loop.

Current Sense Gain: Sets the gain of the current feedback-sensing device. Units are in amps/amp.

Input: The amplifier input must be a positive voltage. The amplifier saturates beyond the range of 0 to 100% modulation. To avoid operation beyond saturation the input should be limited between 0 V and the supply voltage divided by the amplifier gain.

Current Limit: Sets the maximum average output current the amplifier will provide. Units are in amps.

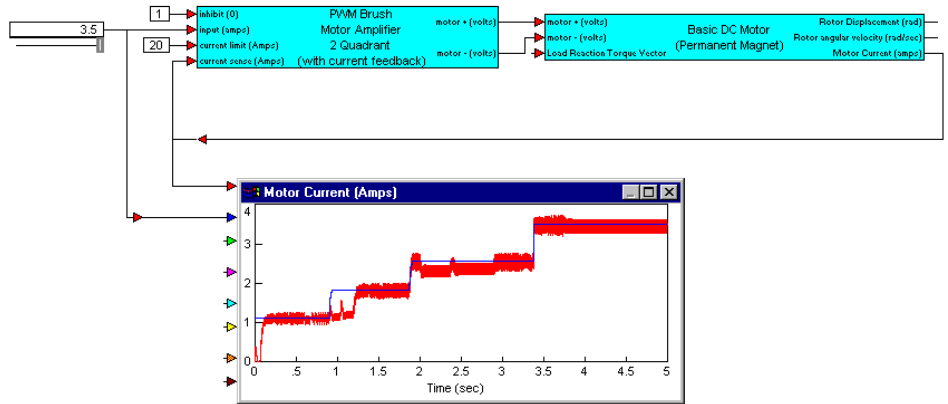
Inhibit: Provides immediate inhibiting of the output signal and can be used with limit switches for out-of-range control. Active when input is less than 0.3 V (active low).

Example

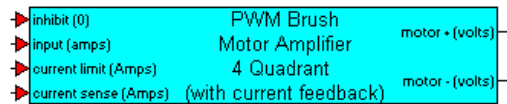
Motor current regulation

(..\MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_2Q_WCF_EXMPL.VSM)

This example illustrates how the PWM amplifier can be used to directly regulate a set current in a DC motor. Such a control loop may be used to control against torque disturbance since torque is directly proportional to motor current. Here the supply voltage is set at 180 V, and PWM frequency is 9000 Hz. Proportional gain is set at 0, so the control is pure integral with a gain of 50. The current sense gain is 1, so the average controlled current should approach the set current of 1.5 amps.



PWM Brush Motor Amplifier (4Q, Current Feedback)



Example Model: ...MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_4Q_WCF_EXMPL.VSM

Block Category: Amplifier

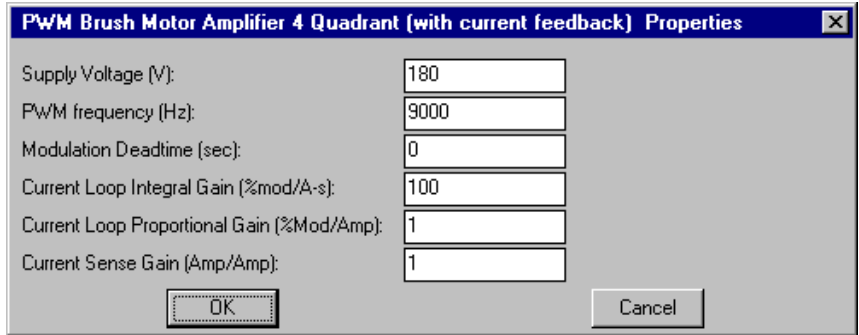
The PWM Brush Motor Amplifier (4Q, Current Feedback) block simulates a brush DC motor bipolar PWM amplifier (4 quadrant) with current feedback that can be used for bi-directional motor control. The control logic simulates an H bridge inverter that allows PWM voltage swing between plus and minus the specified supply voltage over a 0 to 100% modulation range. The amplifier is biased at 50% modulation, which represents an average 0 output voltage. This allows bipolar input current control. An inhibit control is provided to disable amplifier output for a control voltage less than 0.3 V (logic low). Control of the PWM is accomplished through current feedback control. Motor current is fed back through a current sense (resistor). The controller for the current loop is PI; the integral and proportional gains can be specified by you. By selecting appropriate values of PI proportional and integral gain, current loop dynamics can be adjusted for any particular motor. Logic is provided that clamps the output to limit load current according to the specified current limit. Anti-windup control is provided in the integrator.

Deadtime that may occur between pairs of switching devices in the H bridge can be simulated by selecting an appropriate deadtime, or it can be set to 0 to disregard this affect.

Motor (+) and Motor (-) output provide the PWM control signal. To connect a single terminal device, the net modulation effect can be simulated by using a summingJunction block to create a difference between the bipolar output ((+) - (-)).

To provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Direct access to the PWM can be accomplished by setting the current sense gain and current loop integral gain equal to 0, and the current loop proportional gain equal to 1. In this situation current limiting is still available as long as motor current output is connected to the current sense input.



Supply Voltage: Sets the voltage level output of the PWM in volts.

PWM Frequency: Sets the carrier frequency for the PWM in hertz. Note that to provide adequate simulation of PWM behavior, the simulation step size should be considerably smaller than the inverse of the selected PWM frequency; preferably at least 100 times smaller.

Current Loop Integral Gain: Typically sets the speed and tracking accuracy (bandwidth) of the current control loop. Units are 1/sec.

Current Loop Proportional Gain: Typically sets the damping of the current control loop.

Current Sense Gain: Sets the gain of the current feedback-sensing device. Units are in amps/amp.

Input: The amplifier input can be a positive or negative voltage. The amplifier saturates beyond the range of 0 to 100% modulation. To avoid operation beyond saturation, the input should be limited between plus or minus the supply voltage divided by the amplifier gain.

Current Limit: Sets the maximum average output current the amplifier will provide. Units are in amps.

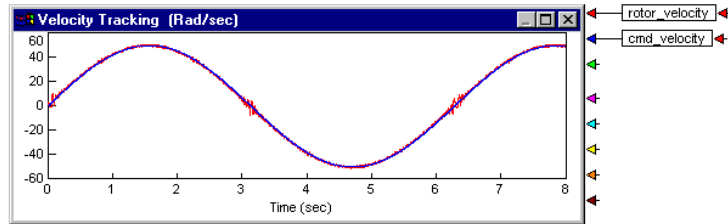
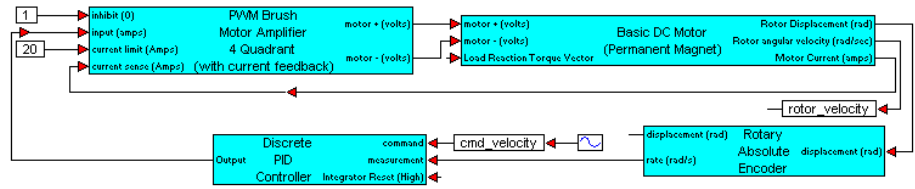
Inhibit: Provides immediate inhibiting of the output signal and can be used with limit switches for out-of-range control. Active when input is less than 0.3 V (active low).

Example

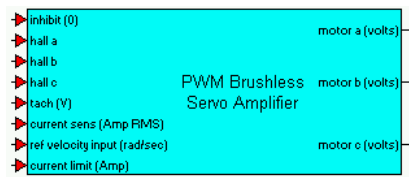
Velocity Tracking with Current Feedback

(..\MOTION\EXAMPLES\BRUSH_PWM_MOTOR_AMP_4Q_WCF_EXMPL.VSM)

In this example, the 4-quadrant PWM amplifier is used with current feedback and with a cascaded velocity loop to track a reference sinusoidal speed trajectory. The velocity PI controller is set with integral gain at 10, proportional gain at 0.1 and feed forward gain at 0.1. Current loop PI control is set with integral gain at 100, and proportional gain at 1. Current sense is at a gain of 1, and PWM frequency is at 9000 Hz. A 4000 line optical encoder is used to close the velocity loop. The response shows stiff tracking of the reference signal. An interesting result occurs near 0 velocity that may be due to a dropout in the rate estimation from the encoder or possibly not enough modulation depth in the PWM either by design or by inadequate selection of the simulation step rate.



PWM Brushless Servo Amplifier



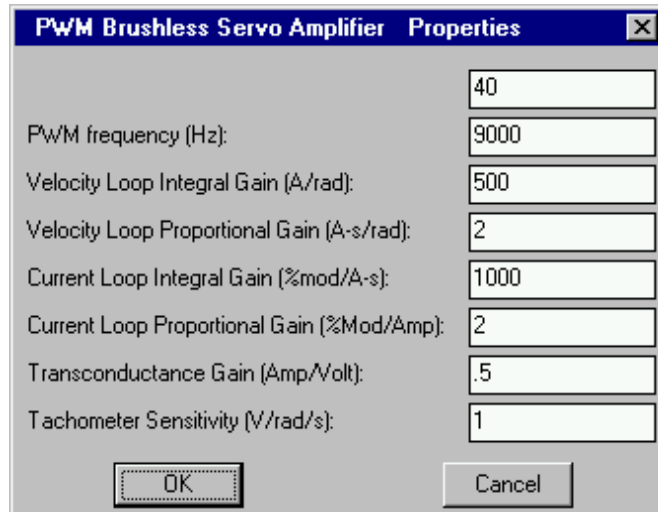
Example Model: ...MOTION\EXAMPLES\BRUSHLESS_PWM_SERVO_AMP_EXMPL.VSM

Block Category: Amplifiers

The PWM Brushless Servo Amplifier block simulates a brushless PWM servo amplifier used to control the velocity of a brushless DC motor. The control logic provides a transformation that decodes Hall sensed motor shaft position and direction to stator field commutation (six-step control). An inhibit control is provided to disable amplifier output for voltage input less than 0.3 V (logic low). All 3-phase voltages are provided with full PWM control (0-100%). A value of 100% represents no modulation. This allows variable speed control. Control of the PWM is accomplished through cascaded loops.

The inner loop is current feedback control. Current sense is fed back through a current sense (resistor) or transconductance gain. The controller for the current loop is PI; you can specify integral and proportional gains. The PI output is clamped between 0 and 100% modulation to prevent control windup.

The outer loop is velocity feedback control. A tachometer input (V) provides the measured shaft velocity that is converted to engineering units (rad/sec) by the tach sensitivity and compared against the reference velocity input (rad/sec). The controller is PI; you can specify integral and proportional gains. The PI output is clamped by the current limit control that limits the current loop reference.



Supply Voltage: Specifies the voltage level for PWM pulses in volts. A positive voltage produces clockwise rotation of the motor. A negative voltage produces counterclockwise rotation.

PWM Frequency: Sets the carrier frequency for the PWM in hertz. Note that the reciprocal of the simulation step size (simulation frequency) must be equal to or greater than the carrier frequency to perform a proper simulation.

Velocity Loop Integral Gain: Indicates the velocity loop PI controller integral gain. This option typically sets the steady-state tracking accuracy of the speed control. Units are in amps/rad.

Velocity Loop Proportional Gain: Indicates the velocity loop PI controller proportional gain. This option typically sets the speed (bandwidth) of the velocity control. Units are in aAmp-sec/rad.

Current Loop Integral Gain: Indicates the current loop PI controller integral gain. This option typically sets the steady-state tracking accuracy of the current control. Units are in 1/sec.

Current Loop Proportional Gain: Indicates the current loop PI controller proportional gain. This option typically sets the speed (bandwidth) of the current control.

Transconductance Gain: Sets the current to voltage gain of the current controller. Units are in amps/V.

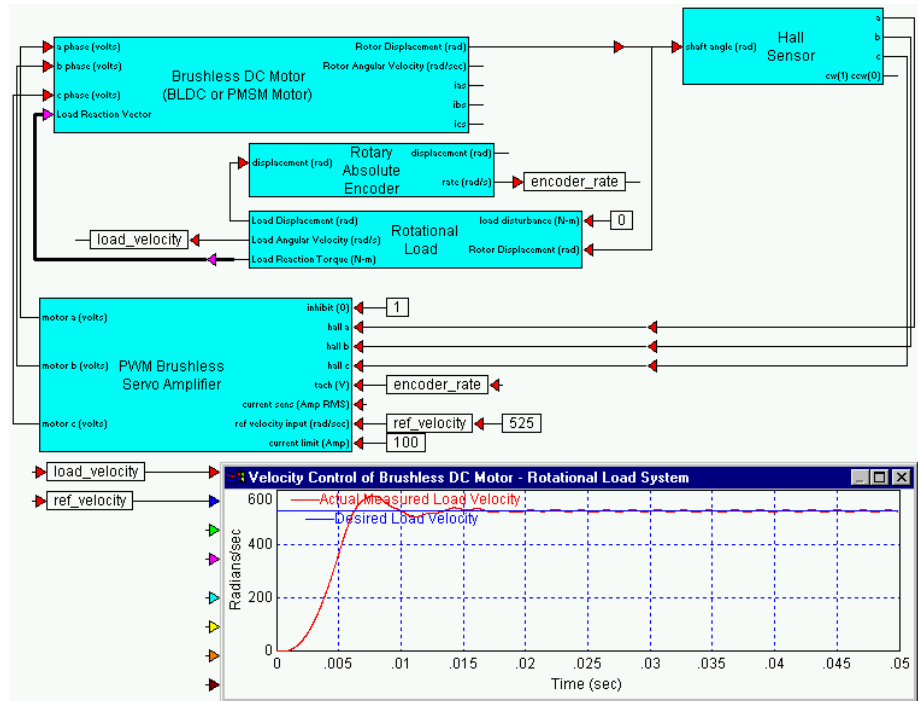
Tachometer sensitivity: Lets you rescale tachometer readings from V to rad/sec. For example, if the known tachometer sensitivity is 200 V/rad/sec, enter this number to scale the reading back to rad/sec.

Example

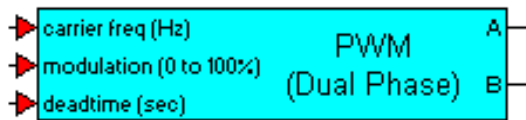
Velocity control of a brushless DC motor using PWM servo amplifier and encoder feedback

(..\MOTION\EXAMPLES\BRUSHLESS_PWM_SERVO_AMP_EXMPL.VSM)

The following example demonstrates the application of PWM Brushless Servo Amplifier block being used to control velocity of a brushless DC motor in feedback with an encoder. Hall sensors are used for commutation of the motor and are connected between the motor shaft and the PWM amplifier. Note that the encoder measures rotor shaft displacement and velocity is estimated for feedback.



PWM (Dual Phase)



Example Model: ... \MOTION\EXAMPLES\PWM_DUAL_PHASE_EXMPL.VSM

Block Category: Controllers

The PWM (Dual Phase) block provides a 2-phase pulse width modulated signal based on carrier frequency and modulation percent. Modulation is created using the triangular wave – threshold comparator technique. The three inputs are:

Carrier Freq: Indicates the base frequency of modulation carrier in hertz.

Modulation: Indicates the percentage of carrier frequency cycle in which pulse is active (high) from 0 to 100%.

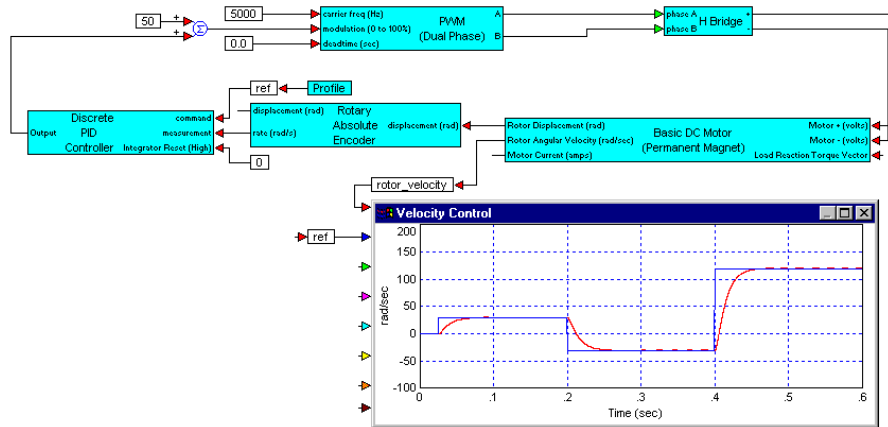
Deadtime: Sets a symmetric deadtime between the A and B PWM phases. Units are in seconds.

Example

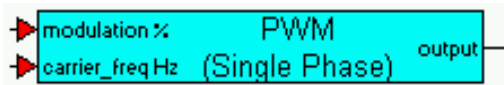
Bi-directional PWM velocity control

(... \MOTION\EXAMPLES\PWM_DUAL_PHASE_EXMPL.VSM)

The following example illustrates how the 2-phase PWM can be used to control bi-directional velocity in a closed loop. The two PWM phases are fed to an H bridge power amplifier with voltage rails at ± 28 V. Modulation is biased at 50% from the output of a PI controller limited to $\pm 50\%$ to prevent windup in the control. An encoder with rate estimation is used for feedback.



PWM (Single Phase)



Example Model: ...MOTION\EXAMPLES\PWM_SINGLE_PHASE_EXMPL.VSM

Block Category: Controllers

The PWM (Single Phase) block provides a pulse width modulated signal based on the input carrier frequency and % modulation.

Modulation%: Indicates the percentage of carrier frequency cycle in which pulse is active (high) from 0 to 100%.

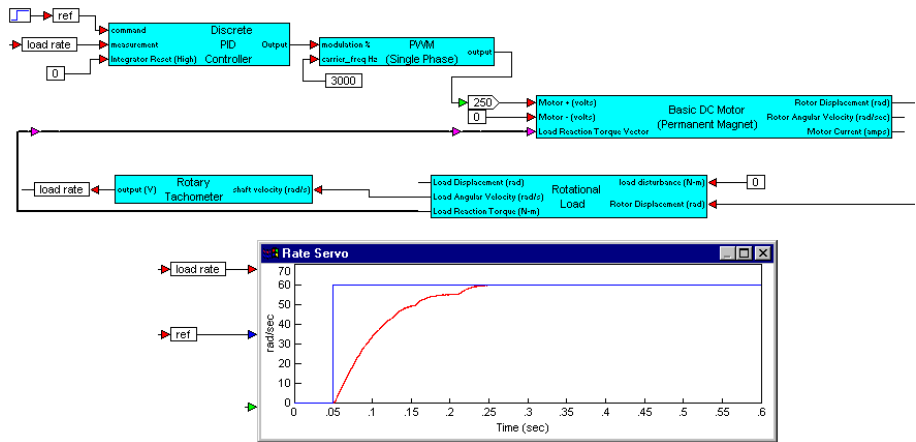
Carrier_freq: Indicates the base frequency of modulation carrier signal in hertz

Output: Indicates the pulse width modulated output scaled 0 to 1.

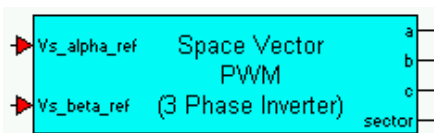
Example

Unidirectional PWM speed controller for DC Motor (...MOTION\EXAMPLES\PWM_SINGLE_PHASE_EXMPL.VSM)

In this example, the PID Controller (Digital) block is configured as PI and saturation limits are set for 0 to 100 for anti-windup control with the PWM logic. A modulation frequency is chosen at 3000 Hz, and the gain of 250 represents voltage switching at 250 V, controlled by the PWM. Velocity is measured from the load using a tachometer and fed back to the PI compensator.



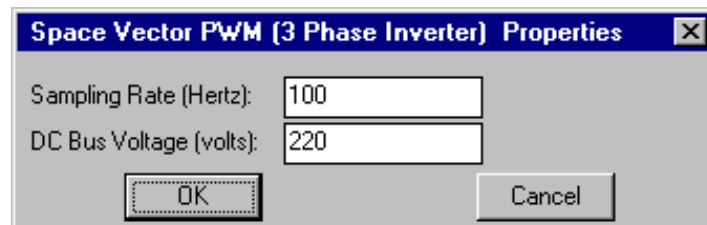
PWM (Space Vector)



Example Model: ... \MOTION\EXAMPLES\PWM_SPACE_VECTOR_EXMPL.VSM

Block Category: Controllers

The PWM (Space Vector) block produces PWM pulses from orthogonal 2-phase input signals using space vector modulation techniques



Sampling Rate: Defines the processor rate for PWM calculations in hertz. Note that this frequency is typically chosen to be large (8000 Hz or greater). The simulation rate should be at least 10 times the PWM rate to assure proper operation of the simulation.

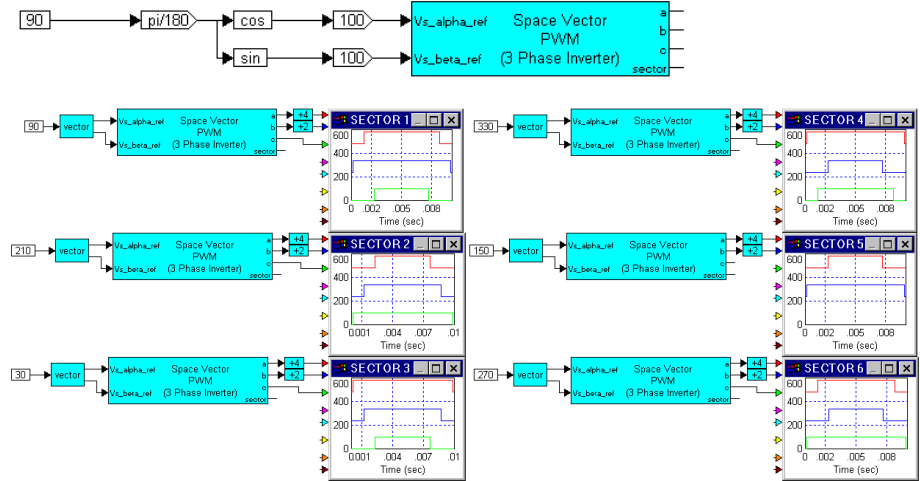
DC Bus Voltage: Defines the output voltage level of the 3-phase PWM signals in volts.

Example

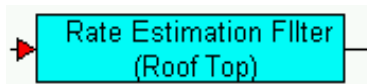
SVPWM sector modulation patterns

(... \MOTION\EXAMPLES\PWM_SPACE_VECTOR_EXMPL.VSM)

Modulation patterns in SVPWM occur based on the location of the timing vector in 6 possible sectors in one complete revolution. The diagram is constructed and duplicated 6 times, changing the input angle in 60 degree increments for each diagram copy that places the timing vector mid-sector for each case. Output phases are then connected to plot blocks to display the patterns for each sector, as shown in the following diagram.



Rate Estimation (Rooftop) Filter

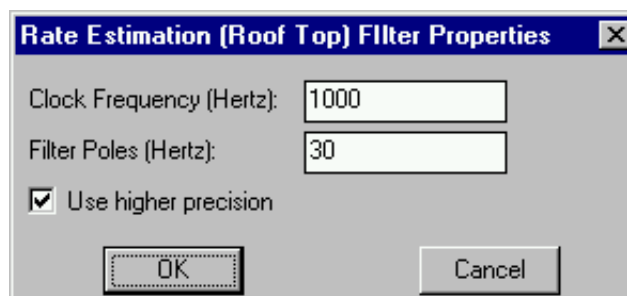


Example Model:

... \MOTION\EXAMPLES\RATE_ESTIMATION_ROOF_TOP_FILTER_EXMPL.VSM

Block Category: Filters

The Rate Estimation (Rooftop) Filter block provides a band limited derivative of a signal by band-pass filtering. This method of obtaining the derivative is optimal in a sense that it maximizes derivative bandwidth as a compromise to upper band noise attenuation. The rooftop filter is a 2-pole band pass filter in which the poles at p rad/sec are equal. If the filter is scaled by p^2 , the low frequency band of the filter well below the filter poles corresponds to the response of a pure derivative with gain equal to ω and a phase shift of plus 90° . As the frequency nears p rad/sec, phase shift becomes less than 90° and the derivative response is diminished. Frequency content above p rad/sec is attenuated at -20 dB/decade. A general rule is to select the filter pole 30 times larger than the highest frequency at which derivative response is to be applied. Pole selection provides a trade-off between derivative bandwidth and higher frequency noise attenuation.



Clock Frequency: Defines the processor rate for rate estimation in hertz.

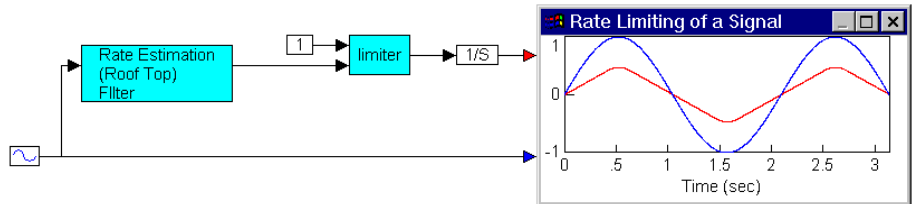
Filter Pole: Defines the center frequency of the rooftop filter in hertz. Select a frequency 30 times larger than the highest frequency for estimation of derivative.

Example

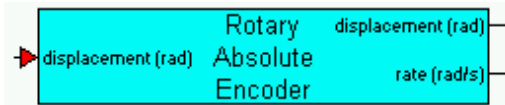
Rate limiting a signal

(...\MOTION\EXAMPLES\RATE_ESTIMATION_ROOF_TOP_FILTER_EXMPL.VSM)

The following example demonstrates how to use a rooftop filter to limit the rate on a signal. The rooftop filter is first used to determine the derivative (rad/sec) of the angular displacement input signal (rad). The rate is limited, and then converted back to radians by integration. The resulting signal shows limiting when the input signal exceeds the rate limit.



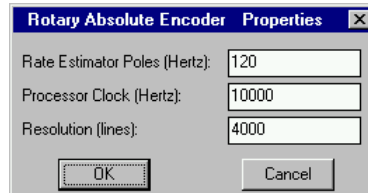
Rotary Encoder



Example Model: ...\MOTION\EXAMPLES\ROTARY_ABSOLUTE_ENCODER_EXMPL.VSM

Block Category: Sensors

The Rotary Encoder block models quantized displacement measurements obtained from a generic encoding device such as a rotary optical encoder or similar device. This block provides simulation of processor rate estimation from quantized signal.



Rate Estimator Poles: Defines the bandwidth of rate estimation. This option typically requires a trade-off between rate estimation accuracy and noise. Units are in hertz.

Processor Clock: Defines the processor clock rate for rate estimation.

Resolution: Indicates the total number of lines per revolution.

Displacement (input): Indicates the physical angular displacement connection. Units are in rad.

Displacement (output): Indicates from where the quantized displacement measurement is read. Units are in rad.

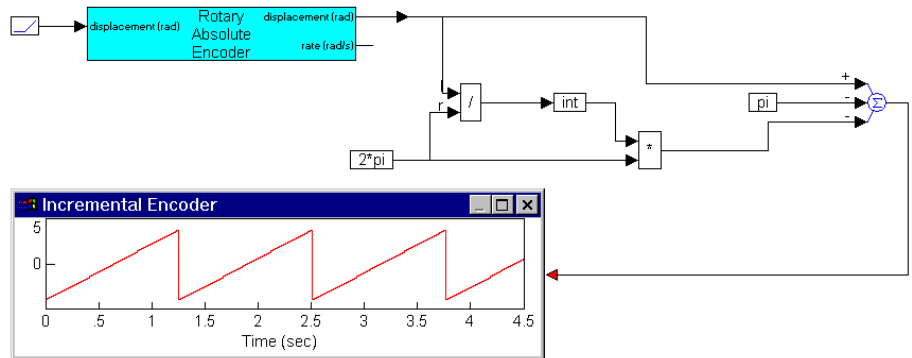
Rate: Indicates from where the estimated rate is read. Units are in rad/sec.

Example

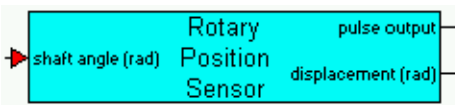
Simulating a true absolute encoder

(...\MOTION\EXAMPLES\ROTARY_ABSOLUTE_ENCODER_EXMPL.VSM)

The Rotary Encoder block actually simulates an incremental encoder with an assumed built-in counter providing undisturbed unlimited displacement output. A true absolute encoder that uses some type of binary coded scales will reset to 0 position after one complete revolution, and will usually issue an index signal at the 0 position to signal an external counter or processor. In cases where a simulation of an absolute encoder is needed, a simple block construction can be made to calculate absolute angular position using a modulo 2π conversion. The following simulation example illustrates this method. In this example, π is subtracted at the output to provide $\pm\pi$ measurements.



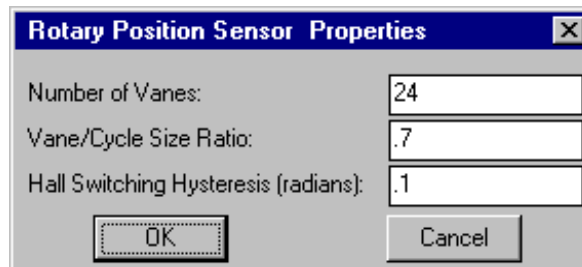
Rotary Position Sensor



Example Model: ...MOTION\EXAMPLES\ROTARY_POSITION_SENSOR_EXMPL.VSM

Block Category: Sensors

The Rotary Position Sensor block models the behavior of a vane-operated rotary position sensor that uses digital (ON/OFF) Hall sensors. Shaft angle motion creates pulse output as vane interruption between Hall sensor and magnet complete a flux path and the Hall sensor shuts off (it is normally on). A pulse counter simulates electronics or software that accumulates pulses for displacement measurement. Phasing determines UP/DOWN count. You specify the number of vanes on the wheel and the corresponding window/vane ratio. A cycle is the size of the combined window and vane.



Number of Vanes: Indicates the number of vanes on the interrupter wheel.

Vane/Cycle Size Ratio: Indicates the proportion of space occupied by the vane interrupter vs. total cycle. A cycle is the size of the combined window and vane.

Hall Switching Hysteresis (radians): Sets the Hall device switching hysteresis characteristic. Units are in radians.

Shaft Angle: Specifies the displacement of input device to be measured.

Pulse Output: Indicates the pulse output corresponding to passage of a cycle over a Hall sensor.

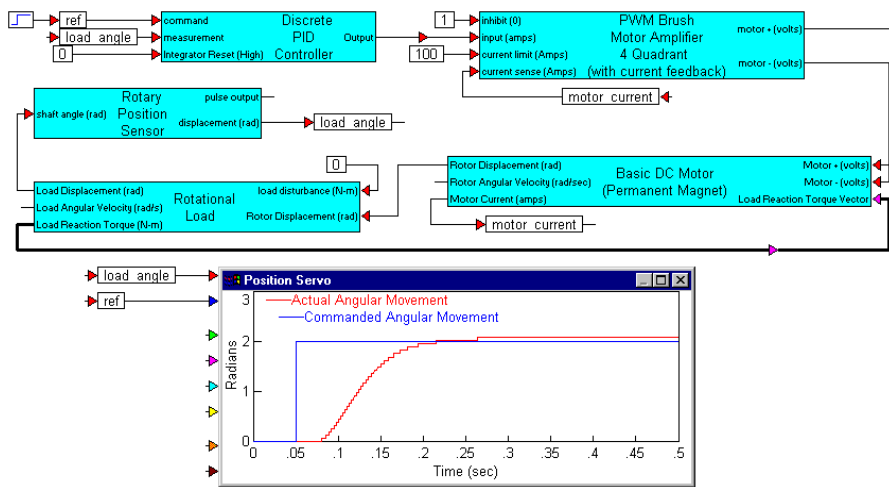
Displacement: Indicates the counter output scaled by number of vanes per 1 rotation of the shaft to provide angular displacement in radians.

Example

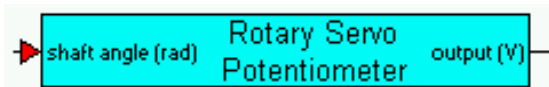
Position control of a brush DC motor

(...\MOTION\EXAMPLES\ROTARY_POSITION_SENSOR_EXMPL.VSM)

The example below shows a brush DC motor model controlled by a PI compensator in feedback with the Rotary Position Sensor block.



Rotary Servo Potentiometer

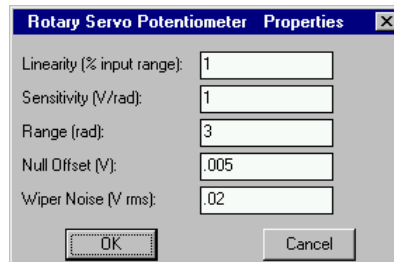


Example Model:

...\MOTION\EXAMPLES\ROTARY_SERVO_POTENTIOMETER_EXMPL.VSM

Block Category: Sensors

The Rotary Servo Potentiometer block models the behavior of a servo potentiometer in terms of linearity and wiper noise. A rate estimator is used to switch wiper noise off at zero velocity. Hard stops are provided.



Linearity: Specifies the maximum deviation from true linear response in terms of a percentage of input range. A curvilinear distortion, symmetric about null is used to model non-linearity (S-curve).

Sensitivity: Indicates the gain of the potentiometer in V/rad.

Range: Indicates the total angular stroke of the potentiometer in rad. If the input exceeds the allowable range, the output is limited to a constant value as though a slip-clutch coupling were used.

Null Offset: Indicates the offset voltage at 0 input position in volts.

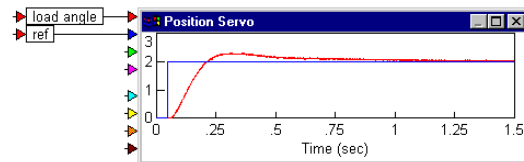
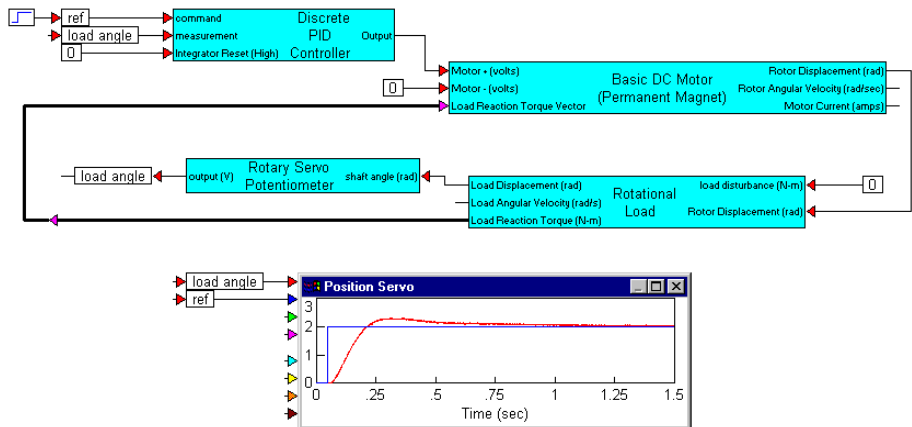
Wiper Noise: Indicates the RMS voltage wiper electrical noise induced by contact motion between wiper and resistive element. Units are in volts RMS.

Example

Position feedback control of a permanent magnet DC motor

(...MOTION\EXAMPLES\ROTARY_SERVO_POTENTIOMETER_EXMPL.VSM)

A fairly rough position control system is simulated using inexpensive components including a 10 to 1 gear reducer with 1/2 degree backlash, and a noisy servo potentiometer with 1% nonlinearity. The load has no intrinsic restoring forces. Consequently, a PID control block is used to configure a PI controller. The results show that even with the inexpensive components, fairly decent positioning control can be obtained with minor overshoot.



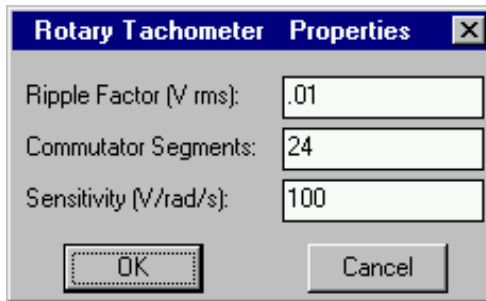
Rotary Tachometer



Example Model: ...MOTION\EXAMPLES\ROTARY_TACHOMETER_EXMPL.VSM

Block Category: Sensors

The Rotary Tachometer block models the behavior of a tachometer generator using the standard ripple model. This block creates output voltage signals proportional to input shaft velocity with ripple frequency components. These components are a nonlinear function of the input shaft angular velocity and number of commutator segments in the tachometer.



Ripple Factor: Indicates the tachometer ripple factor in volts RMS.

Commutator Segments: Indicates the number of tachometer commutator segments.

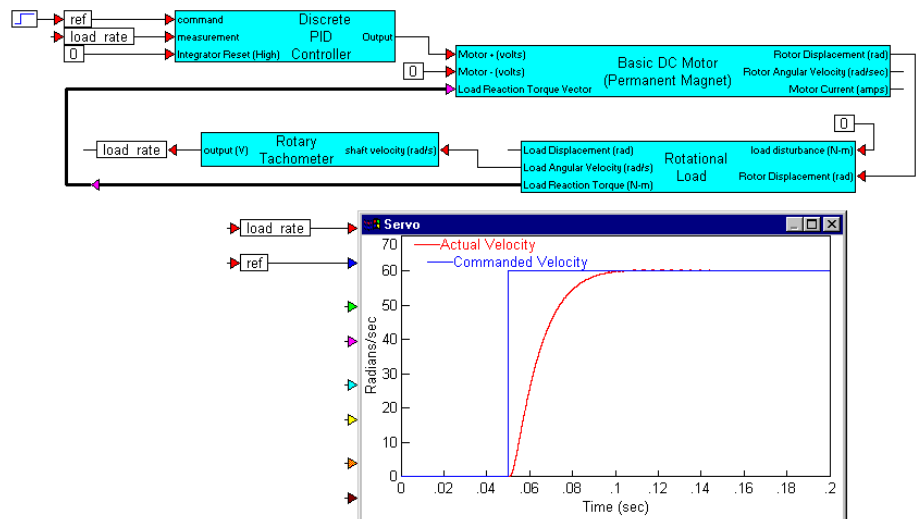
Sensitivity: Indicates the tachometer gain in V/rad/sec

Example

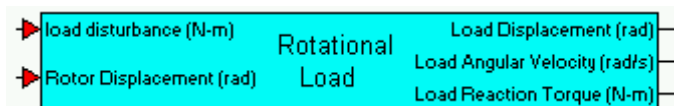
Effect of tachometer ripple on closed loop velocity control

(...\MOTION\EXAMPLES\ROTARY_TACHOMETER_EXMPL.VSM)

The following simulation example shows velocity feedback control of a permanent magnet DC motor. A load is coupled to the motor by a gear ratio of 0.5, thus multiplying motor torque by a factor of 2. Load velocity is measured by a tachometer with a ripple factor of 0.1 VRMS and 24 commutator segments. The plot zooms in on the velocity as it approaches steady state (60 radians/sec). The ripple can be seen superimposed on the response.



Rotational Load



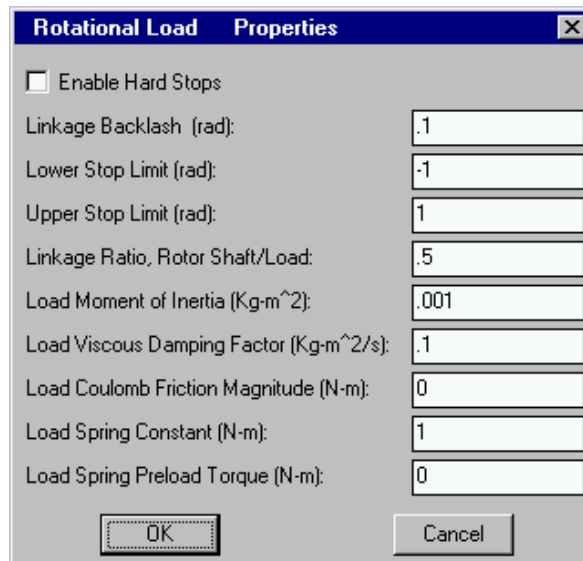
Example Model: ... \MOTION\EXAMPLES\ROTATIONAL_LOAD_EXMPL.VSM

Block Category: Loads

The Rotational Load block represents a mechanical rotational load model.

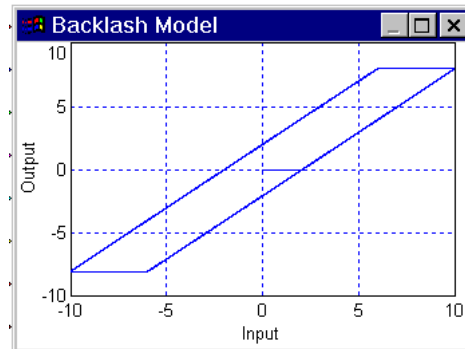
The Rotational Load block models the dynamic and static behavior of a single degree of freedom, rotational rigid body connected by a linkage mechanism. The Rotational Load block provides flexible definition over a variety of generic load configurations used for motor applications in speed, torque and position control. The generic linkage ratio represents any type of mechanical advantage that multiplies either torque or speed. This represents a large class of mechanisms, including gear trains, pulleys and levers. The linkage can either include or exclude hysteretic backlash.

The Rotational Load block requires connection with a motor block model to complete the load dynamics. Load reaction torques are reflected back into motor dynamics using a load reaction torque vector wire connection. Besides hysteresis, the Rotational Load block includes Coulomb friction, viscous friction, load inertia, torsional spring torque, spring preload torque, and hard stops. All of these properties can be used simultaneously; or by choosing a 0 value for a particular property, it can be eliminated from the Rotational Load block. A disturbance torque input is provided that may be applied from other system component models, such as bearing noise, and wind gust torque models.



Enable Hard Stops: When activated, this block enables hard stops on rotational motion at the specified upper and lower stop limits.

Linkage Backlash: Introduces hysteretic backlash in linkage mechanism in units of radians. Enter 0 for no backlash.



Lower Stop Limit: Indicates the minimum hard stop limit in radians. This option is available only when Enable Hard Stops is activated.

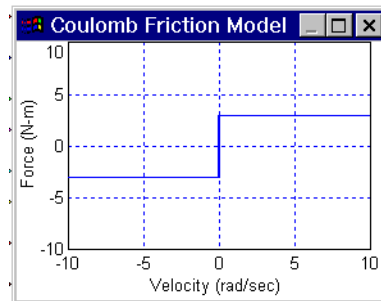
Upper Stop Limit: Indicates the maximum hard stop limit in radians. This option is available only when Enable Hard Stops is activated.

Linkage Ratio, Rotor Shaft/Load: Defines the mechanical advantage factor between the rotor shaft and load. Enter 1 for direct coupling between rotor shaft and rotational load; enter a number less than 1 to multiply torque; and enter a number greater than 1 to multiply speed.

Load Moment of Inertia: Indicates the inertia of load about the axis of rotation in $\text{kg}\cdot\text{m}^2$.

Load Viscous Damping Factor: Indicates the proportional factor that relates the angular velocity to the dissipative torque. Enter 0 for no viscous damping.

Load Coulomb Friction Magnitude: Indicates the constant directional sensitive torque (Coulomb model) in N-m. Enter 0 for no Coulomb loading.



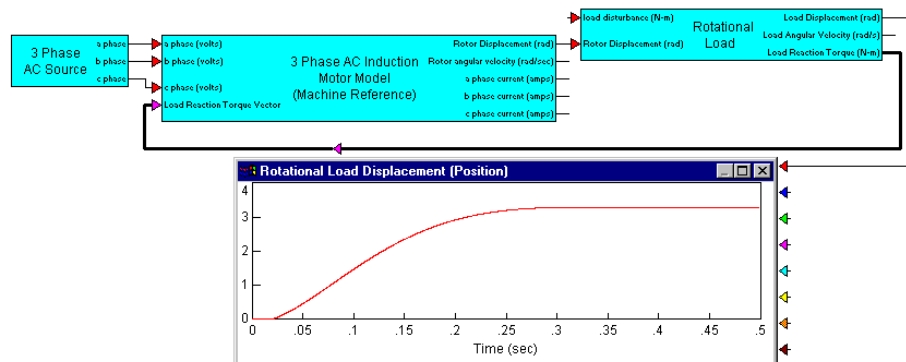
Load Spring Constant: Indicates the proportional spring factor that relates the angular position of the load to the restoring torque. Enter 0 if there is no torsional spring in the load.

Load Spring Preload Torque: Indicates the initial restoring (bias) torque introduced by load torsional spring. Enter 0 if there is no preload in the torsional spring.

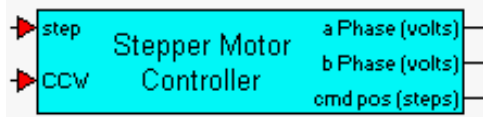
Example

Measuring output zero speed torque of a 3-phase AC induction motor (... \MOTION\EXAMPLES\ROTATIONAL_LOAD_EXMPL.VSM)

The Rotational Load block is configured with a torsional spring with a 1 N-m/rad spring constant. The motor is started and allowed to settle to its final equilibrium position at 0 velocity. Since the spring constant is unity, the final angle of the shaft is a direct measure of the output torque in N-m.



Stepper Motor Controller



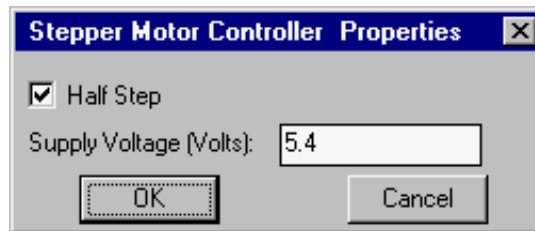
Example Model:

...\MOTION\EXAMPLES\STEPPER_MOTOR_CONTROLLER_EXMPL.VSM

Block Category: Controllers

The Stepper Motor Controller block simulates a typical full/half step bi-directional controller for operating incremental motion of 2-phase permanent magnet stepper motors. The controller can be used for open-loop positioning or speed (slew) control applications, or used as a component in a step motor closed-loop control system. Logic inputs provide step activation and directional control. A counter output is provided to track the current commanded (accumulated) position in steps.

Note that the slew rate (speed) of the motor in steps/sec depends on the rate of input pulses. Actual shaft (angular rate) depends on this pulse rate and also the characteristics of the stepper motor (number of rotor teeth). Full step resolution of the motor is $360/(\text{number of rotor teeth})$ in degrees. Selecting half step operation can increase incremental resolution.



Half Step: When this checkbox is checked, the controller issues pulse sequences that step the motor in half step increments. When this checkbox is not checked, the controller issues full step increments.

Supply Voltage: Defines the voltage level of phase a and b pulses. Units are in volts.

Step: Specifies logic transition input (0 to 1). The rising edge of this input shifts the control sequence one increment, causing the motor to move by one step (one half step if the half step checkbox is selected).

CCW: Specifies logic level input (0 or 1). When CCW is true (1), the control sequence moves the motor in a counter-clockwise (CCW) direction. When CCW is false (0), the control sequence moves the motor in a clockwise (CW) direction.

a Phase: Connects to “a” phase input of stepper motor block. Units are in volts.

b Phase: Connects to “b” phase input of stepper motor block. Units are in volts.

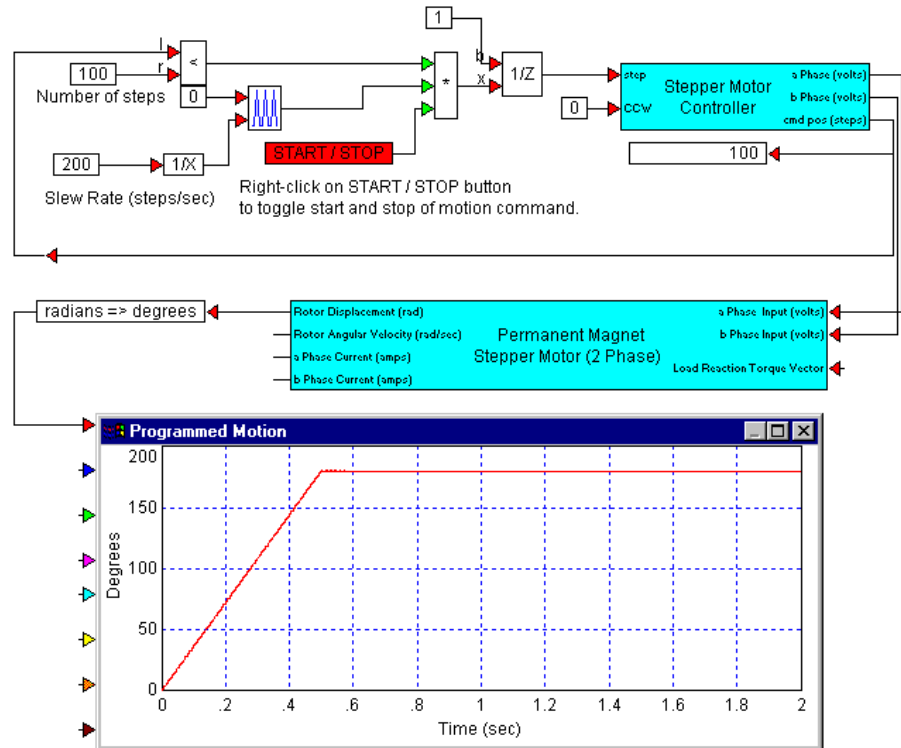
cmd pos: This output provides the current commanded step position. Units are in steps.

Example

Programmed positioning at specified slew rate

(...\MOTION\EXAMPLES\STEPPER_MOTOR_CONTROLLER_EXMPL.VSM)

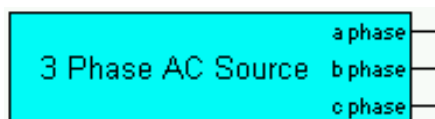
This example illustrates a typical application where a processor commands a step motor to increment to a specified step count at a specified slew rate.



Here the counter (cmd pos) output is used to compare against the set value of 100 steps. When the counter reaches the 100th step, the conditional < block output becomes zero. This closes the gate (multiplier block) for any further pulses issued by the pulseTrain block. The Time Between Pulses parameter for the pulseTrain block determines slew rate, and the operation is executed manually by clicking the Start button. The unitDelay block, used after the multiplier, allows perfect matching of counter to targeted step position, as well as preventing an algebraic loop. In this example, the stepper motor was specified having 50 rotor teeth making the full step 1.8°, and the controller was set for half stepping (0.9°) Thus total angular displacement is 90°.

The advantage of open-loop control, as shown by this example, is that it is inexpensive; that is, it does not require sensors for feedback. The disadvantage is that positioning errors can occur if external shaft torque exceeds designed limits, causing the motor to lose steps. Open-loop systems may sometimes require periodic homing maneuvers that check actual shaft position against hard stops, limit switches, or sensors. Such systems and devices can also be simulated using VisSim blocks.

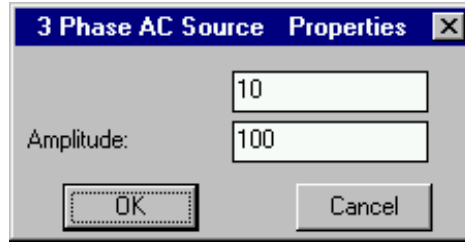
Three Phase AC Source



Example Model: ... \MOTION\EXAMPLES\3P_AC_SOURCE_EXMPL.VSM

Block Category: Sources

The Three Phase AC Source block provides a 3-phase signal that can be used as either a current or voltage source for driving asynchronous motors. As outputs, phase a leads phase b, and phase b leads phase c. All phases are separated 120°.



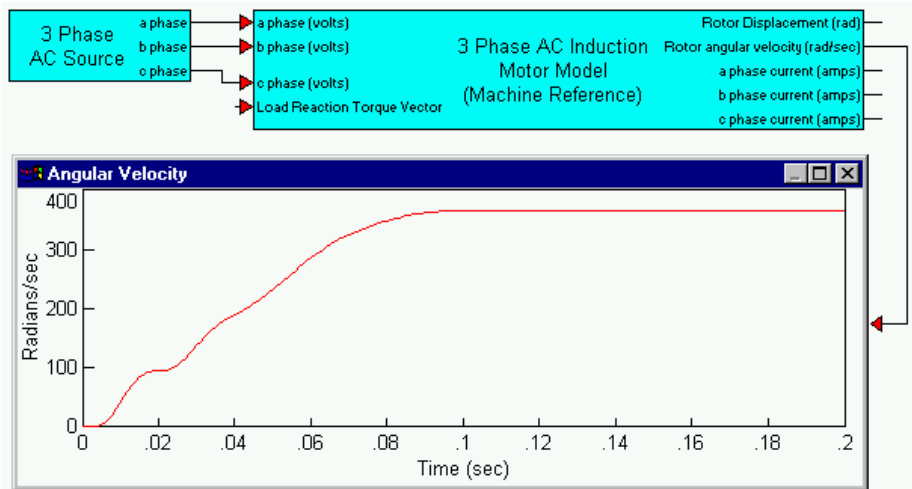
Frequency: Represents the signal frequency in hertz.

Amplitude: Represents the signal amplitude (arbitrary units).

Example

Open loop control of a 3-phase AC induction motor measuring slip (...\MOTION\EXAMPLES\3P_AC_SOURCE_EXMPL.VSM)

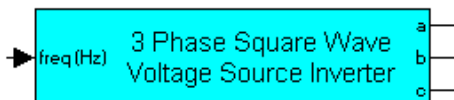
Driven by a 155 VRMS, 60 Hz 3 phase source (220 V peak amplitude), the 3-phase AC induction motor rotor angular velocity without load connected is driven to 371 rad/sec steady state.



Field rate is $2\pi \cdot 60 = 377$ rad/sec. The slip is calculated by:

$$s = \frac{377 - 371}{377} = 0.016$$

Three Phase Square Wave Inverter

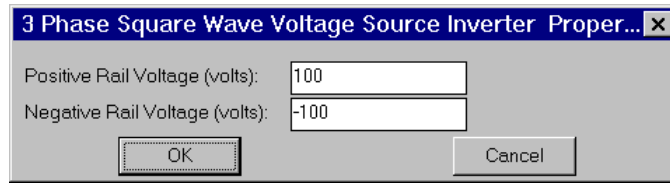


Example Model:

...\MOTION\EXAMPLES\THREE_PHASE_SQUARE_WAVE_INVERTER_EXMPL.VSM

Block Category: Sources

The Three Phase Square Wave Inverter block provides bipolar 3-phase square wave output from an input frequency.



Positive Rail Voltages: Defines the upper voltage rails for inverter supply

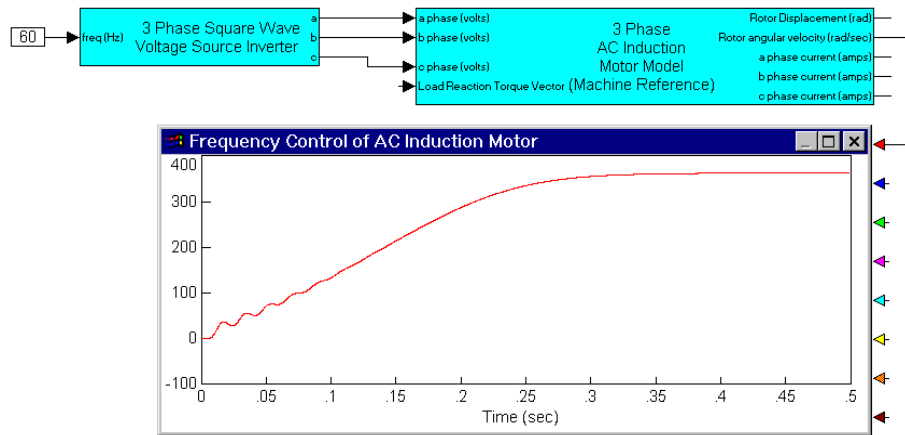
Negative Rail Voltages: Defines the lower voltage rails for inverter supply.

Example

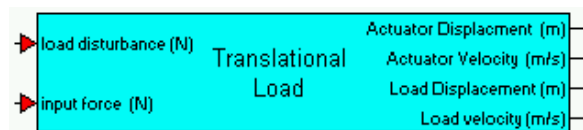
Frequency control of a 3-phase AC induction motor

(...\MOTION\EXAMPLES\THREE_PHASE_SQUARE_WAVE_INVERTER_EXMPL.VSM)

The following illustrates a simple example of frequency control of a 3-phase AC induction motor. Typically, the stator voltage is adjusted in proportion to frequency to maintain a constant flux. This provides easier adjustment of speed.



Translational Load



Example Model: ...MOTION\EXAMPLES\TRANSLATIONAL_LOAD_EXMPL.VSM

Block Category: Loads

The Translational Load block models the dynamic and static behavior of single degree of freedom translational rigid body dynamics of two bodies connected by a linkage mechanism. The linkage mechanism is generic in the sense that it can represent a lever arm or similar device.

This block allows separate specification of load and actuator moving mass properties. Load dynamical properties are reflected back into actuator properties using the linkage ratio. The Translational Load block includes hysteresis, Coulomb friction, viscous friction, static friction, and mass, linear spring, spring preload, and hard stops. All of these properties can be used simultaneously, or by choosing a 0

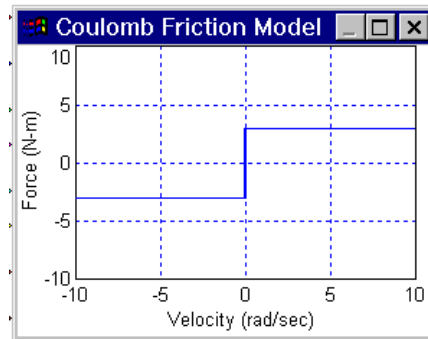
value for components can eliminate those components. A disturbance force input (load side) is provided, and can be used to link the Translational Load block with external forces that may be applied from other system components, such as bearing noise and wind gust model. Note that this input is coupled in through the load.

Parameter	Value
Spring Preload Force (N):	0
Actuator Mass (Kg):	.1
Actuator Viscous Damping Factor (N/m/s):	.01
Linkage Ratio, Actuator/Load:	.5
Load Mass (Kg):	.3
Load Coulomb Friction Factor (N):	.1
Lower Stop Limit (m):	-4
Load Viscous Damping Factor (N/m/s):	1
Upper Stop Limit (m):	4
Actuator Spring Constant (N/m):	2
Load Spring Constant (N/m):	.2
Backlash (m):	.1
Stiction Factor (N):	.01
Actuator Coulomb Friction Factor (N):	.01

Linkage Ratio Actuator/Load: Defines the mechanical advantage factor between the actuating device and load. Enter 1 for direct coupling between actuator and load; enter a number less than 1 to multiply force; and enter a number larger than 1 to multiply speed.

Spring Preload Force: Indicates the initial restoring (bias) force introduced by load spring. Enter 0 if there is no preload in the spring.

Load Coulomb Friction Factor: Indicates the constant load directional sensitive force (Coulomb Model) in N. Enter 0 for no Coulomb loading.



Actuator Mass: Indicates the actuator moving mass in kg.

Lower Stop Limit: Indicates the minimum hard stop limit in m.

Upper Stop Limit: Indicates the maximum hard stop limit in m.

Actuator Viscous Damping factor: Indicates the proportional factor of the actuator that relates the velocity to the dissipative force. Enter 0 for no viscous damping.

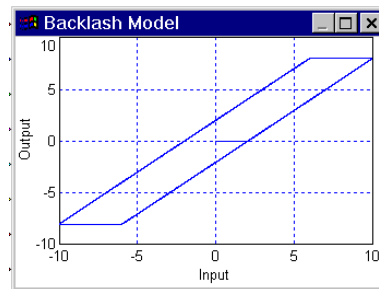
Load Mass: Indicates the load moving mass in kg.

Load Viscous Damping factor: Indicates the proportional load factor that relates the velocity to the dissipative force. Enter 0 for no viscous damping.

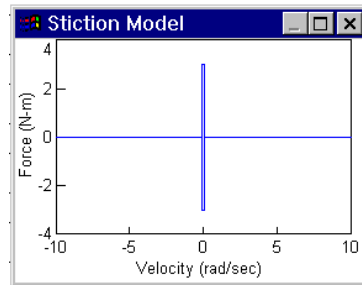
Actuator Spring Constant: Indicates the proportional spring factor that relates the position of the actuator moving mass to the restoring force. Enter 0 (zero) if there is no spring in the load.

Load Spring Constant: Proportional spring factor that relates the position of the load to the restoring force. Enter 0 if there is no spring in the load.

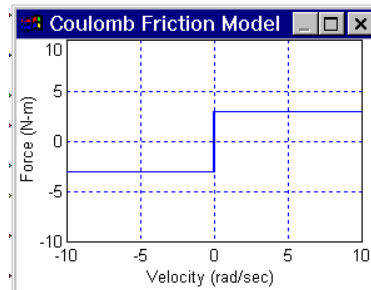
Backlash: Introduces hysteretic backlash in linkage mechanism in units of m. Enter 0 for no backlash.



Stiction Factor: Sets the 0 speed, static friction force of the actuator in N.



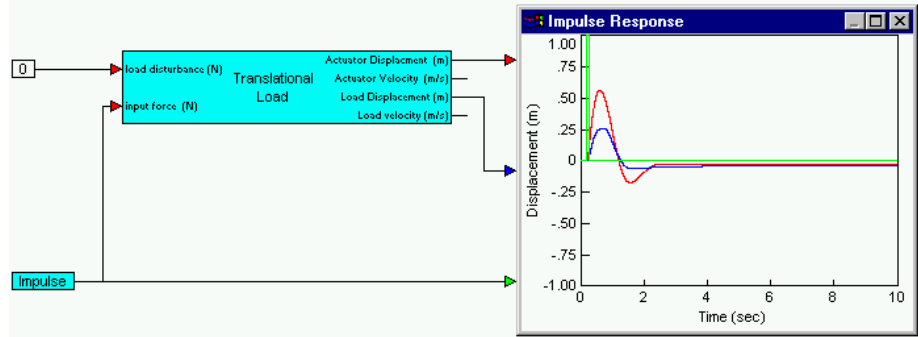
Actuator Coulomb Friction Factor: Indicates the constant directional sensitive force (Coulomb Model) in N. Enter 0 for no Coulomb loading.



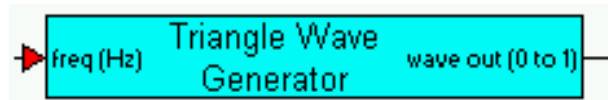
Example

Impulse response of translational masses coupled by a gear box
(...\MOTION\EXAMPLES\TRANSLATIONAL_LOAD_EXMPL.VSM)

The example shown below is the impulse response of translational masses connected by a gear of ratio of 1:2 with backlash. Each mass is suspended by springs and bearing surfaces that are modeled by a combination of Coulomb, stiction and viscous friction forces. At 0.2 sec a 10 N impulsive force is exerted on the actuator mass and the assembly is allowed to oscillate freely under its own suspension. The topping off of the geared mass is caused by the backlash. Rather than settling to the 0 equilibrium position, both stiction and backlash cause a steady-state offset in the position of the masses.



Triangle Wave Generator



Example Model: ...MOTION\EXAMPLES\TRIANGLE_WAVE_GENERATOR_EXMPL.VSM

Block Category: Sources

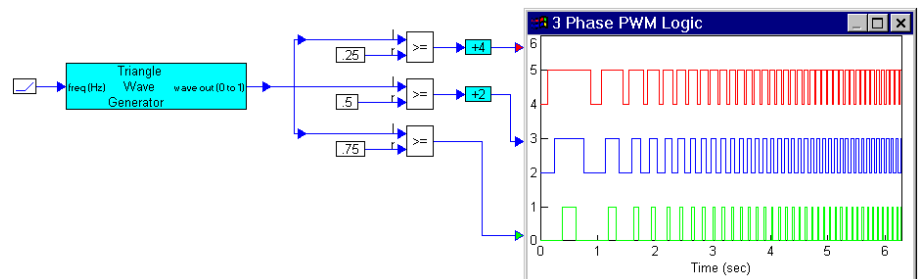
The Triangle Wave Generator block provides a symmetric triangular wave signal of amplitude 1 from a specified input frequency. This block is generally used as a basic element in motion control applications for generating various pulse-modulation schemes for motor drives.

Example

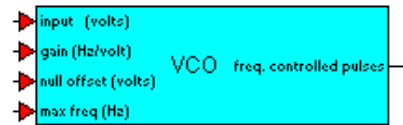
Three phase pulse width modulation logic

(...MOTION\EXAMPLES\TRIANGLE_WAVE_GENERATOR_EXMPL.VSM)

In this example, three comparators are used to create logic states for 3-phase PWM from the triangle wave generator. A null offset of 1 Hz was used in this example.



VCO



Example Model: ...MOTION\EXAMPLES\VCO_EXMPL.VSM

Block Category: Tools

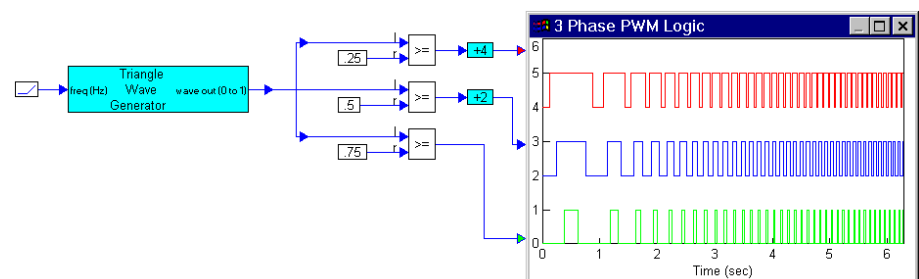
The VCO block models a programmable voltage controlled oscillator. It produces a series of clock pulses whose frequency is controlled by the input voltage. The VCO block has four inputs:

- **Input:** Input in volts. The output pulse width and frequency are determined by the magnitude of this input. This value is expected to be a positive non-zero quantity.
- **Gain (Hz/volt):** The oscillator frequency gain. The frequency of the output will be the product of the input voltage and the oscillator frequency gain, as long as it does not exceed the maximum frequency.
- **Null Offset:** The minimum value the input voltage must reach for the VCO to produce output. In other words, the VCO will ignore all input values that are less than the null offset value.
- **Max Freq:** Defines the maximum output frequency of the VCO. This will set the upper limit on the raw output frequency, which is the product of the input voltage and the gain.

Example

Closed-loop positioning of a stepper motor (...MOTION\EXAMPLES\VCO_EXMPL.VSM)

For a detailed description, see the example under the Microstep Controller block. This example is a replica of the Microstep Controller block example, except that the output of the VCO is plotted in a plot block.



Index

A

AC Induction Motor (DQ) 25
AC Induction Motor (Machine Reference) 29
AC induction motor application: power train drive system 11
AC Induction Motor Current Model (FOC) 27
AC induction motor tutorial: speed control of a machine tool lathe 5

B

Basic (Permanent Magnet) DC Motor 31
Block Reference 25
Brushless DC (BLDC/PMSM) Motor 33
Brushless DC (BLDC/PMSM) motor tutorial: target tracking system 12

C

Clarke Transform 35
Commutator (Six Step) 36
Configuring the PID compensator 10
Customizing the Volts/Hz Controller block 9

D

Designing the volts/frequency controller for the motor 8
Discrete Integrator 37

F

Field Orientated Controller (FOC) 38
Final configuration requirements 22
Frequency Demodulator 41

H

Hall Sensor 42

I

Interactive webinars 4
Introduction 1
Inverse Clarke Transform 43
Inverse Park Transform 44

L

Linear Encoder 45
Low Pass Filter 46
LVDT 47

M

Microstep Controller 49
Motor Specifications 12

O

Other applications 24
Other things you can do with VisSim/Motion 24

P

Park Transform 51
Park Transform (Stationary Frame) 52
Permanent Magnet Stepper Motor (2 Phase) 53
PID Controller (Digital) 54
PID Controller (Ideal) 56
PWM (Dual Phase) 69
PWM (Single Phase) 70
PWM (Space Vector) 71
PWM Brush Motor Amplifier (2Q) 60
PWM Brush Motor Amplifier (2Q, Current Feedback) 63
PWM Brush Motor Amplifier (4Q) 61
PWM Brush Motor Amplifier (4Q, Current Feedback) 65
PWM Brush Servo Amplifier 58
PWM Brushless Servo Amplifier 67

R

Rate Estimation (Rooftop) Filter 72
Resources for learning VisSim/Motion 4
Rotary Encoder 73
Rotary Position Sensor 74
Rotary Servo Potentiometer 75
Rotary Tachometer 76
Rotational Load 77

S

Sample diagrams 4
Setting parameter values 6
Setting the simulation properties 21

Setting up the motor, load, and encoder 6
Setting up the plot blocks 19
Simulation development 13
Simulation results 23
Stepper Motor Controller 80

T

The VisSim product family 2
Three Phase AC Source 81
Three Phase Square Wave Inverter 82
Training 4
Translational Load 83
Triangle Wave Generator 86

U

Using the Tutorials 5

V

VCO 87

W

What is VisSim/Motion 1
Wiring Volts/Hz Controller to the overall simulation 10