AN1770

Using the Microchip Motor Model Library for Simulink®

Author: Jason Sachs
Microchip Technology Inc.

OVERVIEW

Microchip's Motor Model Library is a set of components ("blocks") that can be used with the Simulink® simulation tool from MathWorks®. These blocks are intended to allow engineers to model a system with Permanent Magnet Synchronous Motors (PMSMs) under closed-loop control of an embedded microcontroller, such as Microchip's dsPIC® Digital Signal Controllers (DSCs). Model-based engineering can be used to assist in the development, testing and understanding of motor control systems. Simulation can be used to confirm the behavior of real systems, and can also be used to explore system behavior under conditions that are impossible or cumbersome to test with a real system.

The Microchip Motor Model Library is provided for use within the MATLAB® and Simulink environment. This library contains a continuous-time simulation model of a Permanent Magnet Synchronous Motor (PMSM). The PMSM model is designed to help engineers understand PMSM dynamics through simulation, and to facilitate developing control loops for these motors.

This library was developed in collaboration with the Linz Center of Mechatronics (LCM) in Linz, Austria.

MOTOR MODEL LIBRARY SETUP

General Requirements

MATLAB and Simulink can run on the latest versions of Microsoft® Windows®, Apple® Mac OS® X and Linux® operating systems. Refer to the **"References"** section for the latest system requirements.

Using this library requires MATLAB and Simulink, R2012b or later. No other toolboxes are required.

We recommend that users have some familiarity with MATLAB and Simulink prior to using the Motor Model Library. The MathWorks, Inc. web site provides information and tutorials to help get started. (See the **"References"** section.)

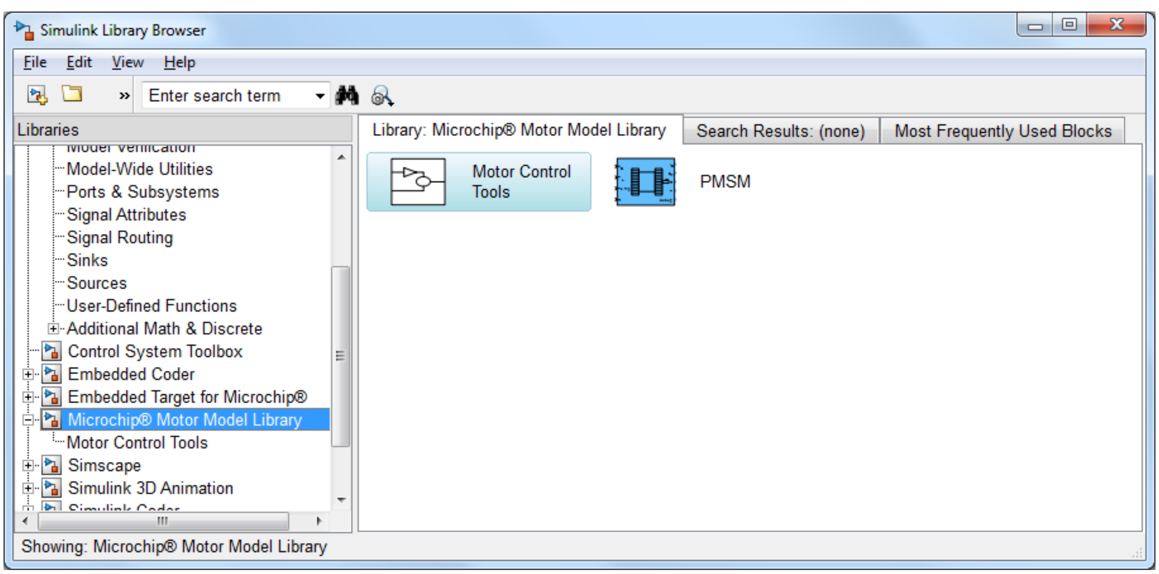
Model Setup and Initialization

LIBRARY INSTALLATION AND CONTENT

The motor modeling package consists of an archive file, `MicrochipMotorModelLibrary.zip`, which contains the Simulink library file and some example files.

To use, unzip this package into a directory of your choice and add this directory to the MATLAB search path. Once you have done this, the Motor Model Library (Figure 1) will appear in the Simulink Library Browser the next time you start MATLAB.

FIGURE 1: SIMULINK® LIBRARY BROWSER WITH THE MOTOR MODEL LIBRARY



The library includes one motor model block (PMSM for the basic linear Permanent Magnet Synchronous Motor model), as well as some auxiliary blocks for reference transforms. Further details are given later in this application note.

To use one of these blocks in a new Simulink model, click on the desired block to select it, then right-click and select "Add to a new model (Ctrl-I)".

To use one of these blocks in an existing Simulink model, click on the desired block to select it, then drag it into the model.

LIBRARY BLOCKS

The Motor Model Library contains the following blocks (see Table 1):

TABLE 1: MOTOR MODEL LIBRARY BLOCKS

Block Name	Description
PMSM	Permanent Magnet Synchronous Motor, basic linear model.
abc-to-alphabeta	Clarke Transform: Convert from three-phase measurements (abc-coordinates) to the two-element, orthogonal stationary reference frame (αβ-coordinates).
abc-to-alphabeta0	Clarke Transform: Convert from three-phase measurements (abc-coordinates) to the three-element, orthogonal stationary reference frame (αβ0-coordinates).
alphabeta-to-abc	Clarke Transform: Convert from the two-element, orthogonal stationary reference frame (αβ-coordinates) to three-phase measurements (abc-coordinates). This is sometimes called the Inverse Clarke Transform. (The Park and Clarke Transforms were originally used for analysis only, to transform measured phase voltages and currents into the non-measurable synchronous reference frame. With the advent of field-oriented motor control, the inverse transforms are now used to convert from the synchronous reference frame back to per-phase quantities.)
alphabeta-to-dq	Park Transform: Convert from the two-element, orthogonal stationary reference frame (*αβ*-coordinates) to a rotating reference frame (*dq*-coordinates).
alphabeta0-to-abc	Clarke Transform: Convert from the three-element, orthogonal stationary reference frame (*αβ0*-coordinates) to three-phase measurements (*abc*-coordinates).
cstheta	Given an input angle *θ* (theta), computes a two-element vector [*cosθ sinθ*] for use with Park Transform blocks.
dq-to-alphabeta	Park Transform: Convert from a rotating reference frame (*dq*-coordinates) to the two-element, orthogonal stationary reference frame (*αβ*-coordinates). This block is sometimes called the Inverse Park Transform.
vector mixer	(Also known as cross product.) Given two inputs, A and B, each of which is a two-element vector, treat the inputs as complex numbers and compute the product, with optional conjugation. This is used in the Park Transform blocks for vector rotation and may also be used for quadrature modulation/demodulation in Phase-Locked Loops (PLLs).

INITIALIZING LIBRARY PARAMETERS

The PMSM block in the library has mask parameters which need to be set appropriately. These parameters are described in Table 2.

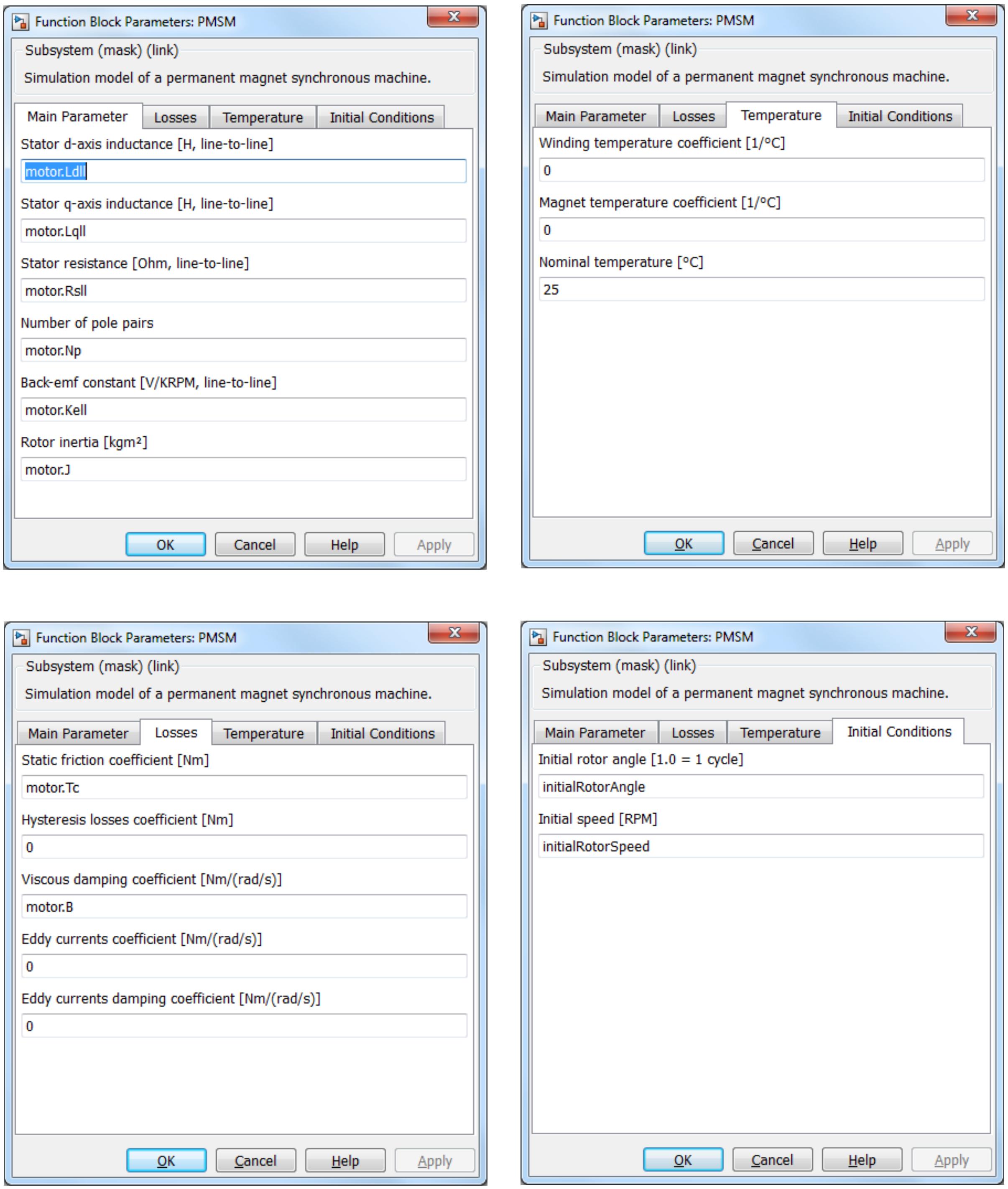
To enter these parameters, double-click on the PMSM block to open the parameters dialog (Figure 2).

TABLE 2: LIBRARY BLOCK PARAMETERS

Parameter	Description	Units	Comments
Main Motor Parameters			
Ld	Stator d-axis inductance	H, line-line	
Lq	Stator q-axis inductance	H, line-line	
Rs	Stator resistance	Ω, line-line	
n_p	Number of pole pairs		
Ke	Back-emf constant	V/KRPM, line-line, zero-peak	If your motor's data sheet lists the back-emf constant in RMS, to convert to zero-peak, multiply by √2 (≈1.414)
J	Rotor inertia	$kg \cdot m^2$	
Losses			
cf	Static friction coefficient	N·m	
chy	Hysteresis losses coefficient	N·m	Empirically equivalent to cf(1)
d	Viscous damping coefficient	N·m/(rad/s)	
ced	Eddy current damping coefficient	N·m/(rad/s)	Empirically equivalent to d(1)
ded	Flux eddy current damping coefficient	N·m/(rad/s)	This parameter is reserved for future use; set to zero
Temperature Sensitivity			
alphaCU	Winding temperature coefficient	1/°C	
alphaPM	Magnet temperature coefficient	1/°C	
Temp_nom	Nominal temperature	°C	Normally between 20-25°C; this is the temperature at which other parameters are specified or measured
Initial Conditions			
theta_m0	Initial rotor angle	revolutions (1.0 = 1 cycle)	
omega_m0	Initial speed	RPM	

Note 1: The static and viscous friction terms each have components from mechanical and electrical causes. If you are determining these parameters using torque and speed measurements from a sample motor, the mechanical and electrical components are essentially indistinguishable. In this case, we recommend using only cf and d, and setting the other two parameters to zero.

FIGURE 2: DIALOGS FOR CONFIGURING LIBRARY BLOCK PARAMETERS



When the motor parameters are likely to change, using symbolic variables wherever possible, rather than fixed numeric constants, is recommended. These must be defined in the workspace visible to Simulink; in most cases, this is the same top-level workspace visible to the MATLAB Command Window. This allows you to change the values of these parameters from outside the model and place them into scripts (files with the extension, `.m`), which can be version-controlled as necessary. If you use numbers directly in the Simulink model, you won't be able to retain changes without re-saving the model, even though the model's structure may stay exactly the same. Keeping parameters separate allows you to maintain changes in model parameters separately from changes in model structure.

An example of this can be found in the file, `examples/motorAC300020.m` (Example 1). These parameters are obtained from the motor's data sheet.

EXAMPLE 1: SAMPLE MOTOR PARAMETER FILE

```
function motor = motorAC300020()

% MicrochipDirect AC300020
motor.manufacturer = 'Hurst';
motor.manufacturerPartNumber = 'DMB0224C10002';
motor.href = 'http://ww1.microchip.com/downloads/en/DeviceDoc/Hurst%20Motor%20DataSheet.pdf';
% unit conversion factors
KRPM_rads = 0.060/2/pi;         % KRPM/(rad/s):
                                % 2pi rad/s = 1Hz = 60RPM = 0.060 KRPM
Nm_ozin = 7.061552e-3;          % N*m/(oz*in):
                                % see NIST SP811 for reputable reference

% Electrical specifications
motor.Rsll = 4.03;              % line-line resistance, ohms
motor.Lqll = 4.60e-3;           % line-line q-axis inductance, H
motor.Ldll = motor.Lqll;        % non-salient rotor so Ld=Lq
motor.Kell = 7.24;              % 7.24V/KRPM 0-peak, line-line
motor.Np = 5;                   % number of pole pairs

% Mechanical specifications
motor.J = 0.000628 * Nm_ozin;   % 0.000628 oz*in*s^2
                                % convert to N*m*s^2
% Not given by the data sheet:
motor.B = 0;                    % damping coefficient, N*m/(rad/s)
motor.Tc = 0;                   % coulomb friction, N*m
```

INPUTS AND OUTPUTS

The PMSM block has a limited number of inputs and outputs, as listed in Table 3. A selection of internal signals is also made available for debugging purposes.

TABLE 3: LIBRARY BLOCK INPUTS AND OUTPUTS

Parameter	Description	Units	Comments
		Inputs	
u_s	Stator Terminal Voltage Vector	V	This is a three-element vector, one element for each of the three motor terminals, ABC.
T_L	Load Torque	N·m	Positive load torque opposes positive rotation. If the motor produces no electromagnetic torque, a positive load torque will accelerate the motor towards a negative angular velocity.
Temp	Stator Winding and Magnet Temperatures	°C	This is a two-element vector; Temp(1) = Winding Temperature and Temp(2) = Magnet Temperature. Used to determine back-emf and winding resistance using the temperature coefficients in the motor model parameters.
Ω	Rotor Angular Velocity	rad/s	An input to the motor; in many cases, the angular velocity output of this block will be fed back in as this input.
		Outputs	
i_s	Stator Terminal Current Vector	A	This is a three-element vector, one element for each of the three motor terminals, ABC. Positive current flows into the stator terminals.
θ_m	Mechanical Rotor Angle	rad	Integral of the output rotor angular velocity.
Ω	Output Rotor Angular Velocity	rad/s	The PMSM model contains a mechanical model of the rotor inertia. If the mechanical model is used, this should be fed back into the rotor angular velocity input; otherwise, it can be ignored.
T	Motor Torque	N·m	Total torque produced by the motor. Includes electro-magnetic torque, hysteresis and viscous drag torques. Does not include load torque.
		Debug (Simulink® bus with internal signals)	
debug.idq	Synchronous Frame Stator Current Vector	A	This current and the terminal currents, i_s, are related using the Park and Clarke Transforms. The *q*-axis component (2nd element) of the vector produces torque; the *d*-axis alters rotor air gap flux.
debug.udq	Synchronous Frame Stator Voltage Vector	V	This voltage and the terminal voltages are related using the Park and Clarke Transforms.
debug.psidq	Synchronous Frame Stator Flux Vector	V·s	Stator flux, $\psi_{dq} = \psi_m + L_{dq}I_{dq}$, where ψ_m is the permanent magnet flux and L_{dq} is a diagonal inductance matrix.
debug.T_e	Electromagnetic Torque	N·m	Torque produced by the motor due to the cross product of stator flux and stator current.
debug.theta_r	Rotor Electrical Angle	rad	The electrical angle, $\theta_r = N_p \int w_m dt$.
debug.omega_r	Rotor Electrical Frequency	rad/s	The electrical angle, $\omega_r = N_p \omega_m$, where ω_m is the input rotor angular velocity, Ω.

SIMULATION SETUP AND USAGE EXAMPLE

Inside the library directory is a subdirectory, called “examples”, which contains one Simulink model (`FOCopenloop.slx`) and various setup files (Figure 3).

Open this directory in MATLAB, and at the Command Window prompt type the following:

```
>> FOCopenloop
>> FOCopenloop_setup1
```

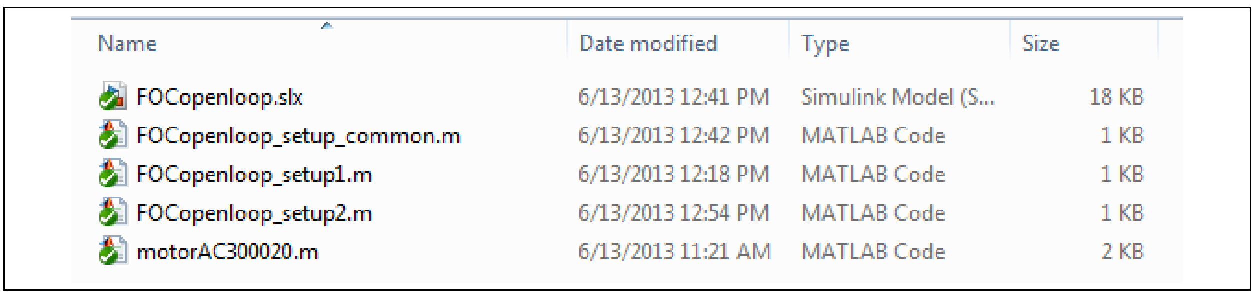
This opens a window with the FOCopenloop model (Figure 4).

The FOCopenloop model represents an open-loop, field-oriented controller that applies stator voltages in the synchronous (*dq*) frame of the motor using perfect information of the motor’s electrical angle. (A real motor would have a position sensor, such as an encoder or Hall sensor, or use a sensorless estimator. In a simulation we can use the exact electrical angle.) The speed can either be determined by the PMSM block itself (which contains a mechanical model) or by an externally imposed speed signal.

This example also shows the use of many of the reference transform blocks found in the Microchip Motor Model. These blocks transform coordinates between the *abc* reference frame (3-element vectors corresponding directly to each of the motor phases), the $\alpha\beta$ reference frame (2-element, orthogonal vector in the stationary frame) and the *dq* reference frame (2-element, orthogonal rotating reference frame). The Park Transform blocks, which transform between the stationary and rotating reference frames, require a 2-element vector of cosine and sine inputs, which can be obtained from the cstheta block. In the FOCopenloop example, the input to the cstheta block is the motor’s synchronous electrical angle, theta_r.

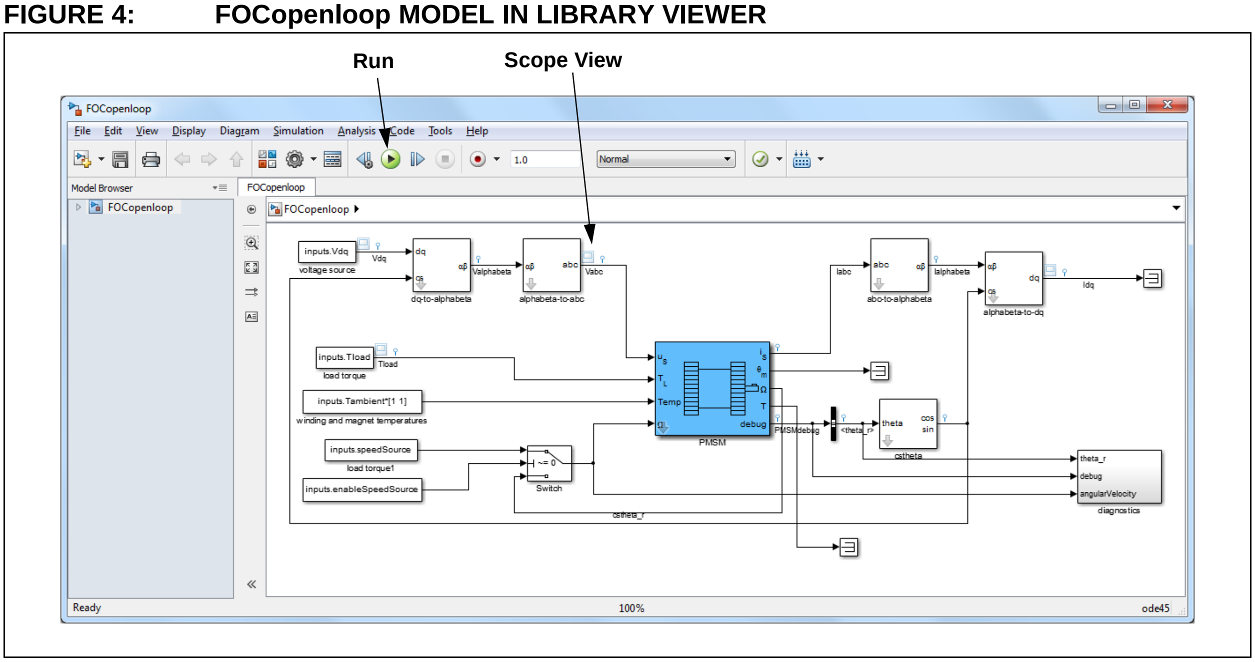
For a more complete description of the PMSM motor model, please see **Appendix A: “The Microchip PMSM Model”**.

FIGURE 3: EXAMPLES DIRECTORY



Name	Date modified	Type	Size
FOCopenloop.slx	6/13/2013 12:41 PM	Simulink Model (S...	18 KB
FOCopenloop_setup_common.m	6/13/2013 12:42 PM	MATLAB Code	1 KB
FOCopenloop_setup1.m	6/13/2013 12:18 PM	MATLAB Code	1 KB
FOCopenloop_setup2.m	6/13/2013 12:54 PM	MATLAB Code	1 KB
motorAC300020.m	6/13/2013 11:21 AM	MATLAB Code	2 KB

FIGURE 4: FOCopenloop MODEL IN LIBRARY VIEWER



The script, `FOCopenloop_setup1.m`, prepares the MATLAB workspace with motor parameters for the Hurst DMB0224C10002 motor (MicrochipDirect Part Number AC300020) and input signals for one particular simulation profile.

The `FOCopenloop_setup1` simulation profile applies a pair of ramps of *q*-axis voltage, with some momentary load pulses after each ramp is complete, and then applies a 3-phase short circuit to the motor. Motor speed is unregulated and decreases when load torque is applied. With open-loop voltage control, the current is not aligned with the back-emf: the *d*-axis component is non-zero and a large portion of the current capability of the voltage source is unnecessarily wasted.

After executing this setup script, press the Run button (see Figure 4). The simulation will run and update some simulation Scope Viewer graphs that will open along with the model. If for some reason, the Scope Viewer window is not displayed, double-click on one of the light blue Scope icons shown in the FOCopenloop model. Scope Viewer graphs are a feature in Simulink meant as a quick check to see the simulation results, but do not include many options to format these graphs. For full control of graphing style, export the data into MATLAB and plot it according to your requirements.

In the Scope Viewer window, clicking the Auto-Scale button (see Figure 5) will provide a combined view (Figure 6).

FIGURE 5: AUTO-SCALE ON TOOLBAR

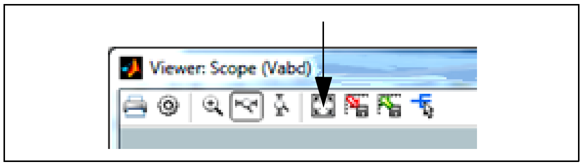
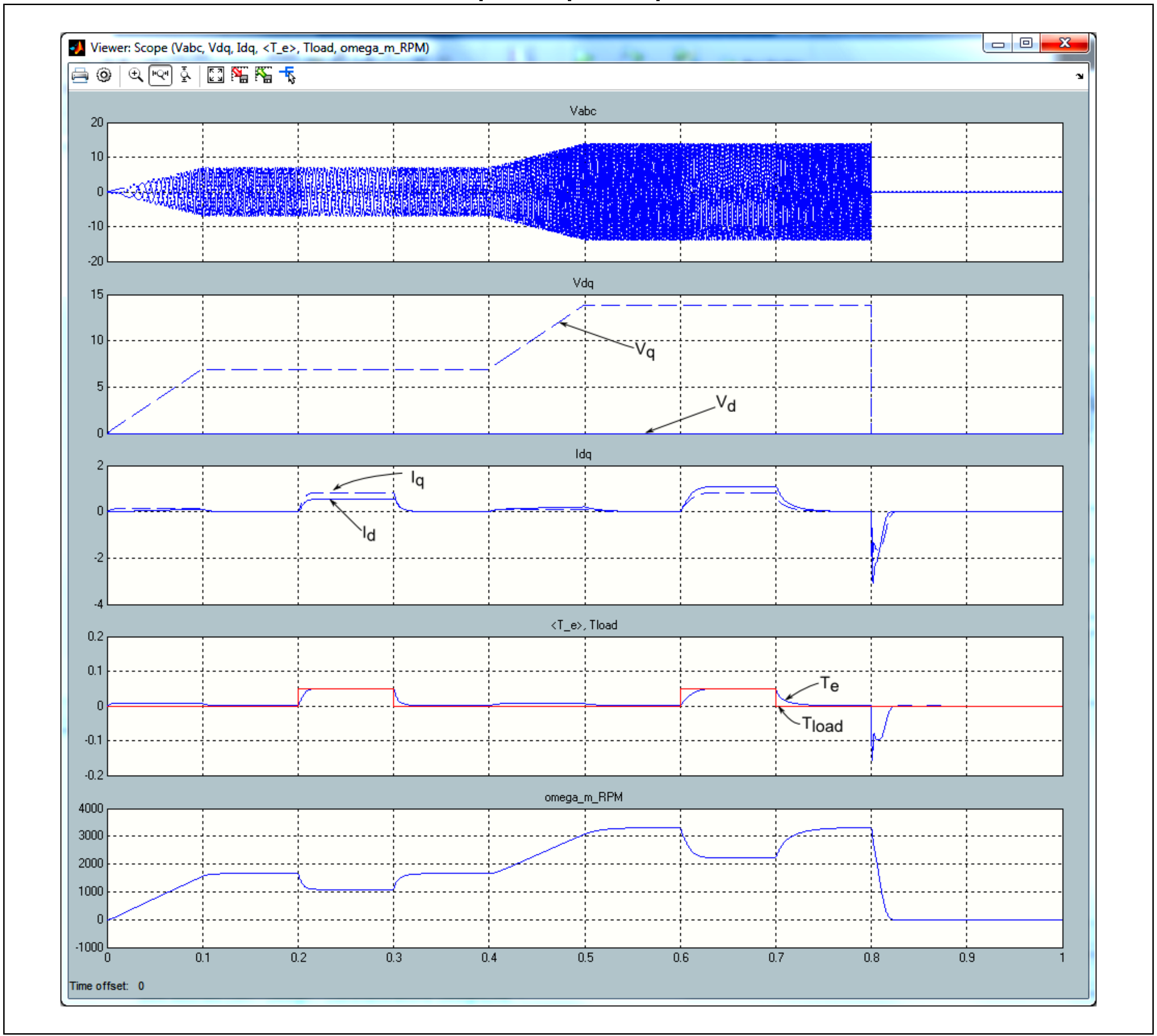


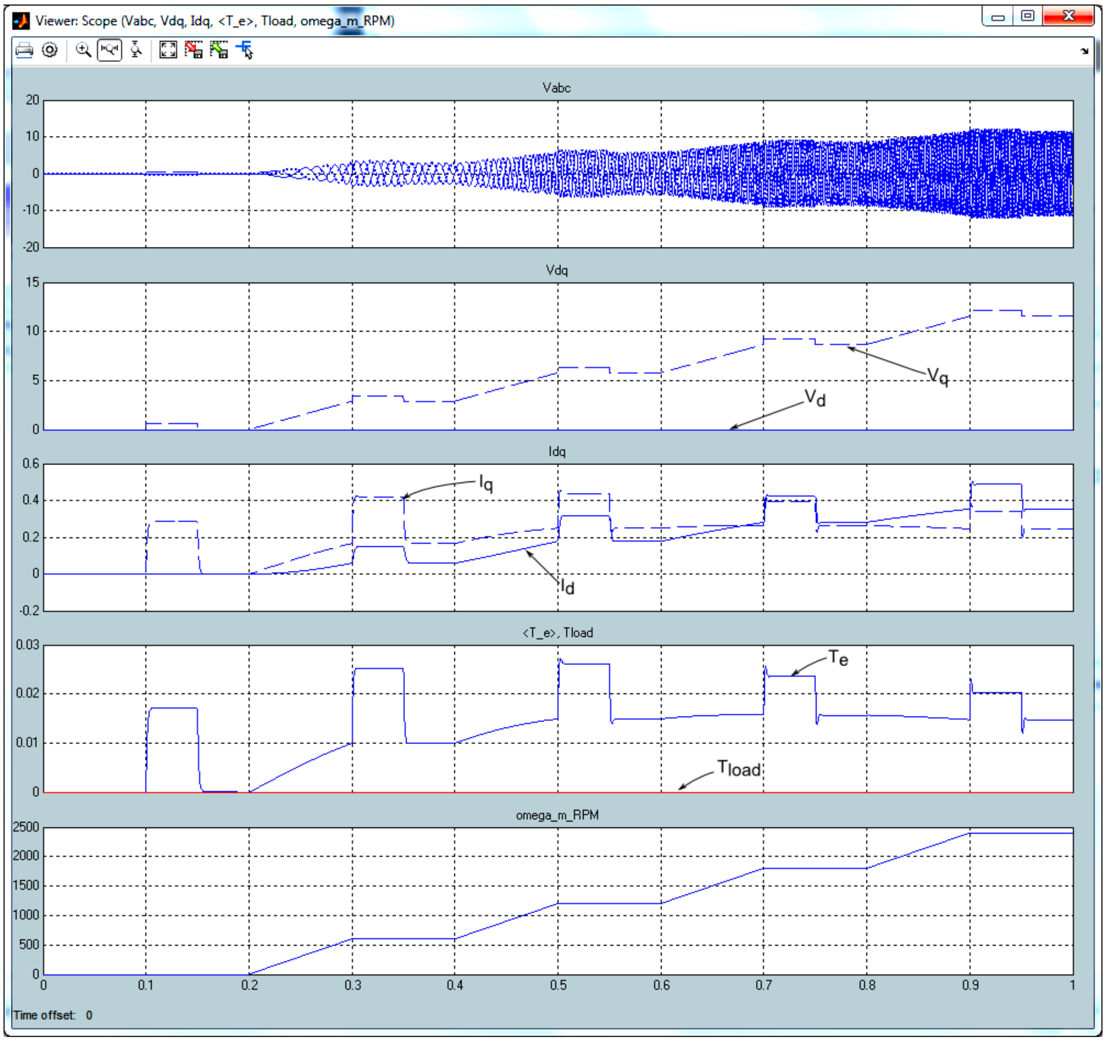
FIGURE 6: SCOPE VIEW FOR `FOCopenloop_setup1`



The other setup file, `FOCopenloop_setup2.m`, defines an externally imposed speed profile. This represents a motor attached to an engine or some other prime mover with large inertia, and is an example of viewing transient responses at various speeds. (This is a useful technique for tuning a current loop in a closed-loop current controller.) Load torque in this profile is zero. The speed in this profile undergoes several plateaus. The motor terminals are driven with a waveform that is synchronous to the motor's electrical frequency, using the electrical angle output of the motor model, and with a ramp that has similar plateaus to the speed waveform. It also has additional pulses of voltage at the beginning of each speed plateau (as shown in the graph of V_{dq} in Figure 7.)

Note that the transient response changes as the speed increases. This is due to increased cross-coupling between the *d*- and *q*-axes as speed increases, due to motor inductance. Figure 7 shows the combined and auto-scaled scope view for this script.

FIGURE 7: SCOPE VIEW FOR `FOCopenloop_setup2`



FREQUENTLY ASKED QUESTIONS

1. **What type of PMSM motors are covered by the model?**

   Any balanced 3-phase electric machine (motor or generator) with a wound stator, and a permanent magnet rotor, can be approximated by this linear PMSM model. These types of motors include most commercially produced motors marketed as "Brushless DC" or "Permanent Magnet Synchronous Motor". Stator and rotor saliency – the degree to which stator and rotor appear as nonuniform magnetic circuits that vary with angle around the rotor axis – are important characteristics to consider.

   The motor model can be used with both surface permanent magnet rotors (magnetically non-salient pole rotors, $L_d = L_q$) and interior permanent magnet rotors (magnetically salient pole rotors, $L_d \neq L_q$).

   While the effects of rotor saliency are included in this PMSM model, the effects of *stator* saliency are *not* included. Most commercial motors have stators which are designed to be as non-salient as possible, so the only unmodeled feature of stator saliency is a small amount of cogging torque, which shows up as a periodic torque disturbance, and is significant only at low rotational speeds.

   Unbalanced 3-phase electric machines (for example, a motor with unequal turns on each phase) are not supported by this model.

   Both star and delta wound electric machines with no exposed neutral connection are possible to use with this model. (Star wound electrical machines are also known as wye wound.) The electric machine is treated as a black box with three terminals. With a delta wound electric machine, there can be circulating currents, and there is no neutral connection, but from the terminals, it is mathematically identical to a star wound electric machine. The circulating currents are not visible at the terminals and a "virtual" neutral can be assumed with a voltage that is the mean value of the terminal voltages.

2. **Does the PMSM model include any nonlinear effects, like iron saturation or back-emf harmonics?**

   Not at the present time. In many cases, the basic linear model is sufficient.

3. **When do I use the internal mechanical model and when do I have to use my own mechanical model?**

   If your system can be modeled simply using an inertia, viscous damping and static friction, along with an externally imposed load torque, use the internal mechanical model, and configure the motor model parameters to use the system inertia/damping/static friction.

   If your system contains more complicated mechanical elements, it may be a better choice to create your own mechanical model in Simulink. In this case, ignore the output rotor angular velocity, and instead, provide the rotor angular velocity of your model as an input into the PMSM motor block.

4. **How do I simulate an open circuit on one or more of the motor terminals, or the effects of external inductance/resistance/capacitance?**

   The PMSM model is a voltage input, current output model. Like most Simulink blocks, it is based on unidirectional signal flow and not network models such as those found in Simscape™. This means that the motor expects 3 voltage sources and will output 3 current measurements. Directly modeling an open circuit with this model is not possible.

REFERENCES

The latest information on MATLAB and Simulink products is available at the web site of The MathWorks, Inc:

www.mathworks.com

For the most current system requirements to run MATLAB and Simulink:

http://www.mathworks.com/support/sysreq/current_release/index.html

For information on getting started with Simulink, please see:

http://www.mathworks.com/help/simulink/gettingstarted-with-simulink.html

For information on configuring the MATLAB search path, please see:

http://www.mathworks.com/help/matlab/matlab_env/what-is-the-matlab-search-path.html

ADDITIONAL TRADEMARKS

"MATLAB" and "Simulink" are registered trademarks of The MathWorks, Inc.

NOTES:

APPENDIX A: THE MICROCHIP PMSM MODEL

The Microchip Motor Model for PMSM used in the Simulink Library is based on a linear model of the motor. The model is hierarchical; you can explore it in Simulink by selecting the PMSM model block, then pressing Ctrl+U to look under the block mask.

The motor model has several components that operate on the synchronous frame. For this reason, the voltage is converted from the per-phase (*abc*) to synchronous (*dq*) reference frame using the Clarke and Park Transforms (Figure A-2).

The Simulink model of these transforms is just an implementation of the following equation:

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad \text{[A1]}$$

FIGURE A-1: OVERALL VIEW OF THE PMSM MODEL

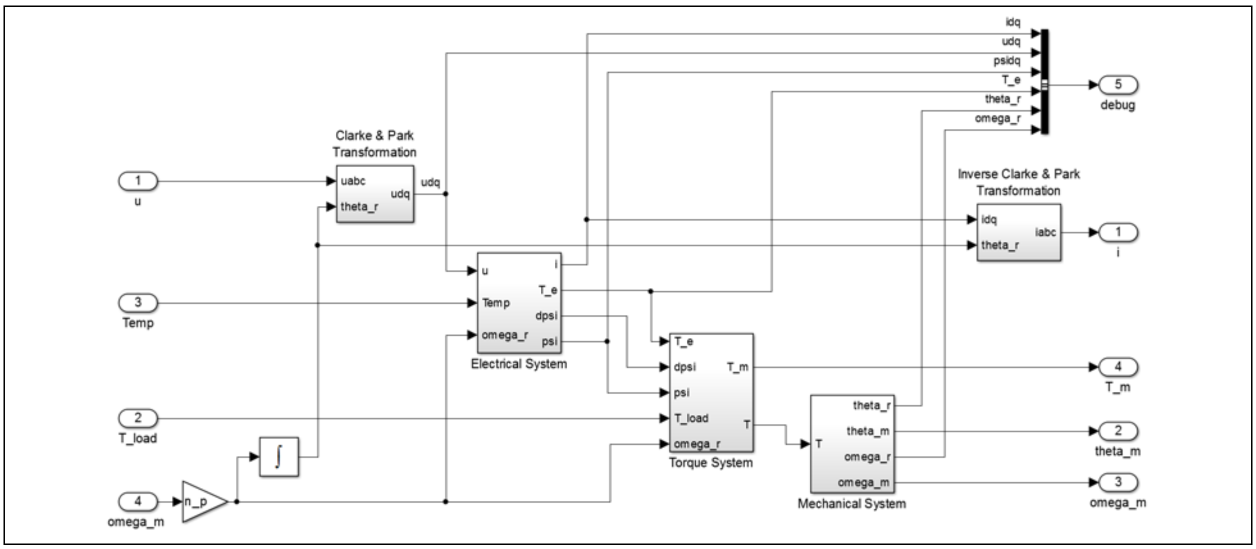
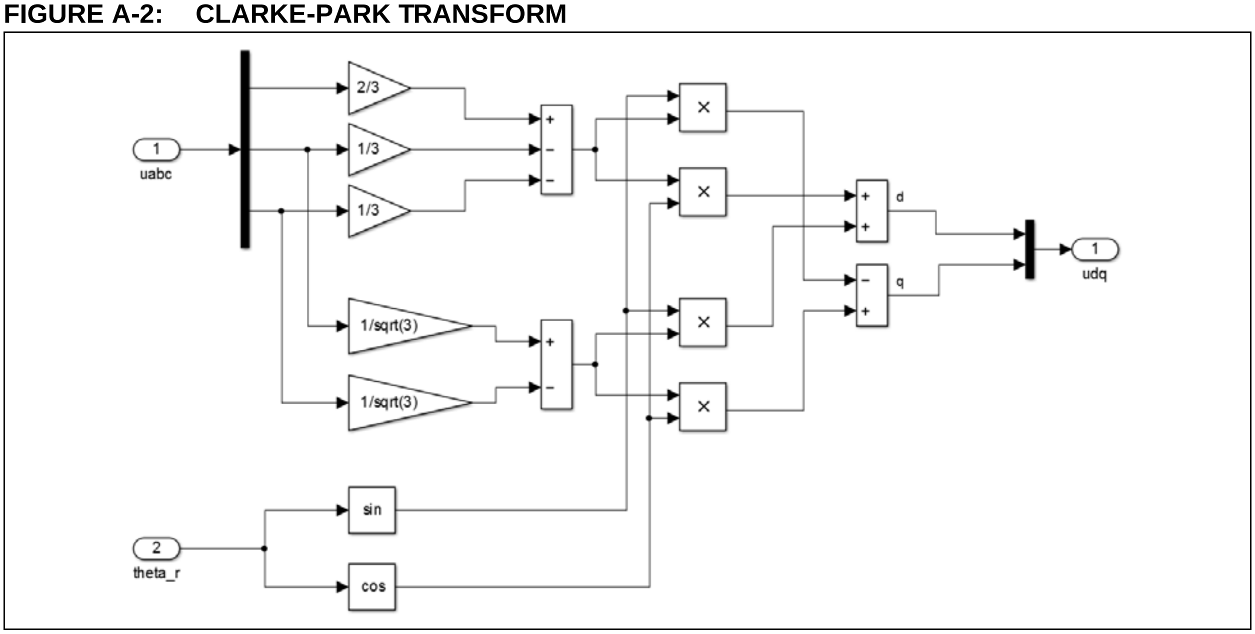


FIGURE A-2: CLARKE-PARK TRANSFORM



The electrical model of the motor (Figure A-3) determines stator current, stator flux and electromagnetic torque from voltage. This simulates the following equations

$$\frac{d}{dt}\begin{bmatrix}\psi_d\\ \psi_q\end{bmatrix} = \begin{bmatrix}u_d\\ u_q\end{bmatrix} - R_s\begin{bmatrix}i_d\\ i_q\end{bmatrix} - J\omega_r\begin{bmatrix}\psi_d\\ \psi_q\end{bmatrix} \quad \text{[A2]}$$

$$\begin{bmatrix}i_d\\ i_q\end{bmatrix} = \begin{bmatrix}\frac{1}{L_d} & 0\\ 0 & \frac{1}{L_d}\end{bmatrix} \cdot \begin{bmatrix}\psi_d\\ \psi_q\end{bmatrix} - \begin{bmatrix}\psi_{PM}\\ 0\end{bmatrix} \quad \text{[A3]}$$

for voltage and current, where

$$J = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}$$

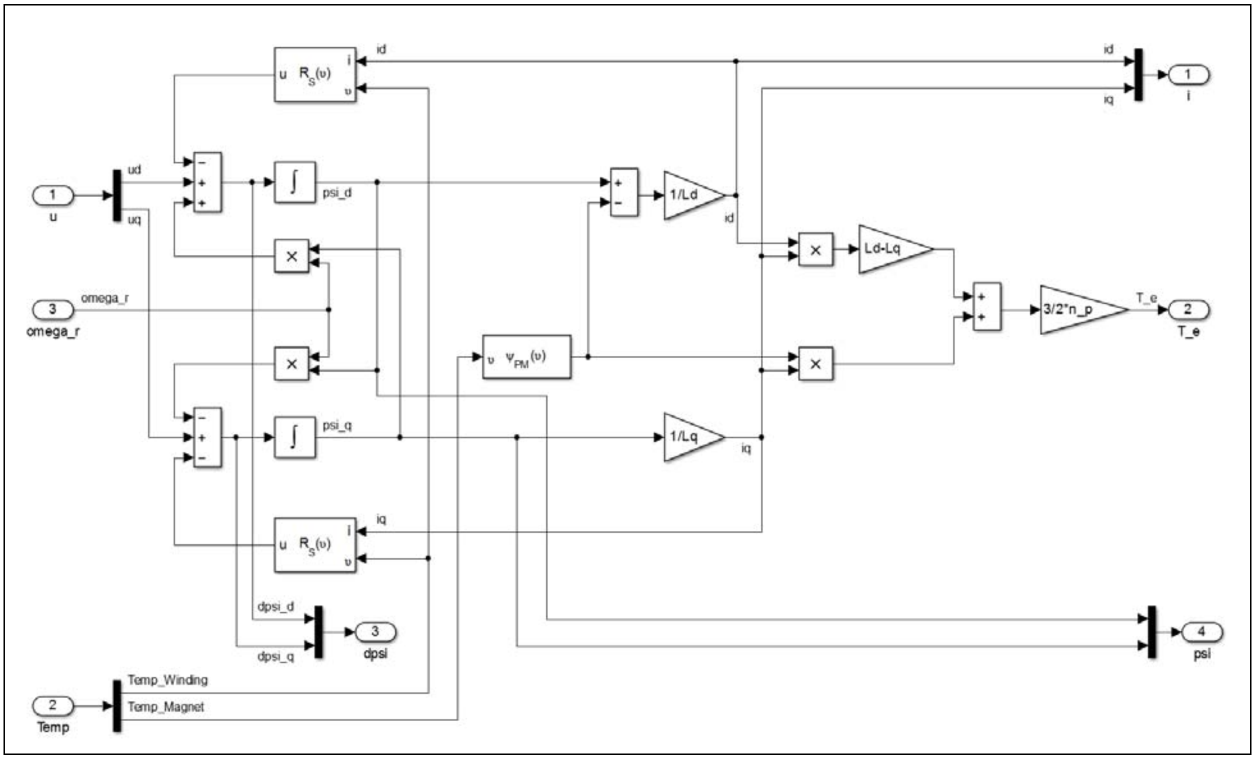
and

$$T_e = \frac{3}{2}n_p\psi_{PM}i_q + \frac{3}{2}n_p(L_d - L_q)i_d i_q \quad \text{[A4]}$$

(= alignment torque + reactive torque)

for torque, where R_s and ψ_{PM} are temperature dependent variables.

FIGURE A-3: ELECTRICAL MODEL



The torque model of the motor (Figure A-4) simulates the equations for internal motor torque and net torque:

$$T = T_m - T_{load} \quad \text{[A5]}$$

$$T_m = T_e - T_{friction} - T_{viscous} - T_{d\psi} \quad \text{[A6]}$$

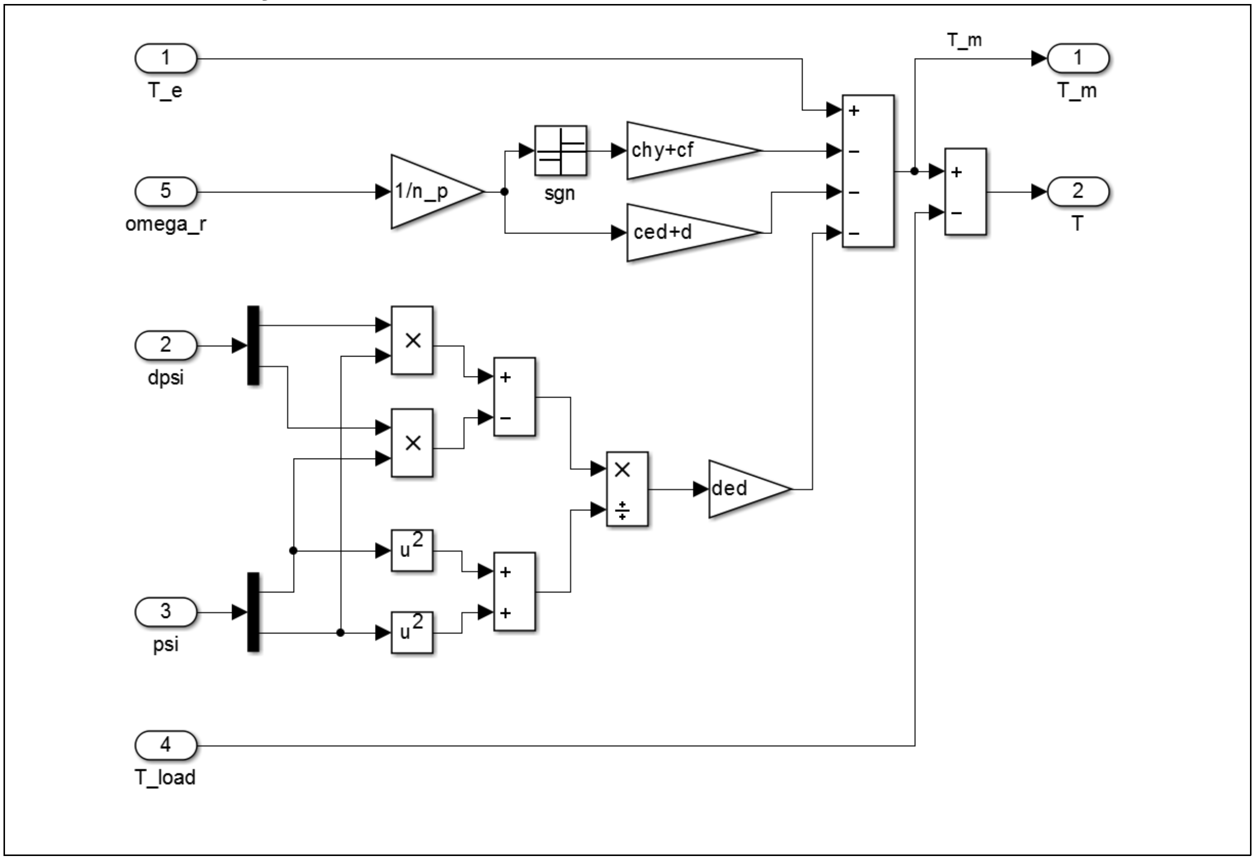
$$T_{friction} = (c_{hy} + c_f)\,\mathrm{sgn}\,\omega_m \quad \text{[A7]}$$

$$T_{viscous} = (c_{ed} + d)\omega_m \quad \text{[A8]}$$

$$T_{d\psi} = d_{ed}\frac{\frac{d\psi_{dq}}{dt} \times \psi_{dq}}{|\psi_{dq}|^2} \quad \text{[A9]}$$

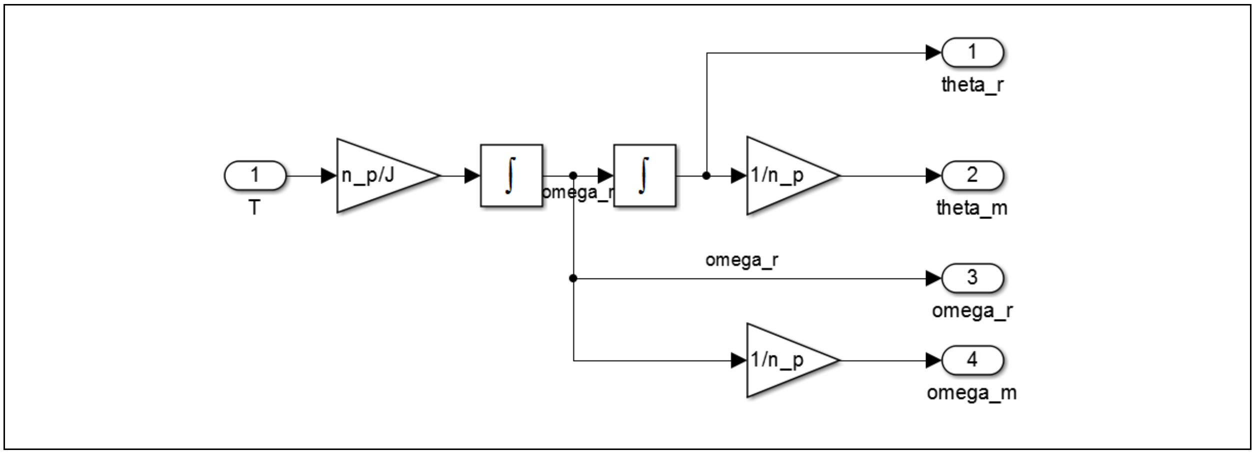
where $T_{friction}$ is the magnetic hysteresis and Coulomb friction, $T_{viscous}$, is eddy current and viscous drag, and $T_{d\psi}$ is drag resulting from time varying flux.

FIGURE A-4: TORQUE MODEL



The mechanical model of the load (Figure A-5) includes the effects of these torques on the motor inertia. As discussed earlier, "m" terms represent mechanical variables (rotor angle and speed for theta_m and omega_m); "r" terms are electrical variables (angle and frequency for theta_r and omega_r), relative to the rotor.

FIGURE A-5: MECHANICAL LOAD MODEL



NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 978-1-63276-307-5

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14