Lecture 3

Motor Simulation Model Development Using MATLAB Simulink

Objectives:

- Introduce MATLAB Simulink as a graphical mathematics simulation tool.
- Rework the motor electrical and mechanical equations to develop a block-diagram model of a PM machine in Simulink.
- Wrap the motor model into a subsystem and apply a subsystem mask.
- Apply a three-phase voltage to our PM machine and simulate its operation.
- Discuss the simulation configuration settings, step-size, and solver.

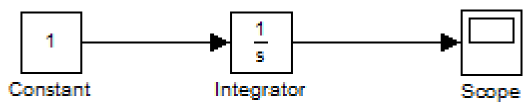
Keywords:

MATLAB Simulink	simulation step size
graphical programming	variable-step simulation
ports & subsystems	fixed-step simulation
source & sink blocks	numerical solver method

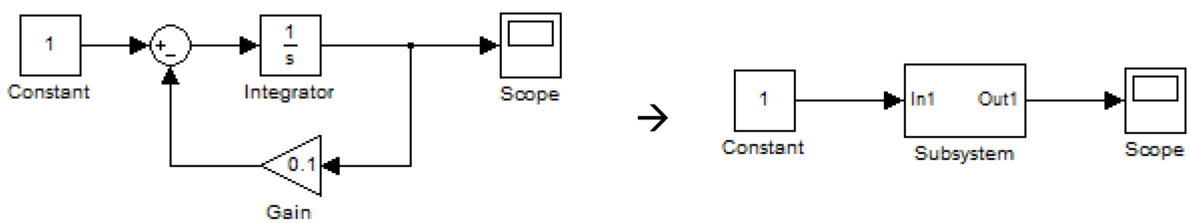
Intro to MATLAB Simulink

Simulink is a subset package of MATLAB – it is a very powerful and widely used numerical simulation tool. Virtually any mathematical system can be constructed and simulated using Simulink.

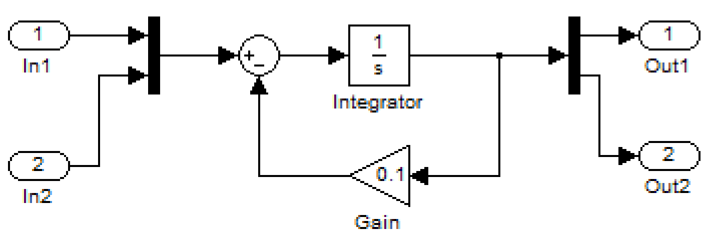
Simulink is a *graphical programming* language, meaning that the "code" involves connecting wires, inputs, outputs, and blocks to form a block diagram. The blocks and the various connections define a mathematical equation.



We can wrap several blocks into a subsystem, by first creating code within a subsystem block and adding input ports and output ports. Simulink can also automatically add the ports if you select a group of blocks and right-click, then select the "create subsystem" option.



Lastly, we can bus multiple mathematical signals together on a single wire by using the multiplexer and demultiplexer blocks. Simulink will NOT indicate the dimensions of a wire by default, unless there is a dimension mismatch.



In this example, the integrator, gain, and summation blocks are copied and applied independently to both signals. Thus, there are two differential equations modeled here.

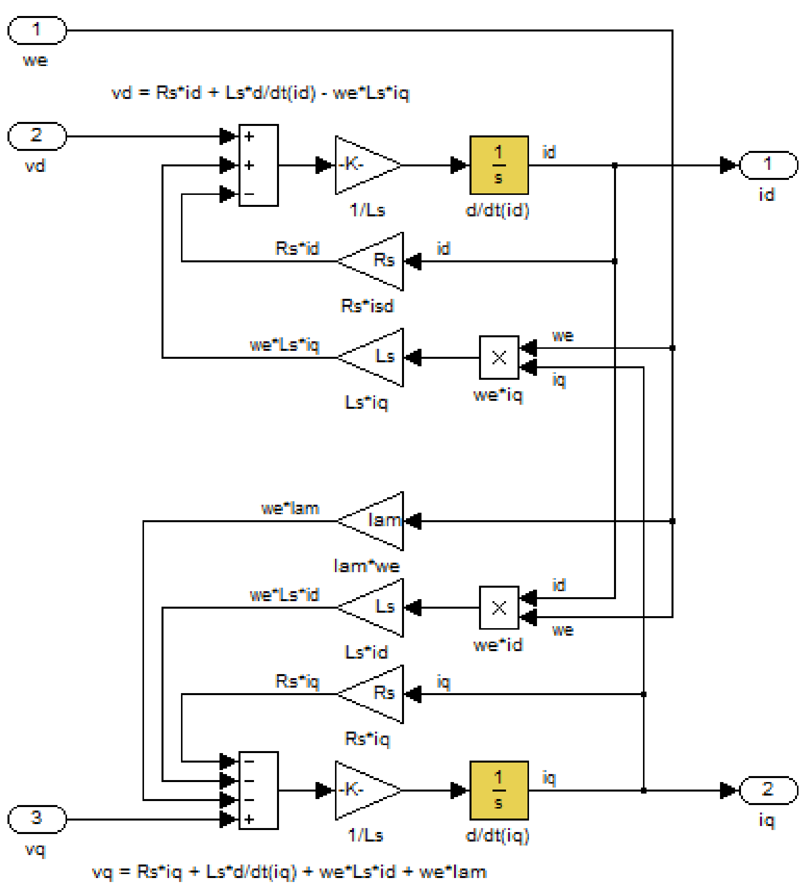
Obtaining Current from the Voltage

Usually, we apply voltages to the machine to control the currents. For simulation purposes, it's useful to rework the rotor-frame stator-voltage equations to solve for the derivative of the current and integrate. This way, we can easily build the block diagram of the machine.

$$\frac{d\,i_d}{dt} = \frac{1}{L_s}\left[v_d - R_s i_d + \omega L_s i_q\right]$$

$$\frac{d\,i_q}{dt} = \frac{1}{L_s}\left[v_q - R_s i_q - \omega L_s i_d - \omega \lambda_{pm}\right]$$

Whenever we're doing block diagram mathematical modeling, it's always easier to start with an integrator block, and build the expression for the derivative input (we can build the voltage equations above). The quantities v_d, v_q, and ω will be inputs, the currents i_d and i_q are derived, and R_s & L_s are constants.



Modeling the Machine Mechanics

Recall that when viewing the machine in the rotor frame-of-reference, our torque equation simplifies quite nicely to:

$$T_{em} = \frac{3}{2}\frac{p}{2} i_q \Lambda_{pm}$$

From the torque equation, and our basic understanding of rotational Newtonian mechanics, we can come up with the differential equations which describe the rotor speed and angle as a function of time and applied current.

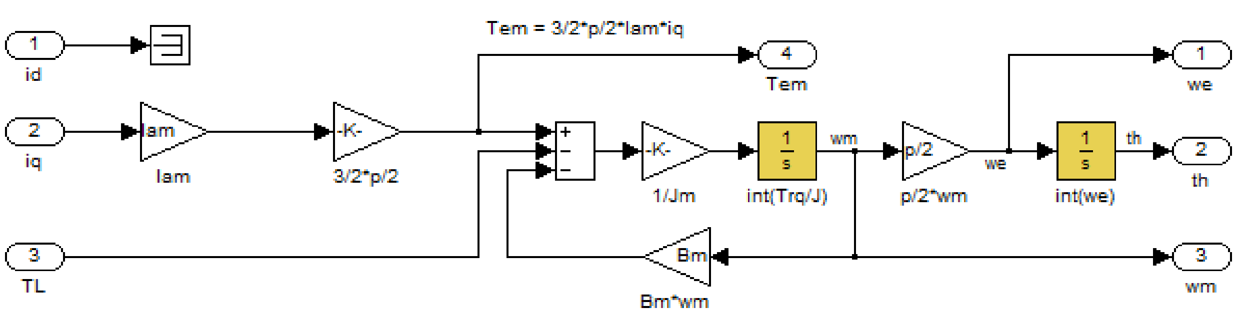
$$T_{em} = B_m \omega_m + J_m \frac{d\,\omega_m}{dt} + T_L \qquad\qquad \omega_e = \frac{d\theta}{dt}$$

where the rotor mechanical speed and rotor electrical speed are related by

$$\omega_e = \frac{p}{2}\omega_m$$

Reworking the mechanical dynamics equations above, we get

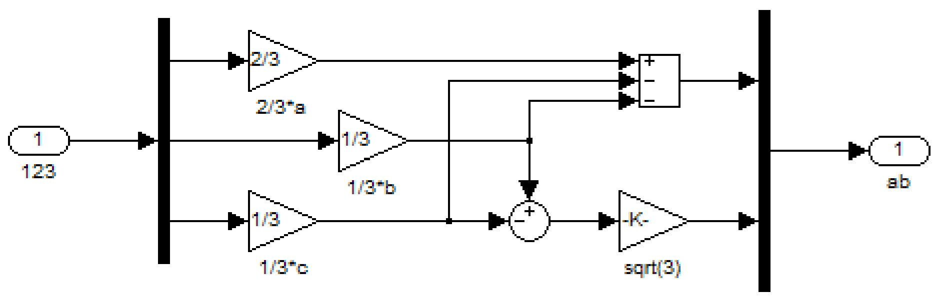
$$\frac{d\,\omega_m}{dt} = \frac{1}{J_m}\left[T_{em} - B_m \omega_m - T_L\right] \qquad\qquad \frac{d\theta}{dt} = \frac{p}{2}\omega_m$$



Connecting the Three-Phase Voltage Terminals

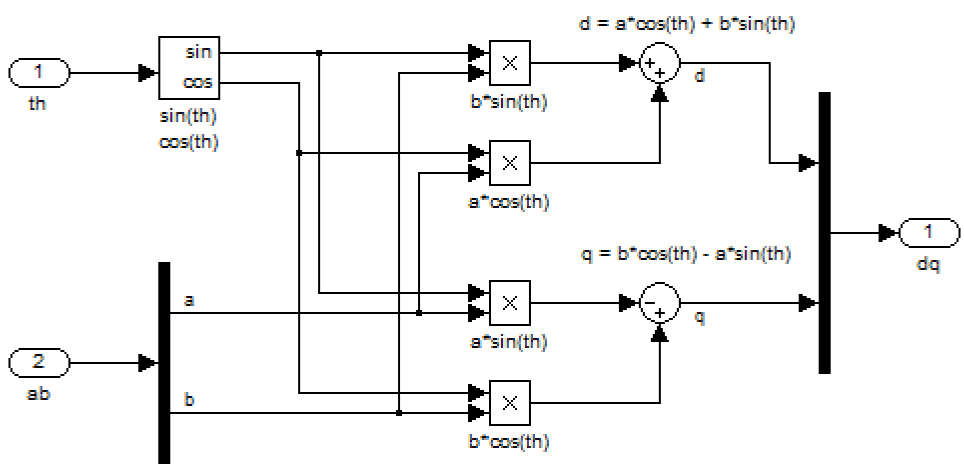
In reality, the machine is controlled by applying a three-phase AC voltage to the terminals. Although the electrical and mechanical models are constructed in the *dq*-frame, we should convert the inputs and outputs to match the real machine's behavior. Thus, we can use the Clarke and Park transforms to convert the applied voltages. The block diagram for the Clarke transform is shown below.

$$T_C = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}$$



The Park transform block diagram is below. Note that it relies on an additional input – the electrical rotor angle. This angle should come from the mechanical section of the motor model.

$$T_P = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

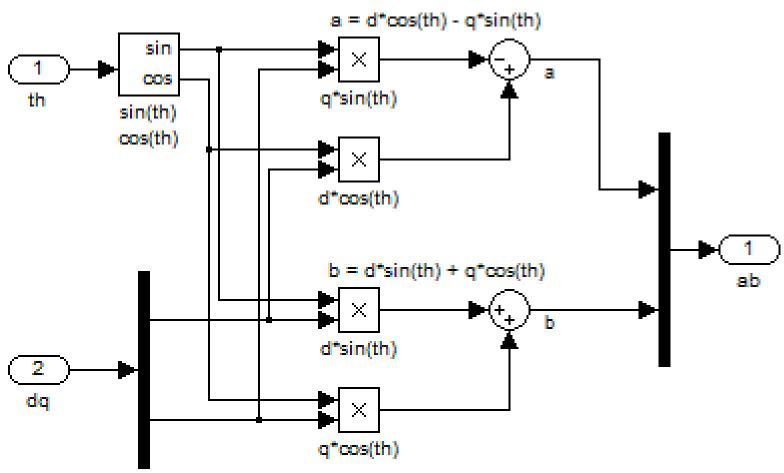


We can ignore the γ-term of the voltage since this only affects the common-mode of all three motor terminals. The machine only responds to a differential line-to-line voltage. If all three voltages increase or decrease together, the voltages applied will not induce any current through the windings.

Connecting the Three-Phase Current Terminals

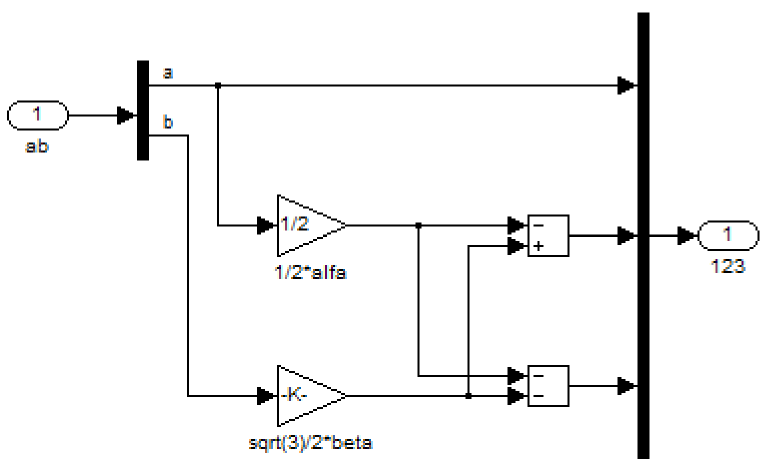
We'll make the current through the machine an output. Thus, to convert the *dq*-currents back to the stationary frame, we should use the Park and Clarke inverse transforms. Below is the inverse Park transform block diagram. Like before, the Park transforms rely on the rotor electrical angle.

$$\mathrm{T_P}^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



Below is the diagram for the inverse Clarke transform.

$$\mathrm{T_C}^{-1} = \frac{3}{2}\begin{bmatrix} 2/3 & 0 \\ -1/3 & 1/\sqrt{3} \\ -1/3 & -1/\sqrt{3} \end{bmatrix}$$

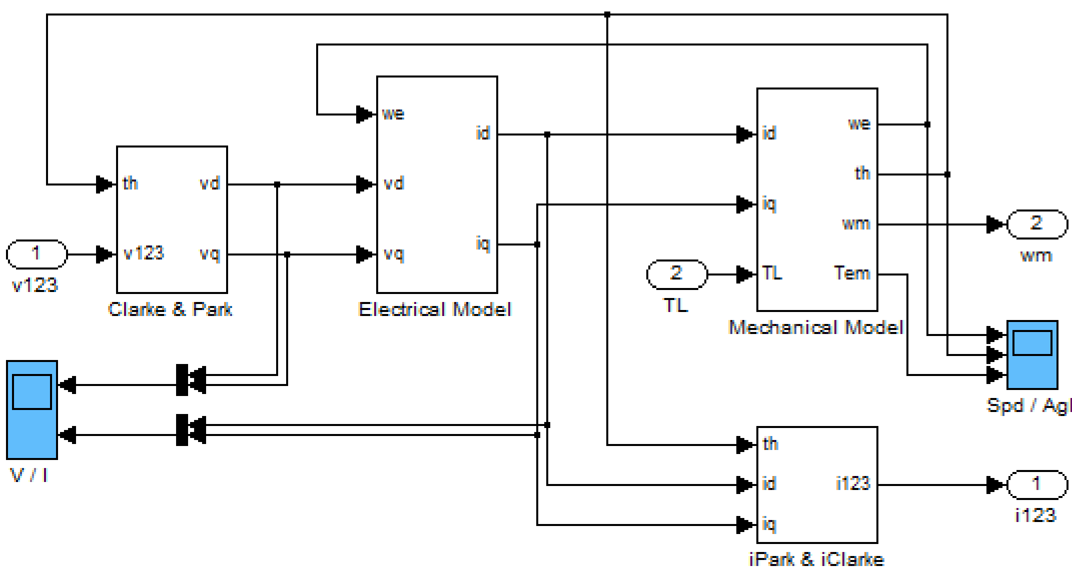


We can ignore the γ-term of the current since we're assuming the three windings are connected internally in a Y-configuration. Thus, it is not possible to have a DC offset value on all three currents.

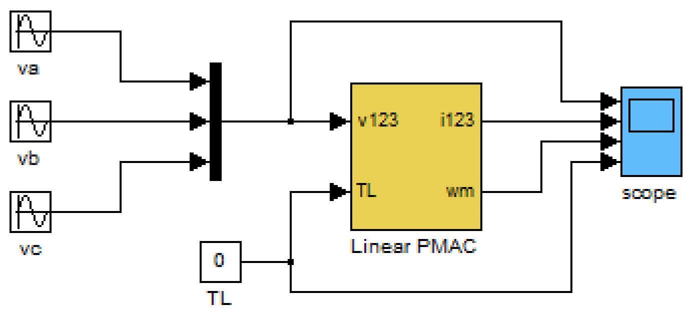
Overall Machine Model

All of the components discussed so far could be connected together and then wrapped into a subsystem which encompasses the entire machine model.

With the way we've modeled the machine dynamics (both electrically and mechanically), we've setup the machine to have inputs of voltage and torque, and outputs of current and speed (and angle). This is illustrated below…



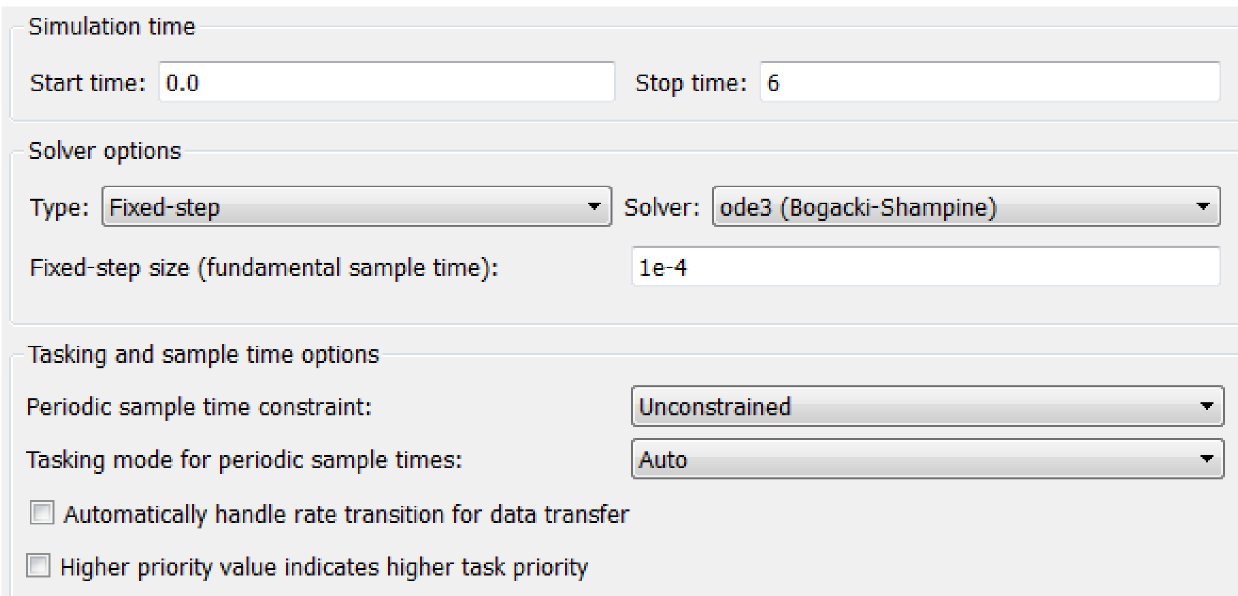
We can apply a three-phase sinusoidal voltage and a load torque by using source blocks. We can then feed the inputs and machine outputs into a scope to view the simulation results. The diagram above has been condensed into a subsystem.



Simulation Configuration Parameters

Before running our simulation, we should set up the configuration parameters.

- Since we don't expect to have any high-frequency harmonics in any of our simulation signals, we can change the time-step to a "fixed-step" solver.
- Furthermore, it's usually a good idea to set the fundamental step size to be around 10x higher than the period of our highest signal frequency.
- On the "Data Import/Export" page, you can uncheck the default variables to be saved to the workspace – we will not use these.



From here, all that's left is to set all of our parameter values of our machine model. Simulink variables will look in the MATLAB workspace for their numeric value. Thus, we can simply run a MATLAB script m-file to initialize all of our model variables. Below are some MATLAB commands to initialize the parameters for a small ~2kW 48V motor (these are the parameters for an electric scooter).

```
Rs = 25e-3;  Ls = 100e-6;  lam = 0.01667;
Jm = 1.0;    Bm = 0.01;    p = 46;
```