

## Electric Drives Experiment 6

### **Analysis and Control of a Permanent Magnet AC (PMAC) Motor**

#### **6.1 Objective**

The objective of this activity is to analyze and control a permanent magnet AC (PMAC) motor. This activity was derived from those developed and distributed by the University of Minnesota (UMN).

#### **6.2 Background, Overview**

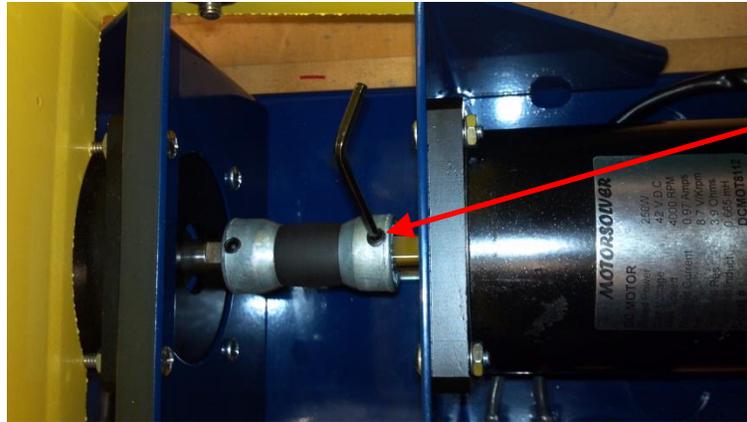
In labs 1-5 you worked with a permanent magnet DC (PMDC) motor, labeled “DC Motor” in the MotorSolver kit. This PMDC motor had permanent magnets mounted on the **stator** and armature coils located on the rotor. As the rotor spun, commutation brushes would connect the input wires to different coils on the stator, sending current to the coil(s) that would produce the most torque for that particular rotor orientation. The number commutation options and the coils’ winding characteristics resulted in a nearly constant (and maximum) torque as the motor rotated.

For the permanent magnet AC (PMAC) motor - which has the same mechanical construction as a brushless DC motor but different power electronics - the permanent magnets are mounted on the **rotor** and the armature coils are located on the stator. One advantage of this configuration is that you no longer need brushes or commutation to provide current to rotating coils, and this reduces maintenance issues and electromagnetic noise. However, in place of the PMDC motor’s mechanical commutation you will now need a control system to determine which PMAC coils should be supplied current to produce the maximum torque given the rotor’s (i.e., the magnets’) orientation.

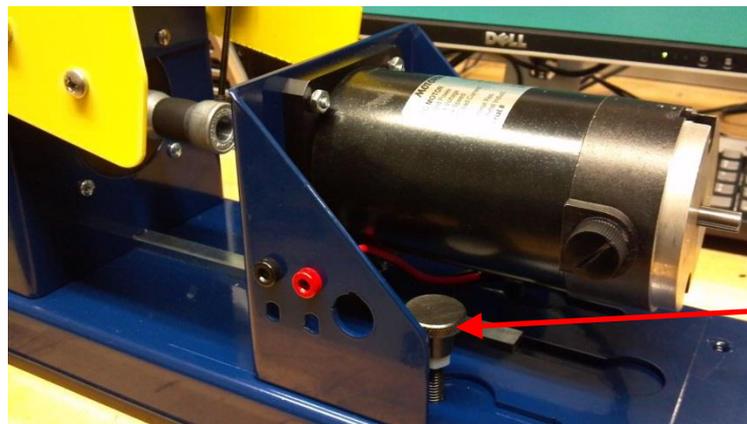
In the first part of this lab, you will analyze the back-EMF produced by the PMAC motor to determine how many magnets the motor contains, as well as the magnets’ orientation with respect to the motor’s encoder reading. This information and the motor’s basic electrical parameters will be used in the second part of the lab to create and test a PMAC controller. The same technique and equations that were used to create a control system for the PMDC motor will again be used to design both the speed and current proportional-integral controllers for the PMAC motor. However, the PMAC motor’s control system will have some added complexity due to the need to continually monitor the magnets’ locations and alter the amount of current sent to each set of coils to create the maximum torque at each moment. This will involve forward and backward conversions between the three current phases and the stator current space vector.

#### **6.3 Reconfiguring the MotorSolver Kit**

Get a MotorSolver Kit, a separate DC Brushless Motor, and an Allen wrench (also known as a hex key) that fits the screws on your kit’s coupler. As shown in Figure 6.1, loosen the screw that connects the DC Motor’s shaft to the coupler, and then hand-loosen the screws that hold the DC motor to the kit’s platform. Pull the DC Motor backwards to slide it out of the coupler, and lift it



(a)



(b)



(c)

**Figure 6.1:** Changing motors on the Motorsolver kit:

- (a) Loosen the hex screw which holds the DC Motor's shaft to the kit's coupler.
- (b) Loosen the screws which hold the DC Motor to the kit's platform, slide the motor back, and lift the motor off the platform.
- (c) Place the DC Brushless (also PMAC) Motor onto the platform, align the **flat side** of its shaft with the hex screw, slide the motor into the coupler, and gently tighten the hex screw as well as the screws which hold the motor to the kit's platform.

off the platform. Reverse the process to mount the DC Brushless (which is also a PMAC) Motor: Place the DC Brushless motor onto the kit's platform, rotate the motor's shaft so that its **flat side** is aligned with the coupler's screw, slide the motor into the coupler, gently tighten the hex screw (snug but not so tight that it's difficult to remove), and hand-tighten the screws that hold the Brushless DC motor to the platform.

## 6.4 Hardware Setup

Return the PMDC motor used in labs 1-5 along with the Allen wrench to the electric drives cabinet, and collect the following items:

- Inverter board
- dSPACE CP1104 board
- +/- 12V power supply / adapter
- Digital power, PWM ribbon cable (37-pin DSUB cable)
- Encoder cable
- Grounding wrist strap

Also obtain a cable with a BNC connector on one end and either banana cables or alligator clips on the other end.

Set the lab bench's DC power supply output to 40 V as usual, and then turn it off.

Put on and ground the wrist strap, and connect all of the hardware components **except the encoder cable and the BNC cable** in the same manner that you've done in previous labs, being careful that you don't supply any power to the board (including the digital power provided by the ribbon cable) until the last step. Use banana cables to connect the Phase A1 output on the power electronics board to the **black** connector on the DC Generator as usual, and connect the Phase B1 output to the DC Generator's **red** connector.

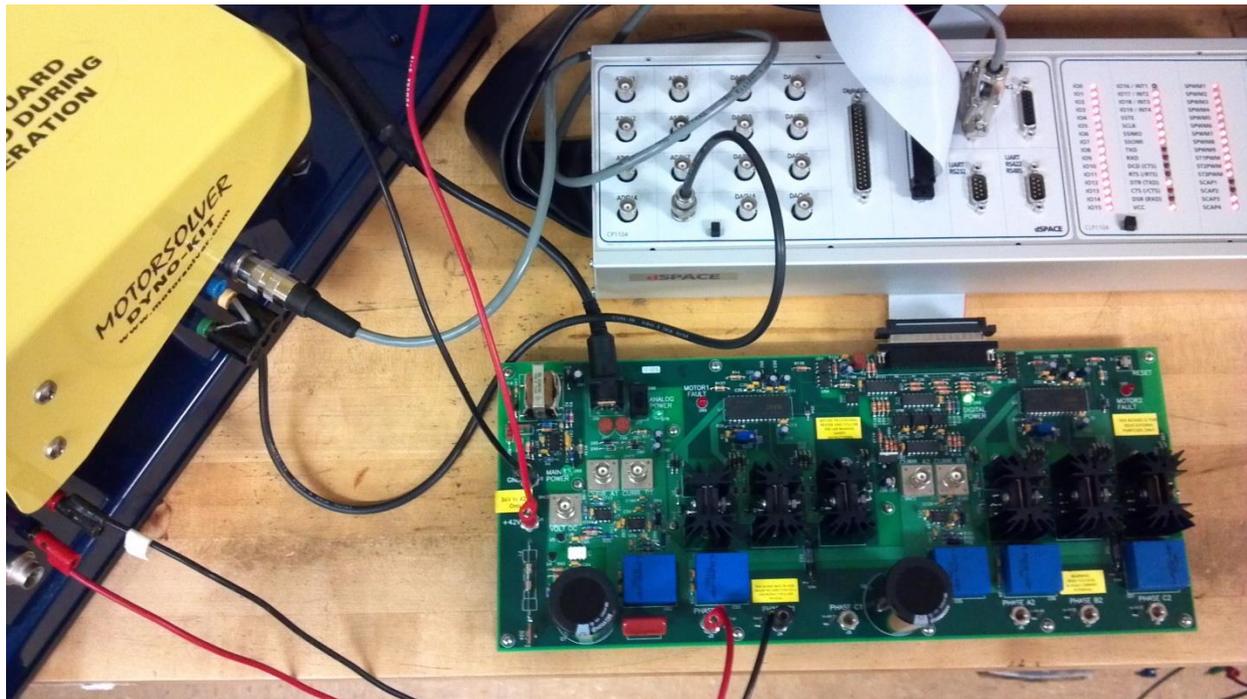
Finally, connect the encoder cable between the **DC Brushless / PMAC** motor and the CP1104 board. In addition, connect the BNC connector to ADC port #8 on the CP1104 board, and connect its other ends to the yellow and green ports on the PMAC motor, as shown in Figure 6.2. (The locations of your motor's ports may be different from those shown in the figure, but the colors you connect to will remain the same). The cable's ground lead should connect to the green port on the PMAC motor. Note that you are connecting between two of the motor's three phases, so your measurements will represent phase-to-phase (or line-to-line, L-L) values. Your hardware setup should look similar to Figure 6.3.

## 6.5 Determining the Number of Magnets and their Orientation on the PMAC Rotor

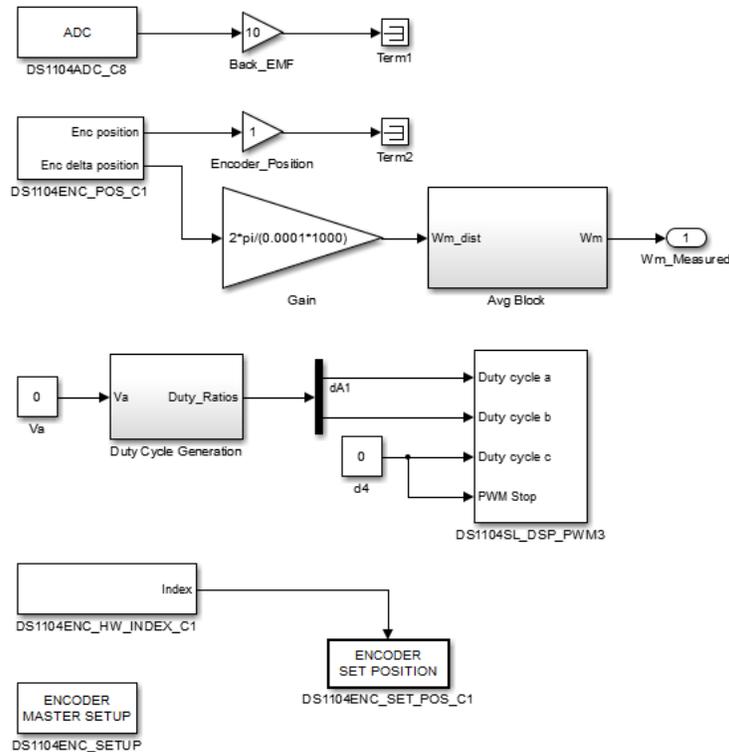
Log in to a lab computer, and create a desktop folder called "Exp6". Download the "Exp6\_Part1.mdl" file from the location specified by the TA or instructor. Rename the file to remove any parentheses caused by the download process. Open the file in the 32-bit version of MATLAB, and the result should look similar to Figure 6.4. Note that Simulink is instructing the DS1104 board to read the value from the 8<sup>th</sup> analog-to-digital port and multiply it by 10 to obtain



**Figure 6.2:** PMAC (DC Brushless) motor connections for the first portion of this lab.



**Figure 6.3:** Overall hardware setup for the first portion of this lab.



**Figure 6.4:** MATLAB Simulink model for the first portion of this lab.

the actual value of the EMF induced between phases ‘a’ and ‘b’ of the PMAC motor. The encoder’s position is also obtained by the DS1104 board so that you can determine the relationship between the magnets’ locations and the encoder’s readings. In addition, the DS1104 board will create PWM signals to apply the drive motor’s armature voltage that you specify in ControlDesk. The last section of the Simulink model is resetting the encoder’s position to zero after each rotation.

Type Ctrl-B to create, compile, and link the C-code associated with the Simulink model, and ensure that the executable is successfully uploaded to the DS1104 board.

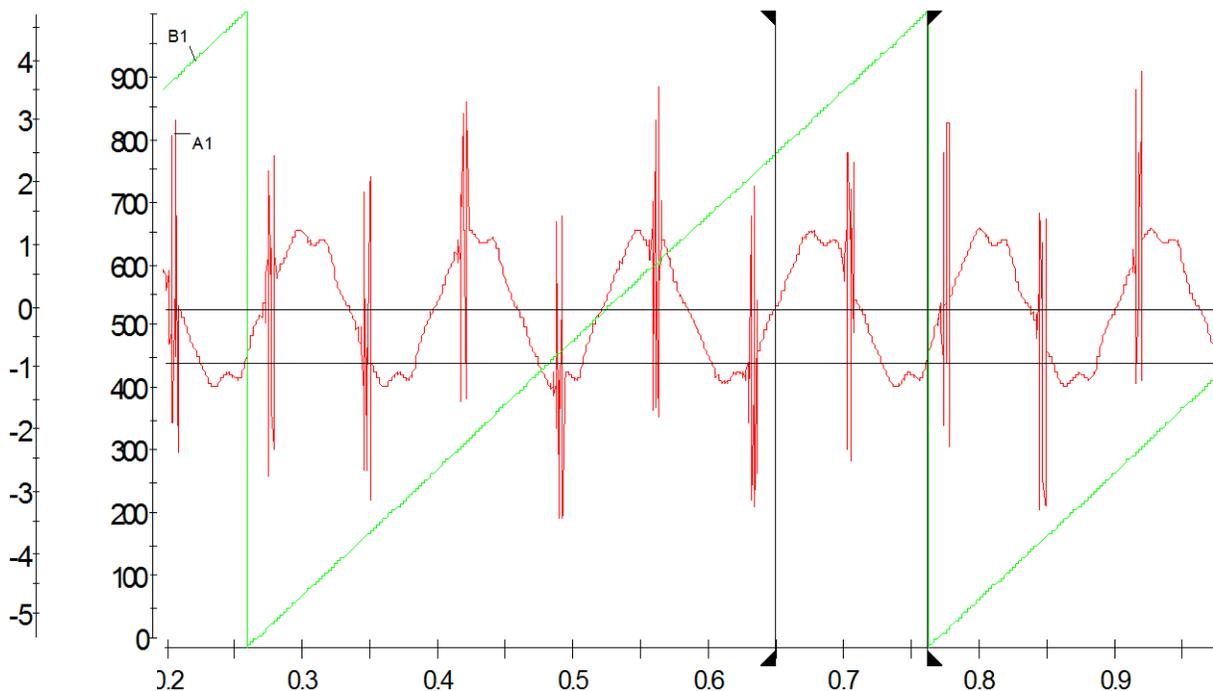
Open ControlDesk and create a new project and experiment, importing the “exp6\_part1.sdf” variable description file as part of the process. Create a plotter window, and display the “Back\_EMF” and “Encoder\_Position” variables in that window, with **different y-axes** being used for the different variables (drag each variable into the window’s graphing area). Set the time duration of the plotter to 3 seconds (accessed via the Measurement Configuration tab...).

Create a slider for the “Va” variable so that you can change the voltage applied to the drive motor, and set the limits of this slider from -5 to 0 (on the Instrument Properties window, unclick “Use variable range” and then set the “Min Range” to -5 and the “Max Range” to 0). The voltage applied to the drive motor is negative so that the PMAC motor will be rotated in its positive direction, which is defined as clockwise rotation as observed when looking down a shaft toward the motor. The voltage magnitude is limited to 5 V so that the line-to-line EMF induced on the

PMAC motor does not exceed the maximum 10 V which can be read by the CP1104 board's ADC ports. Lastly, create a display instrument for the "Wm\_Measured" variable.

Select the "Start Measuring" icon in ControlDesk, turn on the toggle switch associated with the inverter board's analog power supply, and turn on the 40 V DC power supply. Slowly change the  $V_a$  variable until the motors spin smoothly, and *record both the armature voltage and the speed values*. Select the "Stop Measuring" icon when the plotter contains at least a few cycles of the encoder's sawtooth position waveform. Zoom in until you see just a few cycles of the encoder's waveform, and *save an image of the plotter for your report*. Given the number of sinusoidal AC cycles observed in the EMF waveform for each full rotation of the motor (i.e., for one cycle of the encoder position data), *determine how many magnets (poles) are mounted on the rotor*. Remember that one cycle of an EMF waveform is created by two magnetic poles (one pole-pair).

Zoom in to the plotter data so that you can more clearly measure the time between a rising zero-crossing of the sinusoidal EMF waveform and the encoder's zero position, similar to the example shown in Figure 6.5. Use the ControlDesk plotter's XY Cursor capability to measure the time between the rising EMF zero crossing and the encoder's zero position. The y-position values displayed below the plotting window can help you more accurately determine when the EMF waveform crosses zero. *Save an image of this plotter view, and record the time offset ( $X_2 - X_1$ ) measured by your cursors*.



**Figure 6.5:** ControlDesk plotter window demonstrating how to measure the time between the rising zero-crossing of the line-to-line EMF waveform and the encoder's zero position.

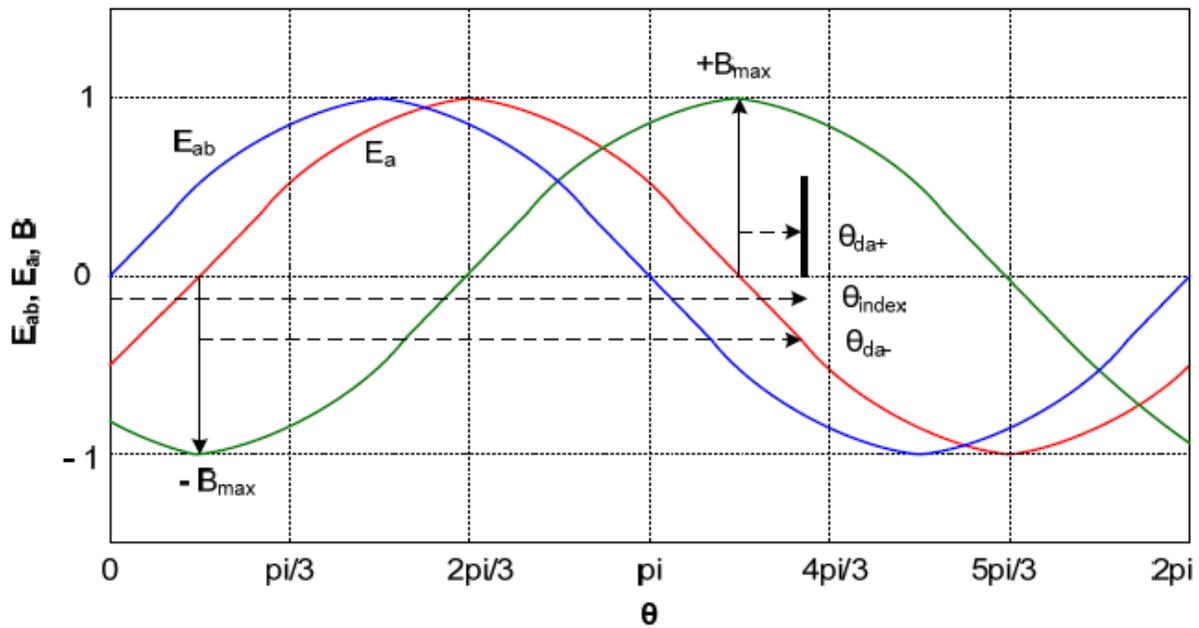
As demonstrated in Figure 6.6, calculate the value of  $\theta_{index}$  as your recorded time offset multiplied by the number of magnetic pole-pairs on the PMAC rotor (poles / 2), multiplied by the rotational speed that you recorded (this process essentially converts the time offset to a phase offset in radians):

$$\theta_{index} = t_{offset} \times pole\_pairs \times \omega_m$$

Calculate and record the phase offsets between the encoder's zero position and the first maximum and minimum magnetic flux locations on the PMAC's rotor:

$$\theta_{da^+} = \theta_{index} - \frac{7\pi}{6}; \quad \theta_{da^-} = \theta_{index} - \frac{\pi}{6}$$

These values will be used by your controller. Note that the  $\pi/6$  value comes from the 30 degree offset between a phase-to-phase waveform and a phase-to-ground waveform.



**Figure 6.6:** (From UMN) Diagram demonstrating the phase offsets between the PMAC motor's normalized phase-to-phase EMF waveform ( $E_{ab}$ ), phase-to-ground EMF ( $E_a$ ), and magnetic flux passing through the phase 'a' coil. Note that the EMF induced in the phase 'a' coil is zero when a magnet is directly aligned with the coil. The phase difference between the magnet's alignment and the encoder's zero reference,  $\theta_{da^+}$ , is needed for the PMAC motor controller.

## 6.6 Determining the PMAC Motor's Voltage Constant

Use a cursor to measure the amplitude of the EMF waveform displayed on the ControlDesk plotter. This should represent the maximum of the sinusoidal waveform, not a sudden peak caused by a noise spike. Be careful to read the value from the correct y-axis scale (that is, double-check whether your measured value makes sense for an EMF voltage). Save an image of

this cursor measurement, and also explicitly state the EMF amplitude in your report. (Use the cursor’s measured value of the EMF amplitude, rather than a graphical estimate.) Calculate the PMAC motor’s voltage constant,  $k_E$ , by dividing this maximum phase-to-phase EMF value by the rotational speed that you recorded earlier:

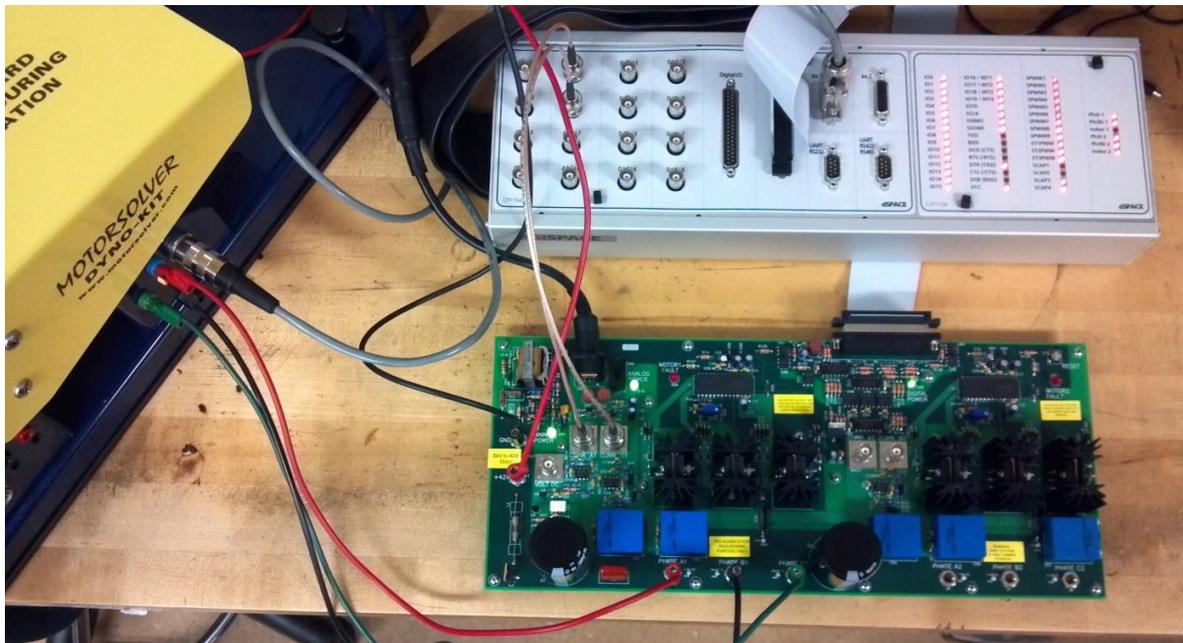
$$k_E = \frac{\hat{e}_{ab}}{\omega_m}$$

Record the voltage constant listed on the PMAC (i.e., DC Brushless) motor’s nameplate. Convert that value to units of V-sec/radian and compare it to the value you determined experimentally.

In ControlDesk, set the armature voltage back to zero, and then close ControlDesk. Turn off the 40 V DC power supply, and turn off the analog power toggle switch on the inverter board.

### 6.7 Reconfiguring the Hardware Setup to Control the PMAC Motor

As shown in Figure 6.7, remove the Phase A1 and B1 connections from the “DC Generator” (drive) motor and remove the BNC cable that connected the PMAC motor to the CP1104 board’s ADC #8 port. Connect the inverter board’s Phase A1, B1, and C1 voltage outputs to the blue, yellow, and green ports on the PMAC motor, respectively. (The relative locations of these ports may be different on your motor than shown in the picture – make the connections according to the connector’s color rather than its location on the motor.) In addition, connect the inverter board’s “Curr. A1” sensor output to the CP1104 board’s ADC #5 port, and connect the “Curr. B1” sensor to the CP1104 board’s ADC #6 port. Note that for three phase power flow,  $i_a(t) + i_b(t) + i_c(t) = 0$ , so the controller can calculate  $i_c$  as  $i_c(t) = -i_a(t) - i_b(t)$  and avoid the use of a 3<sup>rd</sup> current sensor. Your hardware setup should look similar to the one shown in Figure 6.7.



**Figure 6.7:** Hardware setup for controlling the PMAC motor

## 6.8 Designing and Modeling a PMAC Motor Current Controller in MATLAB Simulink

The PWM switching frequency for this lab will be increased to 50 kHz. The crossover frequency for the current controller is desired to be **400 Hz**, far below the PWM switching frequency so that the switching frequency noise is damped. The crossover frequency for the speed controller should be **10 Hz**, well below the current controller's crossover frequency to avoid interference between the controllers.

Given these crossover frequencies, the PMAC motor's voltage constant that you calculated in the previous section, the **typical PMAC motor parameters shown in Table 6.1**, and the same equations used in Lab 4 to calculate the PMDC controller values, *calculate the proportional and integral constants for the PMAC motor's current and speed controllers:  $k_{iI}$ ,  $k_{pI}$ ,  $k_{i\Omega}$ , and  $k_{p\Omega}$* . Show your work in calculating these values. Your values of  $k_{iI}$ ,  $k_{pI}$ ,  $k_{i\Omega}$ , and  $k_{p\Omega}$  should be near but not exactly equal to 40, 0.03, 10, and 0.3, respectively. If your values are more than a factor of two different than these approximate values, double-check your calculations and ask the TA or instructor for assistance before proceeding.

**Table 6.1:** Typical Motorsolver PMAC Motor Parameters

| Parameter | Value                                  |
|-----------|--|
| $R_a$     | 0.625 $\Omega$                         |
| $L_a$     | 0.45 mH                                |
| $J_{eq}$  | $5.0 \times 10^{-4}$ kg m <sup>2</sup> |

Download the "Exp6\_Current.mdl" file from BbLearn, and load it into the 32-bit version of MATLAB Simulink. The Simulink model should look similar to the one shown in Figure 6.7. For positive motor rotation, change the model's "Theta\_da\_Initial" entry from zero to:

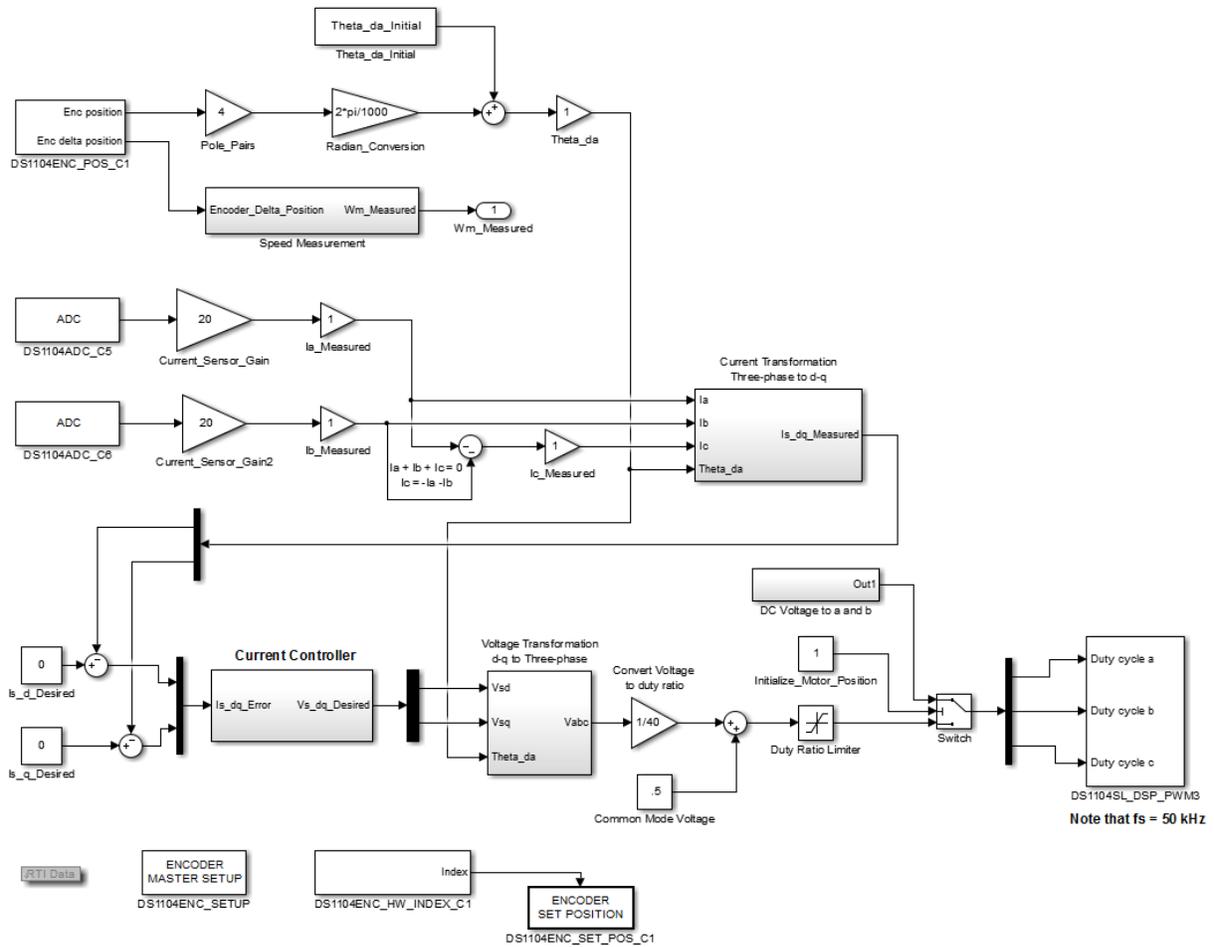
$$Theta\_da\_Initial = \theta_{da^+} + \pi / 2$$

where you previously calculated  $\theta_{da^+}$ . Record your calculation for "Theta\_da\_Initial". In addition, change the entries of Ki\_i and Kp\_i in the Simulink current controller from zero to the values that you calculated earlier. Type Ctrl-B to create an executable of this model and upload it onto the DS1104 board.

## 6.9 Experimental Testing of the PMAC Motor Current Controller

Open ControlDesk and create a new project + experiment, uploading the "exp6\_current.sdf" parameter file as part of the process. Add the following instruments to the project:

- Slider **and** Numeric Input for the Theta\_da\_Initial variable
- Numeric Input for Is\_q\_Desired
- Display for Wm\_Measured
- Check Button for Initialize\_Motor\_Position
- Plotter for Ia\_Measured, Ib\_Measured, and Ic\_Measured where the same y-axis is used for all three variables (drag the second two variables directly onto the y-axis)
- Plotter for Isq and Isd (found under the "Current Transformation" list), same y-axis



**Figure 6.7:** MATLAB Simulink model used to test the PMAC motor’s current controller.

Change the limits of the Theta\_da\_Initial slider to +/- 7.0. Change the time duration of the plotter windows to 4 seconds (...this trigger option is located in the Measurement Configuration window).

Select the “Start Measuring” icon in ControlDesk, turn on the analog power toggle switch on the power electronics board, and turn on the 40 V DC power supply. Enter your value of Theta\_da\_Initial into the associated numeric input instrument. Your motor will not move yet.

Check the “Initialize\_Motor\_Position” button, which will send an equal amount of current to the motor’s phase ‘a’ and phase ‘b’ coils, creating a stationary combined magnetic field halfway between those coils. This will attract the rotor magnets to that location – the location which will give you the largest initial torque when you start the motor. Without this step, the rotor could be in a location that wouldn’t move when you start the motor.

Uncheck the “Initialize\_Motor\_Position” button, and the plotter should indicate that all currents are now zero. Now you’ll apply the three current phases which will create the rotating magnetic field which will apply torque to the PMAC motor. Increase the desired quadrature current

( $I_{s\_q\_Desired}$ ) to 2.0 Amps, and your motor should but may not start spinning. (You may need to push the “Reset” button on the inverter board if the red “Motor Fault” LED is lit, but your motor still may not spin, depending on your  $\Theta_{da\_Initial}$  value.)

Slowly move the  $\Theta_{da\_Initial}$  slider until you find the range of  $\Theta_{da\_Initial}$  values that produces positive motion. *Record the limits of this range, and record the center of that range as the  $\Theta_{da\_Initial}$  value which should produce the maximum speed.* For example, if you obtained positive motion for  $\Theta_{da\_Initial}$  values between 0 and 0.8 radians, choose 0.4 radians as the  $\Theta_{da\_Initial}$  value which should produce the maximum speed. *How does this value compared to the  $\Theta_{da\_Initial}$  value that you calculated earlier?* In ControlDesk, enter that central value for  $\Theta_{da\_Initial}$ , *and record the motor’s approximate speed at that setting.*

Press the “Stop Measuring” icon. Zoom in to the  $I_a$ ,  $I_b$ , and  $I_c$  waveforms until you see a couple periods of each waveform. *Save an image of this view for your report.* (Do not be alarmed if the waveforms are not perfectly sinusoidal – cogging, friction, and other issues create some significant distortions in the waveforms.) *Does phase ‘a’ lead phase ‘b’ by  $120^\circ$ , as expected, and does phase ‘b’ lead phase ‘c’ by  $120^\circ$ ?*

Reduce the desired quadrature current to 1.3 Amps, and *record the motor’s average speed.* Given that this weaker current creates a weaker magnetic field for the stator, and your controller is keeping the stator’s magnetic field at quadrature from the rotor’s magnetic field for maximum torque, *does the change in the PMAC motor’s speed after a decrease in  $I_q$  make sense? Explain.*

Return  $I_q$  to 2.0 Amps, and change the value of  $\Theta_{da\_Initial}$  until the motor stops spinning. Find the lower and upper range of the  $\Theta_{da\_Initial}$  values where the motor does not spin, and *record the limits and center values of that range.* *How far, in radians, is this center value from the center  $\Theta_{da\_Initial}$  value which produced positive rotation? Theoretically, how far of a phase shift from the quadrature alignment (the maximum positive speed condition) would cause the rotor’s magnets to be perfectly aligned with the magnetic field generated by the stator, resulting in no motion? Is this theoretical expectation close to the difference between your theta values which produced the maximum positive and zero speeds?*

Change  $\Theta_{da\_Initial}$  until you reach a region of negative speed (you may need to explore the full range of slider values), and *record the limits and center  $\Theta_{da\_Initial}$  values of that range.* (If your motor won’t move at any  $\Theta_{da\_Initial}$  value, check that the Motor\_Fault on the inverter board has not been triggered. Otherwise, try selecting the “Initialize\_Motor\_Position” checkbox in ControlDesk to set the motor to the best starting location, and then uncheck the Initialize\_Motor\_Position box and try changing  $\Theta_{da\_Initial}$  again.) *How does this center measurement compare to the value of  $\Theta_{da\_Initial}$  which is predicted to produce the maximum negative rotation (there may be a phase difference of  $2\pi$ ):*

$$\Theta_{da\_Initial} = \theta_{da^-} + \pi / 2$$

*How does the measured center angle compare that which resulted in the maximum positive speed, and does that difference match the theoretical expectation?* You may want to refer to Figure 6.6. **Reduce the  $I_{s\_q\_Desired}$  value in ControlDesk to zero. Turn off the 40 V DC power supply, and close ControlDesk.**

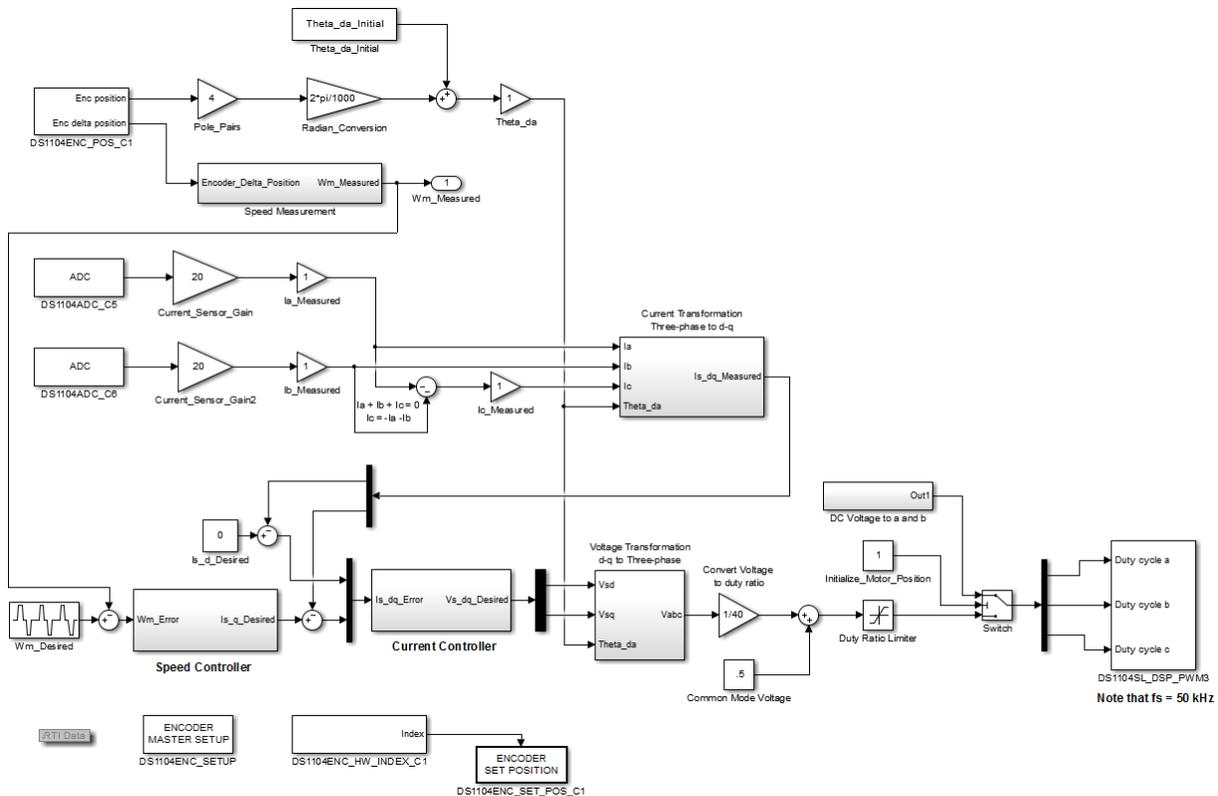
## 6.10 Testing a Cascaded PMAC Motor Controller

Download the “Exp6\_Speed.mdl” file from BbLearn, and load it into the 32-bit version of MATLAB Simulink. The Simulink model of a cascaded PMAC speed and current controller should look similar to the one shown in Figure 6.8. Change the proportional and integral constants in **both** controllers from zero to the values that you calculated earlier, and change Theta\_da\_Initial to the center value which produced the positive speed in the previous section. Type Ctrl-B to create an executable of this model and upload it to the DS1104 board.

Reopen the ControlDesk experiment that you used in the previous section of the lab. Select the “Project” tab, and right-click on the “exp6\_current.sdf” source for the variable descriptions. Select the “Replace Variable Description...” option from the resulting pulldown menu, and select the “exp6\_speed.sdf” file that you just created in MATLAB,

Create a plotter that shows “Wm\_Desired” and “Wm\_Measured” on the same y-axis. Increase the time duration of the plotter window to 10 seconds.

Select the “Start Measuring” icon, and turn on the 40 V DC power supply. Check the “Initialize\_Motor\_Position” box to set the motor to the best starting location for your next experiment. Then uncheck the Initialize\_Motor\_Position box. If the “Motor Fault” error occurs, you may need to hold the “Reset” button on the inverter board until the motor starts spinning.



**Figure 6.8:** MATLAB Simulink model used to test the PMAC motor’s cascaded controller.

Select the “Stop Measuring” icon when the plotter shows a complete cycle of the desired speed waveform. *Save an image of the plotter showing the desired and actual speeds. Analyze the quality of the PMAC speed controller.* For example, you might discuss the quality of the rise time, overshoot, settling time, or steady state error of the actual speed as compared to the desired speed waveform. Note that your controller’s behavior may be significantly different from that experienced by other teams, since each motor has slightly different characteristics and you used the example parameter values provided in Table 6.1 instead of measuring the motor’s precise values. That being said, if your controller is performing poorly, double-check all of the values that you entered into the Simulink model, and ask the TA or instructor for assistance.

Close the ControlDesk and MATLAB software programs.

### 6.11 Shutdown Procedures

Remember to turn off and remove the three power sources first to reduce the probability of damaging the inverter board.

- **Turn off the high-power DC power supply first, without reducing its output voltage.**
- Turn off the toggle switch associated with +/- 12V power supply, and unplug this power supply.
- Remove the ribbon cable between the inverter board and the CP1104 board.
- Remove the black cable from the DS1104 PC board (at the back of the lab computer).
- Remove the BNC cables and encoder cables.
- Remove all banana cables and store them at the lab benches as usual.
- Replace the inverter board inside the protective electrostatic cover.
- Carefully return all of the specialized electric drives components to the lab’s cabinet.
- Check that all of the test bench equipment is turned off, and leave the test bench area clean and organized. Your lab grade will be lowered if you are not careful with the lab equipment and/or if you leave trash in the test bench area.

### 6.12 Lab Report

The group of students at each lab station should submit a lab report. The report should focus on presenting and analyzing the data collected in this lab, although your instructor or TA may provide additional questions or requirements for your report. The requested data were often emphasized in the procedures with italicized text. The report should have section headings to help organize your information. Each figure should have a label and a brief description placed below the image, as demonstrated in these procedures, and each figure should be referenced in the text. Your analysis should be written in prose (i.e., complete sentences); the report should not just contain a collection of images and numbers. Think about writing the report such that you can

use it later in the semester as a study guide. Show the equations or processes used to calculate values that were not directly measured. Your instructor or TA will announce the due date and method for submitting your report.

Checklist of items requested in these lab procedures:

- Record the DC armature voltage applied to the drive motor while measuring the PMAC motor's induced EMF. Also record the PMAC motor's speed in this configuration.
- Save an image of the ControlDesk plotter showing a few cycles of the PMAC motor's encoder and induced EMF waveforms.
- Given the number of AC cycles observed in the EMF waveform for one complete rotation of the motor, determine how many magnets (poles) are mounted on the rotor of the PMAC motor.
- Save an image of the ControlDesk plotter showing the time difference between the rising EMF zero crossing and the encoder's zero position. Also record the time offset.
- Calculate the value of  $\theta_{index}$  for your lab station's PMAC motor.
- Calculate the angular offsets between the encoder's zero position and the first maximum and minimum magnetic flux locations on the PMAC's rotor.
- Use a cursor to measure the maximum value (i.e., amplitude) of the EMF waveform displayed on the ControlDesk plotter. Save an image of this cursor measurement, and also explicitly state the maximum EMF value in your report.
- Calculate the PMAC motor's voltage constant,  $k_E$ .
- Record the voltage constant listed on the PMAC motor's nameplate. Convert that value to units of V-sec/radian and compare it to the value you determined experimentally.
- Calculate the proportional and integral constants for the PMAC motor's current and speed controllers:  $k_{iI}$ ,  $k_{pI}$ ,  $k_{i\Omega}$ , and  $k_{p\Omega}$ . Show the relevant equations and your work in calculating these values.
- Record your calculation for Theta\_da\_Initial.
- Record the limits and center of the range of Theta\_da\_Initial values which produced positive rotation for your PMAC motor. How does this measured value compared to the Theta\_da\_Initial value that you calculated earlier?
- Record the motor's average speed at the center Theta\_da\_Initial value.
- Save an image showing a few periods of the  $I_a$ ,  $I_b$ , and  $I_c$  waveforms. Does phase 'a' lead phase 'b' by  $120^\circ$ , as expected, and does phase 'b' lead phase 'c' by  $120^\circ$ ?
- Reduce the desired quadrature current to 1.3 Amps, and record the motor's average speed. Does the change in the PMAC motor's speed after a decrease in  $I_q$  make sense? Explain.

- Record the limits and center of the range of Theta\_da\_Initial values where the PMAC motor did not spin. How far, in radians, is this center value from the center Theta\_da\_Initial value which produced positive speed? Theoretically, how far of a phase shift from the quadrature alignment would cause the rotor's magnets to be perfectly aligned with the magnetic field generated by the stator, resulting in no motion? Is the theoretical expectation close to your measurement?
- Record the limits and center of the range of Theta\_da\_Initial values that produced a negative speed in the PMAC motor. How does this center measurement compare to the value of Theta\_da\_Initial which is predicted to produce the maximum negative rotation (there may be a phase difference of  $2\pi$ )? How does the measured center angle compare to that which resulted in the maximum positive speed, and does that difference match the theoretical expectation? You may want to refer to Figure 6.6 in your explanation.
- Save an image of the plotter showing the desired and actual speeds when the speed controller was added to the model, and analyze the quality of the PMAC speed controller.