Iowa State University
Electrical and Computer Engineering
E E 452. Electric Machines and Power Electronic Drives

Laboratory #8
Three-phase inverter with Space Vector Modulation
Hardware Implementation

Summary

In this lab you will implement a three-phase inverter to generate a waveform based on space vector modulation (SVM). This will be implemented using the TI F28035 ControlCARD. The embedded application will be developed using MATLAB/ Simulink Embedded Coder.

Learning objectives

- Use the TI MF28035 to implement a space vector modulated inverter.
- Introduce the squirrel-cage induction machine.

Background material (should be read before coming to the lab)

- Trzynadlowski chapters 7 and 4
- Lecture notes on three-phase inverters and space vector modulation
- *TI C2000 Digital Motor Control Library Details* (*DMC Library Details_spru485a.pdf*)
- TMDSHVMTRPFCKIT IGBT Datasheet (*ps21765_IGBT_Module_HVMTRKIT.pdf*)

Exercises and Questions

Instructions: every student should deliver his/her own report at the end of the lab session, even though the experiments are conducted in groups. You may want to answer the questions as you go along the exercises. Time yourselves according to the recommendations below.

1. Pre-lab assignment

The TI C2000 Digital Motor Control (DMC) library, is a set of code templates for implementing common motor drive functions, including SVM. Embedded Coder supports use of the TI DMC library, by offering configuration of these functions through Simulink blocks.

Review the details of the *SVGEN_DQ* block, in *DMC Library Details_spru485a.pdf*. This block will be used to generate the required signals.

Hint: Check the definition of the Direct and Quadrature axes, and angle between the 3-phase set and qd-set. All DQ blocks in the TI motor control library use this DQ definition.

DELIVERABLE 1: What differences do you see between the SVM developed in class, and the SVM provided by TI? Provide details.

2. Space Vector Modulation with F28035 using Embedded Coder [90 minutes]

Open a new model in Simulink. Add the *Target Preferences* block and select your board. Additionally, add *ePWM*, *ADC*, and *Digital I/O* blocks to configure the inputs and outputs of your application. The ADC will be used to sense the DC-bus voltage, the ePWM blocks will drive the six switches, and the Digital I/O blocks will provide application status.

The ADC should be configured to measure the DC-bus voltage. Add the necessary math blocks to determine the DC-bus voltage. You should have this from Labs 5 and 6.

The ePWM blocks should be configured for *Up-Down* count mode, with the action of ePWMxB as opposite of ePWMxA. There should be three inputs to each PWM block, including the CMPA register value, Software Forcing for ePWMA, and Software Forcing for ePWMB. Set the PWM switching frequency to a value appropriate for your power switching devices, somewhere between 1 kHz and 15 kHz.

Add dead-time to each PWM channel, so the high- and low-side switches are never ON at the same time. Within the ePWM configuration blocks, go to the *Dead-Time* tab. Select to add dead-time to both ePWMxA and ePWMxB channels. Select the *polarity* as *Active High Complimentary (AHC)*. Set the number of clock cycles to inject between switching events, to a value appropriate for your switching device. Check the IGBT datasheet for switching time details (*ps21765_IGBT_Module_HVMTRKIT.pdf*).

Configure the *ePWM Trip-Zone* (TZ) to turn off the switches on the event of an IGBT fault. Use *One-Shot TZ1*. Make sure to configure the pin configuration of the TZ unit; this is done in the *Target Preferences* block, just like in Labs 5 and 6.

Configure a digital output to show the application's status; this should be similar to Labs 5 and 6.

Add the *Space Vector Generator* block from the *Embedded Coder/Embedded Targets/TI C2000/Optimization/C28x DMC* library. This block generates the switching signals for the high-side switches of a three-phase H-Bridge. The inputs to this block are *Q*-axis and *D*-axis reference commands, in the stationary reference frame. The output is the duty cycle to send to each complimentary switching pair. This block is used on a per-unit basis; the input will be on the scale of 0 to 1, and the output will be on a scale of 0 to 1.

Add a *multiply* block between the SVM_DQ bock and the ePWM blocks, to scale the duty cycle to a value appropriate for the ePWMx *PRD* register. The duty cycle input (determined by *CMPA*) should be in the range of [0, PRD].

Add two *Float to IQN* blocks from the TI DMC library; set the *Q value* to *15*. These blocks produce a number with the proper format required for the *SVGenDQ* block. The input data type to the *Float to IQN* must be a "single" format.

Create a subsystem with the above blocks, such that the inputs are the desired V_{ds} and V_{qs} components, and the outputs are the duty cycles for each switch. The subsystem should be similar to that shown in Figure 1.

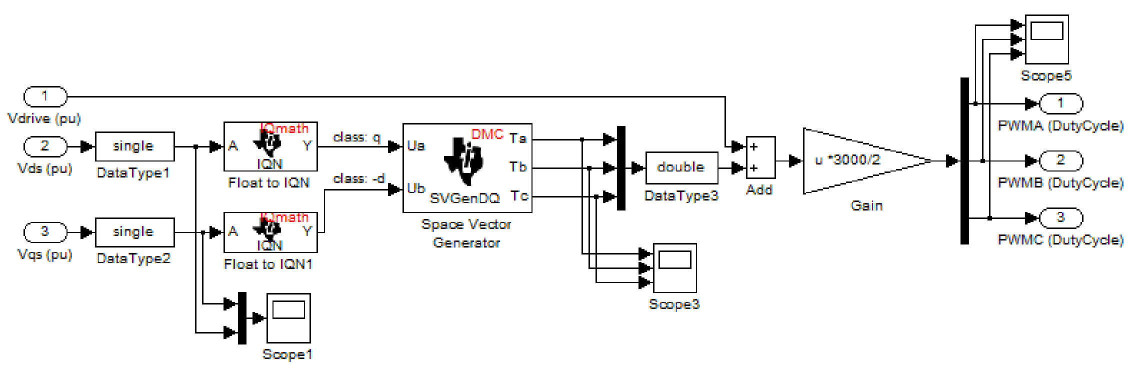

Figure 1. Implementation of space vector modulation to drive the F28035 ePWM modules.

To generate the reference command signals, create a subsystem to generate the commanding waveforms.

Because the SVM_DQ block generates the duty cycle on a per-unit basis, a 100% duty cycle will always maximize use of the DC-bus voltage; $V_{ll\_rms} = V_{dc}$/sqrt(2). If the DC-bus voltage is larger than the nominal value, for any reference V_{ll}, it is necessary to scale the input to the SVMGen_DQ block. In this lab, we will use a V/Hz control strategy; the DC-bus voltage is measured, and compared to the nominal DC-bus voltage, for a given V_{ll_rms}, to scale the input to the SVM_DQ. Appropriate scaling is done to generate the desired output.

Example: The desired nominal line-to-line voltage is 240 V rms. The minimum DC-bus voltage required to generate this output, with a modulation index of m = 1, is V_{dc} = 240*sqrt(2). This is the nominal DC-bus voltage. For any DC-bus voltage larger than 240*sqrt(2), the modulation index must be m < 1 to generate V_{ll_rms} = 240 V.

Note: If the actual DC-bus voltage is less than the nominal DC-bus voltage, it will be impossible to generate the desired V_{ll_rms}.

Add a *Multiply* block at the output of the sine sources. This block will implement the per unit scaling to account for the actual DC-bus voltage. It should implement the equation

$$V_{ref} = \left(\frac{DCBUS_{nom}}{DCBUS_{act}}\right), \quad (1)$$

To generate the space vector that rotates with speed ω_e, integrate the speed to obtain the time-changing angle of the vector. You will need to use the *Discrete-Time Integrator*. Configure the integrator to reset the *State and Output* to 0 on a *Rising* event. Check the box to *Show state port*, and use the state to determine when to reset the block. Make the output vary between 0 and 2π.

The Reference Signal Generator should be similar to that shown in Figure 2.

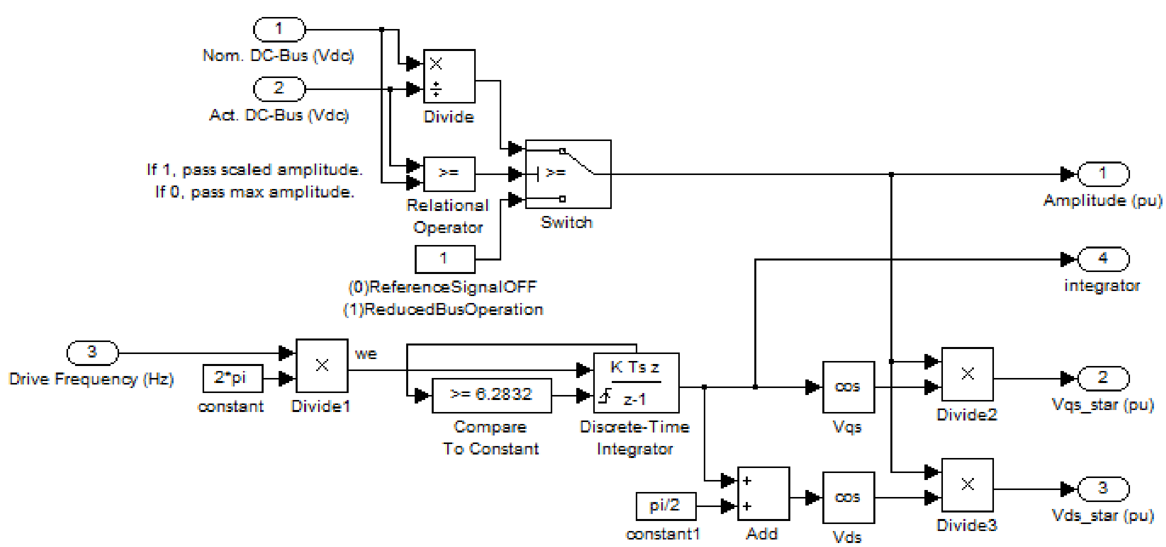

Figure 2. Reference signal generation for implementation of the constant V/Hz control.

Add another subsystem to implement the V/Hz control strategy. This control strategy is commonly used for AC induction machines. As the applied frequency changes, the amplitude of the line-to-line voltage also changes, by the ratio defined by V_{rated}/F_{rated}. For low frequencies, generally under 10 Hz, it is usually necessary to increase the voltage a little bit, to maintain sufficient magnetizing flux; this is commonly referred to as "Boost" voltage. We will not consider low frequency boost in our motor drive.

Add *Logic, Math*, and *Routing* blocks to your model to implement the V/Hz control. The only reference command, input by the user, should be the drive frequency. The nominal DC-bus voltage should be calculated based on the V/Hz control. You should have something similar to that shown in Figure 3.

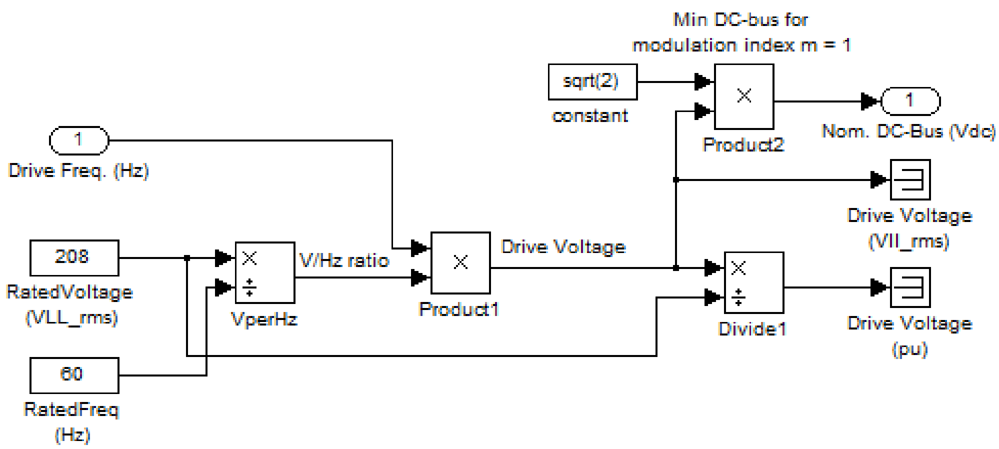

Figure 3. Implementation of V/Hz control using Simulink blocks. The nominal DC-bus voltage is calculated based on the variable input drive frequency and the specified V/Hz ratio.

Add *Constant* and *Scope* blocks as needed, to simulate your model, and verify all equations.

DELIVERABLE 2: Email your completed model (.mdl file) to the T.A. Include your group member names in the email.

When your model is done, it should be similar to that shown in Figure 4.

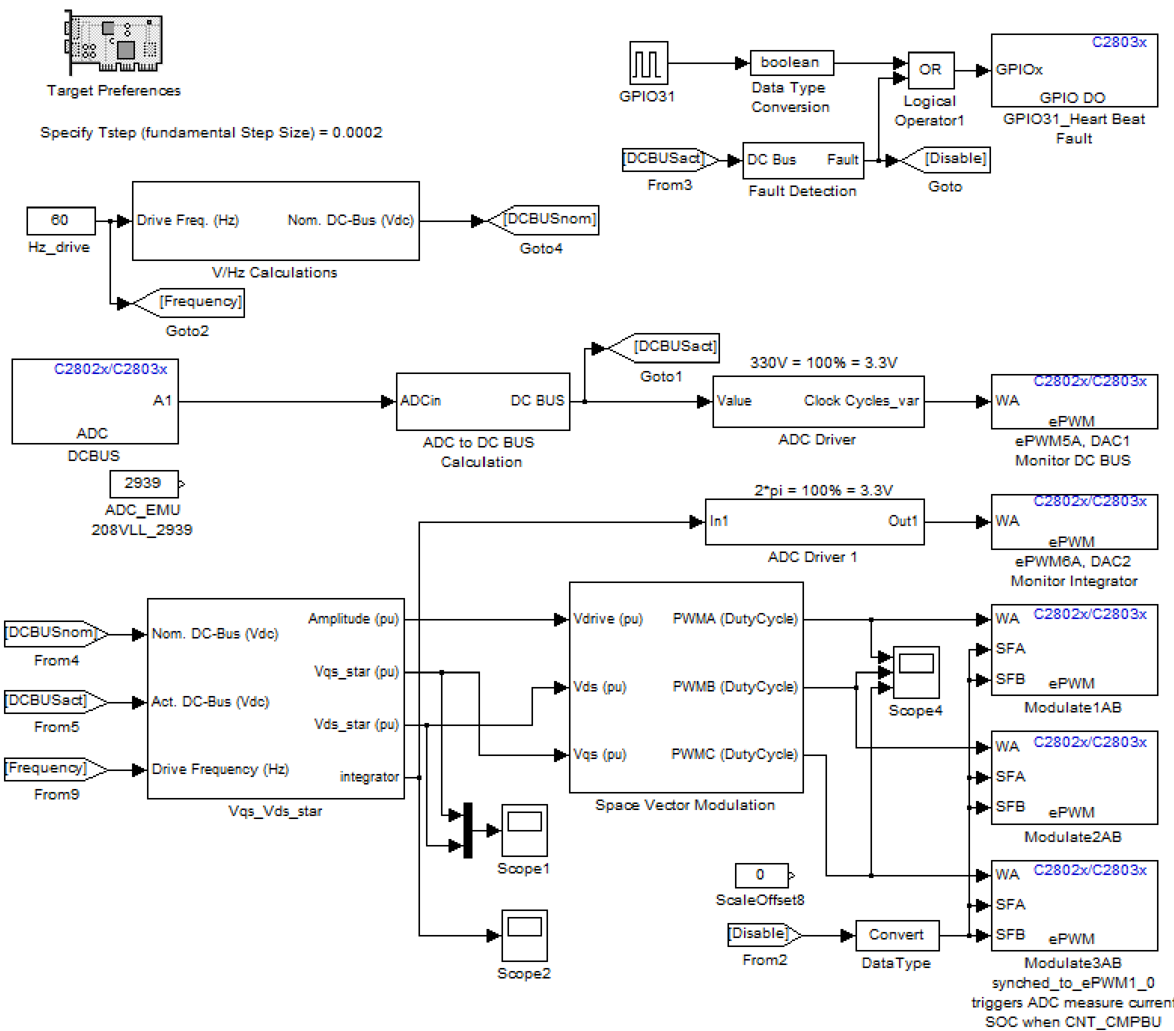

Figure 4. Simulink model for the three phase AC induction motor drive, using space vector modulation and open-loop V/Hz control.

In the *Simulation Configuration Parameters*, set the *Fixed Time Step* to 0.0002 s. Check the box to *Automatically handle rate transition for data trasnsfer*. On the *Code Generation* tab, set programming *Objectives* for *RAM efficiency* and *ROM efficiency*. Click to *Check model…* For each warning that displays in the Code Generation Advisor, click the box near the bottom to *Modify Parameters*.

Note: not every objective will be satisfied.

Set the drive frequency to 60 Hz. When the model is finished and equations are verified, *Build* the model to generate the .out executable file.

3. **Space Vector Modulation with an R-L Load** [30 minutes]

Operation of the controller will be verified with an R-L load before running the motor. Ensure all safety precautions are considered. Cover the TMDSHVPFCMTR kit before powering the circuit. Wear safety glasses while working near powered equipment.

Configure your hardware for a three-phase R-L load, connected in Y configuration, where R_Y = 1.2 kΩ and L_Y = 0.8 H. Connect the Lab-Volt DC-bus (*Pins 7-N*) to the inverter outputs; turn the power supply dial to the minimum. Connect a 1.2 kΩ resistor in parallel with the DC-bus, to discharge the DC-bus when powering off. DO NOT TURN ON POWER AT THIS POINT.

Note: There will be many banana plug wires used in this lab. Choose cable lengths and colors appropriately, to avoid confusion.

CAUTION!
Verify your hardware and software setup with your group partners, and your T.A.

Turn on the TMDSHVMTRPFC kit, *Debug* the application using CCSv5, and *Run* the program. LED *LD3* should be blinking on the ControlCARD, when the code is running.

While measuring the load voltages and currents, slowly turn up the DC-bus voltage. If you notice anything out of the ordinary, stop, and fix any problems. The frequency should be fixed at 60 Hz, and the amplitude will rise with the DC-bus; remember, the reference voltage amplitude is set to 1.0 pu when the DC-bus voltage is less than the nominal value.

DELIVERABLE 3: Observe and record the line-to-line voltage and phase current waveforms using the oscilloscope. Do the RMS values match what you expected?

4. **Three-phase Inverter driving a Squirrel-cage Induction Motor** [60 minutes]

Now that you have developed a controller to run on the F28035, and verified its operation, it is time to implement the three-phase AC induction motor drive.

Pull the Lab-Volt ACI motor out and examine its physical structure. Notice there are no electrical connections to the rotor, and that the squirrel cage bars, which extend the length of the rotor, are shorted together at the ends of the machine; currents will circulate through these bars as the stator magnetic field rotates around the rotor.

In this lab, we will run the machine at one voltage and one frequency, following the V/Hz relationship.

Configure your control model for a constant drive frequency, of your choice, between 20 Hz and 60 Hz. Make sure the drive voltage is set accordingly. Build the model to generate the .out executable file.

The DC supply on the Lab-Volt power supply only goes to 170 V DC. A larger DC-bus voltage is required to generate the rated line-to-line voltage for the Lab-Volt ACI machine. The three-phase source and diode rectifier will be used to generate the DC-bus voltage, as shown in Figure 4.

Ensure the Lab-Volt supply is turned OFF. Connect the variable three-phase outputs of the Lab-Volt power supply (*pins 4, 5, 6*) to the three-phase diode rectifier (*pins 4, 5, 6*). Connect the diodes to generate a DC-bus between pins 1 and 7. Connect this DC-bus to the input of the TMDSHVMTRPFC (*pins BS5 to BS6*). Also, connect a parallel resistor (1.2 kΩ) to discharge the DC-bus after the drive is turned off.

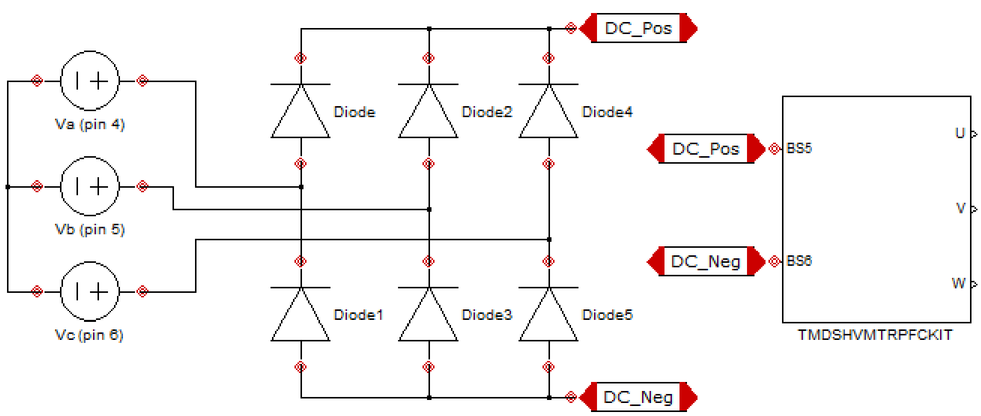

Figure 4. Three-phase diode rectifier for a 300 V DC-bus.

WARNING! The DC-bus (-) terminal is **NOT** connected to Earth ground. **DO NOT** connect the oscilloscope ground directly to any part of the TI kit; use only isolated probes, or you will damage the oscilloscope.

Connect the *U, V, W* outputs of the TMDSHVMTRPFCKIT to the Lab-Volt squirrel-cage motor. Connect the motor windings in *WYE* configuration.

Connect the Lab-Volt dynamometer to the induction motor via the supplied belt. Set the dynamometer for Neg CT Prime Mover/Brake; a counterclockwise torque will be produced (drive the ACI machine in the clockwise direction). Ensure the control dial is at the minimum setting.

Connect voltage and current probes to measure the DC-bus voltage, output line-to-line voltage, and phase current. Beware of the machine's rated values.

CAUTION!
Verify your configuration with your group members and your T.A.

Download and run your application on the F28035. The LED should be blinking when the application is running.

Slowly increase the DC-bus voltage, while monitoring the line current. Be aware that the line current may be larger than the machine's rated value while starting the machine (the time of which you are increasing the DC-bus voltage). Do not operate the machine in this condition for an extended period of time; you run the risk of burning the machine.

If everything is okay, increase the DC-bus voltage to at least the nominal value you calculated for your drive frequency. As the motor starts spinning, the current should reduce to a steady-state value within the machine's rating. If it does not, turn off the power supply and fix any problems.

Once the machine is spinning at a steady state condition, you can increase the dynamometer torque. Observe the line current. Do not operate the machine beyond the rated current for an extended period of time.

DELIVERABLE 4: How does the current change as you increase the torque?

Reduce the dynamometer torque and DC-bus voltage, and turn off the drive.

5. **Conclusion** [15 minutes]

Write about one or two things you learned in this lab that you think are important or interesting, and why.