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

Laboratory #13
Squirrel-cage Induction Motor – Field Oriented Control (Part 1)

Summary

A closed-loop speed controller using Field Oriented Control (FOC) will be implemented. The TMDSHVMTRPFCKIT comes with a set of CCS example projects, including a FOC application pre-tuned for the Marathon Electric 5K33GN2A squirrel-cage induction motor. This project will be incrementally built using CCSv5, and will run on the F28035 MCU. Machine response to a load step change will be analyzed.

Learning objectives

- Learn about Field Oriented Control through a CCS example project.
- Use CCS to incrementally build the project, and view system variables in real time.
- Analyze the dynamic response of the machine for a varying load.

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

- *TMDSHVMTRPFCKIT How to Run Guide (HVMotorCtrl+PFCKit_HowToRunGuide.pdf)*
- *Sensored Field Oriented Control of 3-Phase Induction Motors (Sensored FOC of ACI.pdf)*
- Krause chapter 6.11 *(Introduction to Field-Oriented Control)*

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

This lab follows the example project described in *Sensored FOC of ACI.pdf*. Please refer to that document, and any other documents referenced there, for further details regarding the material contained in this laboratory manual. All documents and project files can be found in ControlSUITE.

Note: All figures and page numbers referenced in this laboratory manual refer to those figures and pages in *Sensored FOC of ACI.pdf*.

Read through *Sensored FOC of ACI.pdf*. Be sure to understand the objective of field oriented control, and how this is accomplished. The basic scheme of FOC is shown in Figure 7, on page 8.

DELIVERABLE 1: Describe Field Oriented Control. In Figure 9, describe the purpose of the *Current Model*. What equations does this block contain?

2. Hardware Setup [15 minutes]

This lab requires the use of all of the peripherals we have studied so far, including the eQEP, ADC, and ePWM. Machine parameters are also necessary to implement FOC of an ACIM. An overview of the system is shown in the block diagram of Figure 9.

Follow the instructions of page 16, *Hardware Setup Instructions,* to configure the TMDSHVMTRPFCKIT for running this project. Make sure the jumper from *BS1* to *BS5* is NOT installed. Generate the DC-bus from the Lab-Volt 3-ph variable power supply. Connect a 1.2 kΩ resistor in parallel with the DC-bus, to discharge the bus upon powering down the circuit.

Note: Connect the motor and encoder to the kit only after build level 1 is complete.

Warning! Install the plastic cover before applying the DC-bus voltage.

Warning! The DC-bus ground is not at Earth ground. Do not connect the oscilloscope ground to the DC-bus ground. Damage to the oscilloscope will result. Use isolated probes for the duration of this project.

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

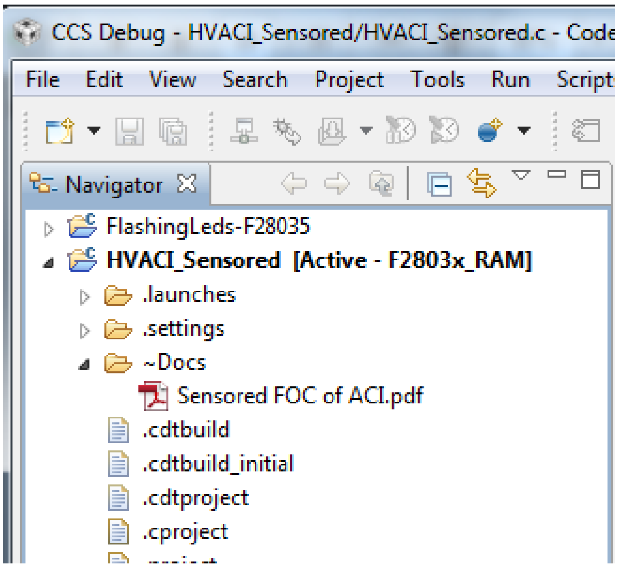
3. **Software Setup** [15 minutes]

Setting up the CCS environment follows *HVMotorCtrl+PFCKit_HowToRunGuide.pdf.*

Open CCSv5. Go to File/Import to import the example project into the workspace. You will need to click on *Code Composer Studio/Existing CCS/CCE Eclipse Projects*, and then browse to

C:\TI\controlSUITE\development_kits\HVMotorCtrl+PfcKit_v1.6\HVACI_Sensored.

Click "Finish" to import the project into the workspace. You should see the project in the Navigator window, as shown below.



Right-click on the project and set it as the active project, it should turn bold.

To set the *Build Level*, open the file *HVACI_Sensored-Settings.h*, and scroll to line 27, where you can then set the build level appropriately. This value will need to be changed for each step in the incremental build process.

Right-click on the project and set the *Build Configuration* as *F2803x_RAM*. Build the project and launch a debug session as described in the document.

Click *Enable silicon real-time* mode and *Enable polite real-time mode* as described in *HVMotorCtrl+PFCKit_HowToRunGuide.pdf.*

Warning! Disable these two real-time modes before performing a *Reset* to avoid connectivity problems.

To watch the value of system variables, go to View/Expressions and add the variable you wish to monitor. Enter the variable names as described in Table 1 on Page 19.

Add two graphs as described on Page 19, and click to *Continuously Refresh*. Make sure the *Start Address A* and *Start Address B* have the following entries:

DLOG_4CH_buff1
DLOG_4CH_buff2
DLOG_4CH_buff3
DLOG_4CH_buff4.

Continue following the instructions of *Sensored FOC of ACI.pdf* to complete all five build levels.

4. Continue with the Incremental Build Process [140 minutes]

Follow the instructions of *Sensored FOC of ACI.pdf* to complete the incremental build process, and verify the contents of the field oriented control application.

Attention: Do not get ahead of yourself. There are many steps in this procedure, and it is critical that you pay attention to details. If something does not seem to be working properly, go back and make sure that what you expect to happen is what is truly meant to be happening for the build level you are on. This example project is designed to work with the exact kit and motor you are using!

DELIVERABLE 2: In Build Level 2, what happens to the *Alpha* and *Beta* components of the current command (inputs to space vector modulator, viewed on CCS plots) when *VqTesting*and *VdTesting*arechanged? Hint: pay attention to the current magnitude when the change takes place. Does the phase separation of the *qd* components change? Try to explain what is happening.

DELIVERABLE 3: In Build Level 3, what are the *Proportional, P,* and *Integral, I,* gain values you chose?

DELIVERABLE 4: In Build Level 3, what is the slip, in pu? Reduce the value of *IdRef* to 0.01pu. What is the new slip, and why did it change the way it did? What happens if you set *IdRef* to 0, and why?

5. Conclusion [10 minutes]

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