pkarplus@gmail.com Paul Karplus December 10, 2012

LPMC (Low Power Motor Controller)

EE152 Final Project Report

Summary:

For my final project, I designed a brushless motor controller that operates with 6-step commutation with a PI speed loop. There are two main reasons why I chose this project: first, I think I will work in electric vehicles or robotics after I graduate and brushless motor control is critical to both; second, I am currently working in a robotics lab designing a robotic hand and none of the off-the-shelf brushless motor controller I have found are small enough. The robotic hand will need three motor controllers and the smallest motor controllers offered by Maxon, Elmo, or Advanced Motion Controls are about three times too big.

Accomplishments
Inverter PCB designed and manufactured
Brushless motor and controller MATLAB model
Motor dynomometer designed and manufactured
PI speed loop implemented on controller

Inverter PCB:

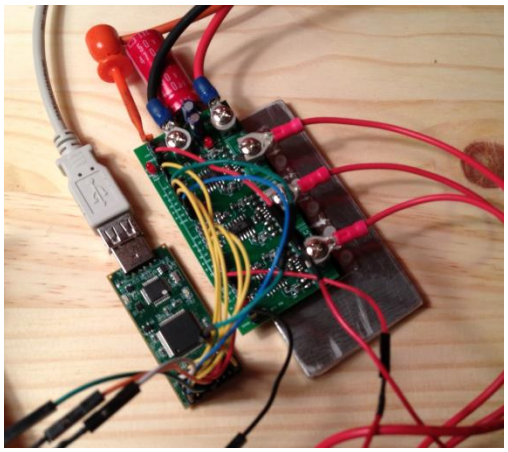

Figure 1: Inverter PCB and Piccolo controlSTICK

IC Part List		
Part Type	Part	Purpose
Microprocessor	TI F28069 Piccolo	Generate PWM for gate driver. Floating point processor. Run control algorithm. Handle user interface.
Gate Driver	IRS21864	4A source sink bootstrap gate driver. No deadtime built in. Separate high and low side PWM inputs
MOSFET	IRLB3036	60V 195A Power MOSFET
Hall Effect Current Sense	ACS711	+/- 12.5A Isolated Linear Current Sensor to measure phase current. 3.3V supply

Gate Drive:

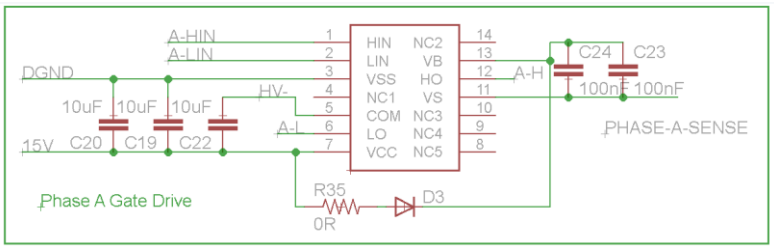

Figure 2: Gate Drive Schematic

The gate drive circuit it taken directly from the IRS21864 data sheet. The zero ohm resistor in series with the bootstrap capacitor acts as a place holder in case the inrush current to charge the bootstrap capacitors is too high and a resistor is required to limit it. The 100nF capacitors are sized to be 100x the gate capacitance of the IRLB3036 MOSFETs. The bypass capacitors on the input are sized to be 10x the gate capacitors. The bootstrap diode is sized for very fast reverse recovery (<20ns) and breakdown voltage greater than battery voltage (50V max) plus gate drive voltage (15V).

Half-Bridge:

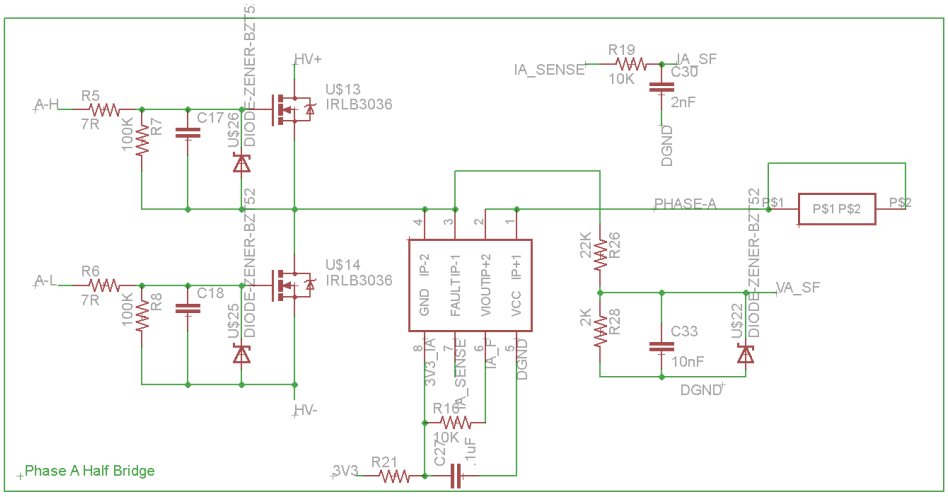

Figure 3: Half-Bridge Circuit

R5 is the gate resistor to limit the current into the MOSFET. It is an 0805 resistor and is sized to allow 2A of gate drive current (15V/7Ohm). R 7, C17, and U$26 are placeholders and will not initially be stuffed.

The ACS711 is a linear hall-effect current sensor and will measure the phase current. The phase-voltage will be measured with a resistor divider and protecting zener.

PCB:

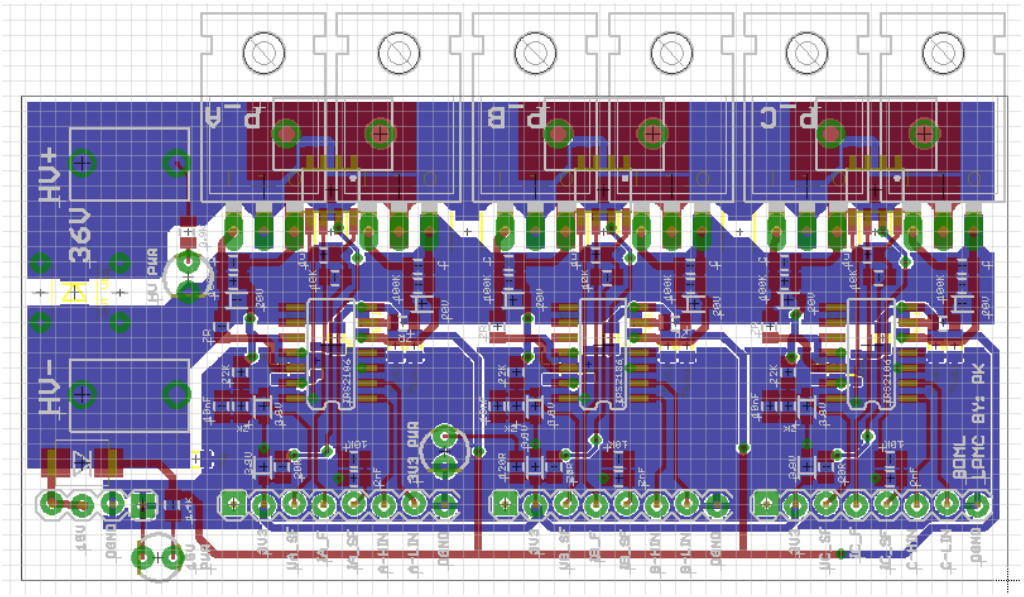

Figure 4: LPMC V1 PC

MATLAB Model:

Motor Model:

MATLAB will be used to build a 3-phase brushless motor model. The model will include electrical (phase currents and voltages), mechanical (speed and torque), and thermal (temperature) responses.

To the right you can see the motor model results. The motor parameters are from a 4-pole Maxon EC32 Flat brushless outrunner motor. The motor is fed a 1 Vpp voltage with frequency equal to the motor speed.

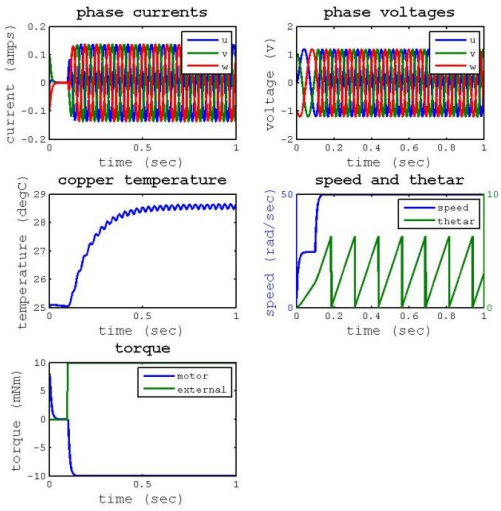

Figure 5: Brushless Motor Simulation

The motor started with zero current at zero RPM with zero Nm of torque applied. By .05 seconds, the motor quickly accelerates to 25 rad/sec. At .1 seconds, a 10mNm step in torque is applied to the motor. The motor accelerates until it can generate enough current to resist the external torque. the current also ramps up and reaches 28.5 degC steady state due to the .13 App current passing through the motor.

Controller Model:

The controller was modeled as a voltage source that is connected to HV+ when the high switch is on and HV- when the low switch is on. This causes the current waveform to be a saw-tooth instead of a smooth sine wave. I implemented a 6-step algorithm. Figure 8 shows a diagram of the 6 states. Figure 6 shows one phase current from the model and Figure 7 shows the same phase current from the scope. The scales are not equivalent between the model and scope trace but the shapes are roughly the same.

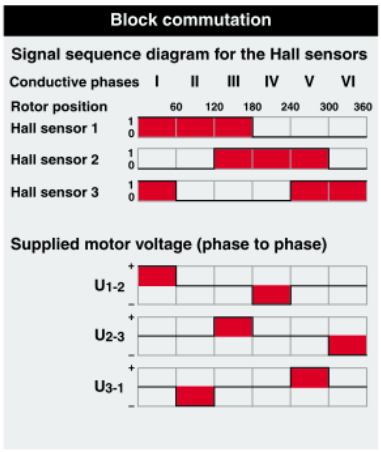

Figure 8: 6 State Diagram

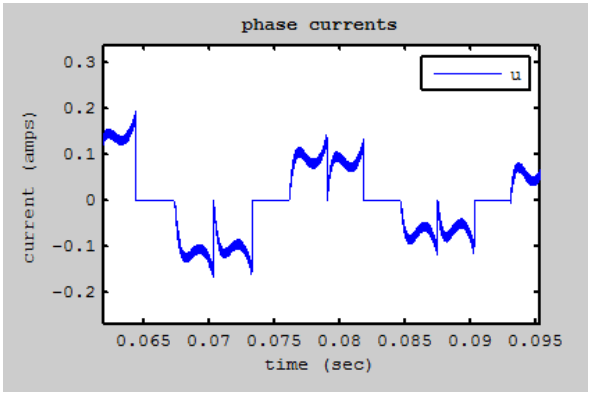

Figure 6: 6-Step Phase Current from Model

Figure 7: 6-Step Phase Current from Scope

The controller model was very helpful during the debugging process because I gave me an idea of the current and voltage waveforms that I should be getting.

Motor Dynomometer:

In order to test the motor controller under load, it is necessary to have a motor dynomometer. A motor dynomometer consists of two motors that are mounted opposed to each other and coupled together. I used a Maxon EC32 Flat as the test motor and a large Maxon brushed motor as the load motor. Torque can be varied by shorting the two terminals of the brushed motor through different sized power resistors.

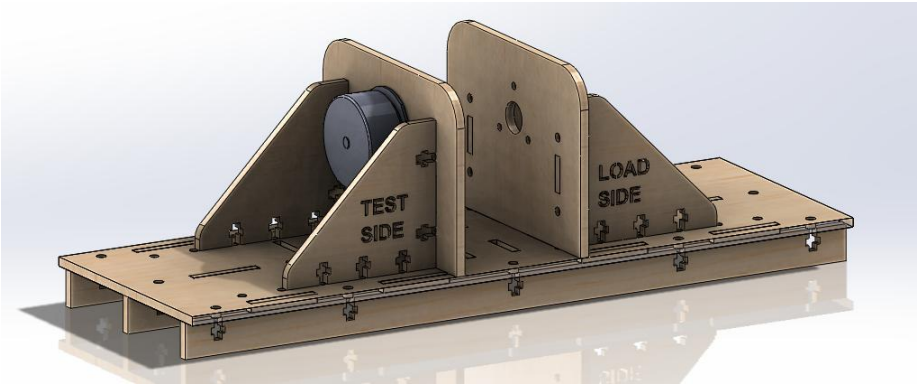

Figure 9: Motor Dynomometer Model

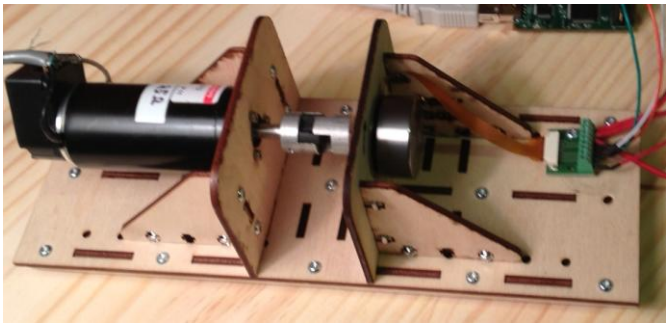

Figure 10: Motor Dynomometer

Learnings:

I could fill several pages with what I learned from working on this project. However, here is a condensed list.

- How to produce center aligned PWM with an F28069.
- How to measure the motor speed with a GPIO rising and falling edge interrupt.
- How to implement a multi-layered controller. Duty cycle is updated every PWM cycle (30Khz). Speed PI controller updates duty cycle every 1Khz.
- How to design the hardware for a three-phase inverter.
- How to use EAGLE PCB layout software.
- How to trigger ADC measurements with the F28069.

Key Results:

I used a PI loop to control the motor speed. Figure 11 shows the current resulting from a step change in target speed. It took a while to tweak the Ki and Kp gains to get the desired fast but over-damped performance. The target RPMs can be set via the a serial command line interface.

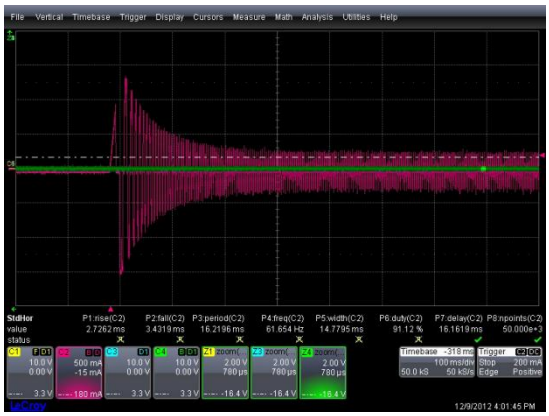

Figure 11: Phase Current from a Step Change in Target Speed

Next Steps:

The next step for this project is to close a PI loop around phase current. There are several aspects to this. First I need to measure the phase currents with the ADCs in the microcontroller and make sure the signals are clean. I tested this and Figure 12 shows the resultant ADC readings. Readings were taken every PWM cycle (30Khz) and the calibration is incorrect. However, the shape is smooth and matches the shape of the oscilloscope current measurement and the noise seems tolerable.

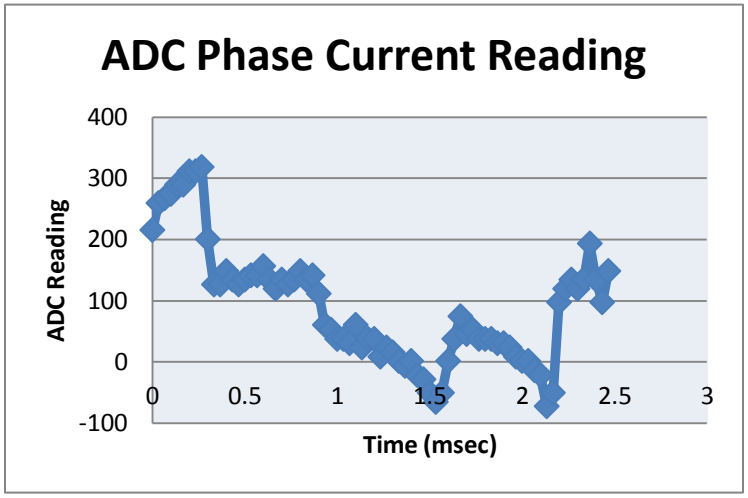

Figure 12: ADC Phase Current Reading

After implementing current control, I would like to implement sensorless control. The way the hobbyist motors do sensorless control is a lot simpler than I thought. In each of the six steps, one phase is pulled low, one is PWM, and one is allowed to float. By measuring the voltage of the floating node, it is possible to estimate the BEMF of the motor and from BEMF you can estimate speed.

After that, I would like to implement sinusoidal commutation where I use the hall sensors to get started and then switch to sinusoidal commutation after I have a good lock on motor speed. This should greatly reduce torque harmonics that cause acoustic noise.

After sinusoidal commutation, I would like to try field oriented control where I transform the stator currents into the rotor frame and use a PI loop to control them.

I find brushless motor controllers a fascinating topic hope that I will have the opportunity in the future to continue working on them.