1 Base Goal

The FPGA motor controller is a very hardware–dependent project. As such, there is a nonzero probability that the hardware will not work at all. Fortunately, a substantial portion of the project can be developed purely in simulation. The field oriented control loop can be tested by writing a Verilog module that emulates the behavior of a brushless motor. The motor model will take in simulated voltages and rotation speed and output phase currents and a resolver reading. The internal resistance, self–inductance, and Kv can be parameterized to match a real motor.

This motor simulation is an extremely valuable tool for debugging the control loop because it allows for changes to be made and tested entirely in software. To test performance, artificial delays or noise could be added to sensor data. Additional features like variable reluctance can be added to the motor for a more powerful model.

2 Minimum Viable Product

The minimum viable product is to make a motor controller that can actually control a motor. With knowledge of the rotational inertia of the motor, the torque can be measured by applying different target currents to the controller and analyzing the behavior of the phase currents and the motor velocity. This measurement can be done both with external tools to measure the motor performance or by outputting internal data through the ILA. The motor should also sound qualitatively satisfying. It should spin up cleanly, rather than sounding like it ate a bag of assorted fasteners.

3 Stretch Goal(s)

We do not plan to implement all of the stretch goals listed here. The most practical goals for us to achieve are parameterizing Q and D currents to optimally achieve field weakening and sensorless control, as these would be the most practical for an electric scooter–mounted motor.

The brushless motor we are using is an internal permanent magnet motor (IPM), where the rotor magnets are buried inside of a ferromagnetic rotor. Since IPMs have ferromagnetic material between the magnets and the coils, varying the D current through the motor can adjust the internal field strength, meaning that IPMs field weaken extremely well. Parameterizing characteristics of our motor with different rotation speeds and Q and D currents could lead to high efficiency over a wide range of speeds.

Once torque control works for the motor controller, there are many possibilities for making the motor controller more exciting. One option is to add sensorless control to the motor controller. By indirectly measuring the back–emf, the motor controller can detect the phase angle of the motor without using an angular resolver when the motor is spinning. Sensorless control can be combined with resolver data so that sensorless control kicks in at higher rotational speeds, when the resolver lag starts becoming noticeable.

Another option is to add servo control to the motor. By measuring position and velocity of the motor, the motor can be commanded to different positions and velocities with high accuracy by using the torque as a setpoint. We could potentially connect a large inertial mass to the motor and have it move the mass to a desired position.

We would like to eventually have a serial interface to control the motor, allowing us to setup the motor controller parameters from an external device, rather than reprogramming it. The serial interface could also be used to provide torque and speed commands to the motor controller.