Project: PID motor control

EE 554

11.8.2010

Matthew Aurand
Andrew Murray

Introduction

The purpose of this project was to design an embedded motor control system capable of regulating the speed of a supplied DC motor. This control system was implemented on the Altera FPGA DE0 demonstration board. Altera Quartus II and NIOS II Embedded Design Suites were used to create the System on a Programmable Chip (SOPC) controller which is programmed into the FPGA. The motor interface is provided by a Pulse Width Modulation (PWM) subsystem. The PWM signal is connected to the DC motor via a MOSFET to facilitate high-current output. The signal from the PWM is what influences the speed of the motor. When the duty cycle is set to 100% the motor will be running at full speed. The speed of the motor will decrease according to the PWM duty cycle.

In order to monitor the actual rotation speed, an optical encoder is used. The encoder generates 30 pulse outputs per complete rotation. The output signal of the encoder is connected to an input-capture subsystem programmed on the FPGA board. The microprocessor SOPC does the necessary calculations in order to determine the speed of the motor.

Without compensation, the system described above is an open-loop controller. This means that the system will not be very accurate and be susceptible to disturbances. In order to correct this, a control system needs to be programmed. A closed-loop control monitors the process output and actively adjusts the PWM signal to reduce or eliminate error.

The type of control system used is Proportional-Integral-Derivative (PID). The basic algorithm for PID control is:

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t)$$

This equation consists of 3 parts; the proportional section, integral section and derivative section. The proportional gain term, K_p, linearly adjusts the output based on present error. The second gain term, K_i , is the integral gain. This term is multiplied by the accumulated past errors of the system. This term speeds up correction and eliminates the steady-state error. The final term, K_d, is the derivative gain and is multiplied by the difference in error over time. This slows down the rate of change of the output and commonly reduces the overshoot present in a system. By adjusting the values of the gain terms you can tune the response of the system to perform optimally.

Hardware

The motor system controlled consists of a slotted aluminum disk mounted to the output shaft of a common & inexpensive Johnson can-type motor. This slotted disk provides velocity feedback to the controller by means of an optical emitter-detector system. There are 30 total slots in the disk, resulting in the following relationship:

$$RPM = 2/T$$

Where T is the interval time (in s) between each slot. The motor is interfaced to the control system with a PWM-driven N-Channel Logic Level MOSFET.

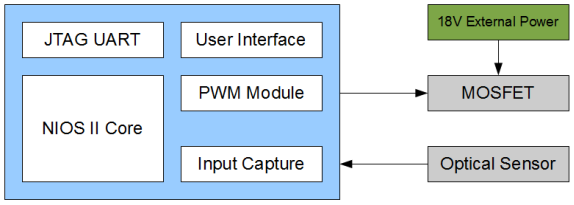

Figure 1: Hardware Configuration.

The PWM module was constructed on the FPGA fabric using Quartus II to provide 8-bit resolution, with an output frequency of approximately 10KHz. 10KHz was selected as adequate for most motor control applications. Higher rates can easily be used with little hardware modification, although more attention to driving the MOSFET gate capacitance should be exercised at large switching frequencies.

The input capture module was created around the measured motor characteristics. The minimum open-loop interval measurement (maximum rotational speed) was about 1ms, and the maximum interval (minimum rotational speed) was approximately 30ms. For a 1% worst-case measurement resolution, a period of 10.24us was used for clocking purposes. Overflow and stall detection hardware is provided to ensure accurate information is fed into the controller.

A crude user interface was created to allow the controller to be modified in-situ. A bank of switches provides 8-bit speed setpoint manipulation, and allows test or run modes to be entered. Run modes simply track a user-setpoint (either PWM or RPM depending on PID enable bit). Test modes output data from autonomously-performed step characterizations.

Characterization

The motor system was initially characterized by a range of measurements to determine the linear operational range of the mechanical system. The deadband extends from 0V to about 1.5V. Outside the deadband, the voltage constant was measured as 110rpm/V. This changes by 10% over the [roughly] linear range of applied input voltages [3-18V].

Next, an open-loop step response was applied from 0V to 12.6V [0-70%].

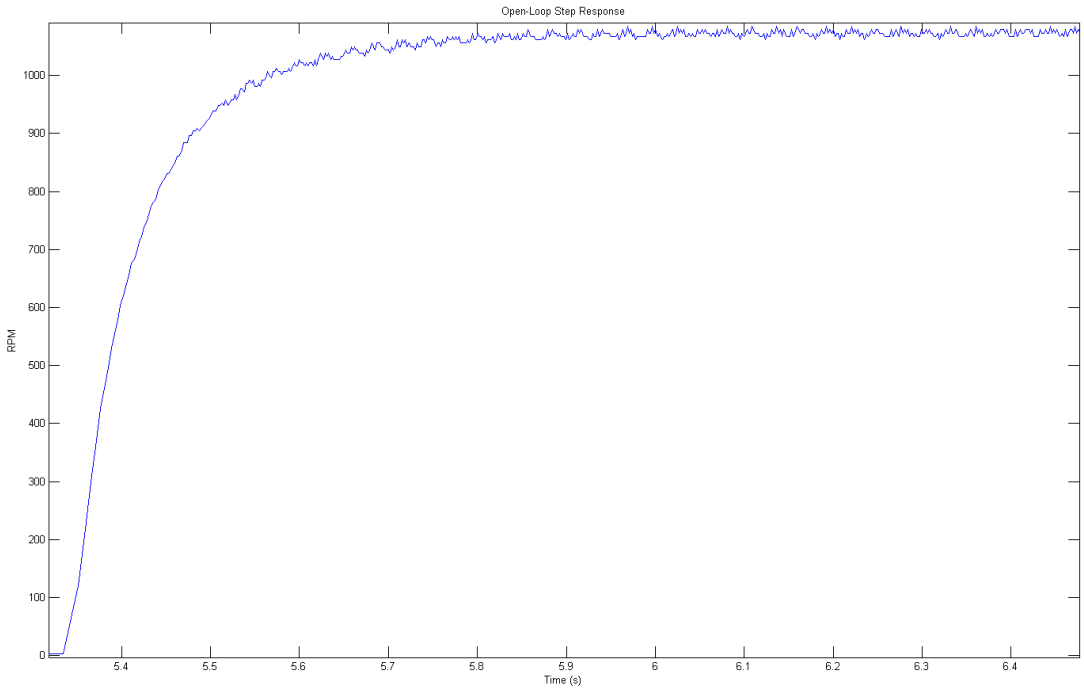

Figure 2: Open-loop step response, 0-12.6V.

From this response, the time constant of the motor was found to be approximately 100ms. This leads to a open-loop plant transfer function of:

$$G(s) = 10 / (s + 10)$$

Control

The controller is a standard-form discretized PID. This function is executed at each slot edge detection, maximizing temporal resolution. This technique has implications not encountered with fixed sample-rate controllers, as the system gains effectively change as a function of measurement frequency. In a higher-order implementation, the system coefficients could be scaled linearly as a function of measured interval frequency to optimize response over a wide range of setpoints. As implemented, the controller works best for mid-range setpoints (where it was compensated). Results are still acceptable elsewhere, but could be improved if necessary. An alternative to linearly-variant gain coefficients is to use gain-scheduling, which is easier to implement and very common in PID-based systems.

The closed loop response was tailored in MATLAB until suitable results obtained, at which time the necessary coefficients were transferred to the controller.

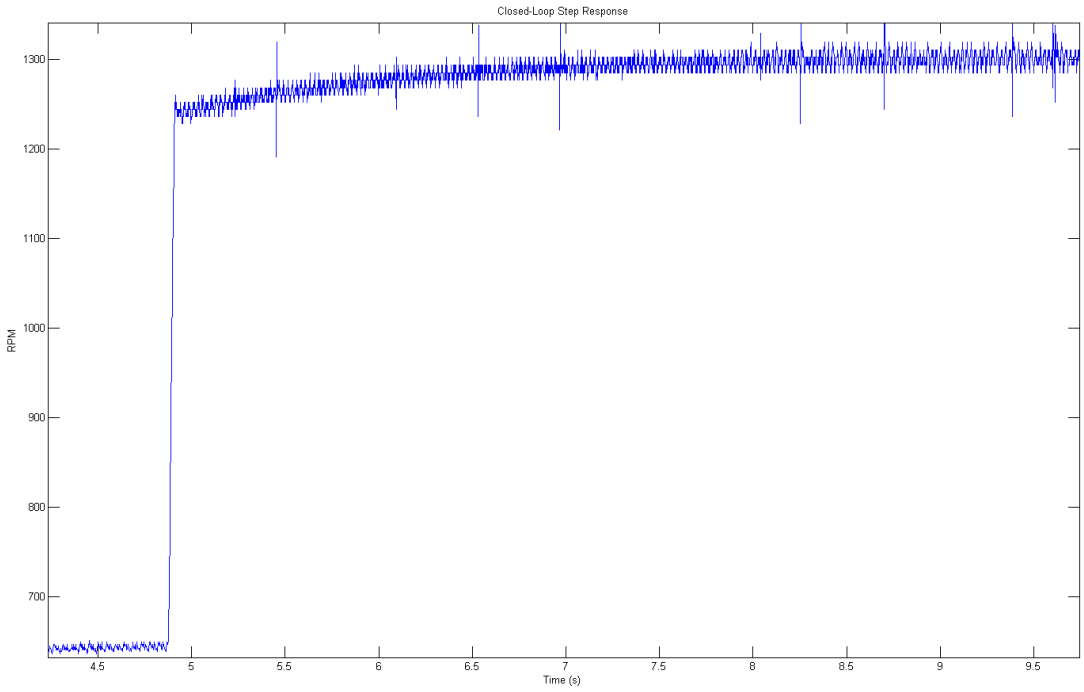

Figure 3: Closed-loop PID control of the system responding to step input from 640 to 1300rpm. K=7, I=0.1, D=0.01

The response is slightly under-damped. The selected coefficients provide no overshoot and no steady-state error. Settling time to 10% is less than 1 second, with 1% settling after 2 seconds. The system is stable for all setpoints (60 to 2000rpm), with optimized response at setpoint 1000rpm. The gains could be increased to reduce settling time at the expense of stability across the operational range.

Summary

The DC motor controller is an interesting problem. The hardware used for the demonstration has a steep learning curve, with a serious time-investment required for success. After learning the hardware, however, the outstanding flexibility of the FPGA/embedded controller became evident.

PID control appears simple until users have applied and attempted to tune systems manually. What appears mathematically and conceptually trivial is actually fairly involved. It is good experience to posess.