EE 554 Midterm Project:

Implementation of an Embedded Control System on an FPGA

Ian Johnson, Scott Freund, Luke Euker, Tyler McCracken

October 31, 2010

Introduction

The objective of this project was to implement an embedded control system on an FPGA. Specifically, the motor speed of a DC motor was controlled using a control system implemented on an Altera DE0 board. It was first required that the motor be characterized. Once that was complete, the actual control system was implemented on the FPGA and interfaced with the motor. Finally, the results were verified graphically and through demonstration.

Procedure

The characterization of the motor involved calculating the time constant of the motor. By observing the time between successive edges of a square wave resulting from an input voltage to the motor, the corresponding RPM value can be found. This value, along with a timestamp, can be plotted and used to calculate the time constant(τ) by observing the 2/3 rise time. Our determined value of τ was 0.08. Another consideration was that of the sampling rate. Too high a sampling rate would result in too much data for NIOS to handle but too low a sampling rate would produce too few data points creating a poor rise time for the motor characterization. Using an assumed RPM of 3000 and the characteristics of the motor, we determined the frequency of the motor to be 50 Hz. This value was then multiplied by the number of values obtained per revolution to obtain a value of 800 Hz. We then sampled at ten times that value, or 8 kHz.

A PID controller was selected for this particular project because of the versatility of the controller. The PID controller consists of three parts, the proportional term, the integral term, and the derivative term and these terms are in turn influenced by their respective gain constants. By tuning these variables, we can control the response of the motor to a desired input. The proportional gain controls how much the system responds to the current error. Tuning of the proportional term is important because too small a value results in inadequate response to a disturbance while too high a value can cause the system to overcompensate for error and leads to instability. The next term, the integral term, sums the instantaneous error over time, producing an offset that should have been corrected previously. The importance of the integral gain comes from its control over the steady state error of the system; a high gain value will reduce the deviation of the system from the desired input, but too high a value can cause an increase in overshoot and rise time. The final term, the derivative term, influences the rate of change of the system and controls the magnitude of the overshoot. This means that higher values of derivative gain will decrease overshoot but increase the time the system takes to settle. Determining the optimal values of these three gains was vital to ensuring a properly functioning system.

For this project we elected to solve for our gains in terms of K_d. Using the equations $K_p = K_d * (2 * Zeta * W_n)$ and $K_i = K_d * (W_n)^2$ we controlled the various parameters of the output such as rise time and percent overshoot. Values of Zeta = 0.7 and W_n = 17.857

were used for all calculations. The value of zeta was chosen based off the defaults used in class and W_n was then calculated.

Results

Once the code was complete and the motor was interfaced, we ran several trials with the DC motor. In the tests, a target speed was input and the controller would respond by driving the motor to the desired speed and then maintaining that speed for the duration of the test. In order to test the functionality of our controller, we periodically printed the speed value to the console. These values were then plotted in MATLAB so that we could observe the behavior of the motor over the test period.

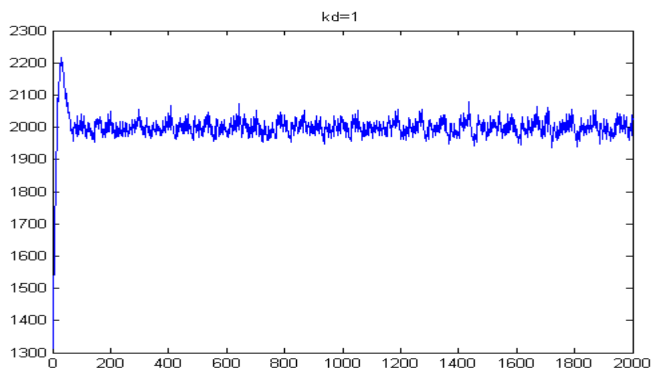

Figure 1: Sample motor output (Kd = 1)

The above figure plots the motor RPM on the y-axis and the sample number (5ms spacing) on the x-axis. As the title states, this data was produced with a K_d value of 1. It is clear from the graph that a low K_d indeed results in a very fast rise time but a relatively long settling time. In addition, the low K_d value results in a manageable overshoot relative to the steady state value. Another observation made is the effect of the K_i value on the output. Because our K_i value is directly dependent on our K_d value, it is also low for this test and is evidenced by the ringing, or deviation from the desired speed of 2000.

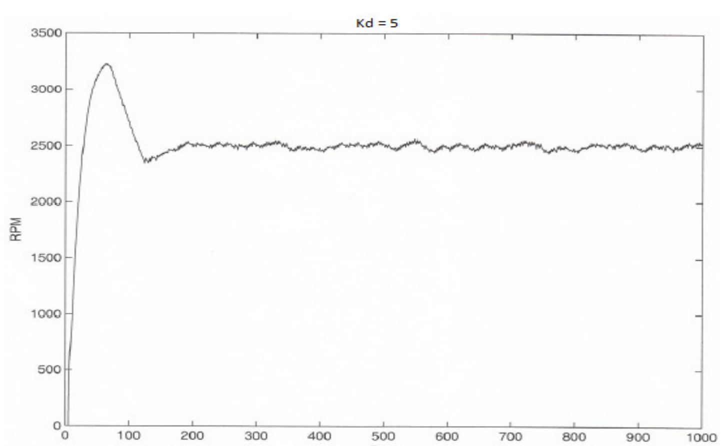

Figure 2: Sample motor output (Kd = 5)

Figure 2 shows the motor response to a desired speed of 2500 RPM and a K_d value of 5. This higher value of K_d results in both a much longer rise time and a longer settling time than a value of 1 as is evident in the graph. However, it also produces a high K_i value which results in a much smoother steady state signal(less ringing). Also evident in this graph is the quick response of the system to steady state error due to a higher K_p value.

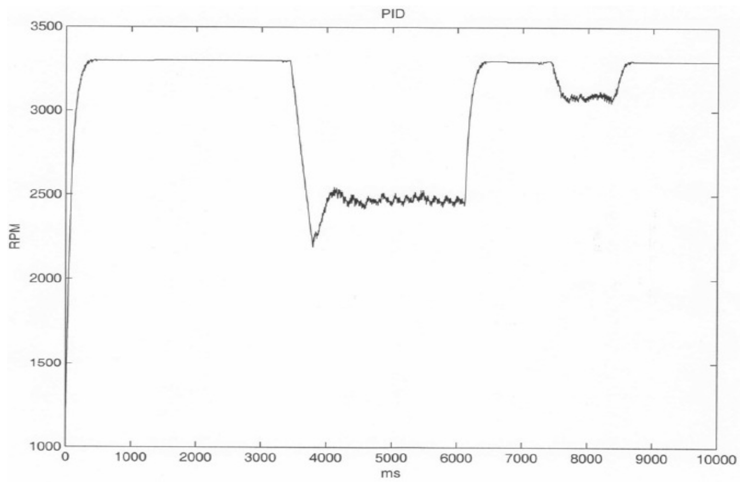

Figure 3: PID response to changing speeds

This final figure shows the system's response to several changes in desired speed over a short period. As the graph shows, the system is capable of rapidly adjusting its speed to accommodate for a dynamic user input.

Summary

In closing, our controller has been successfully implemented and interfaced with the motor. We have also found that varying the gains produces changes that are consistent with the behavior of an ideal PID controller.