

ME 135 SPRING 2008

ASSIGNMENT 5: PID CONTROL OF A SINGLE MOTOR

Due: Tuesday, April 1, 2008

Background

Each of the PWM channels is equipped with both an H-bridge for switching the direction of the motor and an encoder for recording the position of the motor. Operating the Scorbot is simply a matter of directing six of these motors. For this assignment, however, we will start with a single motor. The black box provides the current needed to drive the motors.

As always, make sure that the motor is connected to the PWM channel, that the black box is powered, that it is connected to the ETS computer, and that you have chosen the correct ETS as your target device. Then try the test application before you run your own. Note that you must also specify the ETS computer in the CVI program.

Although it is possible (and possibly easier) to complete this assignment in LabVIEW, you are required to use LabWindows CVI. Because the R Series DAQ cards are designed for LabVIEW, however, a LabVIEW program is used to publish encoder counts to the network in the form of a shared variable and to generate PWM signals based on another shared variable. This LabVIEW program must be running in order for the CVI program to interface with the motor. Note that due to hardware restrictions this setup (using LabVIEW to operate the FPGA and CVI to generate the Windows interface) is the reverse of the desired configuration. Typically you would want to use C on your FPGA for speed and memory and LabVIEW on your Windows computer for the interface.

Task

Your task is to implement a discrete PID controller for position control of a motor. Allow the user to adjust the proportional, integral, and derivative gains manually. Then, plot the motor response against a step input and calculate the 10-90% rise time, 10% settling time, and overshoot on your interface. Simulating the motor response is useful but optional.

Adjust your gains to minimize the rise time, the settling time, and the overshoot. One technique for tweaking the gains is to start by ramping up k_D , then balancing that with k_P , and finally using k_I to eliminate any residual error. Record these three sets of gains and plot the motor response for the respective conditions in your report.

Note that both a LabVIEW and a CVI project have been included in the example. The LabVIEW VI is run remotely on the ETS computer while the CVI is run locally in Windows. You should not have to modify the LabVIEW project, but make sure you understand how it works for future assignments. Again, the LabVIEW project must be running for the CVI to work. Use the NI Variable Manager to monitor the network variables in real time.

Deliverables

A quick start guide explaining how to use your interface. A report including the gains, performance characteristics, and plots when minimizing rise time, settling time, and overshoot.

All of the C/C++ files (.c, .h, .uir, .prj, .cws) needed to make your program run. The LabVIEW files need to be included only if you modified them. Please comment your C code thoroughly, as always. Part of your grade will be based on the clarity of your code.