CS 431 Lab #8
Two-dimensional Discrete PID Control with Dynamic Setpoints

Spring 2015

Demos will happen at the week of April 15th.

1 Overview

In this lab you will extend your lab07 program for PID control to two dimensions. Furthermore, you will add a new capability to dynamically specify the setpoint for both dimensions simultaneously via the joystick. You must write a program that does all of the following:

1. Use PID control to balance the FLEX board platform in two dimensions. Initially, set the setpoints to the middle positions for both the X and Y-axis (50 points).
2. Use the joystick as a way to specify new setpoints for both dimensions. When the joysticks trigger button is pressed, set the setpoints for both the X and Y-axis of the FLEX board platform according to the current position of the joystick for both the X and Y-axis, respectively. Note that you must compute a mapping from joystick position to touchscreen position in order to update the setpoints (35 points).
3. Display the following information clearly on the LCD (15 points):
   - The current position of the ball on the touchscreen for both the X and Y-axis (Cur pos)
   - The X and Y-axis positions of the respective setpoints (Set pos)
   - The tentative X and Y-axis positions of the respective setpoints if the joystick trigger were to be pressed (Joy pos). This value should be converted to its touchscreen position mapping value before being displayed.
   - The PID coefficients K_p, K_d , and K_i for both the X and Y-axis.

2 Procedure

1. Before getting started, review how PID control works. In particular, a good reference for this lab is http://en.wikipedia.org/wiki/PID controller#Discrete_implementation.
2. Use your code/project from a previous lab as a starting point for coding this lab and remove all unnecessary functionality. In particular, you may want to reuse the following code: lab01 for debouncing, lab02 for Joystick trigger interrupt, lab05 for Joystick ADC, lab06 for touchscreen ADC, and lab07 for PID control.

3. A demo program that demonstrates the features that you need to implement is provided in compiled form on the course website.

4. Write *lab08.c* such that it fulfills the requirements specified in the Overview section.

   (a) At the start of your program, perform the steps to calibrate the min and max positions of the joystick as done in lab05.

   (b) Using lab06, manually find the X-axis and Y-axis touchscreen ADC values that represent the middle position of the FLEX board platform and use these values in your code as the initial set points for the PID control.

   (c) Compute a mapping from joystick ADC value to touchscreen ADC value in order to use for the setpoints. Note that the range of joystick ADC values should be linearly proportional to the range of touchscreen ADC values.

   (d) Extend your function from lab07 to balance the ball at the specified X and Y setpoints. You should continue to use a timer interrupt to control the time interval for input and output.

   (e) You need to select floating point PID control constants K_p , K_d , and K_i for both the X and Y-axis and they may be hard-coded into the application. The values do not have to be the same for the X and Y-axis.

5. Experiment with your program to find some "good" values for K_p , K_d , and K_i.

3 Questions to Ponder

1. When varying the K_p , K_d , and K_i values, does the real FLEX board respond like the MATLAB model does in Homework 3? How well do you think the MATLAB simulation models reality?

2. How would you implement this: http://www.youtube.com/watch?v=ML4woERjvlk