TEMPLE UNIVERSITY COLLEGE OF ENGINEERING | ECE 3413 | FALL 2014

Lab 8: PID feedback control of a DC motor using experimental set-up | Man Vs. Machine

OBJECTIVE

THE PURPOSE OF THIS LAB **is to demonstrate the application of PID control in practice. You will implement DC motor speed control with feedback PID controller using your calibration parameters from the previous lab. The PID controller should be tuned to outperform a human operator that is manually changing system input by controlling a numbered slider.**

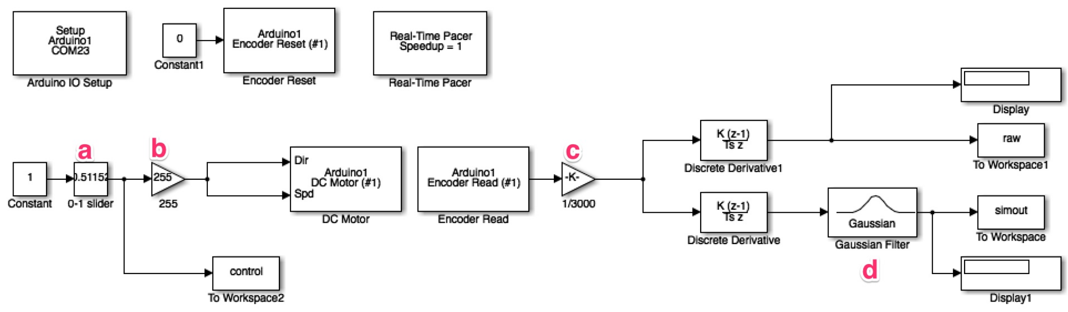
1 BACKGROUND

This lab extends our feedback loop simulation work in lab 6. , We will use our DC lab set-up along with your results from last week's lab to validate our simulation results by observing PID control behavior in experimentation. In this lab we will implement control and compare it's performance with an open loop system controlled manually by an operator.

Figure 1.1 shows the manual control diagram. The block marked (a) is the *Slider Gain* block. It allows you to manually change the gain between two constraints. Gain blocks (b) and (c) are used for normalizing input and output to 1. You will use your results from last weeks lab in place of these. Block (b) scales the 0 - 1 input by 255 - the maximum input to our motor. (e.g. a 1 slider gain on a 1 constant would produce a 255 input to the motor block).

Block (c) normalizes the output to 1 based on an arbitrary top speed of 3000 pps. (e.g. if a top speed output is 3000ppi, it is scaled by a gain 1/3000 which produces 1). Assuming that 1 input produces a 1 out this is true. Use your own polynomial mapping functions to normalize input and output.

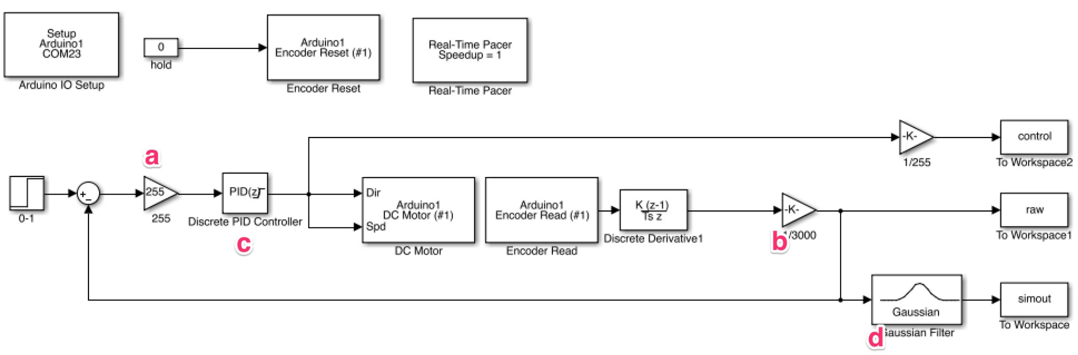
Figure 1.1: Operator-Controlled open loop system



The above also applies to blocks (a) and (b) of Figure 2.2

Note that the derivative blocks in both diagrams convert encoder pulse position to pulse per second (pps) speed.

Figure 1.2: PID controlled feedback system



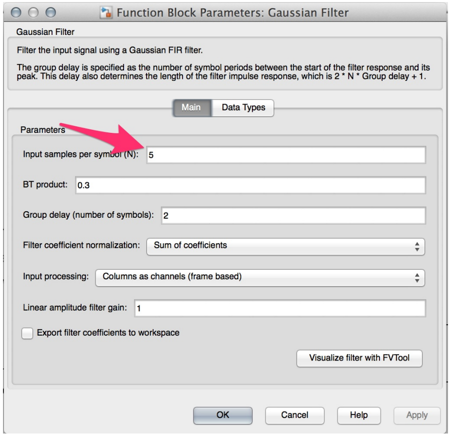
Shown in Figure 2.2 is the feedback system with a PID controller block. (c), the PID controller block acts on the system to change input to produce a desired output. The (d) *Gaussian Filter* block smooths or noisy output for better display and comparison.

2 Experiment

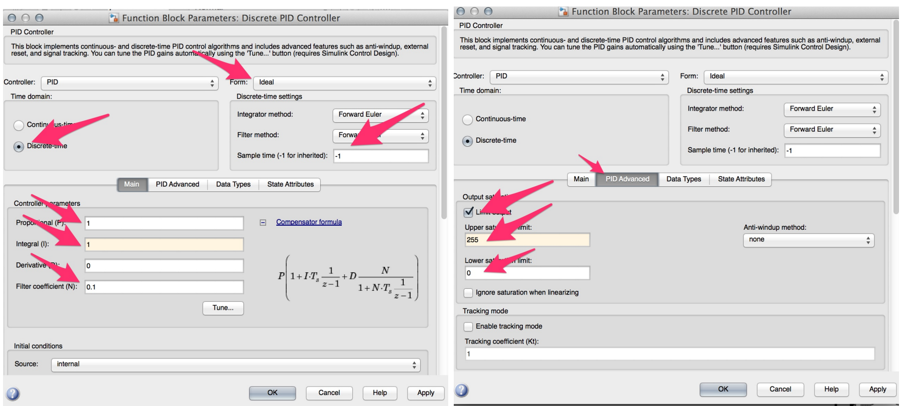
The following configuration parameters should be applied to models:

- In both models, sample time should be set to 0.05 seconds in all ArduinoIO blocks

- In both models, the system input and output should be normalized to 1
- In both models, the gaussian filter block should be set up as follows:



- In model 1.1 slider gain should be limited from 0 - 1
- In model 1.2, the PID controller block should be set up as follows:



- The simulation time for both models should be set to 10 seconds.
- The step time in model 1.2 should be set to 0.
- **The goal reference speed should be 0.5**

1. Download model 1.1 from Blackboard, modify with your calibration block from the previous lab.
2. Construct model 1.2 with your calibration.
3. Tune the P and I parameters in model 1.2 independently to achieve the fastest possible settling time, leave D at 0.

4. Attempt to achieve the 0.5 reference speed with model 1.1 by manually changing the slider gain, try to reach and settle on 0.5 as fast as possible.

3 Deliverables

Include the following in your report:

1. With a proportional gain **P** of 0.1, on a single plot, graph the control signal along with raw and filtered outputs. Do this for different levels of integral **I** gain ranging from 1 - 20.
2. Comment on how proportional controller affects the system.
3. With an integral gain **I** of 1, on a single plot, graph the control signal along with raw and filtered outputs. Do this for different levels of proportional **P** gain ranging from 0.1 - 1.
4. Comment on how integral controller affects the system.
5. Include a single plot, graph the control signal along with raw and filtered outputs of your best response with the fastest settling time for reference 0.5. What were your P and I values to achieve this?
6. Describe the approach you took to determine the P and I values.
7. Using the filtered response, which performed better the manual control or the PID? By how much? Assume that you are measuring from the time input was actually entered.
8. Again, with the filtered response, calculate the percent error of the manual and PID response from the unity step function beginning at zero.