Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Report Cover Page

Course Number: ECE 3412

Lab 5: DC motor model for PID feedback control in MATLAB Simulink

Student Name: Devin Trejo; Robert Irwin
TUid: 914924557; 914980083
Due Date: 3/10/2015

TA Name: Michael Korostelev

Grade: / 100

I. Introduction

Lab 5, we experiment with feedback in a control system using an external controller. Feedback control is important since we often cannot alter the intrinsic characteristics of hardware, i.e. the DC motor used in Lab 4. In software, it is possible to simulate the response with different damping and inertial characteristics, but it is not possible to change the damping ratio or moment of inertia without assembling an entirely different motor. For this reason, it is imperative to experiment with an external controller has the ability to alter the response based on the current point in the simulation, and the intended steady state value. PID controllers do this exceptionally well, and simply adjusting the constants, P, I, and D can alter the characteristics of the response.

PID controllers have the following response in the time domain.

$$u(t) = Pe(t) + I\int_0^t e(\tau)d\tau + D\frac{d}{dt}e(t)$$

Where:

- ***e*** - Represents the error between the current output and the steady state output.
- **P (Proportional term)** - Produces an output that is proportional to the error between the current value and the steady state value.
- **I (Integral term)** - Produces an output that is proportional to the magnitude of the error and the duration of the error.
- **D (Derivative term)** – Is determined by finding the slope of the error.

With these constants, the PID controllers attempt to minimize the error of a system.

II. Procedure

Before we begin the lab it is important to understand the different system responses we may encounter. That way we know the various inputs our PID controller will face.

The response we observe in the time domain is termed the transient response. There are four possible outcomes:

- **Over-damped –** There is lots of damping causing the system to gradually reach steady-state.
- **Critically Damped –** There is enough damping to just stop the transient so there is no transient.
- **Underdamped –** The circuit has oscillation however there is damping that causes the transient to settle to steady state.
- **Un-damped –** The system will oscillate forever since there is no damping

Given the various situation we may encounter what we will test is how each term in the PID controller effects our output. There are certain criteria we wish to minimize or maximize within a system's response. For example the **rise time**, is the amount of time it takes the signal to go from 0.1 to 0.9 of its final value. We also care about the **settling time,** or the time it takes the response to stay within 2% its final value. Lastly, the **percent overshoot** notes the amount your signal goes over your desired final value. When designing a system it is important we know the parameters, and usage case scenarios our PID controller will operate in.

To begin we will test the effect of the P term by changing P and keeping I and D set to 0. There are four cases to test $P = 1, 10, 100, 1000$. To maximize the time we have available to us in lab, we will use a for loop and sim() command to test each case. After we test each case we can plot the individual results on the same graph to see a comparative result.

Next we will experiment with changing our Integral gain. Since proportional gain effects our output the most, we will have $P = 100$ but keep D set to zero. Again using a for loop we can sweep across a set of different I values to see the effect it has on our output. In this instance we test $I = 0.1,1,10$. To see the effects of P we also test a case where $P = 10, I = 1$.

The last term in our PID controller is the derivative gain. Following a similar procedure laid out above we test the effects on our system when $P = 10, I = 1$ and $D = 1, 10, 100$.

III. Results

Below we can see the effect of altering the proportionality term. It is important to note that the constants I, and D, are set to 0 for the following simulations.

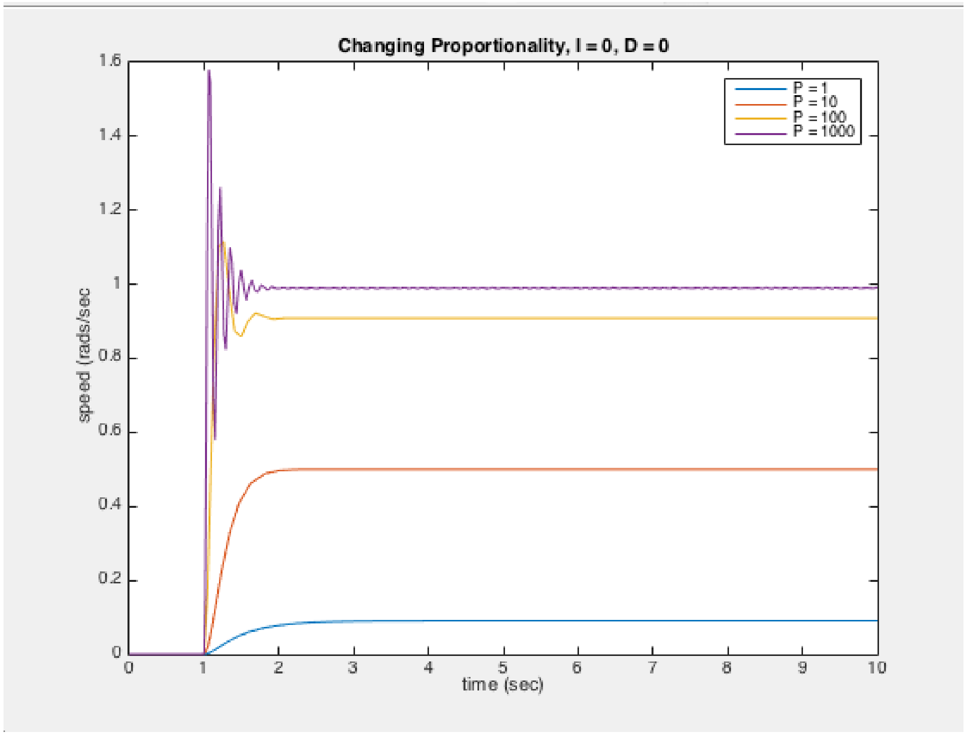

Figure 1: Changing Proportionality Constant

As we can see, if the proportionality constant is too low, the system can never reach its intended steady state value. There is a caveat, however. If the proportionality constant is too high the result is a large overshoot, and a larger settling time.

Now we will look at the effect the integral constant has on the response.

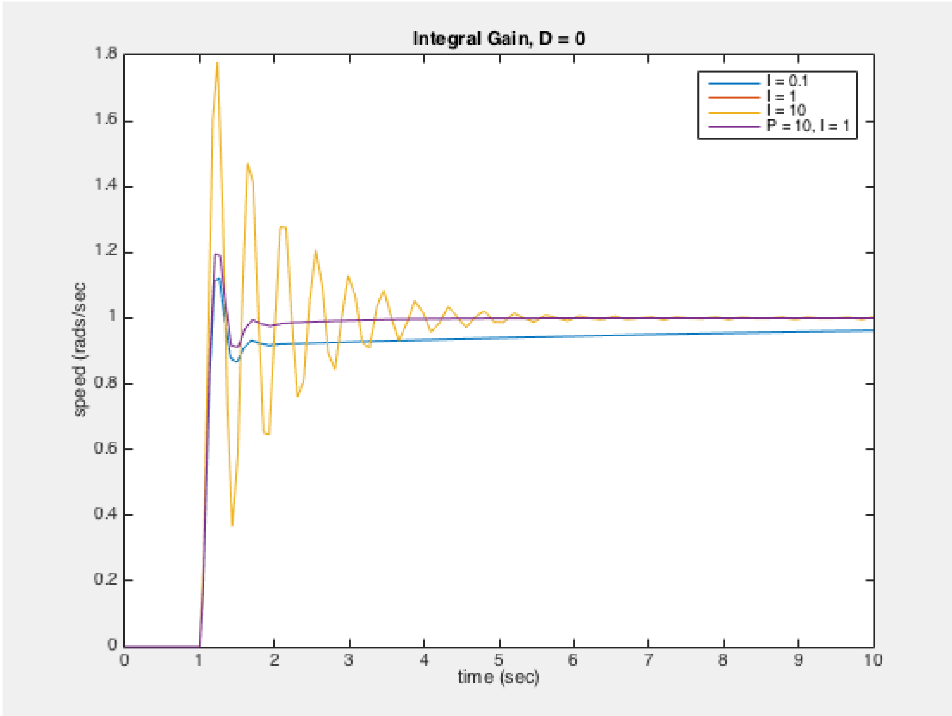

Figure 2: Changing the Integral Gain

In the figure above, the proportionality constant is set to 100, except for the response in purple, which has a proportionality constant of 10. Again, we see that as the integral constant is increased, the overshoot, as well as the settling time are increased.

We also notice that when the proportionality constant is decreased to 10, and the integral gain is kept at 1, the response has a reduced overshoot and settling time.

Below shows the effect when the derivative gain is applied along with the proportionality constant and the integral gain.

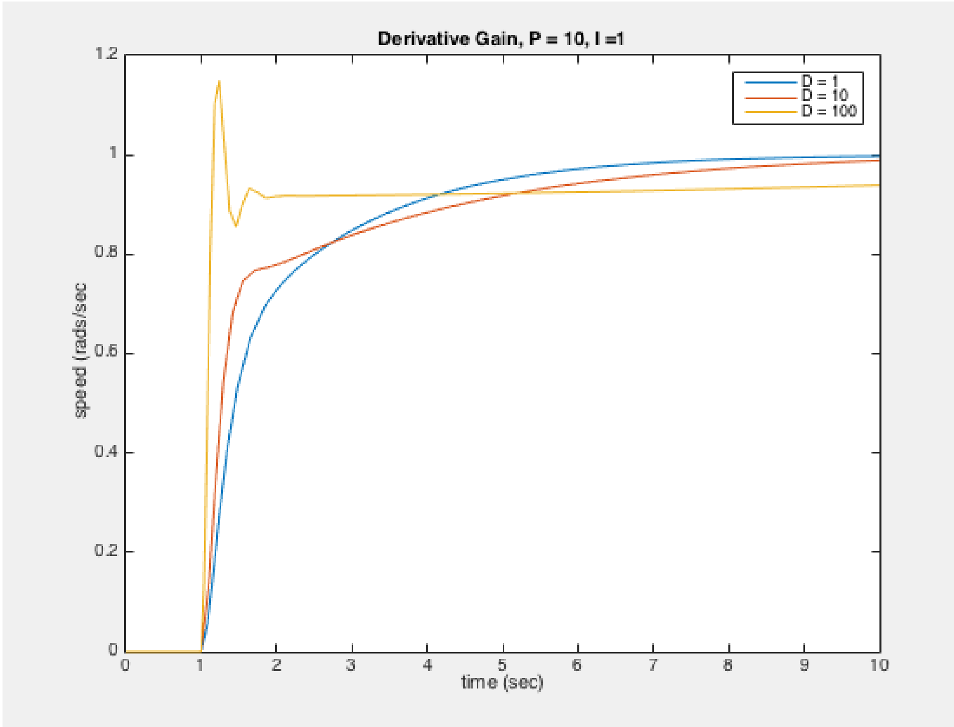

Figure 3: Changing the Derivative Gain

In the figure above, we notice that the best response is when the derivative gain is set to one. It is also important to note that the proportionality constant is set to 10, and the integral gain is set to 1.

From the three figures above, it is evident that by changing the values of P, I and D, one can manipulate the output of a system to achieve the desired response. We notice that P effects the steady state value of the response. If P is too small, the controller will not register an error even though there is a substantial error. If P becomes too large, the system oscillates frequently and the overshoot is extremely high.

The Integral gain affects the transient part of the response. If the Integral gain is large, the system becomes less damped, and oscillates more frequently, and for a longer duration.

The derivative gain also effects the transient response, and makes the output settle in a shorter period of time.

IV. Conclusion/Discussion

Designing a PID controller is important and unique to the situation it will live in. In our scenario we used a PID controller to read in and correct a DC motor's speed in accordance to a desired velocity. It correlates to driving a car with the cruise control on. Say you set the cruise control on going 60mph. A PID controller will take over to make sure the car stays at that velocity. If error is introduced for let's say you going up a hill the PID controller need to recognize the change and correct it. In the scenario of the cruise control you would want a PID system where each term interacts together to create a critically damped transient response. You would not want an underdamped system since your car would become jerky as it speeds up and slows down eventually correcting the error. You do not want no overshoot. On the other extreme however you want to avoid being over-damped since then the person behind you would begin honking their horn. Your car will accelerate to correct and reach you desired 60 mph but it would do so slowly.

The most comfortable system for both the driver and other people on the road would be a critically damped system.

The tradeoffs for making a controller to conform to certain parameters are the three system characteristics. Consider a very fast response where the rise time is very rapid. Relating to the car analogy would mean your motor goes full throttle and you accelerate as quick as possible. Cars can accelerate pretty fast, and next thing you know you're doing 100 mph, but the speed limit is 60. You have overshot your desired speed so now you need to step on the brakes to compensate. Stepping on the brakes at 100mph quickly brings you back down to 30mph. The accelerating and braking will continue to occur and eventually you would settle at 60mph, however the settling time would be tremendous. The system we experiment that best illustrates this example is Figure 2 where our integral gain was 10. The designer the system will need to account for the PID controller they wish to construct, and adjust the terms accordingly.