Temple University
College of Engineering
Department of Electrical and Computer Engineering (ECE)

Student Report Cover Page

Course Number: ECE 3412
Lab 4: DC motor model for open-loop control in MATLAB Simulink

Student Name: Devin Trejo; Robert Irwin
TUid: 914924557; 914980083
Due Date: 2/17/2015

TA Name: Michael Korostelev

Grade: / 100

I. Introduction

For the next lab we experiment working with DC motors. In the previous lab we worked with stepper motors, which operate by supplying power to one of serval magnets surrounding the rotor. The difference provided by the electromagnet causes the rotor to spin in a very precise increments. The DC motor works in similar fashion except with the magnet is permeant (armature) and while the commutator provide a field difference. Due to the rather simple construction of the DC motor we can construct a model for the DC motor using an equivalent circuit consisting of a resistor, inductor, and dynamo. By the end of the lab we will demonstrate the accuracy of our DC motor model via Simulink.

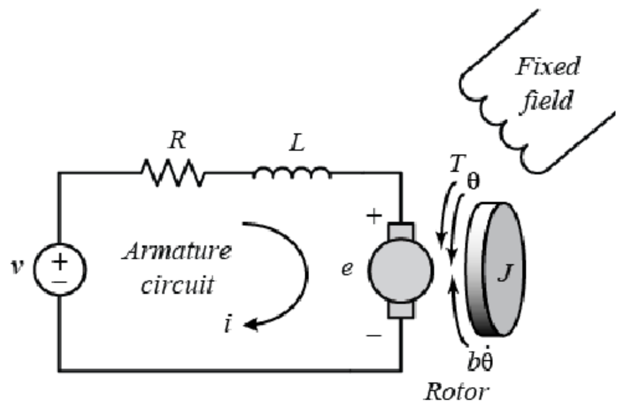

Figure 1: A DC motor model

II. Procedure

To begin designing a control for this system, we must determine the current, I, and $\frac{d\theta}{dt}$, and model the integrals of the rotational acceleration and of the rate of change of the armature current. These are given by:

$$\int \frac{d^2\theta}{dt^2} d\theta = \frac{d\theta}{dt} \; and \int \frac{di}{dt} = i$$

In Simulink, these equations are represented by the following blocks.

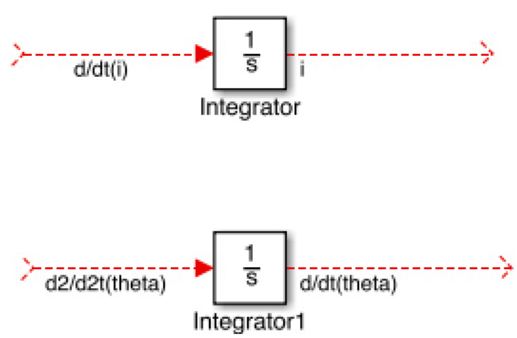

Figure 2: Blocks that Represent the Current and Differential Position

Next, we begin to model Kirchhoff's and Newton's Law. This results in the following equations.

$$J\frac{d^2\theta}{dt^2} = T - b\frac{d\theta}{dt} \quad \rightarrow \quad \frac{d^{2\theta}}{dt^2} = \frac{1}{J}\left(K_t i - b\frac{d\theta}{dt}\right)$$

$$L\frac{di}{dt} = -Ri + V - e \quad \rightarrow \quad \frac{di}{dt} = \frac{1}{L}\left(-Ri + V - K_e\frac{d\theta}{dt}\right)$$

Building off of the two existing blocks, these equations are modeled as follows.

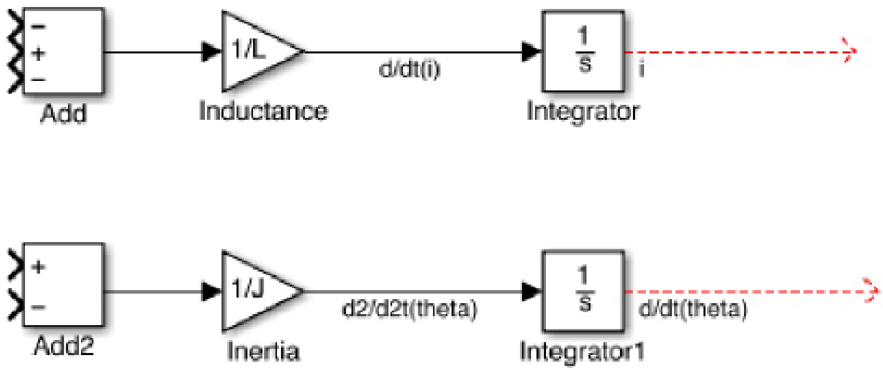

Figure 3

The next thing we must consider is the torque that acts in the DC motor. There are two unique torques: a damping ratio, *b* which slows our angular velocity and an electromotive force constant from the armature K_{ti}. These forces are added into the diagram in the following manner.

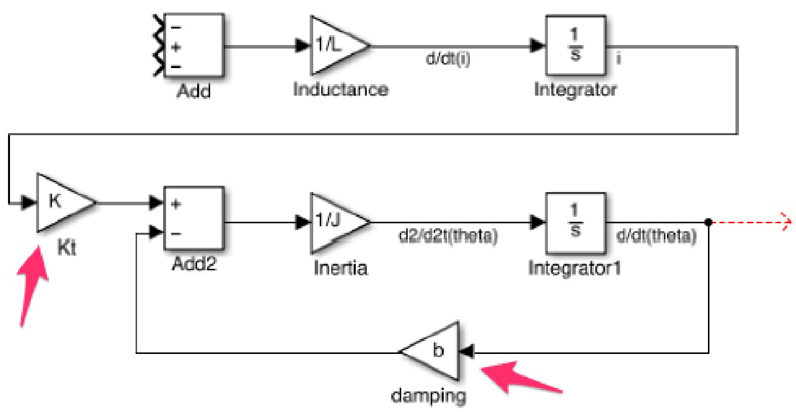

Figure 4

It is now important to add in the voltages that are present in Kirchhoff's Law. We add gain blocks for the voltage drop across the armature resistance and the back EMF from the motor. The final block diagram, with the voltages included is shown below.

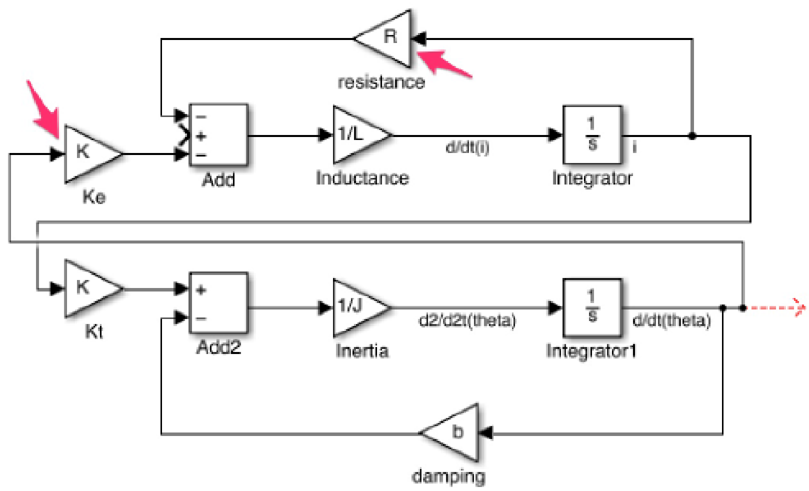

Figure 5: Control System for the DC Motor

We then apply a source voltage to the add block, and an output block to the output of the position integrator. Once this is packaged into a subsystem, we apply a step voltage, and connect a scope and an output block that copies the speed variable into the MatLab workspace. We show the process below:

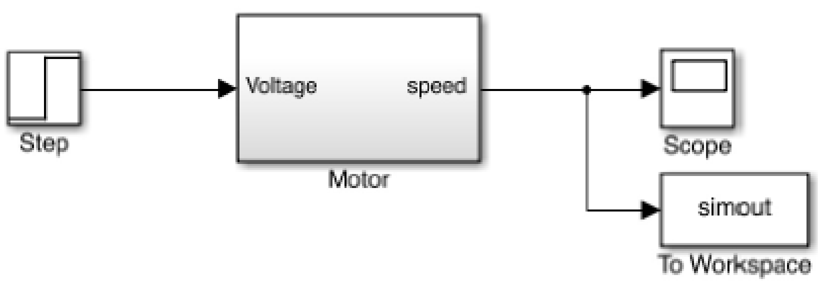

Figure 6

Before running the simulation we must first define the variables in the MatLab workspace. The variables were defined as,

```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
```

Once our variables were defined we could begin simulation. We then compared the outputs of the following combinations of inertia and damping coefficients.

$$J = \{0.01, 0.001, 0.02\}$$

$$b = \{0.1, 0.01, 0.2\}$$

By plotting all combinations of moment of inertia on one plot, and all combinations of damping coefficients on another plot, we were able to compare the outputs.

Because we know that the DC motor moves in a circle, we are interested in its movement on a scale from 0-2π. The final step of this experiment was to translate the position output to be scaled in this range only.

III. Results

The plot below illustrates the effect that the moment of inertia has on the velocity of the DC motor.

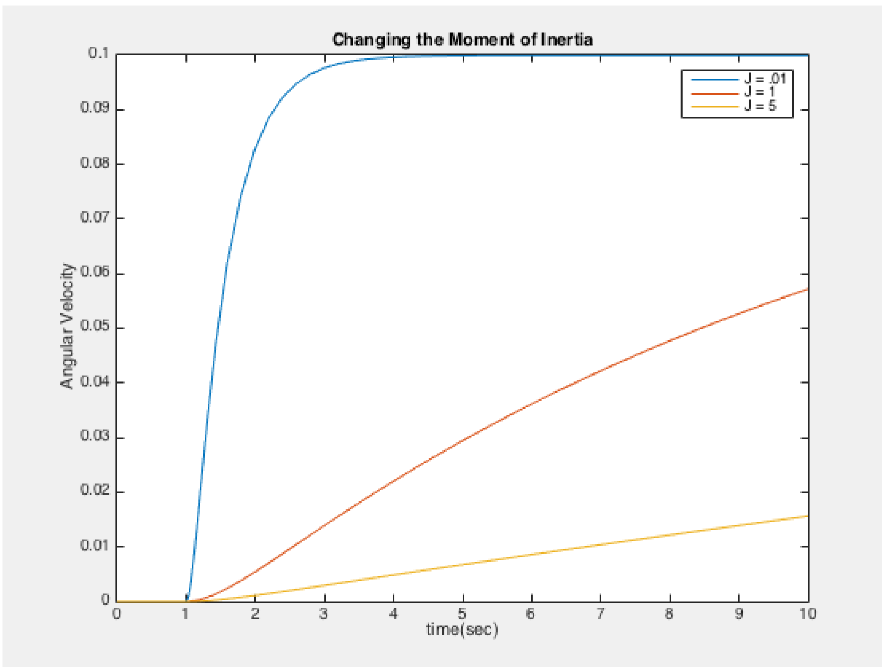

Figure 7: Angular Velocity Output with Different Moments of Inertia

From the plot above, we see that as we increase the moment of inertia, the DC accelerates much more slowly. With a small moment of inertia, we can see that the DC motor reaches its top speed very quickly. It is important to note however, that if the simulation was run longer, the orange curve and the yellow would eventually reach the same speed as the blue curve.

Now we will examine the effect that the damping coefficient has on the velocity and acceleration. This plot is shown below.

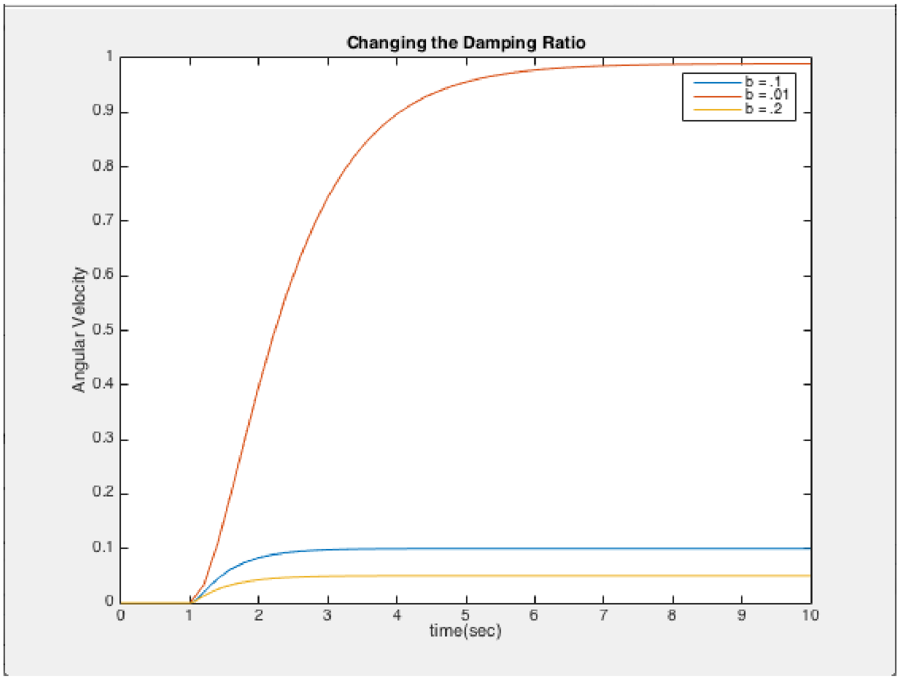

Figure 8: Angular Velocity Output with Different Damping Ratios

Again we see that as we increase the damping ratio, the acceleration decreases. A key difference between this plot and the plot of varying moments of inertia is the fact that the top speed decreases with an increasing damping ratio.

The final part of this lab was to map the position of the DC motor to a value between 0 and 2π. This is done with the following change to the block diagram:

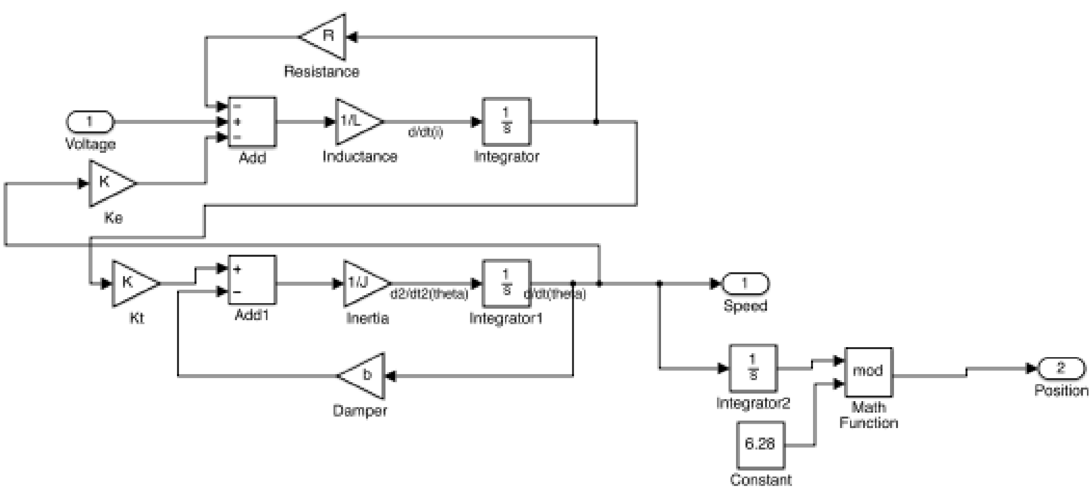

Figure 9: Circuit to calculate position between 0 and 2π

To obtain the position, we simply had to integrate the velocity, which is defined as $\frac{d\theta}{dt}$. This should make sense because $\int \frac{d\theta}{dt} = \theta$, where θ is the position. Now based on the velocity curve, simply integrating the angular velocity would result in a constantly increasing position. We know that this doesn't make sense because the DC motor moves in a circle. To limit the range of possible values between 0 and 2π, we simply used the mod function, which returns the remainder of the division $\frac{\theta}{2\pi}$. This value was then outputted to the MatLab work space. The position and velocity were than plotted simultaneously, resulting in the plot below.

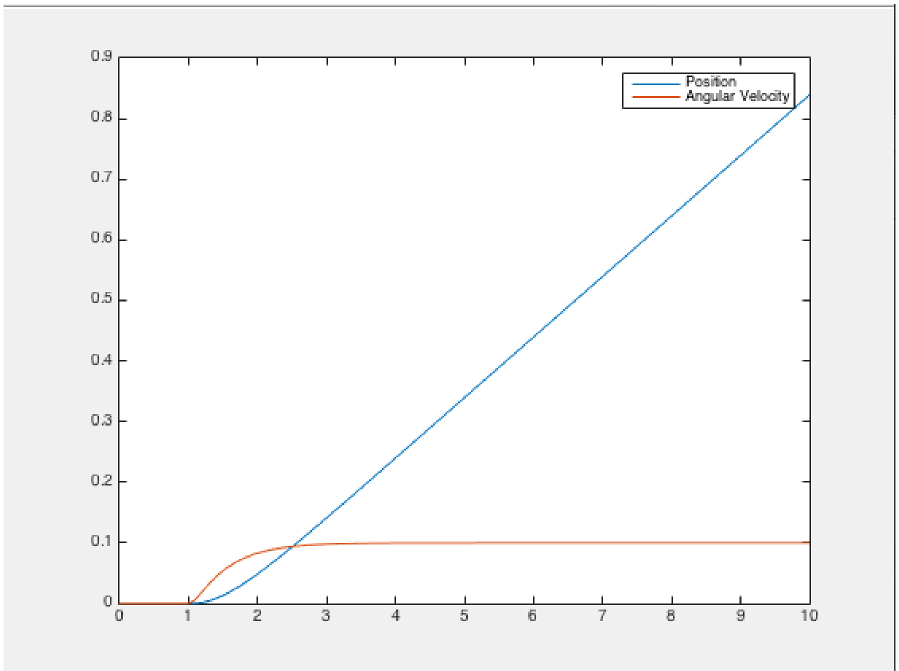

Figure 10: Position and Velocity Plot (Run Time = 10sec)

This plot makes a lot of sense if we understand the relationship between velocity, acceleration, and position. When the velocity is increasing exponentially (1<t<2), the position is also increasing exponentially. This makes sense as the position is the integral of the velocity, and the integral of an exponential is simply a scaled exponential. We also notice when the velocity becomes constant, the position increases linearly. This also makes sense because the integral of a constant is a linear function.

This plot is very telling about the speeding up process of the DC motor. It does not however, give us long-term position statistics. If the simulation is run longer, we can see that the position is indeed a value between 0 and 2π, and that the position is always increasing linearly for a constant velocity. This plot is shown below.

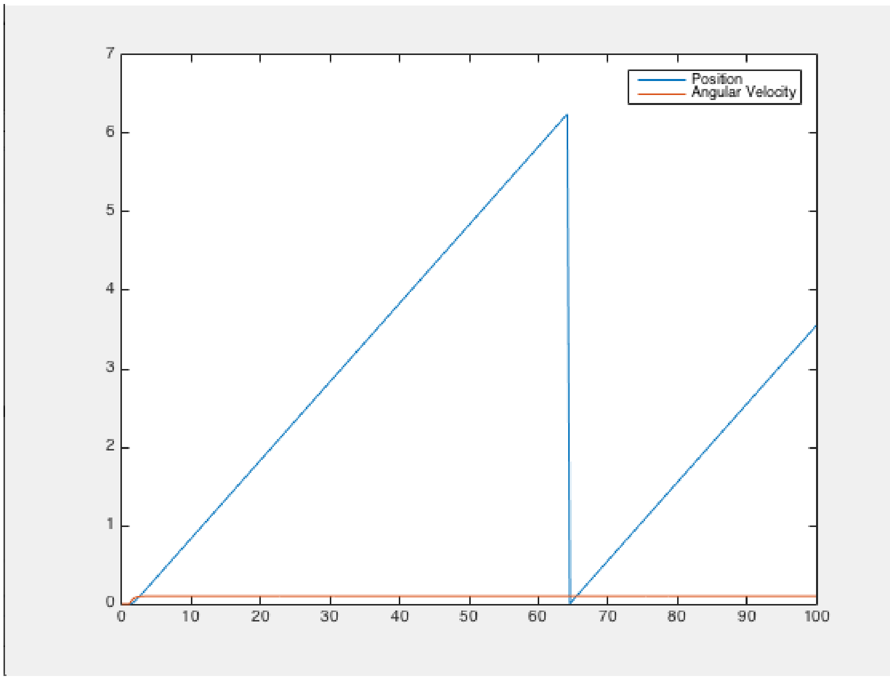

Figure 11: Position and Velocity Plot (Run Time = 100 sec)

From the plot above, it is evident that our position ranges from 0 to 2π. This is in fact what we want, because the DC motor moves in a circle. It is also encouraging to see that our position curve increases linearly for a constant velocity.

IV. Conclusion/Discussion

As we come to an end for the lab we showed we can accurately model a DC motor inside of MatLab. From the plots throughout the lab we showed the functionality available to us via MatLab's Simulink. It is a useful process when you need to experiment with the capabilities of an individual motor. Simulink allows us to individually change the various parameters of our motor.

We started with changing the moment of inertia. Our experiment proved that an increase in moment of inertia correlates to a longer time interval till our motor reaches full speed. A thoughtful analysis confirms our results since inertia corresponds to the objects tendency to keep doing what it is doing. Therefore if you moment of inertia is large it will take longer for you to overcome the motors inertia. On the opposite end we also note that the motor will want to keep rotating longer after we stop supplying power to it. You can correlate the process to a train. A train takes a while to get moving but once it's moving it takes a long time for the train to come to a stop.

Next we changed our damping ratio of the DC motor. We can think of the damping ratio as the inefficiency of the motor. The less damping ratio the faster our motor is allowed to move. Also, the acceleration increases as we decrease the damping ratio.

No matter what parameter we need to manipulate to see the effects on the output we can do so in Simulink. It is an easy approach to finding a motor for you needs rather than buying a box of motors and testing each individually.