Project 1 Technical Report
MAE493 Robotics

9/26/2020

Authors

- Joshua Bukaty
- Birahim Ndiaye
- Benedict Cutri
- Rizve Chowdhury

Summary

This report contains a dynamic analysis of a spring-mass-damper system driven by rack-pinion configuration powered by a variable torque motor. The model was simulated in Simulink, in the presence of friction force disturbances. Design parameters were adjusted to achieve a pre-specified dynamic response. Possible implementation configurations are discussed towards the end.

Task Completed by each member

- Joshua Bukaty, **Motor Selection Presentation, and Technical Report**
- Birahim Ndiaye, **Motor Selection Presentation, and Technical Report**
- Benedict Cutri, **Presentation, Team Oversight and Technical Report**
- Rizve Chowdhury, **Simulink and Technical Report**

Contents

1 Introduction

1.1 Goals

For this project, the primary goals were:

1. Analyze the effects of friction on robotic actuators.
2. Optimize design parameters to achieve desired response.

1.2 Assumptions

In order to analyze and optimize the system, we required a mathematical model of a prismatic robotic actuator. As shown in Figure 1, the robotic actuator was modeled as a spring-mass-damper system driven by a rack-pinion-motor system. The following assumptions were made:

- Spring behavior can be modeled by Hooke's Law.
- Dashpot can be modeled as a linear damper.
- Robotic link can be modeled as rigid block of mass, M.
- Rack and pinion do not have any mass.
- Armature inductance assumed zero and so motor has no back e.m.f.
- Coloumb's model of friction forces were used to account for friction forces.
- Positive voltage input drives the pinion in a counter-clockwise direction.
- Overall gear ratio for the gear system is assumed to be 1.
- Operational amplifier assumed ideal.
- Potentiometers in the non-inverting amplifier setup assumed to be ideal resistors.
- Temperature does not vary in the surroundings.
- Air resistance effects can be neglected and/or accounted for by as a linear resistance (linear drag).
- Bearing friction, b, is assumed to be 0.05.

1.3 System Specifications

Based on system specification, the following are known about the system:

- M = 5 kg (Mass of robotic link)
- k = 3 $\frac{N}{m}$ (Spring constant of spring)
- B = 2 $\frac{Ns}{m}$ (Damping constant of dashpot)
- $X = 0.1\theta$ [i] (Rack and pinion relationship)

[i] X is the linear displacement of the rack and θ is the angular displacement of the pinion

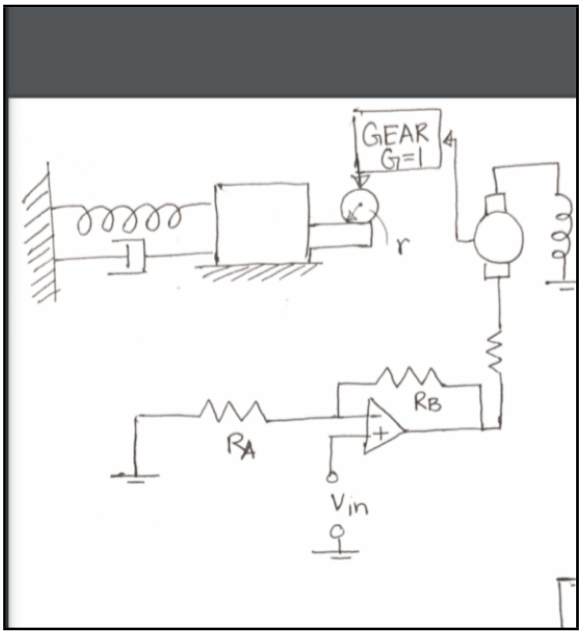

Figure 1: Sketch of mathematical model

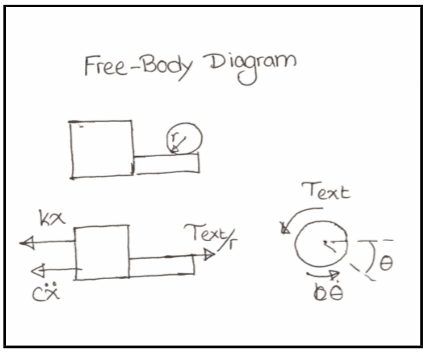

Figure 2: Free-Body Diagram of Mass, M, without friction

2 System Response

To carry out a preliminary study of how the system responds, it is instructive to analyze the transfer function for the system. By definition, the transfer function studies how the system reacts when all the initial conditions, in this case, the position and speed of the mass, are zero. A sketch of the spring-mass-damper system is shown in Figure 1. Based on the free-body diagram of the mass and the shaft as shown in Figure 2, the equations of motion applied to the mass, M, and shaft, S, without, friction, can be written as shown in Equation (1). Applying Kirchoff's Law to the motor circuit shown in Figure 3, we get equation (3). Lastly, the compatibility of the rack and pinion require Equation (4). The following equation form the mathematical model for the entire problem:

$$M * \frac{d^2x}{dt} = -k * x - B * \frac{dx}{dt} + \frac{\tau_{ext}}{r} * G \tag{1}$$

$$J_{shaft} * \frac{d\omega}{dt} = K_T * i - b * \omega - \tau_{ext} \tag{2}$$

$$v_{in} = i * R + K_B * \omega \tag{3}$$

$$\frac{dx}{dt} = r * \omega \tag{4}$$

In the equation above, the following nomenclature is used:

- M is the mass of the robotic link/block
- x is the position of the robotic link/block
- ω is the angular velocity of the pinion connected to the gear reducer.

- τ_{ext} is the torque exerted by the motor on the gear reducer [ii].
- k is the spring constant of the spring
- c is the damping constant of the dashpot
- G is the overall speed ratio of the gear and is assumed to be 1.
- r is the effective radius of the pinion and for this problem is equal to 0.1 [iii]
- v_{IN} is the input voltage delivered by op-amp circuit to the motor.
- R is the internal resistance of the motor.
- b is the bearing friction in the shaft.
- K_T is the torque constant of the motor
- K_B is back-emf constant of the motor

Looking at the equations, we see that there are four unknowns:

- ω, the angular velocity of the pinion
- X, the displacement of the mass
- i, the current in the motor
- v_{IN}, the voltage input to the motor

Since we have four equations to go with these four unknowns, the system of equations (1) to (4) are closed. To solve this system, we take the Laplace transform of the equations, turn them into a set of algebraic equations and solve for the ratio $X(s)/V_{IN}(s)$ which represents the transfer function relating displacement of the block, X, to the input voltage, V[iv]. The resulting transfer function is as follows:

$$H(s) = \frac{X(s)}{V_{IN}(s)} = \frac{C_3}{C_0 * s^2 + C_1 * s + C_2} \tag{5}$$

where C_0, C_1 , C_2, C_3 are the following constants, defined in terms of known system parameters:

$$C_0 = M + \frac{J_{shaft}}{r^2} \tag{6}$$

$$C_1 = B + \frac{1}{r} * \left(\frac{b}{r} + \frac{K_T * K_B}{R * r}\right) \tag{7}$$

$$C_2 = k \tag{8}$$

$$C_3 = \frac{K_T}{r * R} \tag{9}$$

In order to proceed with the design, we had to assume an initial specification of motor constants. For this reason, as an initial choice we choose Moog's *DBE-2000-1ES Matrix*^{TM} *Series* motors for

[ii] In turn, the gear reducer uses this input torque and delivers a greater torque to the pinion, at the cost of slower speed

[iii] This comes from the rack-pinion relationship mentioned earlier. Assuming no slip between rack and pinion, the linear displacement of the rack is equal to the product of the radius of the pinion and its angular displacement

[iv] The transfer function is derived in the Appendix I

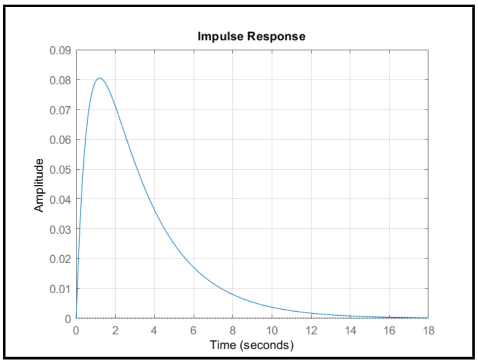

Figure 3: Impulse Response for H(s)

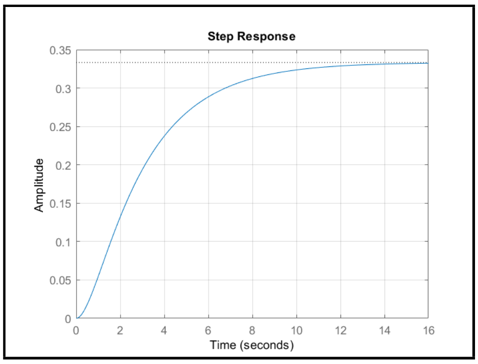

Figure 4: Step response for H(s)

the first iteration of our design. If the resulting design needs alteration, we'll change the choice. But for an initial understanding of the response, we choose a motor based on the following qualitative criteria:

- Easy to procure for early prototyping
- Large peak torque capacity
- Relatively low back-emf

The *DBE-2000-1ES* $Matrix^{TM}$ *Series* motor has the following specifications:

- $K_T = 0.97166 \pm 10\% \frac{N*m}{amp}$
- $K_B = 0.0607 \pm 10\% \frac{V}{rad/s}$
- R = 2.10 $\pm 10\%$
- $J_{motor} = 3.11 * 10^{-6}$

Using the nominal values for these motor constants, we can assign numerical values to the constants in equation 5 and get:

$$C_0 = 5 \tag{10}$$
$$C_1 = 9.8 \tag{11}$$
$$C_2 = 3 \tag{12}$$
$$C_3 = 4.63 \tag{13}$$

Thus the transfer function is:

$$H(s) = \frac{X(s)}{V_{IN}(s)} = \frac{4.63}{5.06s^2 + 9.8s + 3} \tag{14}$$

The impulse and step response for T(s) were plotted in MATLAB and shown in Figures 3 and 4, respectively. From this, we see that the system is already well-damped and displays good tracking behavior without overshoot. However, in order to exert controllability of position on the system, a PID control scheme is put in place.

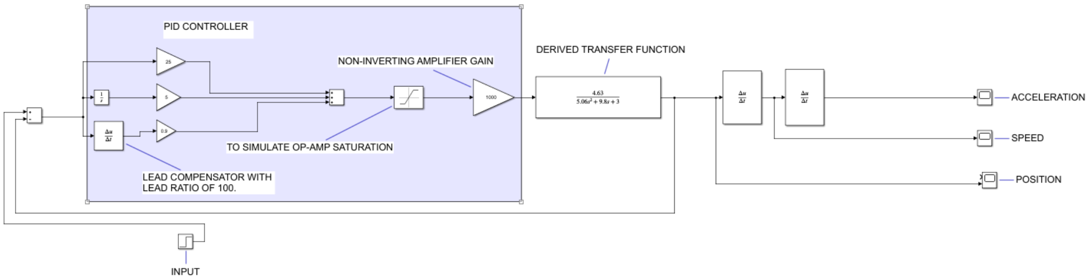

Figure 5: Mathematical Model recreated in Simulink

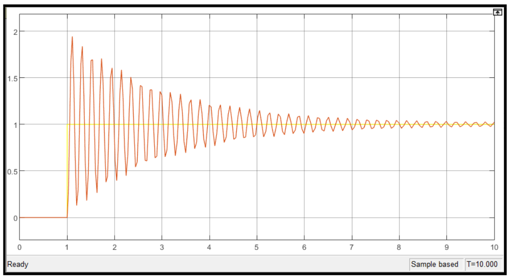

Figure 6: System Response for $K_p = 1$, $K_d = 0$ and $K_I = 1$

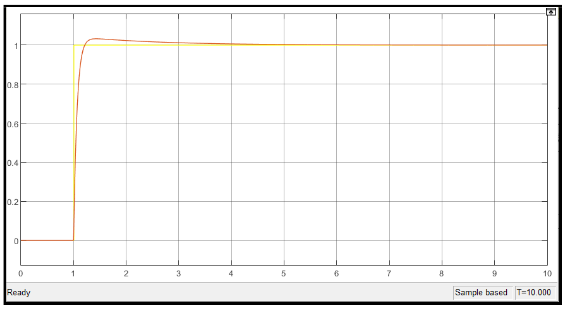

Figure 7: System Response for $K_p = 8$, $K_d = 0.5$ and $K_I = 5$

3 PID Control of System without Friction

For a preliminary study, we simulate the dynamics of the system without friction. To do so, we utilize the derived transfer function and augment a PID controller that calculates error in position and outputs a voltage. This is as shown in the Simulink model above. Also shown on the top are the responses of the system under varying gains. As shown in the figure to the left, with a proportional gain of 1, an integral gain of 1 and derivative gain of 0, the system displays overshoot in its step response . By altering the gains, we can eliminate the overshoot as shown in the figure to the right.

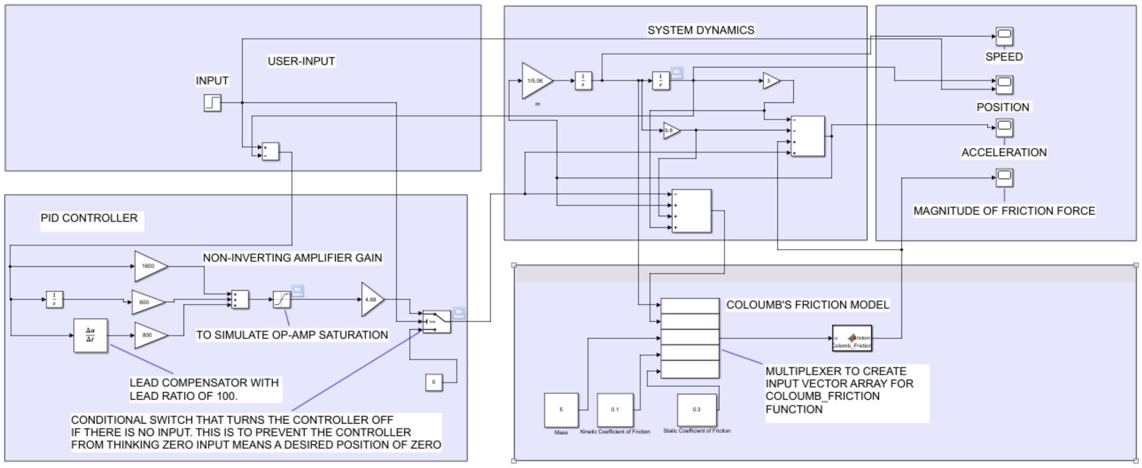

Figure 8: Mathematical Model recreated in Simulink

4 Accounting for Friction

In this section, we explore how friction affects the system. For its mathematical simplicity [v], we resort to using the Coloumb's model of friction. The model is described as follows:

$$F(\frac{dx}{dt}, F_{applied}) = \begin{cases} F_{applied} & if\ \frac{dx}{dt} = 0\ and\ F_{applied} < F_{max} \\ -\mu_k * N * sign(\frac{dx}{dt}) & if\ \frac{dx}{dt} \neq 0\ and\ F_{applied} < F_{max} \end{cases} \tag{15}$$

where μ_k and μ_s are the kinetic and static friction, respectively, $F_{applied}$ is the force applied against friction and F_{max} is the maximum static friction given by $F_{max} = \mu_s * N$.

Since this makes the system non-linear, Simulink [vi] was utilized to simulate system dynamics under the influence of friction. The state dynamics of the system, after plugging in all the constants, is given by:

$$5.06\frac{d^2x}{dt} = -9.8\frac{dx}{dt} - 3x + 4.63 * v_{IN} + F_f \tag{16}$$

where F_f is the force of friction. The block diagram of all the system components as they appear in the Simulink simulation file are shown in Figure 5.

To simulate Coloumbic friction, a MATLAB function block was used. A multiplexer was used to create a vector signal, consisting of speed, net force, mass and the coefficients of friction. Then, equation (7) was written in MATLAB with the following code:

```
function Friction = Colomb_Friction(u)
%The input u is a 6-tuple
%u(1) is the velocity
%u(2) is the net force
```

[v] More physically accurate models of friction like the Dahl and LuGre model could have been used but would have added unnecessary overhead complexity

[vi] Simulink is a registered trademark of MATLAB. If Simulink is out-of-reach, the state dynamics can be propagated with a RKF-4 solver, with the end-result being the same. Simulink utilizes a RKF-4 as the base algorithm for its ODE45 suite which, in turn, is used by Simulink to solve state dynamics.

```
%u(3) is the mass
%u(4) is the kinetic coefficient of friction
%u(5) is the static coefficient of friction
speed = u(1);
TOL = 1E-5;
if (length(u) > 1) %Check if the input is a vector
    Static_Friction = u(5)*u(3)*9.8;
    Kinetic_Friction = u(4)*u(3)*9.8;
    Net_Force = u(2);
if (abs(speed) < TOL))
     if ((abs(Net_Force) < Static_Friction)
         Friction = -Net_Force;
else
Friction = -Kinetic_Friction*sign(u(1));
end
else
        Friction = -Kinetic_Friction*sign(u(1));
     end
else
    Friction = 0;
end
end
```

In order to explore how the system reacted to friction, the system dynamics were simulated with the following initial conditions:

- $x(0) = 0.3\ m$
- $\frac{dx(0)}{dt} = 0\ \frac{m}{s}$

4.1 Frictional Effects on Position

To understand the effects of friction, let's look at three variables, position, velocity and the magnitude of friction force. The position of the block of mass M in this simulation is as shown in Figure 9. As can be seen, friction prevents the block from moving much at all and it remains at its position of 0.3 m. A quick calculation validates this as follows. From the initial conditions, we know that the only force besides friction that acts on the block is the spring restoring force, $-kx$. If $k = 3$ and $x = 0.3$, we know that this force has a magnitude of 0.9. However since the maximum static friction force is 15 N, this restoring force is easily counteracted by friction and the block does not move at all.

4.2 Frictional Effect on Velocity

As the position graph should have implied, the speed of the graph remains at zero and does not budge much from that value as shown in Figure 10. If this were not the case, we would suspect the simulation as speed has to be the derivation of the position.

4.3 Magnitude of Friction Force

From the graph in Figure 8, we see that the friction force discontinuously jumps between 0 and 5. This can be explained by the fact although the speed is small, it does differ from zero. This is

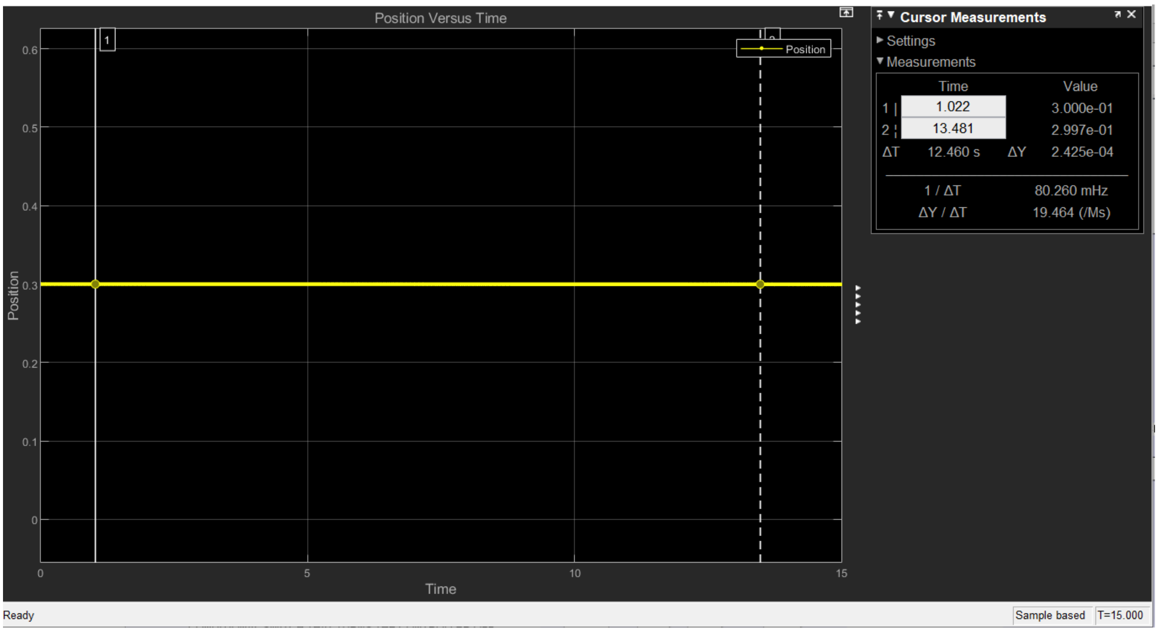

Figure 9: Simulink output of position for the above simulation

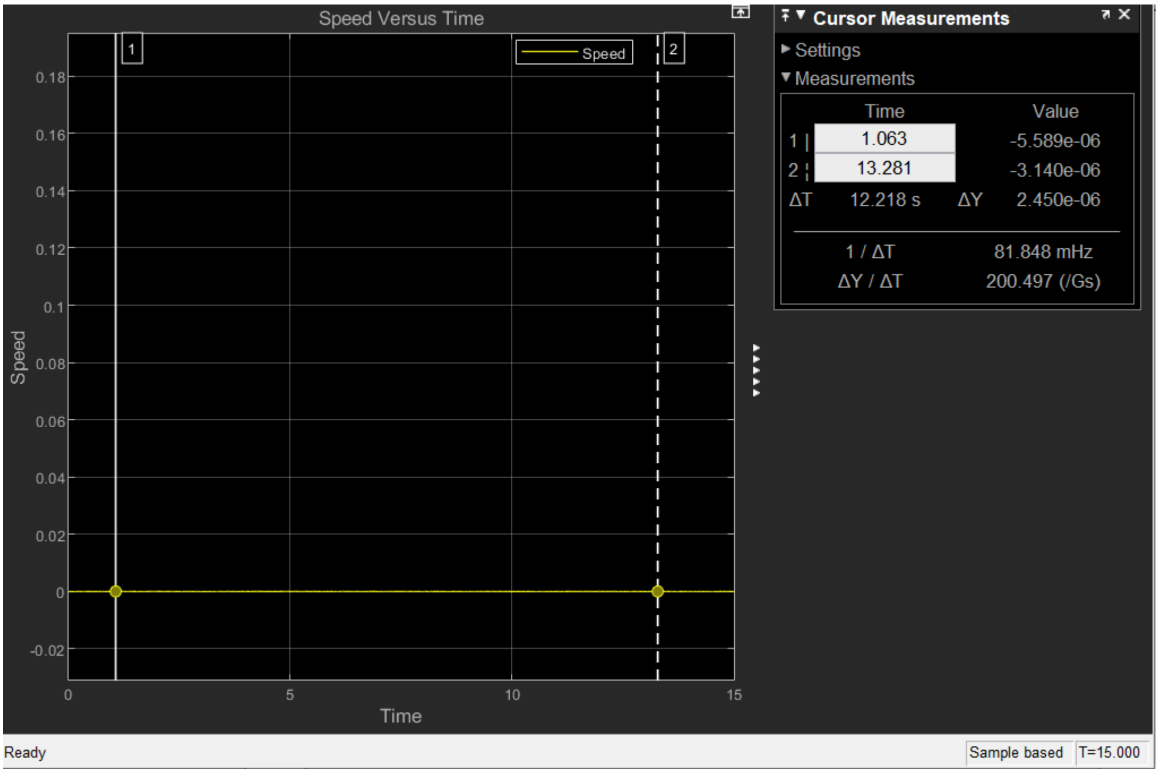

Figure 10: Simulink output of velocity for the above simulation.

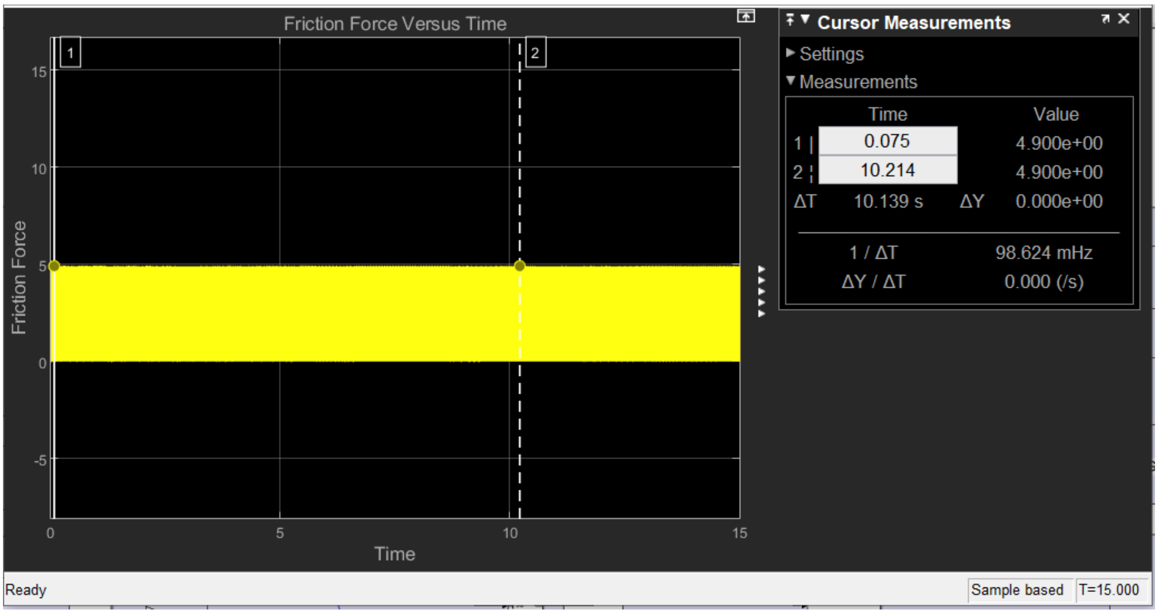

Figure 11: Simulink output of friction force magnitude for the above simulation.

due to Simulink using a numerical solver to compute the dynamics for the system. Since we have a non-zero speed with a magnitude close to zero, the Coloumb model isn't well-defined. Going back to the code, we dealt with this scenario by having a tolerance value under which we can assume that the system has effectively "stopped". Since the small speed fluctuations were greater than this value, the model assumed that the block was under going dynamic friction and predicted a dynamic friction of 5 N as expected.

5 PID Control of System Dynamics with Friction

In order to attain the desired response charateristics, the team opted to utlilize a PID control scheme, partly due to the simplicity of the control scheme and partly due to the non-linearities in the system being bounded.

5.1 Design Criteria and Simulation Settings

To simulate the system, a Simulink model as shown in Figure 9 was used. With this model in place, the gains in the PID controller were adjusted until the following criteria were satisfied:

- Zero over-shoot in step response of 0.3 m
- Zero steady-state error in step response of 0.3 m
- Maximum voltage input limited to ±5 volts (Operational amplifiers cannot produce voltage swings greater than voltage used to power them.)

The simulation was to satisfy the above criteria for the following set of ICs:

1. $x(0) = 0.5\ m$ and $\frac{dx(0)}{dt} = 0$ (Initial Condition 1)
2. $x(0) = \frac{dx(0)}{dt} = 0$ (Initial Condition 2)

The step response were applied after 1 second and a lead compensator in the from $\frac{s}{0.01s+1}$ was used to generate the derivative of the error signal.

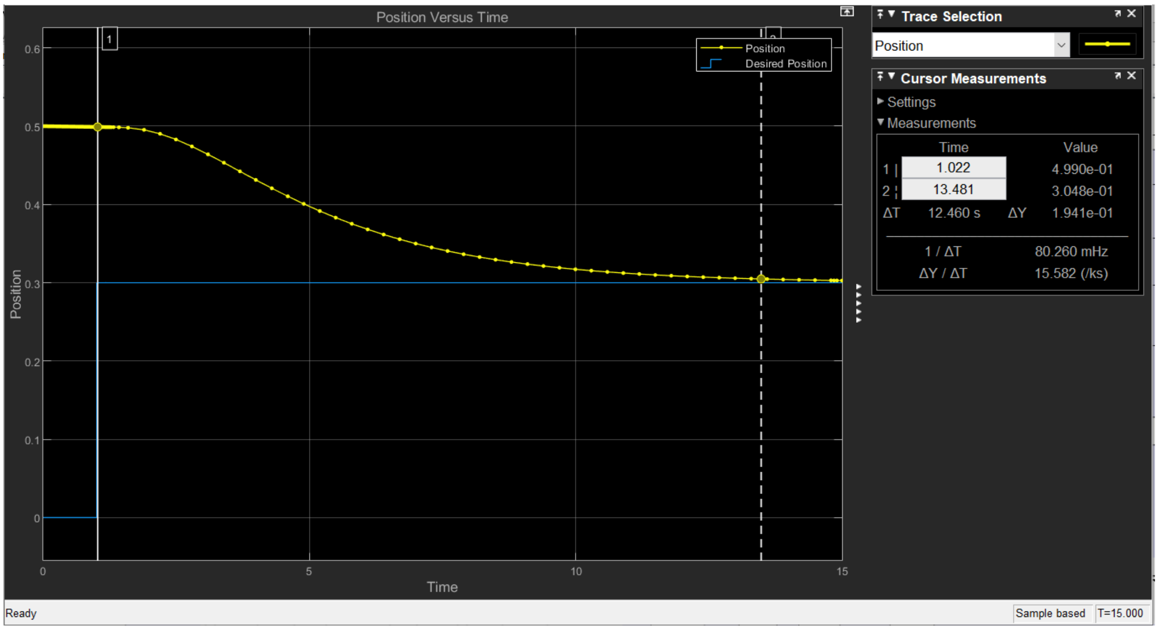

Figure 12: Simulink output of position for the above simulation.

5.2 System Response Under Initial Condition 1

Starting from an initial displacement of 0.5 m and having no initial speed, the system displays the position versus time characteristic shown in Figure 12. As expected, the system starts off from an initial position of 0.5 and when the desired position is entered by the user at time $t = 1s$, the position changes to the desired position of 0.3 m without overshoot. The rise time [vii] for the response, as can be seen from the graph is roughly equal to $t_{rise} = 12.460s$. Since no requirement was set on speed of response and there was no *overshoot*, we consider this a fulfillment of the earlier design criteria.

On the next page, there are also graphs of velocity (Figure 13) and applied voltage (Figure 14). As required by our design criteria, the applied voltage was limited to ±5 volts using a saturation function. This was done to simulate op-amp saturation when powered by a ±5 differential voltage supply. The maximum speed of the block is $0.37\frac{m}{s^2}$ and as implied by the position versus time graph, asymptotically reaches zero. The PID gains used were:

[vii] For the purposes of this simulation, the rise time is the time required to reach 99% of the set-point

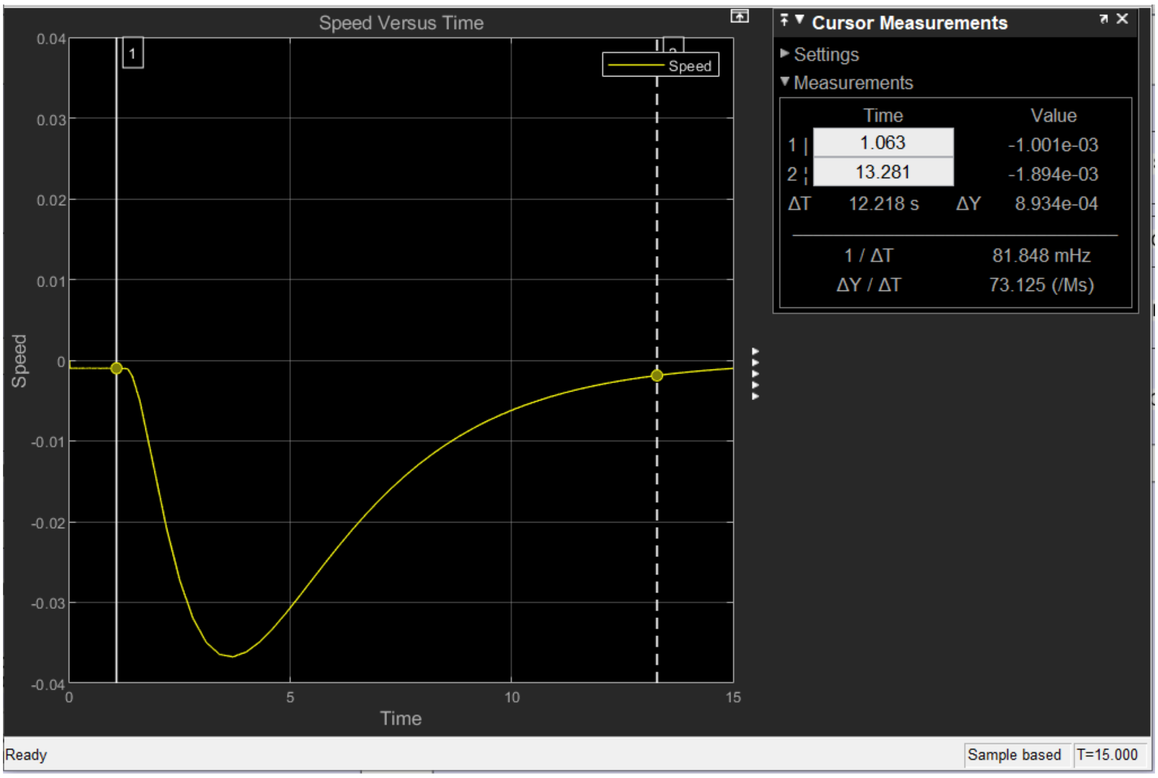

Figure 13: Simulink output of speed for the above simulation.

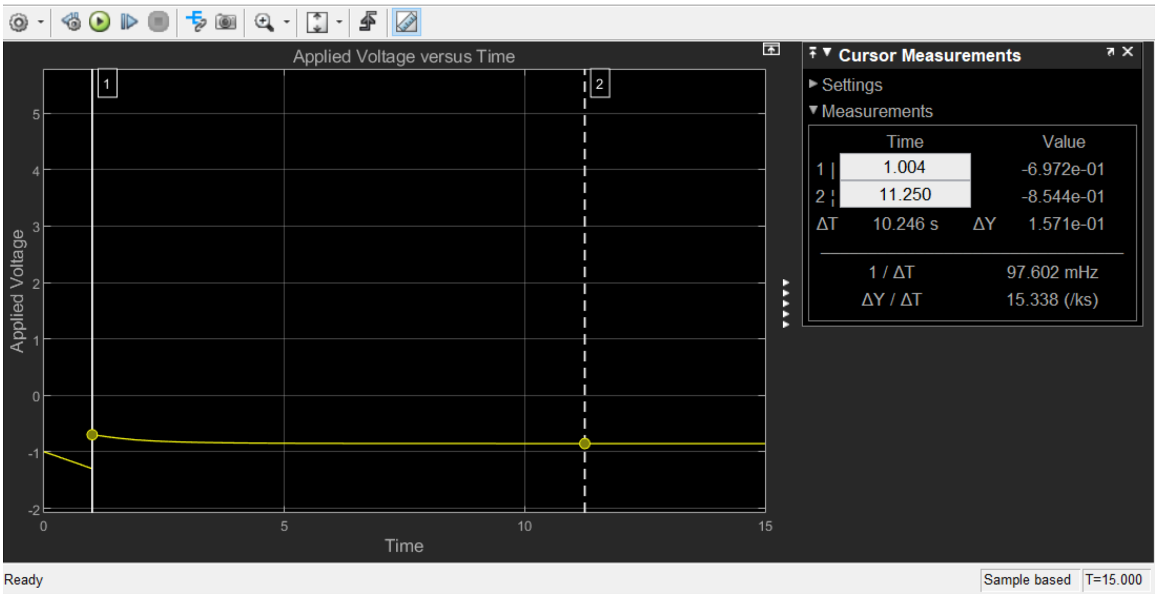

Figure 14: Simulink output of applied voltage for the above simulation.

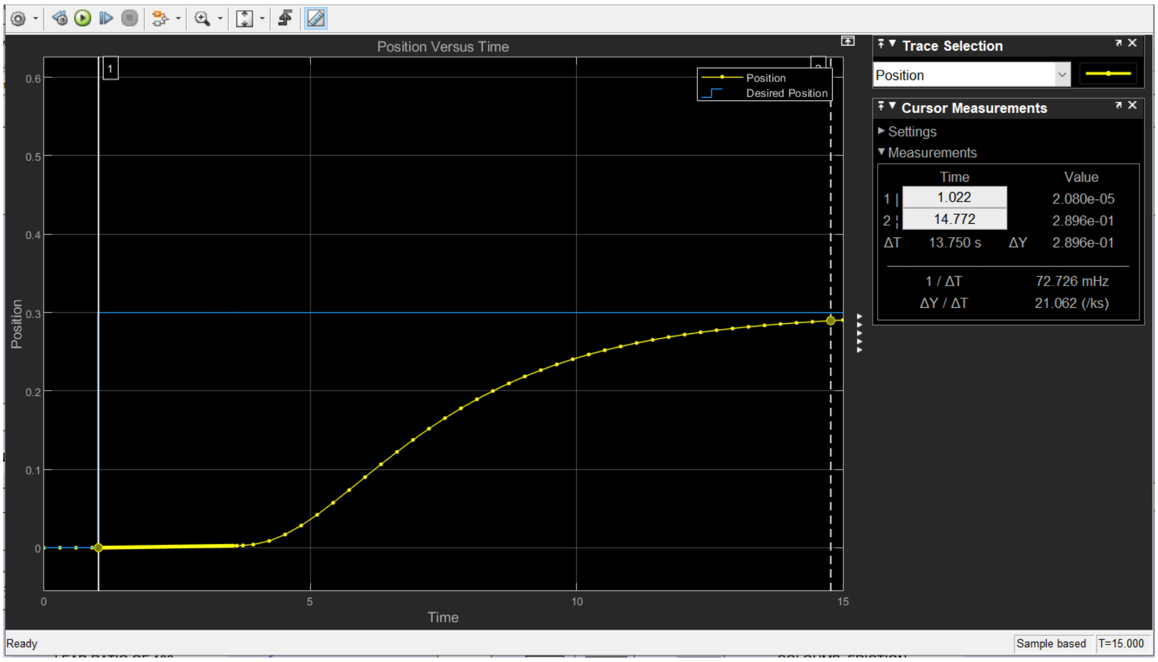

Figure 15: Simulink output of position for the above simulation.

5.3 System Response Under Initial Condition 2

Starting from an initial displacement of 0 m and having no initial speed, the system displays the position versus time characteristic shown in Figure 15. As expected, the system starts off from an initial position of 0 and when the desired position is entered by the user at time $t = 1s$, the position changes to the desired position of 0.3 m without overshoot. The rise time [viii] for the response, as can be seen from the graph is roughly equal to $t_{rise} = 13.75s$, slightly larger than the one with an initial position of 0.5 m. Since no requirement was set on speed of response and there was no *overshoot*, we consider this a fulfillment of the earlier design criteria.

On the next page, there are also graphs of velocity (Figure 13) and applied voltage (Figure 14). As required by our design criteria, the applied voltage was limited to ± 5 volts using a saturation function. This was done to simulate op-amp saturation when powered by a ± 5 differential voltage supply. The maximum speed of the block is $0.546 \frac{m}{s^2}$ and as implied by the position versus time graph, asymptotically reaches zero.

[viii] For the purposes of this simulation, the rise time is the time required to reach 99% of the set-point

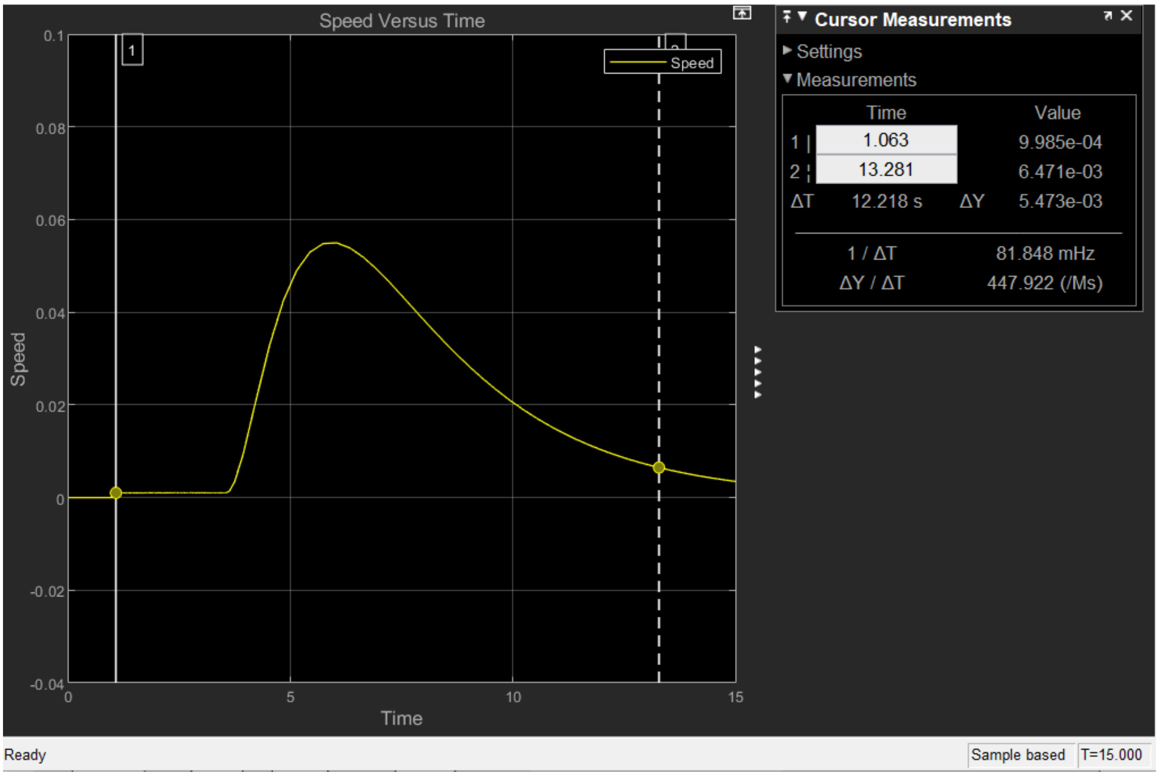

Figure 16: Simulink output of speed for the above simulation.

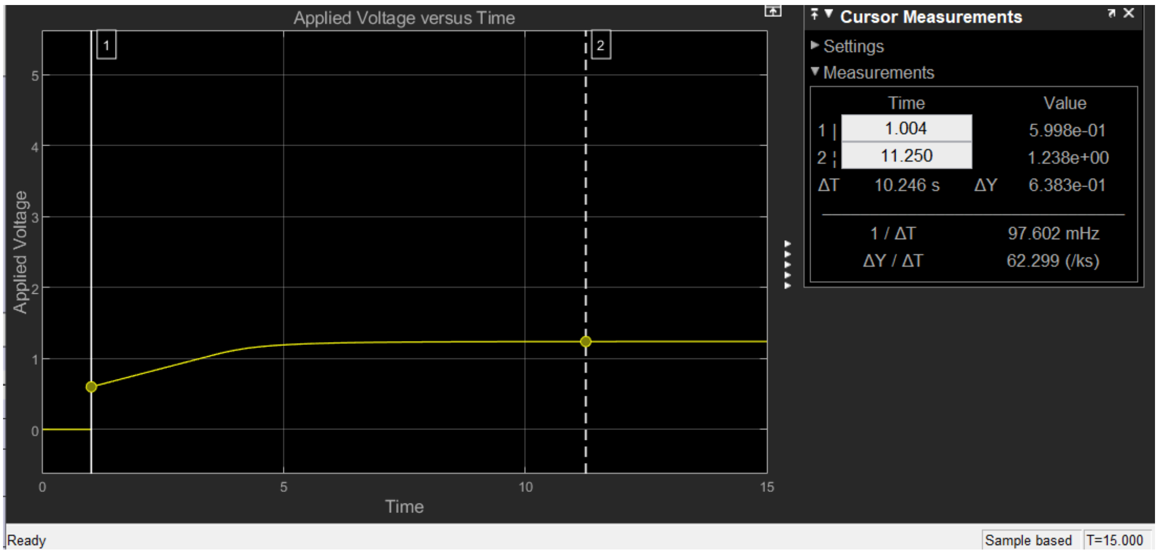

Figure 17: Simulink output of applied voltage for the above simulation.

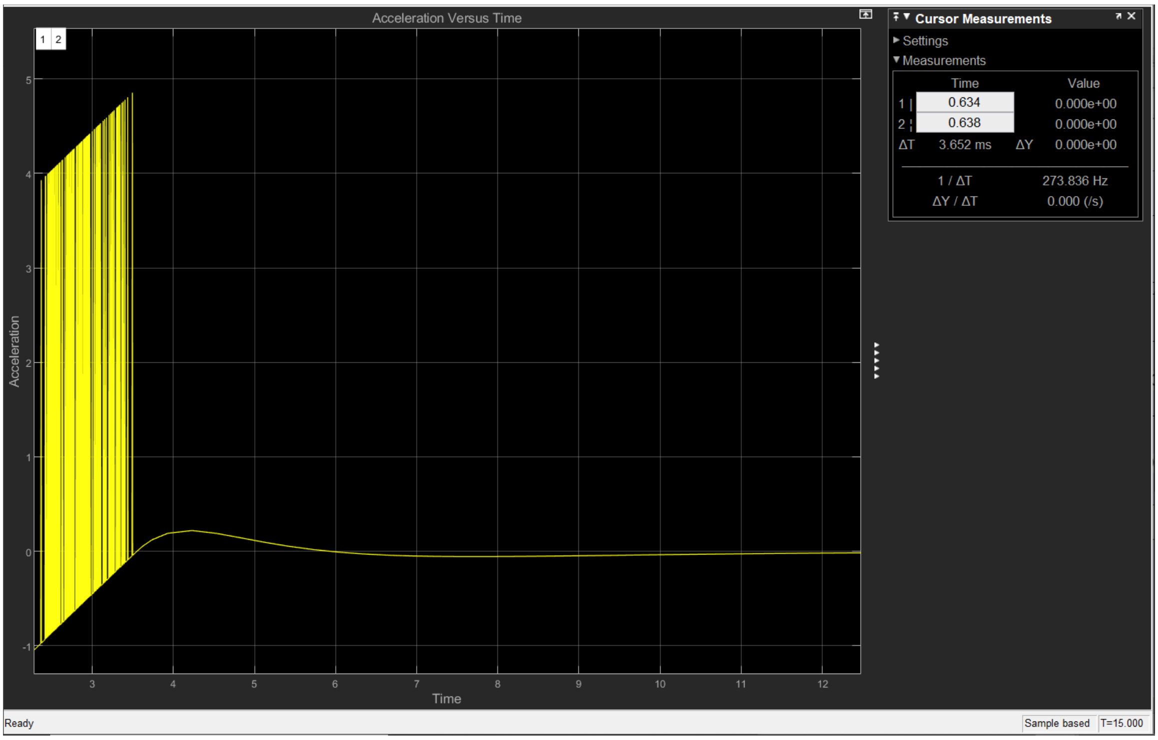

Figure 18: Simulink output of acceleration for the system simulated with initial condition 2.

6 Implementation Specifications

6.1 Motor Selection

At the beginning of this report, we assumed a choice of servo motor based on qualitative criteria. In this section, we refine the estimate to see if we can opt for a better motor. In order to search for a motor, we will focus on three criteria. Each of these criteria will be based on the simulation done in the previous section.
The criteria are:

- **Torque Capacity**:The maximum amount of torque created by the selected motor must *at least* be capable of provide the highest torque that we expect to provide.
- **Power Capacity**: The amount of power that the motor can provide must be equal to power that we expect to provide.
- **Reasonable Cost**: Althiough no budget was specified, the cost should be minimized.

6.1.1 Torque Capacity

Figure 18 shows us the acceleration versus time for the system when released from an initial position of 0 meters and commanded to go to 0.3 meters. Among the two sets of initial conditions, this set

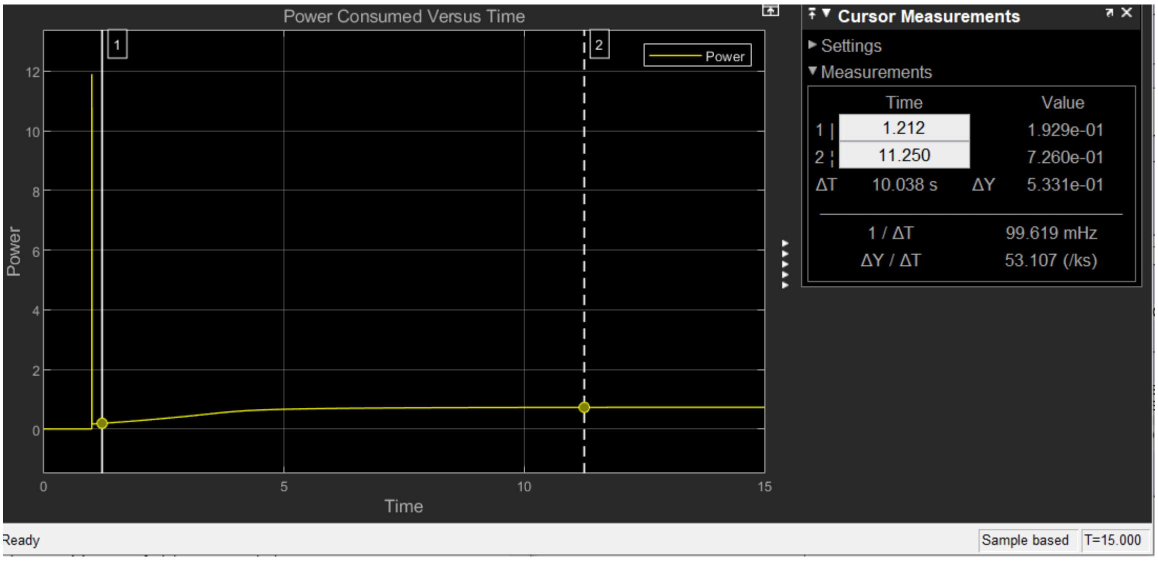

Figure 19: Simulink output of power for the system simulated with initial condition 2.

produces the higest acceleration due to the fact that there is a higher difference[ix] between initial position and set-point and it reaches a higher maximum speed.
With that in mind, we look in Figure 18 and see that the highest magnitude of acceleration is $5\frac{m}{s^2}$. The mass of the block being m, we expect the maximum force that could be exerted on the motor as $F_{max} = 5 * 5 = 25\ N$. Since the radius of the pinion is 0.1 meters, the maximum torque needed is $\tau_{max} = 25 * 0.1 = 2.5\ N * m$.

To account for uncertainties in parameters and simulation errors, we make the estimate for maximum torque more conservative by multplying by a factor of 2. Therefore, we would need a maximum torque of $\tau_{ext} = 5.0\ N * m$.

$$\tau_{max} = 5.0N * m \tag{17}$$

6.1.2 Power Capacity

By utilizing the applied voltage output and equation(3), we can find the power applied as:

$$Power = Voltage * Current \tag{18}$$

From this graph, we see that we expect to use around 0.7260 W of power asymptotically to keep the block at its desired position but that it spikes to 12 W at the instant when the PID controller commands it to move to a desired position. From past experiences with similar simulations, we hypothesize that the spike is due to a simulation error but nonetheless increase our estimate of 0.760 W to 2W to account for that increase.
Therefore the maximum power capacity of our motor should be at least:

$$P_{max} = 2W \tag{19}$$

[ix]For initial condition 1, the difference between set-point and initial position is $|0.5 - 0.3| = 0.2$. Whereas in initiail condition 2, we have $|0 - 0.3| = 0.3$.

6.1.3 Selected Motor

With the criteria above, we chose to utilize a *Motornet DC* series 3000 rpm motor from Parker Hannifin Corporation. For the motor chosen, the pertinent parameters were:

- $\tau_{rated} = 4.4Nm$
- $\tau_{peak} = 15Nm$
- $\omega_{rated} = 3000rpm$
- $P_{rated} = 1382W$
- $K_T = 1.02 \pm 10\% \frac{N*m}{amp}$
- $K_B = 0.59 \pm 10\% \frac{V}{rad/s}$
- R = 0.78 $\pm 10\%$
- $J_{motor} = 0.504 * 10^-3 \frac{kg}{m^3}$

6.1.4 Results of Redesign

If you compare the above parameters to the criteria above, it is noticeable that the parameters are much higher than the minimum. This was intentional and done to ensure that even in the worst-case scenario of external loads not accounted for in the process of systems modelling, the actuator will be able to provide enough power for control applications. With the above parameters, our new system dynamics are, using equations (1)- (4) & equations (6)-(9):

$$5.05\frac{d^2x}{dt} = -84.2\frac{dx}{dt} - 3x + 13.08 * v_{IN} + F_f \tag{20}$$

With these parameters, we can redesign the system simply by changing the gains in the Simulink block diagram as shown in Figure 20. In Figures 21 and 22, we can see that the system tracks the desired position well from both initial positions.

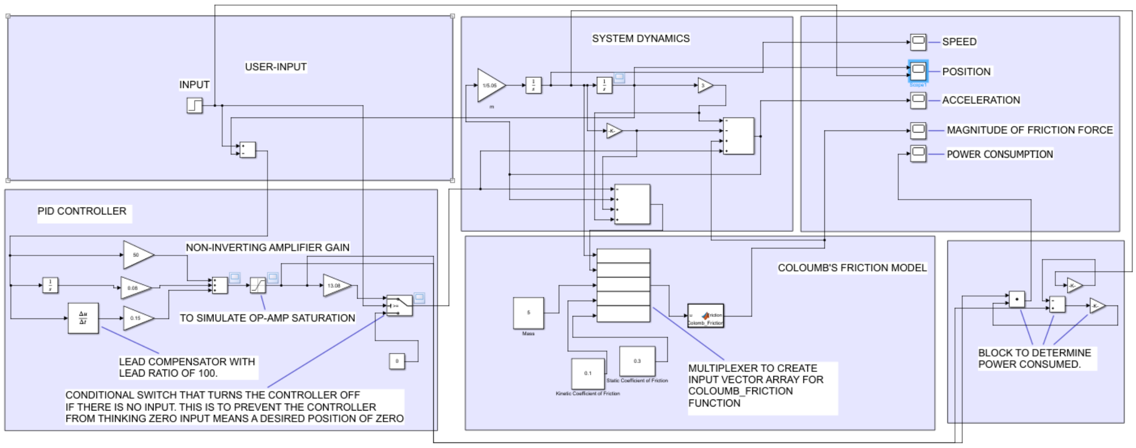

Figure 20: Redesigned Simulink block diagram

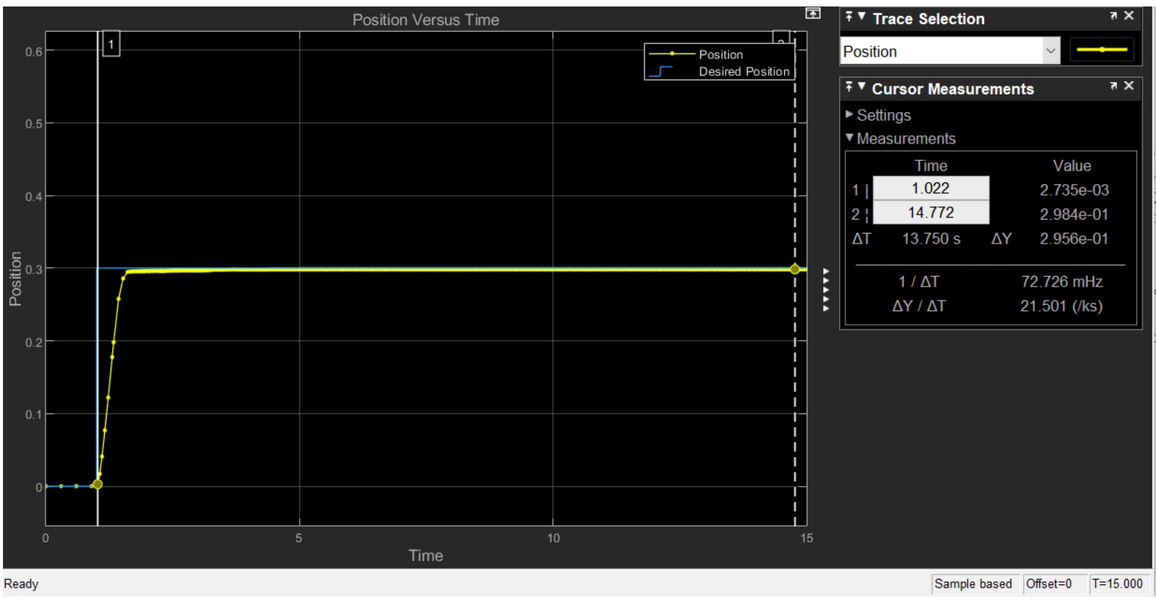

Figure 21: Position Output with Redesigned System and Initial Position of 0 m

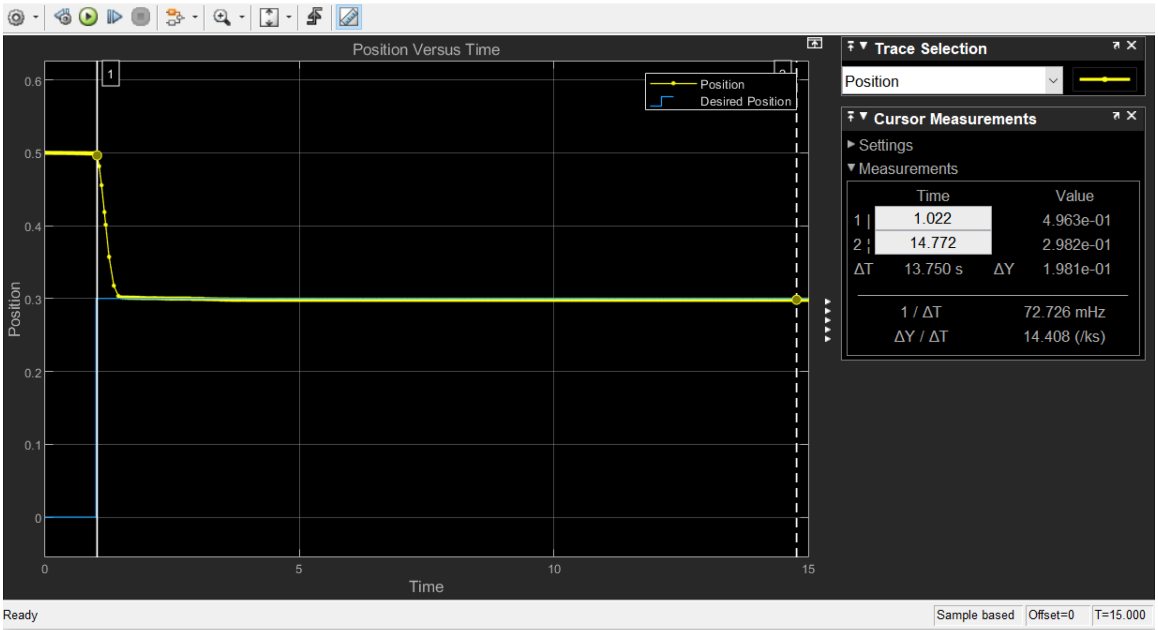

Figure 22: Position Output with Redesigned System and Initial Position of 0.5 m

6.2 Operational Amplifier PID Controller

The PID controller utilized earlier will be implemented through the use of numerous operational amplifiers. With that being written, there are numerous configuration schemes through which this can be accomplished. For the sake of simplicity, the following circuit was designed by meshing together separate op-amps circuits like integrators and summers as shown in Figure 23.

From the left, the highest branch is an integrator, the gain of which can be set by altering the pot R_{A1}. Below that, there is a differentiator, the gain of which is set by altering the pot R_{B1}. Lastly, we have a non-inverting amplifier on the lowest rung, the gain of which can be set by changing resistors R_{C1} and R_{C2}. All of the three branches merge into a summer that sums the inputs and inverts it to obtain V_{out}.

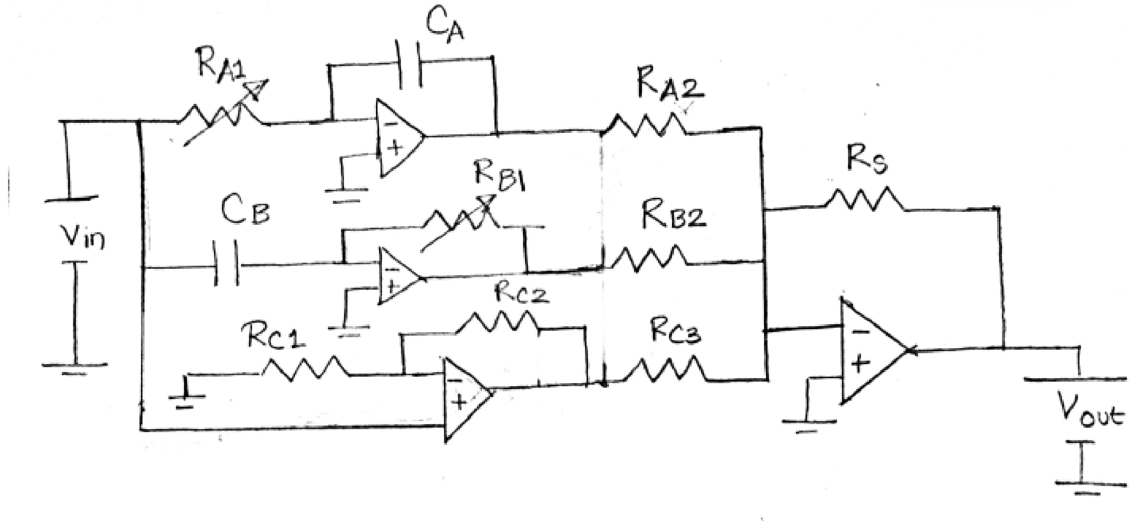

Figure 23: A PID controller implemented using operational amplifiers