**Linear Quadratic Integrator Controller Design for a Voltage Controlled Linear Actuator and Spring Mass System**

Daniel Mesenger

**Abstract –** Linear actuators are a fundamental component of robotics used to generate translational motion in a system. One form of linear actuator is the electro-mechanical linear actuator. It uses a form of electric motor to generate rotational motion which is then converted into translational motion through some type of drive. This type of actuator has a wide variety of applications in which the precise translational position of the actuator is of importance. One such application is to use the linear actuator to control the force of a spring in a spring mass system. An example of this system is a mass that is attached to both a linear actuator and by a spring to a fixed object. By using the linear actuator to displace the mass we can change the spring force, thereby changing the amount of force being applied to the fixed object by the spring. Designing a controller for the spring mass system allows us to use the system to apply a precise force to a fixed object.

System Model

The overall system consists of two models that are cascaded together. The first model is a voltage controlled linear actuator and the second is a spring mass system with a mass connected to a fixed object by a spring. The voltage controlled linear actuator is modeled with an electrical piece and a mechanical piece that interfaces via a DC motor. Figure 1 shows a graphical representation of the system.

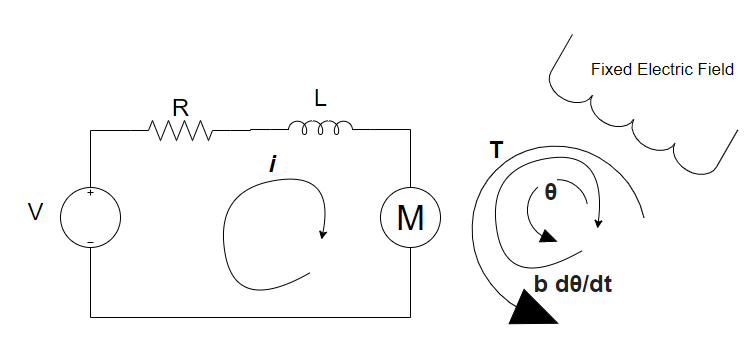


Figure 1

In this system it is assumed that the magnetic field is constant. This means that the torque of the motor is proportional by a constant factor, K, only to the current flowing through the motor as described in equations below. The back emf, e, is proportional to the angular velocity by the same factor of K.

Using Newton’s second law of motion we can derive the first of the equations used to describe the system.

Using Kirchoff’s voltage law, we know the voltage across the inductor, L, and the resistor, R, must be equal to the difference of the input Voltage, V, and back emf

To represent the system in state space, we choose our states to be the current, *i*, the angular velocity, , and the angular position, . Therefore, the system can be described in state space as follows.

The output of this system is the angular position of the DC motor. This angular position is translated to a liner position via a lead screw that is represented as an ideal axle and wheel with radius, w. Therefore, the linear position, x, is:

The second piece of the cascaded model is diagrammed in figure 2. It consists of a mass attached to a fixed object through a spring. The input of the system is the external force applied to the mass and the output of the system is the force of the spring.

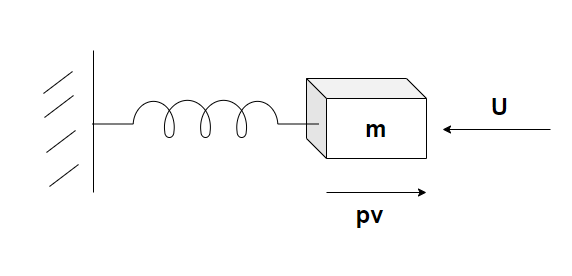


Figure 2

The state space representation for the model can be derived by observing Newton’s second law for the total force in the system. The total force is given by:

To represent the system in state space we choose the states to be the displacement, x, and the velocity v.

We choose the output to be the force in the spring given by:

Therefore the state space representation is:

Controller Design

There are two separate controllers that have been designed to control the overall system. These controllers are cascaded together and use the output from separate outputs from the overall system. The first controller uses feedback from the second system that represents the spring mass system to create an outer feedback loop. The second controller uses feedback from the first system that represents the DC motor to create an inner feedback loop. The outer feedback controller was designed first using the state space model. Design constants were chosen as:

Using the Linear Quadratic Integrator method, we first solve for the control gain with

The full state feedback controller was then simulated where the input to the system is the integral of the error (the difference of the output and the setpoint). The result of this simulation was system that responded much too slowly and oscillated too much. Q and R were then modified and found to satisfy the requirements of settling to a step set point within one second and without overshoot or oscillation.

Once the full state feedback controller was designed, a Kalman filter was added to the design to be used for state estimation. The design parameters used for the inner loop Kalman filter were:

The second, outer feedback controller was designed in a very similar manner to the inner feedback controller. The system design parameters were chosen and using the Linear Quadratic Integrator method again, the control gains were calculated.

Similar to the first controller design, control gains were first calculated with

The controller was then simulated with the input equal to the integral of the difference of the setpoint and the output. The controller was found to perform satisfactory with

J = 0.01;

b = 0.1;

K = 0.01;

R = 1;

L = 0.0015;

%wheel radius

W = 0.05;

A = [0 1 0

0 -b/J K/J

0 -K/L -R/L];

B = [0

0

1/L];

C = [1 0 0];

D = 0;

motor\_ss = ss(A,B,C,D)

Q = [0.001 0 0 0; 0 0 0 0;0 0 0 0;0 0 0 1];

R = [0.0000000000001];

K =lqi(motor\_ss, Q, R)

%Spring mass system controller

%spring constant

k = 4;

row = 0.4;

m = 5;

As = [0 1

-k/m -row/m];

Bs = [0

1/m];

Cs = [k 0];

Ds = 0;

spring\_ss = ss(As,Bs,Cs,Ds);

Qs = [0.4 0 0;0 0 0;0 0 1];

Rs = [0.000000000000001];

Ks = lqi(spring\_ss, Qs, Rs)

Simulation and Analysis

Simulink was used to simulate the performance of the two cascading controllers in the overall system. First, the plant was constructed by cascading the DC motor model and the spring mass model. The output of the motor model is multiplied by the wheel radius to represent the lead screw that is transforming the angular position of the motor to the linear position of the actuator before feeding it to the input of the spring mass model. The output of the DC motor model is also fed back to the DC motor Kalman filter with the DC motor input. The output of the Spring mass model is fed back to the spring mass controller. A setpoint is configured at the input of the overall system. This is the setpoint that the overall design will attempt to maintain at the output of the system. An external disturbance is also summed into the output of the system to represent any disturbance that the controller should attempt to reject. The setpoint, overall system output, and external disturbance are all fed to a Simulink scope in order to analyze performance.

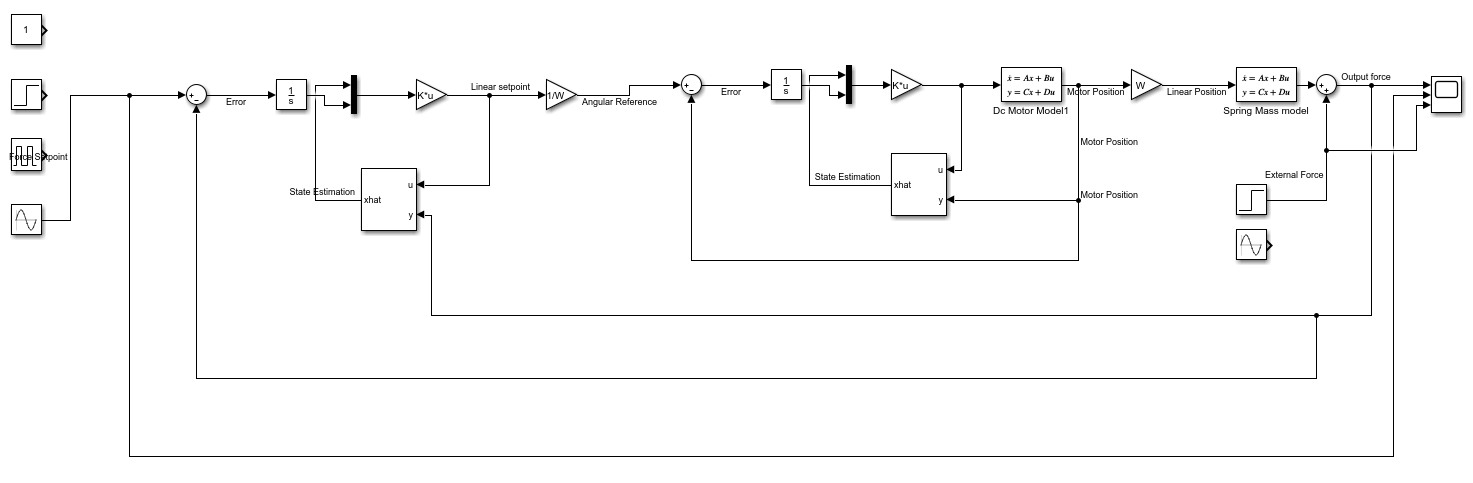


Figure 3

The system was simulated while tracking a constant setpoint, a pulse train, and a sine wave. Each of these setpoints were simulated with no external force, a constant external force beginning at T=5, and a sinusoidal external force beginning at T=5. The results can be found in the figures below.

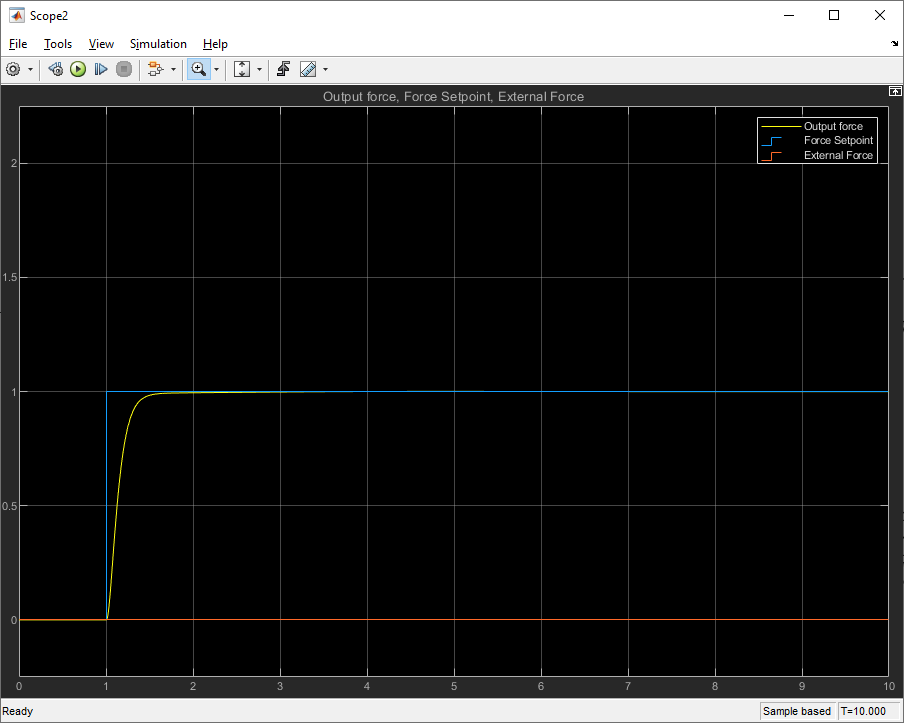


Figure 4

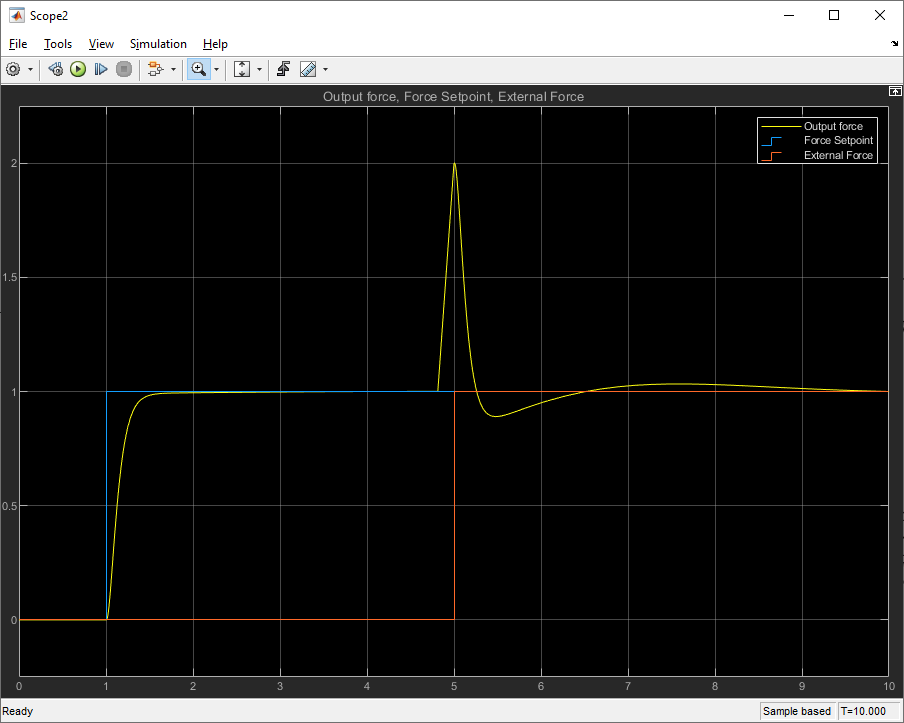


Figure 5

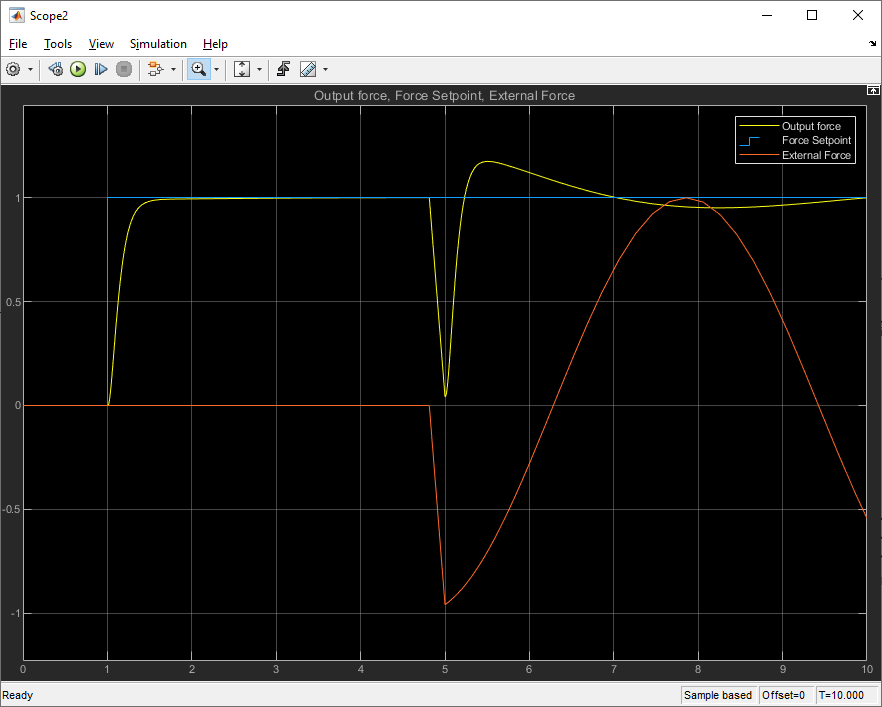


Figure 6

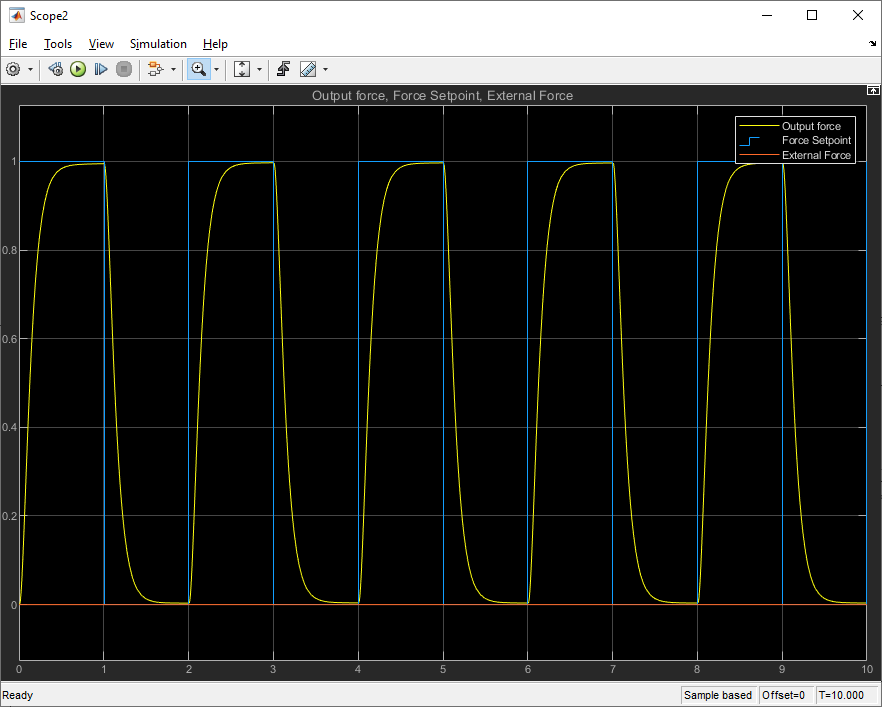


Figure 7

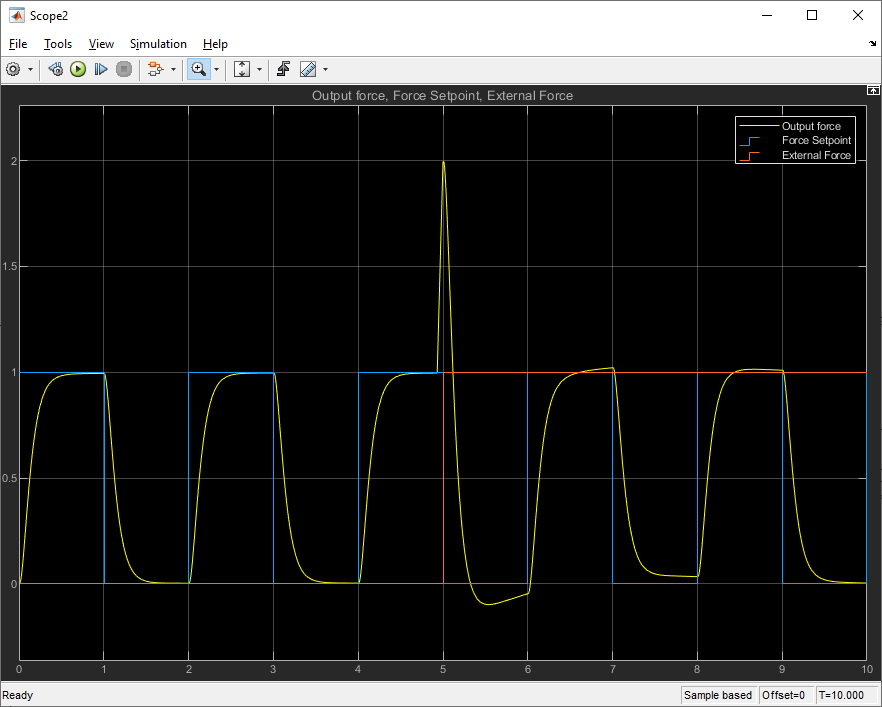


Figure 8

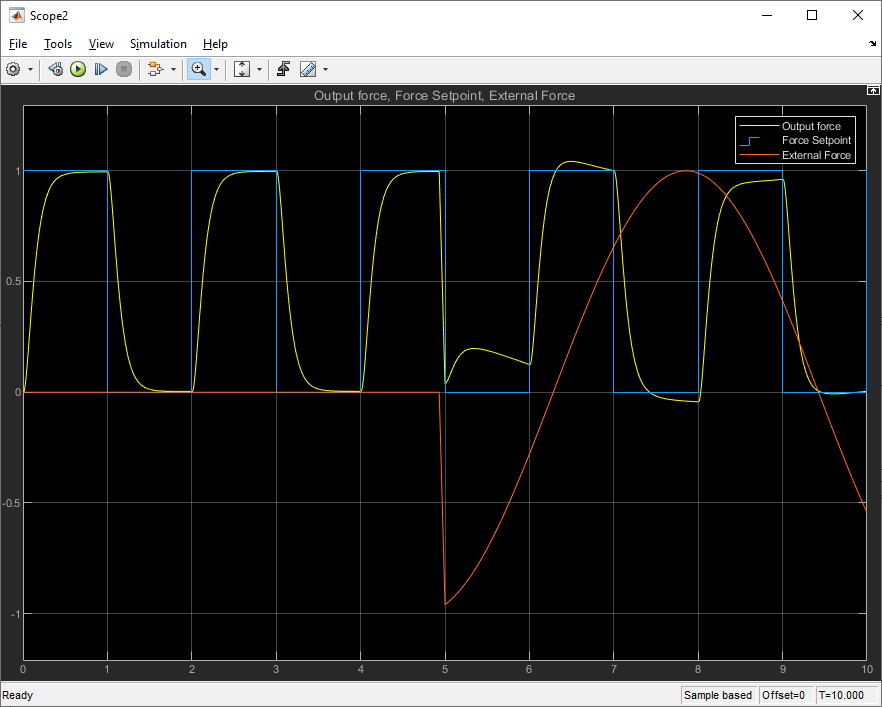


Figure 9

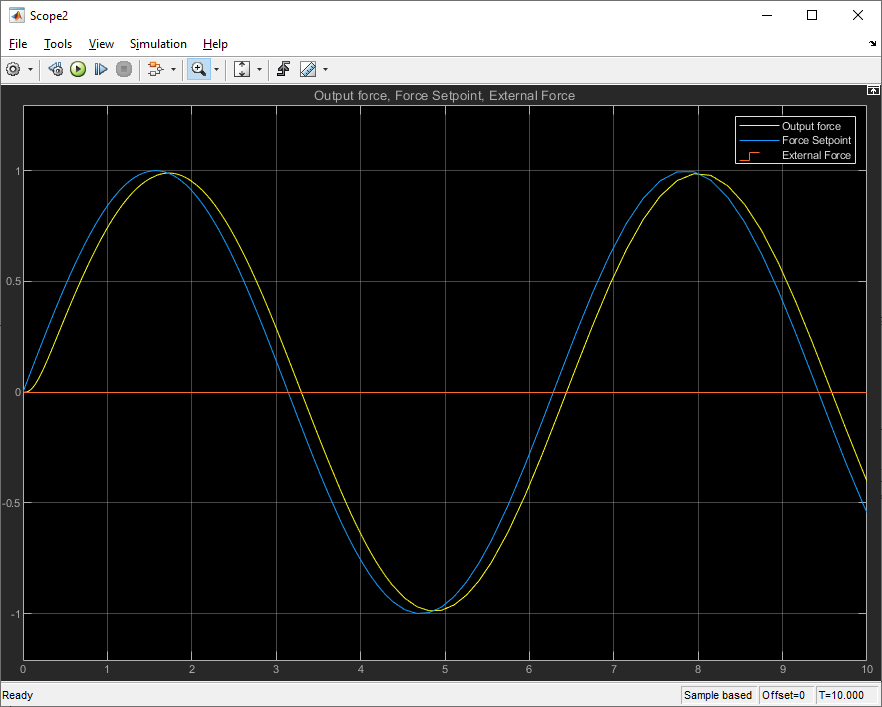


Figure 10

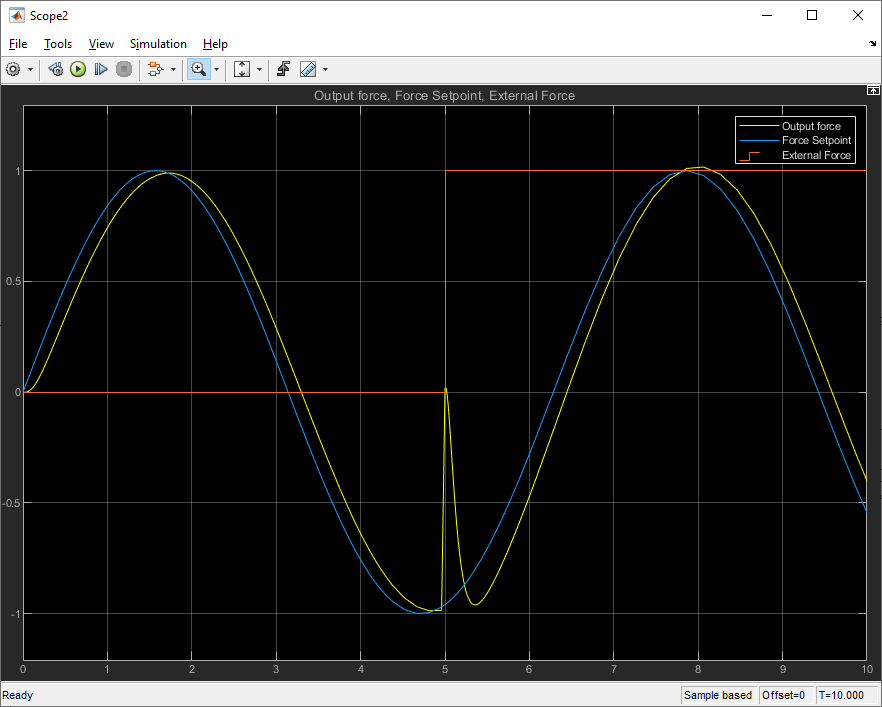


Figure 11

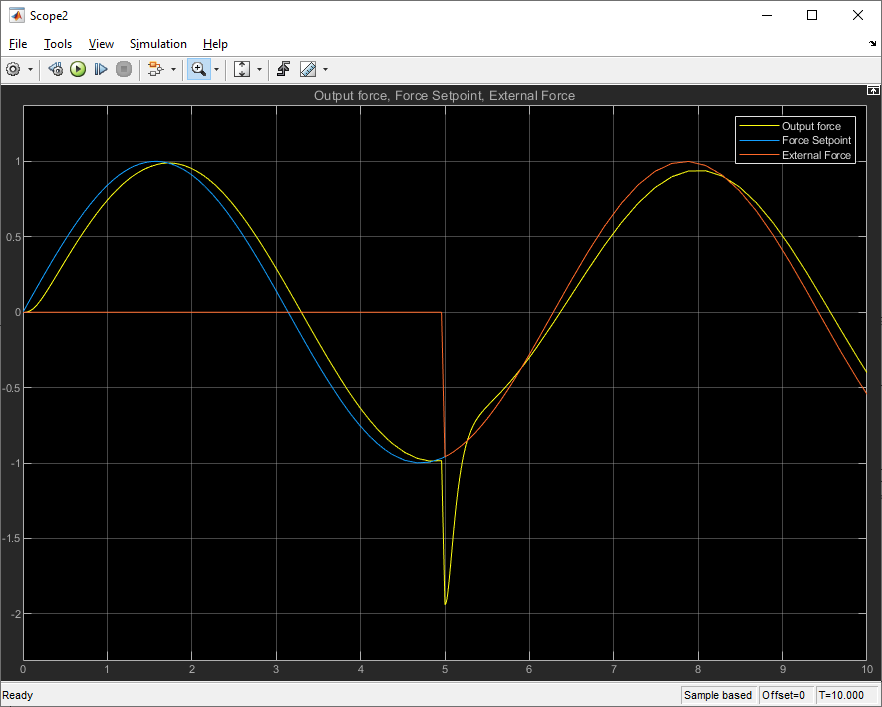


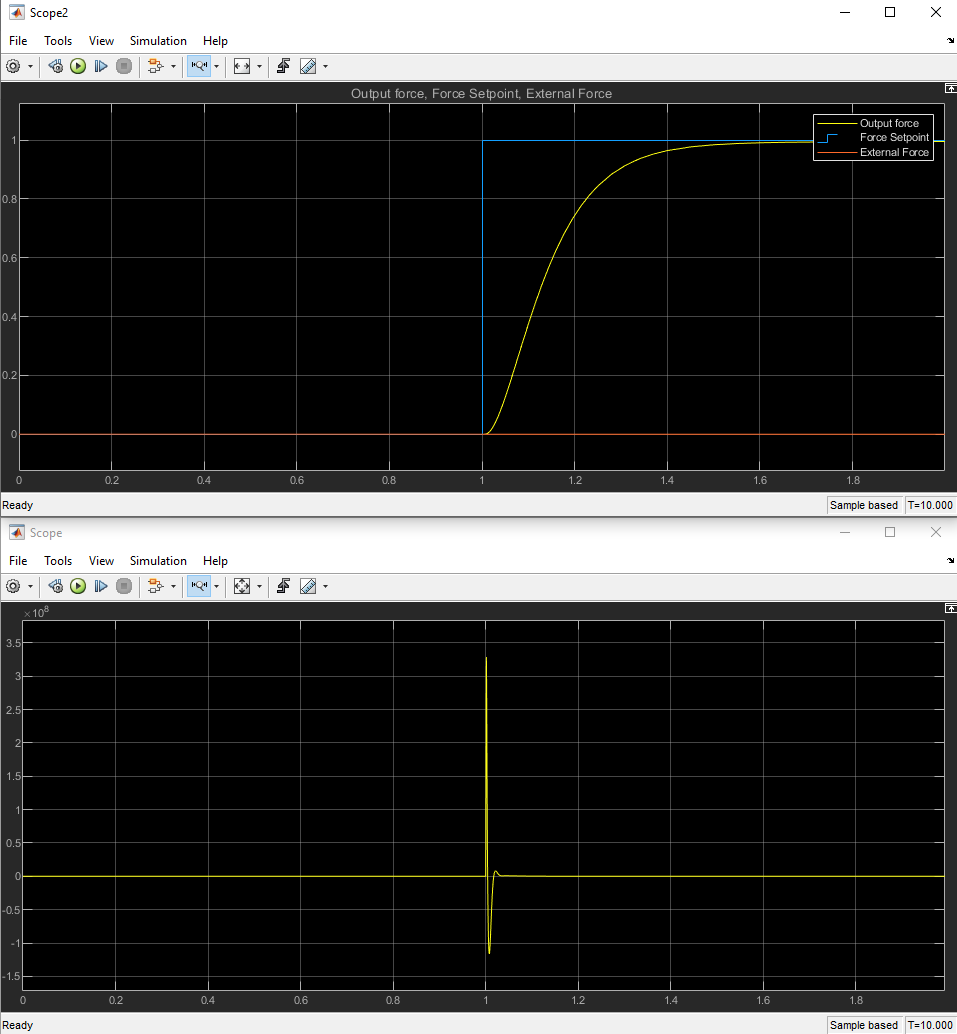
Figure 12

The simulations show that the controller is able to track a step of 1 well within 1 second. It is also capable of tracking a sinusoidal set point very accurately. When external forces are applied at the output, they are quickly compensated for and rejected by the controller. Overall the controller appears to be working as designed.

Conclusion

At first glance the controller seems to be working as it was intended. The controller is tracking the set point of force in the spring very accurately across a variety of conditions including different set points and different amounts of external force. In theory, the controller design would work very well for any application that requires applying a controlled amount of force. This is done by configuring the system so that one end of the spring is fixed to an object that we would want to apply a force to while the other end of the spring is fixed to the actuator. By setting the set point of the controller, we can control the force that is applied to the object, regardless of the dynamic external force that is applied back on the spring by the object.

Upon further investigation, it appears that the overall controller would be impossible to implement under real-world conditions. This is due to the very large control gains in the system. As shown in Figure 13, the result of simulating a setpoint of 1 with no external force being applied, the control input to the system peaks at an enormous 328,700,000 volts.



Figure

This type of voltage would be impossible to produce in the intended applications. In order to implement the controller in a real-world application the controller would need to be redesigned, perhaps at the expense of response time. In conclusion, the controller works as designed in theory and in simulation, but the design would need to be modified to be useful in any practical application.