ECE141 - Principles of Feedback Control

Winter 2018

Final Design Project:
Controller Design for a Fast Tool Servo (FTS)

Due: Wednesday, March 14, 2018

Contents

1 Introduction

In this project, you will design controllers for a Fast Tool Servo (FTS) system. This project is a slightly modified version of the project prepared by Prof. David Trumper at MIT and is based on the work done by X. Lu in his Ph.D. thesis. For more information, please refer to the associated course website:

(+) Link to the FTS Design Project for Course 2.14 at MIT

Also to review Fast Tool Servo technology and applications, as well as recent advances, you can refer to one of their articles here:

(+) Fast Tool Servos: Advances in Precision, Acceleration, and Bandwidth

The design project involves the following tasks:

(i) Design an OpAmp-based high-bandwidth controller for the current control loop for driving the actuator.

(ii) Identify the dynamics of the FTS electromechanical plant from the measured frequency response data.

(iii) Design a position controller for the servo system in order to eliminate the steady-state tracking error for a constant position reference (i.e., step input), and at the same time, minimize the steady-state tracking error for a sinusoidal reference position trajectory, in the presence of sensor noise.

Fast Tool Servos can be used with diamond turning machines and provide the ability to cut ultra-precise non-symmetric contoured micro features at very high speeds. They have found applications in different areas of high-precision industrial manufacturing such as contact lenses, optical films, micro-optical devices and lenses, etc.

Figure 1 shows an FTS used in conjunction with a diamond turning machine, and Figure 2 shows the high-level schematics of FTS:

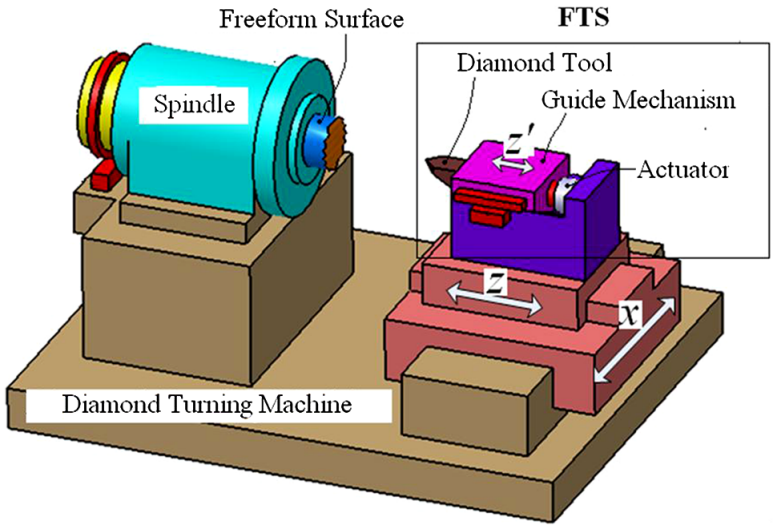

Figure 1: Fast Tool Servo used with a Diamond Turning Machine

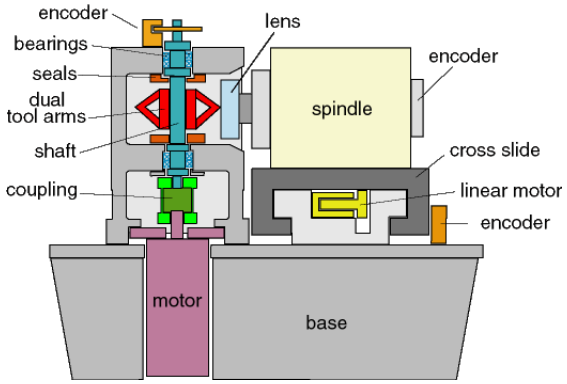

Figure 2: FTS Schematics

2 MATLAB Design Template

All your simulations for this project will be done in MATLAB. For your convenience and for consistent presentation of your results, you are provided with a MATLAB template m-file posted on CCLE: `ECE141_Final_Project_Template.m`
Please note the following:

(i) You must add your own code within the template to perform the different design tasks, and you must then initialize the transfer functions and the variables currently declared as [] within the template.

(ii) The variable names used in the template match the diagrams and the project description here. Do not change any variable names already declared in the template and do not overwrite any of the given values. You can obviously add any additional variables you may need as part of your design process.

(iii) Based on your design, the template will generate all the required plots and report all the required parameters.

3 Fast Tool Servo Control Loop

The overall control loop for the Fast Tool Servo system is shown in Figure 3:

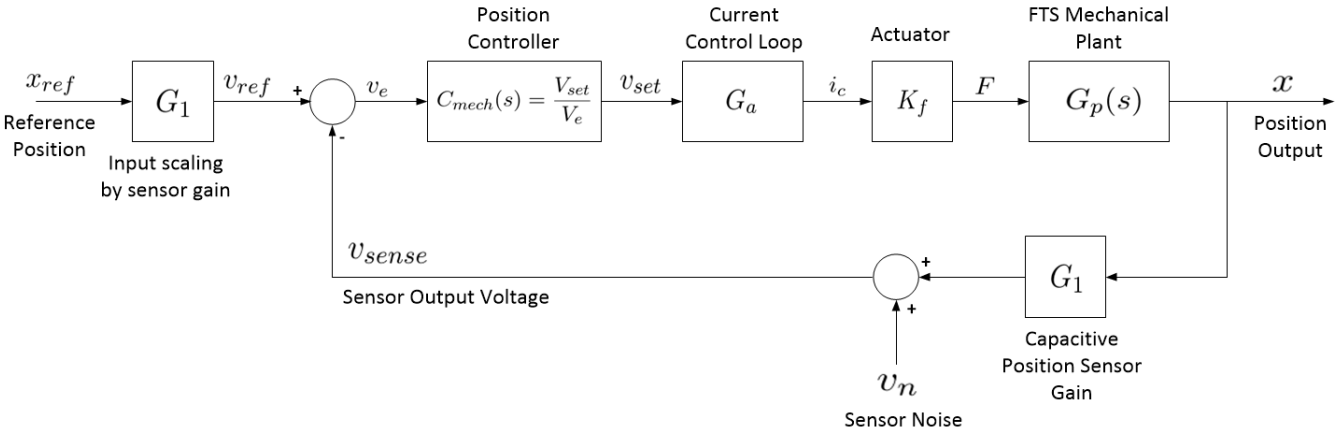

Figure 3: Fast Tool Servo Control Loop

Given the shape of the part to be cut by FTS, the ultimate objective is to control the position of the cutting tool $x(t)$ such that it track the desired reference position trajectory $x_{ref}(t)$ with mimimum error. As shown, the control loop consists of the following blocks:

(i) **FTS mechanical plant** $G_p(s)$: This is the process by which the actuator force $F(t)$ results in displacement $x(t)$ of the cutting tool. You are given the measured frequency response data for this system (`GpPlantFrequencyData.zip` posted on CCLE). Given this data, one of your design tasks is to identify the FTS plant dynamics.

(ii) **Actuator**: This is the motor whose current i_c is controlled such as to generate the required force $F(t)$ for FTS. It is assumed that the dynamics of the actuator can be described by a fixed gain $K_f = 20$ N/A.

(iii) **Current control loop**: This is the loop that, based on its input voltage v_{set}, will control the current i_c to drive the actuator. Your first design task is to design this current control loop. This loop will be designed such that its resulting dynamics will be fast enough compared to the bandwidth of the position loop, so that we can ignore its dynamics and replace the current loop with a constant gain $G_a = -0.5$ A/V as shown in the block diagram.

(iv) **Position Controller** $C_{mech}(s)$: This is the controller for the position loop. It will get the error between the reference voltage v_{ref} and the sensor output voltage v_{sense} and will generate the voltage v_{set} for the current control loop. Once you have identified the FTS plant dynamics, and you have designed the current control loop, your main design task would be to design this position controller and in doing so, *shape* the position loop transfer function such that the closed-loop FTS position control system meet your target performance in tracking the reference position trajectory.

(v) **Position Sensor**: A capacitive displacement sensor (i.e., a capacitance probe) is used to measure the position of the cutting tool. It is modeled by a constant gain $G_1 = 5 \times 10^5$ V/m, and its output is assumed to be corrupted by the measurement noise v_n.

4 Current Control Loop Design

The schematics of the current control loop is shown in Figure 4:

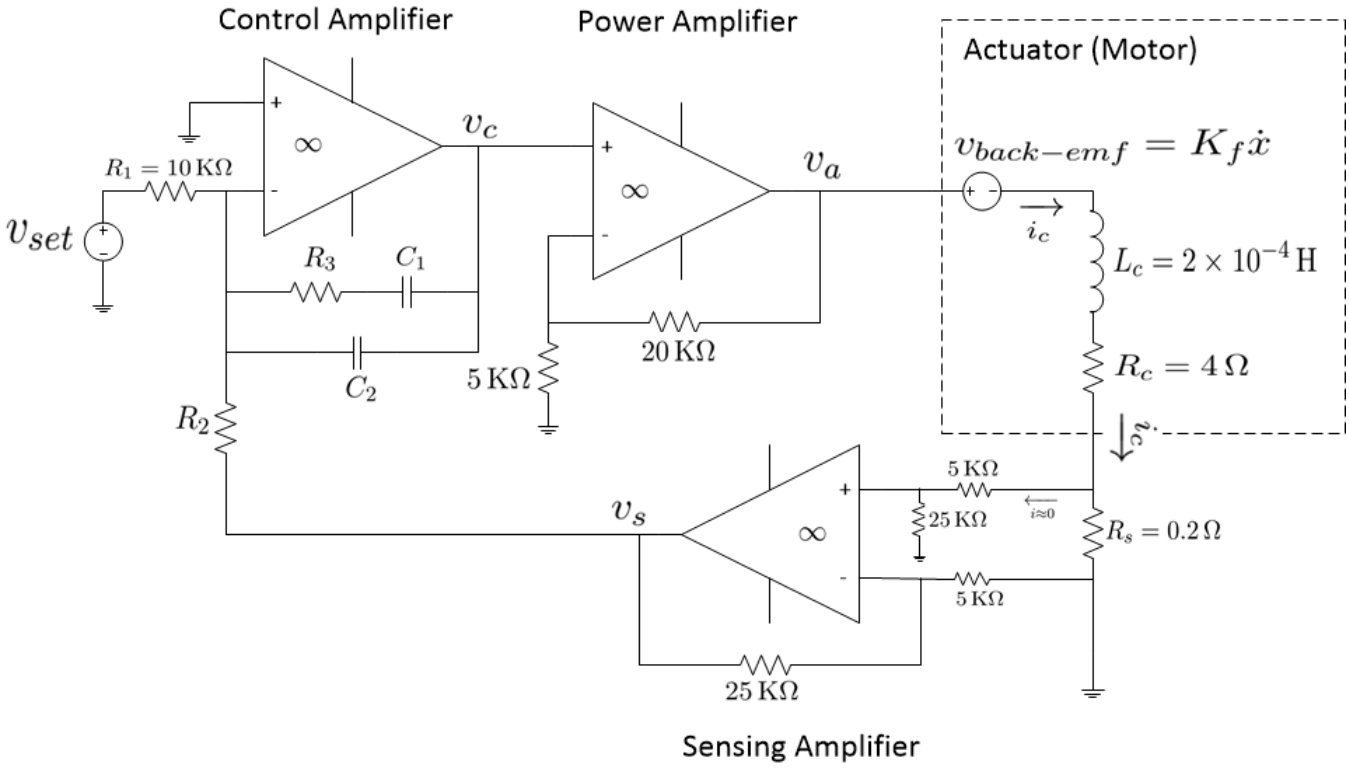

Figure 4: Current Control Loop Schematics

We can represent this current control loop with the block diagram shown in Figure 5:

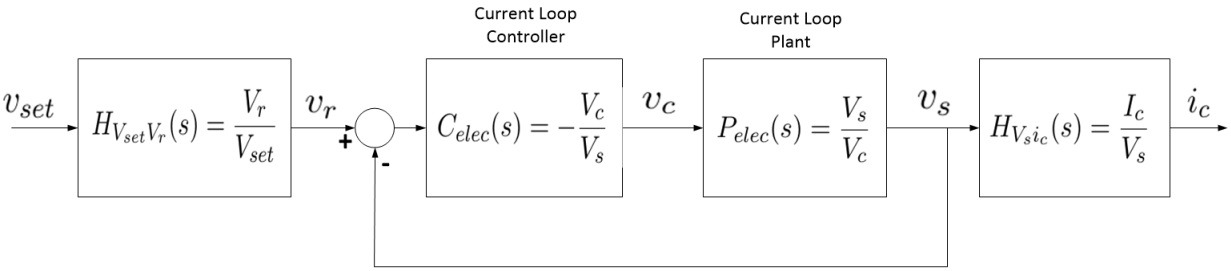

Figure 5: Current Control Loop Block Diagram

As shown on the schematics, the back electromotive force (back emf) from the actuator is modeled as a dependent voltage source, whose voltage is proportional to the velocity of the cutting tool: $v_{back\_emf} = K_f\dot{x}(t)$. Moreover, the force generated by the actuator is assumed to be proportional to the actuator current with the same gain: $F(t) = K_f i_c(t)$. A simple model describing the mechanical dynamics of FTS may be shown as in Figure 6 where the actuator force $F(t)$ is applied to a mass m_1 subject to resistance by a spring k_1 and a damper b_1.

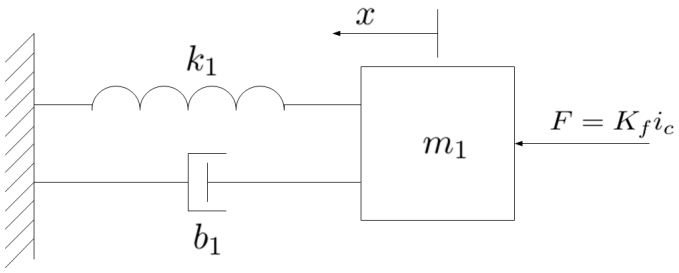

Figure 6: Simple model for the mechanical dynamics of FTS

In the project, assume ideal OpAmps with infinite open-loop gains and also assume the following values:

$$\text{Motor Resistance: } R_c = 4\,\Omega \tag{1}$$

$$\text{Motor Inductance: } L_c = 2 \times 10^{-4}\,\text{H} \tag{2}$$

$$\text{Motor Constant: } K_f = 20\,\text{N/A} \tag{3}$$

(a) Find the plant transfer function $P_{elec}(s) = \frac{V_s(s)}{V_c(s)}$ with the back emf voltage included. Note that, in this case, the FTS dynamics, as modeled in Figure 6, will also be reflected in your plant transfer function.

(b) Find the plant transfer function $P_{elec}(s) = \frac{V_s(s)}{V_c(s)}$ with the back emf voltage assumed to be zero. In this case, the FTS dynamics will obviously have no effect on your transfer function.

(c) Show that at high-frequencies, both transfer functions will be equal. Can you provide a physical explanation for this? This is the reason why back emf voltage may be ignored in designing current control loops for electromechanical systems, as long as nonlinear behavior of the amplifier (e.g., saturation) is not being modeled. As such, you can assume the back emf voltage to be zero in your contoller design in the next step.

(d) Design the current loop controller $C_{elec}(s) = -\frac{V_c(s)}{V_s(s)}$ such that the current loop transfer function has a gain crossover frequency of $\omega_c = 6 \times 10^5$ rad/sec, with a minimum phase margin of PM $\geq 60^\circ$. We also require that an input voltage of $v_{set} = -10$ V will result in a steady-state actuator coil current of $i_c = 5$ A. In other words, the full closed-loop current drive from v_{set} to i_c has a DC gain of $G_a = \frac{I_c}{V_{set}}\big|_{s=0} = -0.5$ A/V. Find the resulting component values of R_2, R_3, C_1, and C_2. Notice that $R_1 = 10\,\text{K}\Omega$. Also given that the sense resistor $R_s = 0.2\,\Omega$ is much smaller that the input resistors of the differential sensing amplifier, you can assume that no portion of the actuator current i_c will enter the inputs of the sensing amplifier.

(e) Provide the Bode plot of the current loop transfer function showing the gain crossover frequency and the phase margin (As mentioned earlier, once you initialize the transfer functions and the component values based on your design, the MATLAB template already has the code to generate all the required plots).

5 FTS Plant System Identification

As shown in Figure 3, the FTS plant $G_p(s)$ describes the dynamics by which the actuator force $F(t)$ yields the displacement $x(t)$ of the cutting tool. The measured Bode plot for $G_p(s)$ is shown in Figure 7:

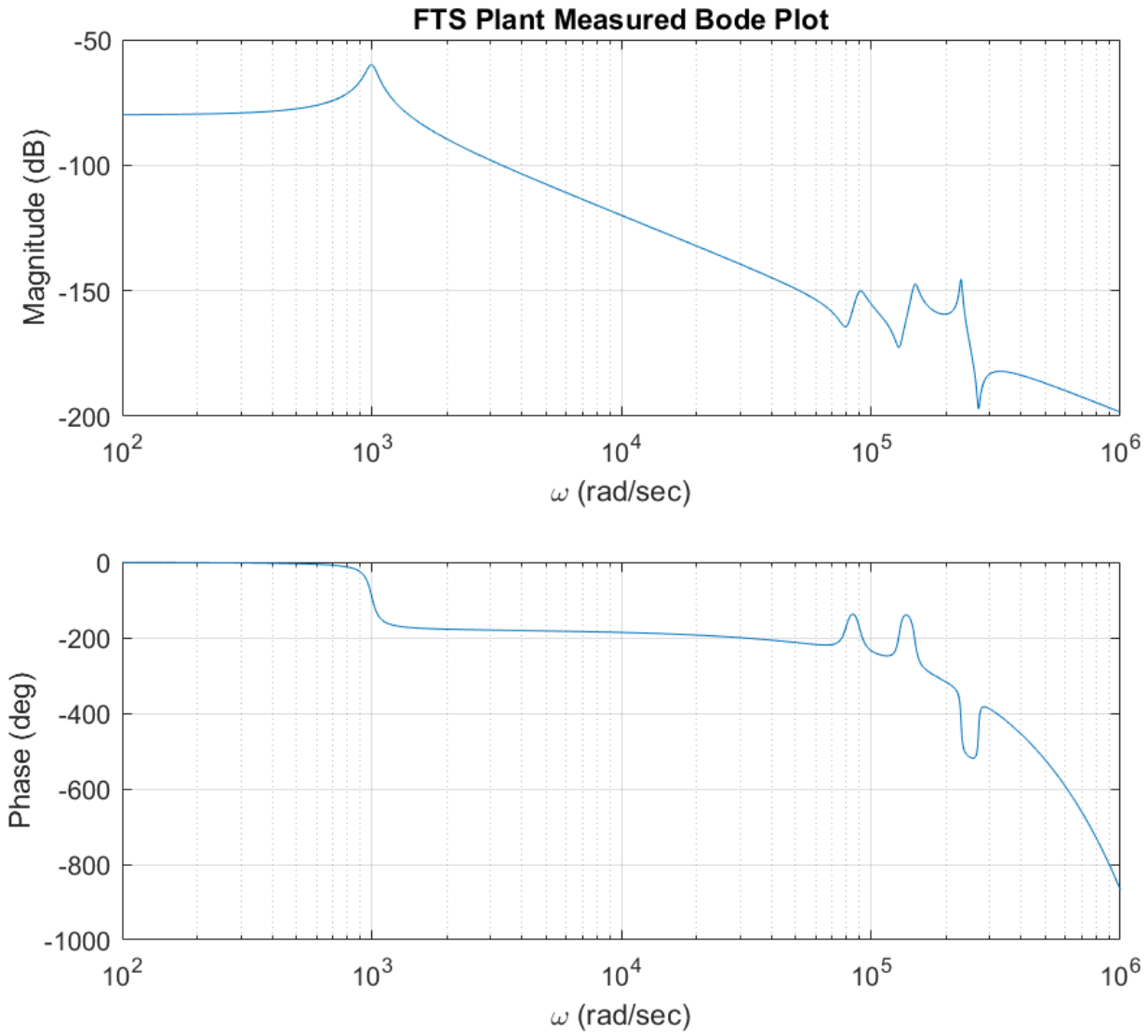

Figure 7: FTS Plant Measured Bode Plot

From the plot, it is clear that the plant dynamics are more complicated than what the simple model in Figure 6 would suggest. Namely, the mass/spring/damper model would only represent the lowest frequency mode seen on the plot, but there are also higher frequency resonance modes as well as a time delay.

The measured frequency response data is posted on CCLE. Please download `GpPlantFrequencyData.zip` and unzip it in the same directory as your project template file. As you will notice in the MATLAB template file, the data will be loaded with `load` command, and you will have both magnitude data `Gpmag` (in linear scale) and the phase data `Gpphase` (in degrees) as well as the associated frequency

vector `ww` (in rad/sec).

(a) On the basis of the lower frequency dyamics, as shown by the first mode, find the values for the mass m_1, the spring constant k_1, and the damper constant b_1 in the simple FTS model shown in Figure 6.

(b) Fit a dynamical model to the measured Bode plot. Provide the transfer function $G_p(s)$, and show its frequency response overlaid on the measured data (Again, this will already be done in the template m-file, once you initialize $G_p(s)$). (Note: You may try to identify the plant with visual inspection of the Bode plot. Alternatively, you may form a frequency response data object by `frd(Gpmag.*exp(1j*Gpphase*pi/180),ww)` function, and then use `tfest` function with a given number of poles and zeros to identify the plant. Or you may use a combination of the two where you would estimate the delay by inspection of the phase plot, and provide it as an input parameter to `tfest` function.)

(c) You will use the transfer function you identify here for the position controller design in the next step. If you find it too challenging to stabilize the closed-loop system and meet the reference tracking requirements, in order to proceed with the design, revisit your plant model, and possibly simplify it by ignoring the higher frequency modes and only modeling the low frequeny mode along with the time delay.

6 Position Control Loop Design

The position contol loop is shown in Figure 3. The task here is to design the position controller $C_{mech}(s)$, which gets the error between the reference voltage v_{ref} and the sensor output voltage v_{sense} and generates the voltage v_{set} for the current control loop. You will have to use the plant transfer function $G_p(s)$ that you identified earlier. Also, as mentioned earlier, given the current control loop that you designed with very high crossover frequency, you can replace it with a constant gain $G_a = -0.5$ A/V when designing your position controller. Finally, assume the position sensor gain to be $G_1 = 5 \times 10^5$ V/m and the motor constant to be $K_f = 20$ N/A.

For the position controller design, we will assume two reference inputs:

$$1\ \mu\text{m Constant Position Trajectory: } x_{ref_1}(t) = 10^{-6}u(t)\ \text{[m]} \tag{4}$$

$$\text{Sinusoidal Position Trajectory: } x_{ref_2}(t) = 10^{-5}\sin(3000t)\ \text{[m]} \tag{5}$$

We will also assume the sensor measurement noise to be modeled as:

$$v_n(t) = 5 \times 10^{-2}\sin(10^5 t)\ \text{[V]} \tag{6}$$

(a) Design the position controller $C_{mech}(s)$ with as good performance as you can achieve. Your objective is to *shape* the position loop transfer function such that:

   (i) the closed-loop FTS system is obviously stable, and

   (ii) the steady-state position tracking error for the step input is almost fully eliminated, and

   (iii) the steady-state position tracking error for the sinusoidal input trajectory is minimized (i.e., as small as you can get), and

   (iv) the steady-state effect of the measurement noise on the output position is minimized, and

   (v) the magnitude of the closed-loop Sensitivity Function stays below +10 dB at all frequencies.

(b) Plot the closed-loop step response to $x_{ref_1}(t)$ and present the transient response characteristics (already done in the template, once you add your controller transfer function). Is the overshoot in your response reasonable? Can you further reduce the overshoot in your step response? What are the trade-offs you have to deal with?

(c) Plot Bode and Nyquist diagrams of the loop transfer function, as well as the closed-loop sensitivity function (already done in the template).

7 Project Deliverables

The project deliverables include:

(i) **Project Report**: A few pages describing the design process you followed including how you came up with your controllers, and how you addressed the design trade-offs and optimized your controllers. Also please include all the relevant calculations, data, and plots.

(ii) **MATLAB m-file**: This will be the MATLAB m-file where you started with the provided template and added all the steps in your design simulation and properly initialized all the required values and transfer functions.

Project Grading:

The project grade will be based on your design effort and the quality of your design in meeting the requirements, as presented in your project report and as shown by your MATLAB file. As such, it is critically important that your MATLAB file be bug free, i.e., it should run smoothly and generate the intended results and plots based on your design. We will NOT debug any MATLAB file that fails to run properly, and you will be given very little, if any at all, partial credit for your project.