<u>Engineering 446</u>

Control Systems Laboratory

Farnam Adelkhani

September 11, 2016

Lab Assignment #2
System Response Using MATLAB
Control Systems Toolbox

Introduction

The fundamental objective when conducting this laboratory assignment is to study the step-response characteristics of first and second-order systems while simultaneously making an introduction to the Control Systems Toolbox. There are various functions available within the domain of the toolbox in the standard command window mode but the goal here is to learn to write our own script files to incorporate the functions we desire to enact. This assignment places priority on the principles of the step-response, however the observer may also like to observe the response of the system to other standard inputs such as a pulse, ramp, impulse and a sin wave. For this lab the user will be required to create the system whose response is desired to be simulated. This laboratory assignment covers use of Matlab publisher to simplify documents for turning in.

Problem Definition

This experiment requires the user to observe the results of system settings and system parameters on the overall step-response of the 1^{st} and 2^{nd} order linear system. The focus of this lab is the study of the step-response. Optionally, the user may want to observe the response of the system to other inputs such as ramp, impulse, pulse and a sin wave. For this laboratory assignment the user will need to create a system with a response to simulate. The tf command, from the control systems toolbox, is used to achieve this desired output. Once this output is achieved, the user will then use functions such as; step, impulse and lsim to simulate the system response to different sets and types of input signals. This lab session is also to be used to familiarize the user with the plotting features of matlab. In particular, common functions such as axis, zoom, title, ect.

Explanation of the experiments

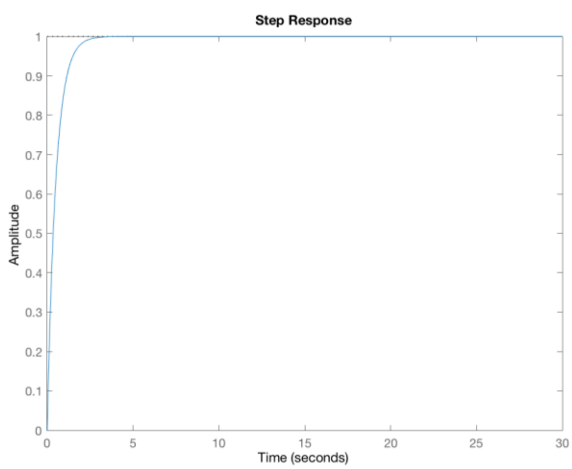
The control system toolbox will be used to acquire the step-response of the process whose transfer function, G(s). The unit step-response will be plotted for this process in order to find the range of parameter values indicated. By merely observing the effect of the time-constant, toa, and gain onto the step response of the 1st order systems.

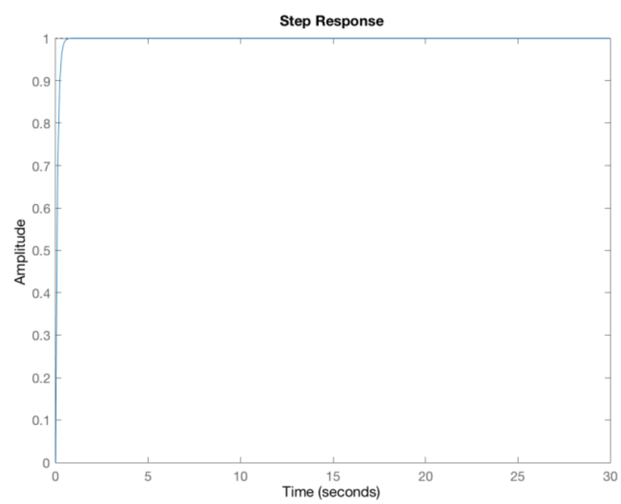
There are four parts to this assignment beginning with representing a given function, G(s) = k/ ts+1, and plotting the function onto a graph. For this particular section toa will be manipulated to produce a set of graphs to compare the output. After this is achieved, the user will post all the graphs onto a single plot for a representation that is more simple to see. The "hold" command is used to achieve this goal. For part b of section 1 of this lab assignment the user is asked to instead manipulate the K value and keep toa constant. The output is then recorded and plotted.

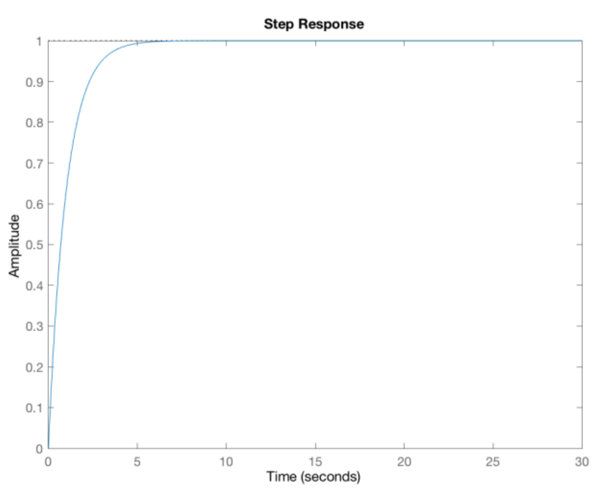
The basic MATLAB commands used are also given as examples throughout the assignment. "s = tf(x) " is a term used to allow the user to create other transfer functions directly in terms of the variable provided in place of x. Next the variables are defined and some need to be defined over a series of values. The final time is set and the figure and hold commands are presented forward. The titles and axes information for the plots is then edited to conform to the values and titles desired. It is also possible to insert text at set points on the plot by using the appropriate commands.

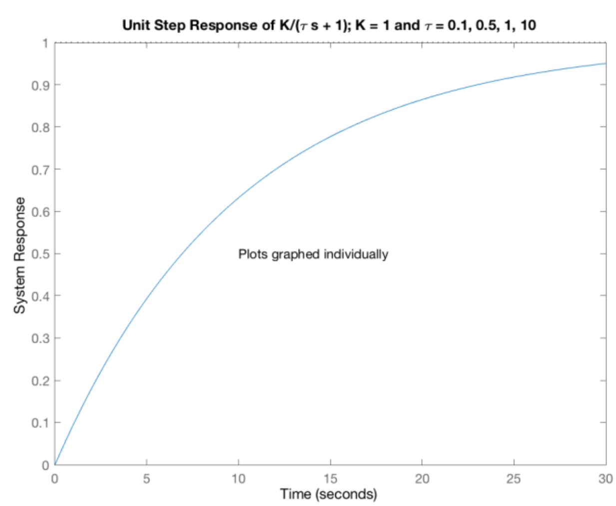
Models/Calculations/ Simulation Results

In **part 1** the user generated a set of plots given the function and desired intervals. The graphs below represent the output of G(s) = k / ts+1 at t=0.1, 0.5, 1 and 10 seconds. The first section of this problem requires the user to plot all 4 graphs on separate domains. The instructions direct the user to also plot all the graphs on a single chart so this is the next goal…

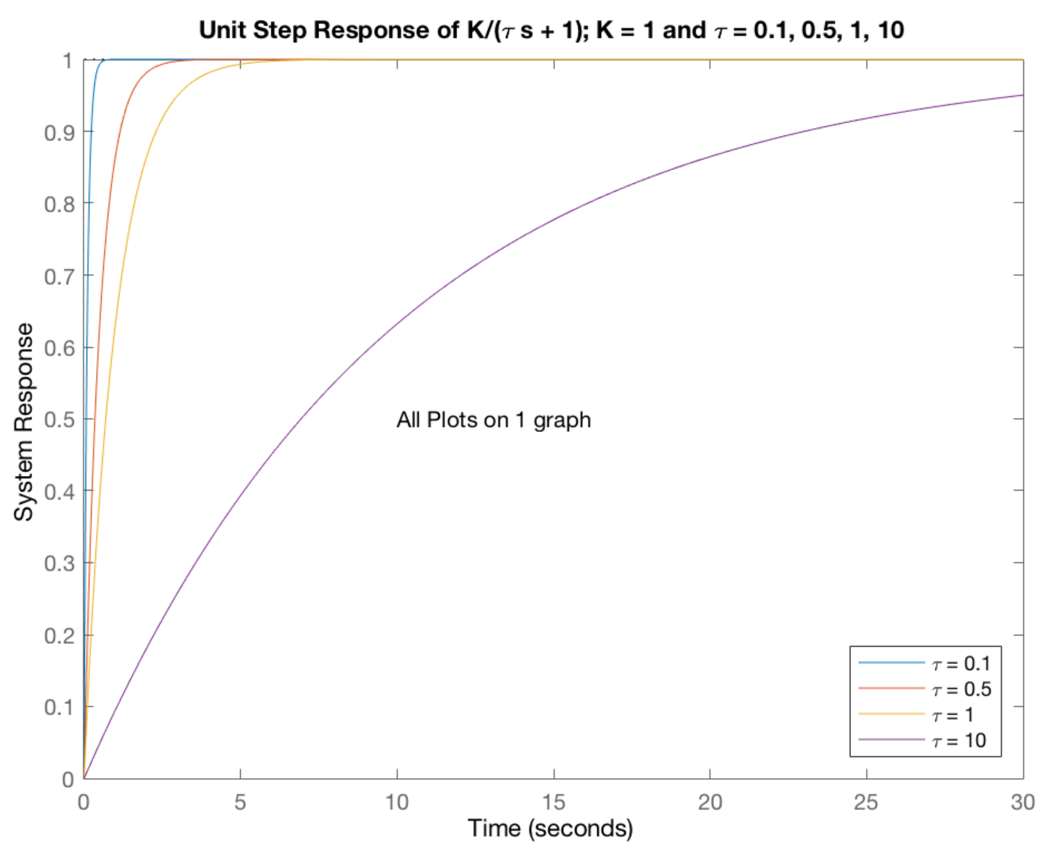




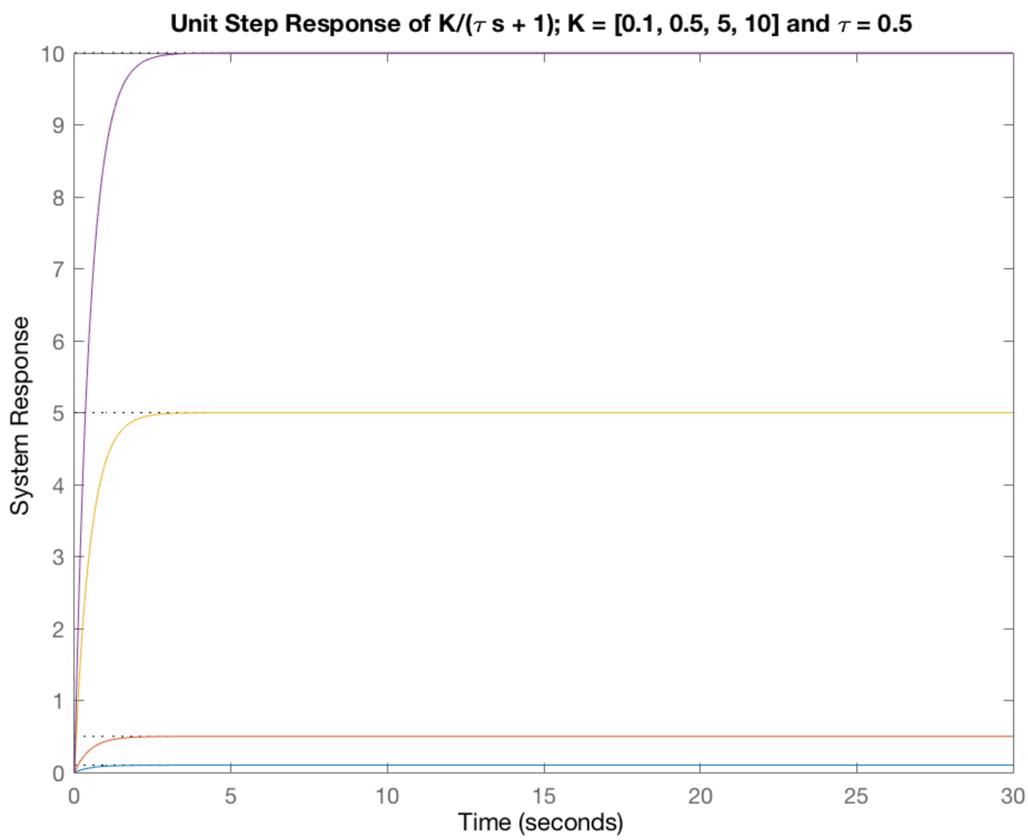




The plot shown characterizes all four charts onto a single graph. This is very helpful to allow the user to compare features of the data and compare the trends. I would imagine that in most cases the user would want to plot the graphs onto a single plane in order to compare results so this is a necessary and essential function to keep in mind. This will be revisited many times, undoubtedly.



For **part 2** of this laboratory assignment we utilize the same function that was being used in part 1, however, we now set a range of K value and just one value for toa. Adjustments must be made to the title and legends of the plot to accommodate the new data, as shown below:



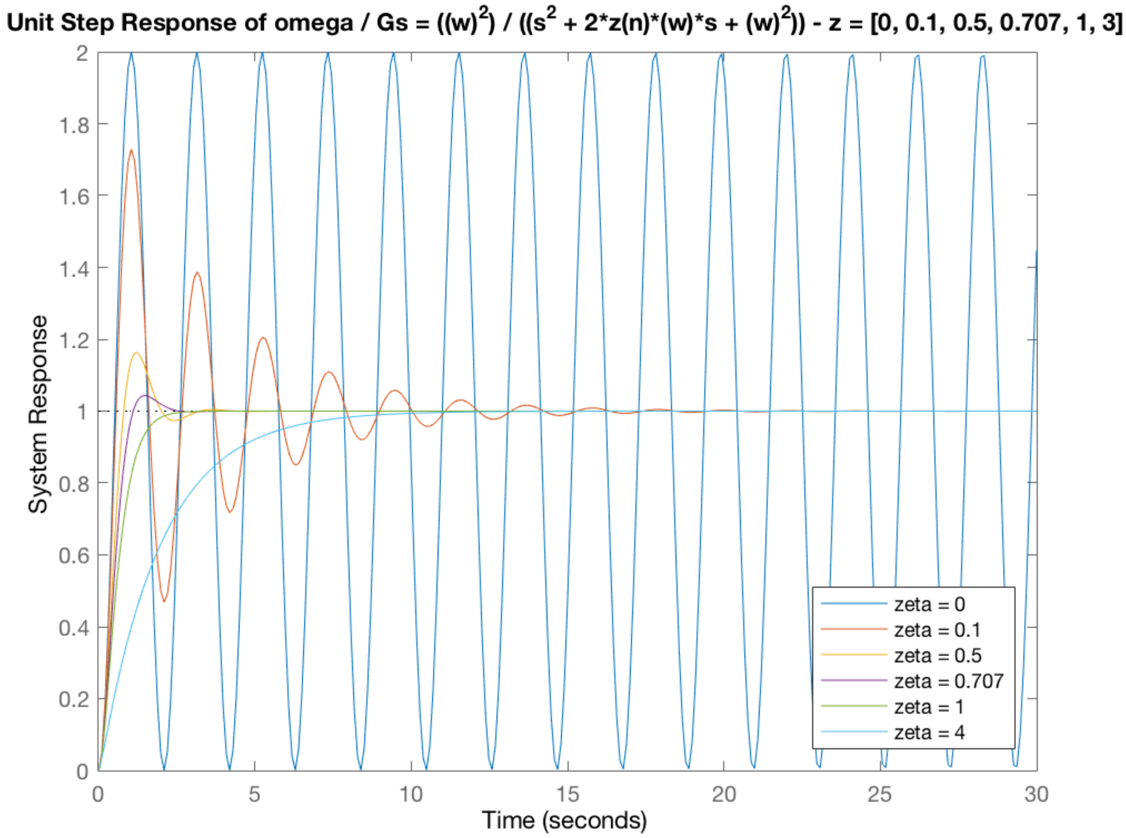
Skipping the step of plotting this function repeatedly over a set of plots, the plots are all generated onto the same axis. This helps to save space and expose some relationships that can be more readily observed.

In **part 3** of this laboratory assignment the user is asked to represent the function;

G(s) = (omega)^2/S^2 + 2z(omega)s + (omega)^2 when …

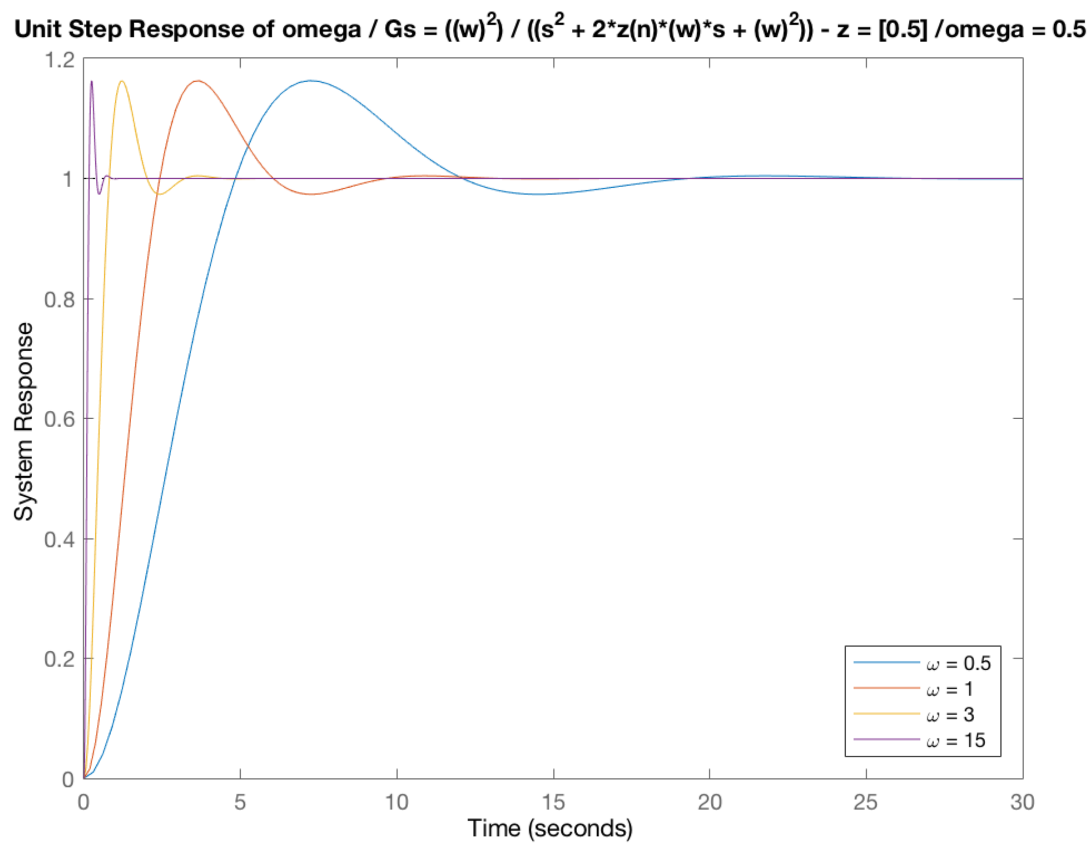
omega=3 rad/sec and z = 0, .1, .5, .707, 1 and 3.

This second order transfer function is represented in the plot shown below.



Changes must be made again, to the title and legend to accommodate changes to the plot.

Finally, in **part 4** of this assignment the user will keep working with the previously described function in order to produce a new plot with varying values of omega and a stationary value of zeta. Again, chart data such as legend and title where manipulated to accommodate the changes.



The plot shown above is accurate in it's representation of the data as well as using different colors, titling the axis correctly, titling the graph correctly and lastly providing and accurate and useful legend to follow.

Conclusion

It is a powerful resource to understand and have at your fingertips once you understand and can become comfortable with MATLAB plotting. As an engineer, we are going to be faced with new tools and techniques as technology and science continue to advance. Matlab has been and continues to be very popular in the fields of science, mathematics and engineering. As engineers working as professionals, it is likely that we will all encounter and will have to use MATLAB at some point, this includes Simulink and other toolboxes. MATLAB is a tool that many engineers find actual use for when entering industry. Many job openings and opportunities exist for users who are fluent and comfortable with using MATLAB.

List of References

"Simulink Basics Tutorial", SFSU 446 Laboratory manual book. Provided by University.

"System Response using MATLAB control systems toolbox", SFSU 446 Laboratory manual book. Provided by University.

Appendix: HTML FILE PUBLISHED IN MATLAB WITH CODE AND GRAPHS

The code below does not include images since they have already been shown on previous pages. The intention of this section is to show the code.

ENGR 447 - Lab2

Matlab tutorial

Contents

- Lab 1_3
- Lab 1_4

Lab 1_1

```
type lab2part1

lab2part1
```

```
%% LAB2 Part i Code



s = tf('s'); % Later create other transfer functions in terms of s

K = 1;

tau = [0.1, 0.5, 1, 10]; % tau?s are put into a vector.

tfinal = 30; % Play with sim. end time to find best fit

figure % Initiate a new figure.

hold; % Hold plotting until several plots are put into the same figure.



% Use a ?for loop? to find the step response for each tau:

% Note: Don?t use i or j as loop index, they are reserved for square root of -1.

for n = 1:4

 Gs = K/(tau(n)*s + 1) % Define the transfer function.

 step(Gs, tfinal)      % Generate the step response.

end

hold; % Release the plot hold.
```

```
      % For adding title, axes labels and legend:

title('Unit Step Response of K/(\tau s + 1); K = 1 and \tau = 0.1, 0.5, 1, 10')

xlabel('Time')

ylabel('System Response')

legend('\tau = 0.1','\tau = 0.5',... % ... means continued on next line.

 '\tau = 1','\tau = 10','location','southeast')

text(10, .5, 'All Plots on 1 graph')


%% Part i code continued - Plotting graphs invidually


s = tf('s'); % Later create other transfer functions in terms of s

K = 1;

tau = [0.1, 0.5, 1, 10]; % tau?s are put into a vector.

tfinal = 30; % Play with sim. end time to find best fit

%figure % comment out - pause figure initiation

%hold; % comment out - hold


% Use a ?for loop? to find the step response for each tau:

% Note: Don?t use i or j as loop index, they are reserved for square root of -1.

for n = 1:4

    figure %Added this

 Gs = K/(tau(n)*s + 1) % Define the transfer function.
```

```
 step(Gs, tfinal)      % Generate the step response.

end

%hold; % Release the plot hold.

      % For adding title, axes labels and legend:

title('Unit Step Response of K/(\tau s + 1); K = 1 and \tau = 0.1, 0.5, 1, 10')

xlabel('Time')

ylabel('System Response')

%legend('\tau = 0.1','\tau = 0.5',... % ... means continued on next line.

 %'\tau = 1','\tau = 10','location','southeast')

text(10, .5, 'Plots graphed individually')


Current plot held


Gs =


      1

  ---------

  0.1 s + 1


Continuous-time transfer function.
```

```
Gs =

      1
  ---------
  0.5 s + 1

Continuous-time transfer function.

Gs =

    1
  -----
  s + 1

Continuous-time transfer function.

Gs =

     1
  --------
```

```
  10 s + 1

Continuous-time transfer function.

Current plot released

Gs =

      1
  ---------
  0.1 s + 1

Continuous-time transfer function.


Gs =

      1
  ---------
  0.5 s + 1

Continuous-time transfer function.
```

```
Gs =

    1
  -----
  s + 1

Continuous-time transfer function.


Gs =

     1
  --------
  10 s + 1

Continuous-time transfer function.
```

Lab 1_2

```
type lab2part2
```

```
lab2part2
```

```
%% Part ii - Same as above with changes to tau and K

% Specifiy a new section with the above command

s = tf('s'); % Later create other transfer functions in terms of s

K = [0.1, 0.5, 5, 10];

tau = 0.5; % tau?s are put into a vector.

tfinal = 30; % Play with sim. end time to find best fit

figure % Initiate a new figure.

hold; % Hold plotting until several plots are put into the same figure.


% Use a ?for loop? to find the step response for each tau:

% Note: Don?t use i or j as loop index, they are reserved for square root of -1.

for n = 1:4

 Gs = K(n)/(tau*s + 1); % Define the transfer function.

 step(Gs, tfinal)      % Generate the step response.

end

hold; % Release the plot hold.

      % For adding title, axes labels and legend:

title('Unit Step Response of K/(\tau s + 1); K = [0.1, 0.5, 5, 10] and \tau = 0.5')

xlabel('Time')

ylabel('System Response')
```

```
Current plot held

Current plot released
```

Lab 1_3

```
type lab2part3

lab2part3
```

```
%% Part iii -

s = tf('s'); % Later create other transfer functions in terms of s

w = 3;

z = [0, 0.1, 0.5, 0.707, 1, 3]; % tau?s are put into a vector.

tfinal = 30; % Play with sim. end time to find best fit

figure % Initiate a new figure.

hold; % Hold plotting until several plots are put into the same figure.



% Use a ?for loop? to find the step response for each tau:

% Note: Don?t use i or j as loop index, they are reserved for square root of -1.

for n = 1:6

    Gs = ((w)^2) / ((s^2 + 2*z(n)*(w)*s + (w)^2)) % Define the transfer function.

    step(Gs, tfinal)      % Generate the step response.

end

a=1;b=2;
```

```
hold; % Release the plot hold.

      % For adding title, axes labels and legend:

title('Unit Step Response of omega / Gs = ((w)^2) / ((s^2 + 2*z(n)*(w)*s + (w)^2)) - z
= [0, 0.1, 0.5, 0.707, 1, 3]' )

xlabel('Time')

ylabel('System Response')

legend('zeta = 0', 'zeta = 0.1', 'zeta = 0.5', 'zeta = 0.707',...

    'zeta = 1', 'zeta = 4', 'location', 'southeast')

Current plot held



Gs =



     9

  -------

  s^2 + 9



Continuous-time transfer function.





Gs =



         9

  ---------------
```

```
  s^2 + 0.6 s + 9

Continuous-time transfer function.


Gs =

        9
  -------------
  s^2 + 3 s + 9

Continuous-time transfer function.


Gs =

          9
  -----------------
  s^2 + 4.242 s + 9

Continuous-time transfer function.
```

```
Gs =

        9
  -------------
  s^2 + 6 s + 9

Continuous-time transfer function.


Gs =

        9
  --------------
  s^2 + 18 s + 9

Continuous-time transfer function.

Current plot released
```

Lab 1_4

```
type lab2part4
```

```
lab2part4
```

```
%% Part iv -

s = tf('s'); % Later create other transfer functions in terms of s

w = [0.5, 1, 3, 15];

z = 0.5; % tau?s are put into a vector.

tfinal = 30; % Play with sim. end time to find best fit

figure % Initiate a new figure.

hold; % Hold plotting until several plots are put into the same figure.



% Use a ?for loop? to find the step response for each tau:

% Note: Don?t use i or j as loop index, they are reserved for square root of -1.

for n = 1:4

    Gs = w(n)^2 / ((s^2 + 2*z*w(n)*s + w(n)^2)) % Define the transfer function.

    step(Gs, tfinal)      % Generate the step response.

end

a=1;b=2;

hold; % Release the plot hold.

      % For adding title, axes labels and legend:

title('Unit Step Response of omega / Gs = ((w)^2) / ((s^2 + 2*z(n)*(w)*s + (w)^2)) - z
= [0.5] /omega = 0.5' )

xlabel('Time')
```

```
ylabel('System Response')

legend('\omega = 0.5', '\omega = 1', '\omega = 3', '\omega = 15',...

    'location', 'southeast')

Current plot held



Gs =



         0.25

  ------------------

  s^2 + 0.5 s + 0.25



Continuous-time transfer function.





Gs =



       1

  -----------

  s^2 + s + 1



Continuous-time transfer function.
```

```
Gs =

        9
  -------------
  s^2 + 3 s + 9

Continuous-time transfer function.


Gs =

        225
  ----------------
  s^2 + 15 s + 225

Continuous-time transfer function.

Current plot released
```

Published with MATLAB® R2016a