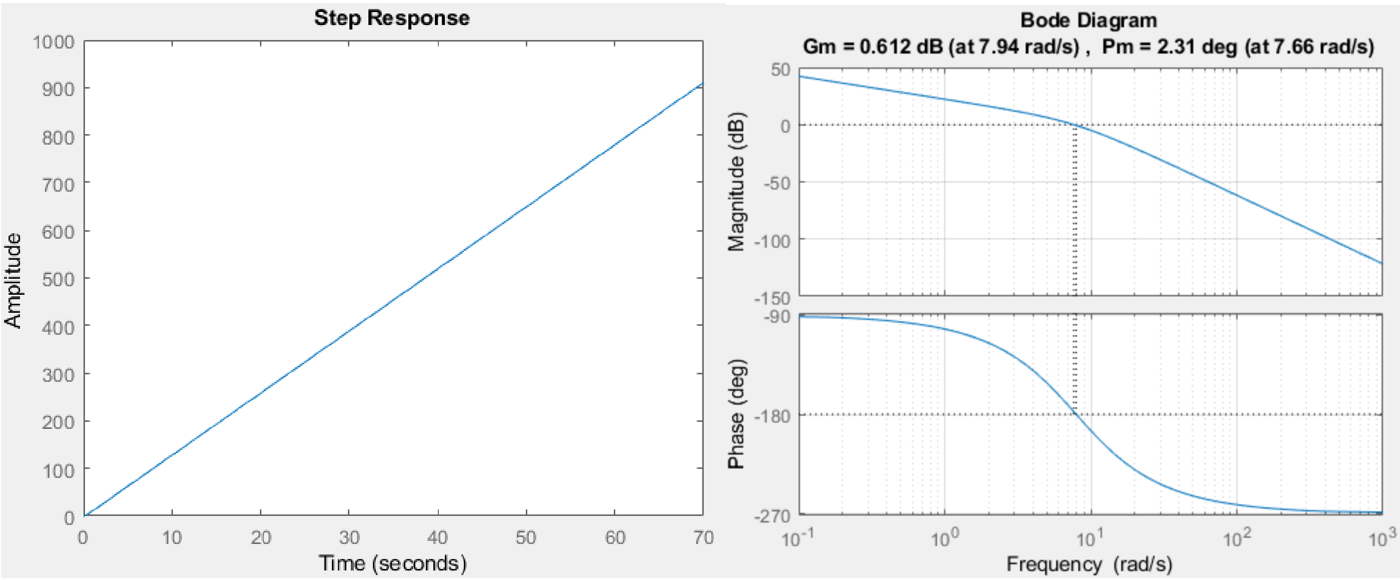
Introduction

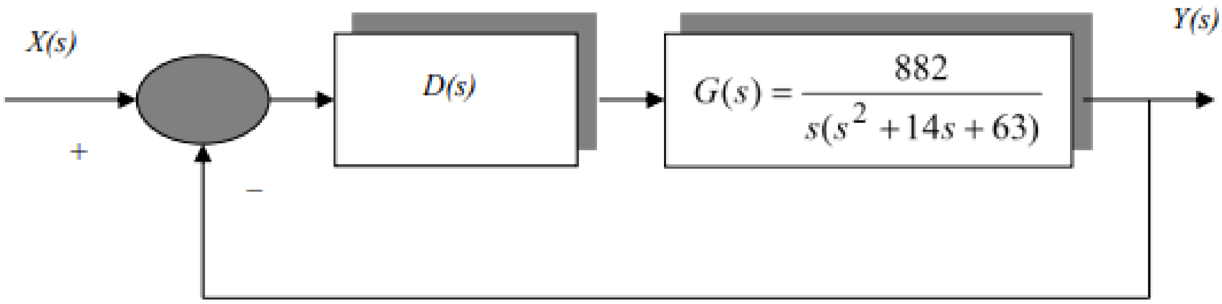
```
G =

          822
  -------------------
  s^3 + 14 s^2 + 63 s

Continuous-time transfer function.
```

System step response showing marginally stable behaviour and bode plot showing poor gain and phase margins.



The task requirements were to design a suitable compensator with the aid of MATLAB/SIMULINK and then assess the performance of the compensator. The design was aimed at satisfying the following specifications:

- Percentage overshoot < 15%.
- Rise-time < 100 msec.
- Settling time < 500 msec to the 5% definition of the settling time instead of the standard 2%
- Zero steady-state error to a step.



S-Domain Compensator design

The first step in designing the compensator was to convert the above time domain specifications into frequency domain specifications:

```
%% Requirements Frequency Domain Calculations
% Percentage overshoot < 15%
% Rise Time < 100ms
% Settling Time < 500ms within 5%

A = log(15);
Rzeta = A/(sqrt(pi^2 + A^2)); % desired damping ratio
CPm = 100*Rzeta; % desired compensated phase magring in degrees

% tan(Cpm) = 8/(Ts*Wc)
%increasing threshold margin
% At Rwc of 7.36, Phase angle = -175
% However i will aim at a number < Ts of 200ms
% to ensure target is met
Rwc = (8)/(0.20*tan(deg2rad(CPm)));

% At Rwc of 18.4056, Phase angle = -227
% additional Phase angle is:
AdPa = CPm + 227 - 180;
```

With the frequency domain specification values of damping ratio (Rzeta) and natural frequency used to calculate desired phase angle and cross-over frequency respectively and I then proceed to design a **Lead** compensator to meet the requirements.

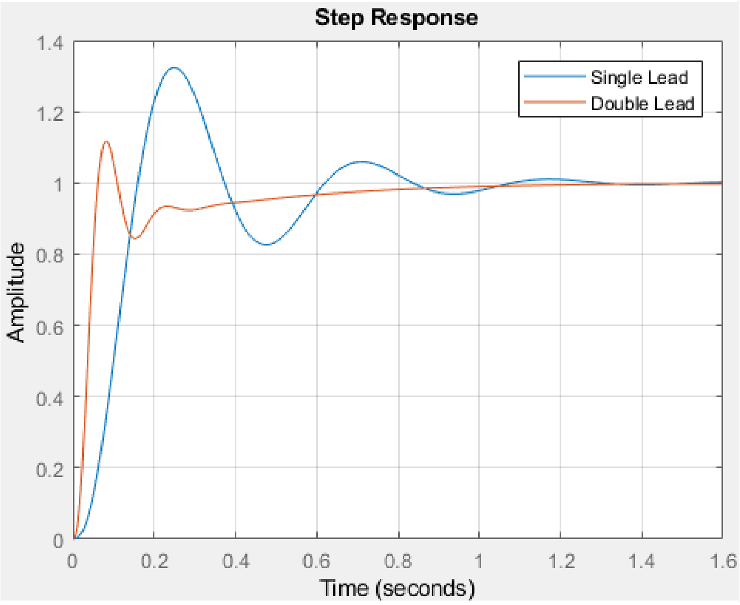
```
%% LEAD COMPENSATOR POLE AND ZERO CALCULATIONS

CPml = deg2rad(AdPa+5); % added 5 degrees
% sqrt(Z*P) = Rwc
alpha = ((1 + sin(CPml))/(1 - sin(CPml)));

zs = Rwc/sqrt(alpha);
ps = alpha*zs;

% lead compensator is
Gcl = tf([1/zs 1],[1/ps 1]);
```

```
Gcl =

  0.2238 s + 1
  -------------
  0.01319 s + 1

Continuous-time transfer function
```

In order to achieve the stipulated specifications, I aimed for slightly better time domain specifications as through experimentation I discovered that it would be difficult to meet the requirements due to attenuation cause by the integrator Lag present in the system.

I aimed for a settling time of **200 milliseconds** instead of the stipulated 500 milliseconds. This increases the phase that could be added on to the system and gives a greater stability margin.

S-Domain Response Plots

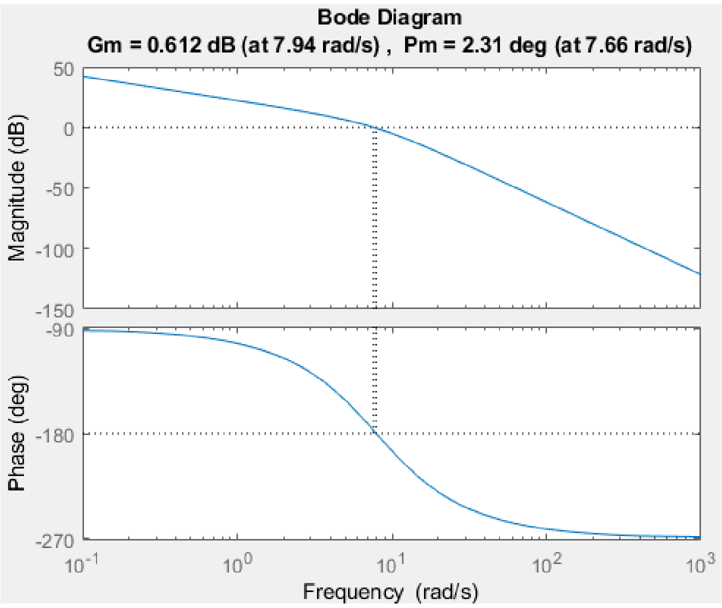


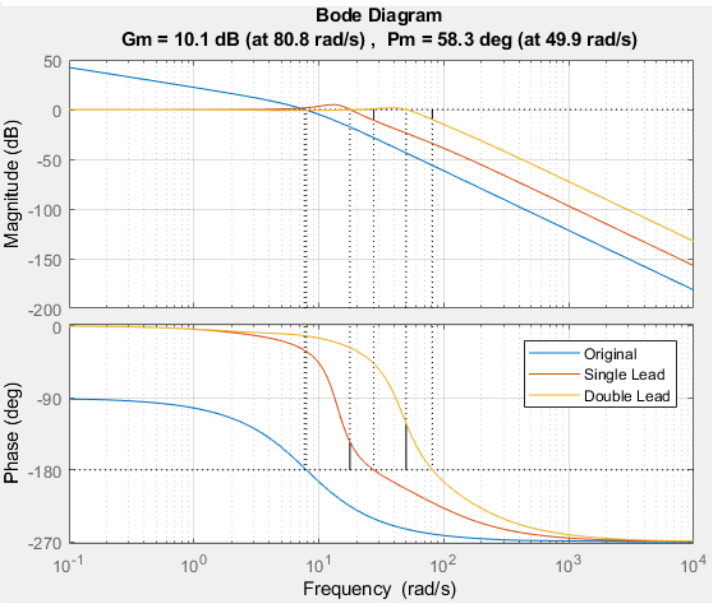

```
stepdataA =

  struct with fields:

        RiseTime: 0.100895291596358
    SettlingTime: 0.750557691349159
     SettlingMin: 0.825593188007421
     SettlingMax: 1.325067114799585
       Overshoot: 32.506711479958497
      Undershoot: 0
            Peak: 1.325067114799585
        PeakTime: 0.247859097886349


stepdataB =

  struct with fields:

        RiseTime: 0.038158553017003
    SettlingTime: 0.447803617497515
     SettlingMin: 0.843696926140655
     SettlingMax: 1.117710277339888
       Overshoot: 11.771027733988770
      Undershoot: 0
            Peak: 1.117710277339888
        PeakTime: 0.082208345878508
```

In implementing the Lead-compensator, I found that in applying a single lead compensator, the performance specifications were not met. Thus, a second Lead compensated was applied in series with the system. This then allowed the specifications to be met as show in stepdataA and stepdataB.

Requirements	Single Lead	Double
Percentage Overshoot (%)	32.5	11.77
Rise Time (s)	0.1	0.038
Settling Time (s)	0.75	0.448

S-Domain Bode and Margins





This Bode diagrams compare the Uncompensated system to that compensated with both single and double Lead compensators. This shows an increase in the Phase Margin of the system from its original 2.31 degrees to 58.3 degrees.

Z and W-Domain System

```
%% DISCRETISATION OF SYSTEM AND ZtoW-DOMAIN

% New Desired cross over frequency in  W-domain
% With New Desired Ts at 300ms instead of 200ms
% used in  S-domain. obtained through trial and error
Rwcw = (8)/(0.30*tan(deg2rad(CPm)));
% The desired system bandwidth
ft = ((4/0.30)*(1/Rzeta)*sqrt(((1-(2*(Rzeta^2)))+sqrt((4*(Rzeta^4))-(4*(Rzeta^2))+2))));
T = 1/(20*ft);
% In order to determine the sampling time the system
% bandwidth is calculated and 20 X bandwidth is used
% to avoid interference and aliasing.
[ZZnum,ZZdnum] = c2dm(822,[1 14 63 0],T);
% The Z-transform of G is taken
minZZnum = [-1*ZZnum(1) ZZnum(2) -1*ZZnum(3) ZZnum(4)];
minZZdnum = [-1*ZZdnum(1) ZZdnum(2) -1*ZZdnum(3) ZZdnum(4)];

[YYnum,YYdnum] = bilinear(minZZnum,minZZdnum,0.5);
% Due to the nature of the bilinear transformation
% -Z is used instead of Z
factor = 1/((-T/2)^3);
WWnum = YYnum.*[(-T/2)^3 (-T/2)^2 -T/2 1]*(factor);
WWdnum = YYdnum.*[(-T/2)^3 (-T/2)^2 -T/2 1]*(factor);
Gw = tf(WWnum,WWdnum);
% In order to translate the bilinear transform into
% the W-domain  each coefficient 1 to 1 multplied by
% increaing powers of -T/3
```

Before discretisation can occur, an appropriate sampling time must first be calculated. This is done by first calculating the desired bandwidth by applying:

$$\omega_B = \frac{4}{T_s \zeta}\sqrt{(1-2\zeta^2)+\sqrt{4\zeta^4-4\zeta^2+2}}$$

$$T_s = settling - time$$

$$\omega_B = Bandwidth$$

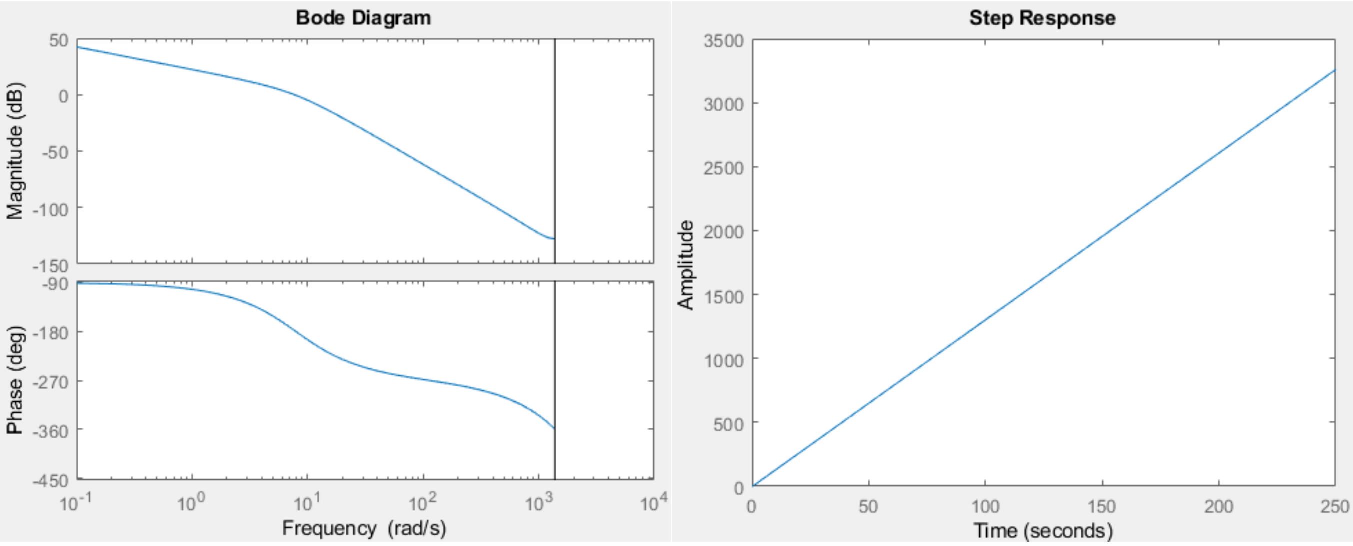
$$\zeta = damping\ ratio$$

This is then multiplied by 20 to adhere to the Shannon-Nyquist sampling requirements and avoid aliasing, and the whole thing is used to divide one to find the sampling time.

This sampling time, T, is then used in conjunction with the MATLAB function c2dm() which is then used to Find the Z-transform of the original uncompensated system. This gives:

```
1.6e-06 z^2 + 6.351e-06 z + 1.575e-06
-------------------------------------
 z^3 - 2.968 z^2 + 2.937 z - 0.9687

Sample time: 0.002275 seconds
Discrete-time transfer function.
```

The discrete Bode and the response to a step input of the Discretised system are show below.



Translating the Z-Transformed system into the W-domain requires the modified use of the Bilinear Transform as it closely maches the conditions for the shift from the Z to W domains.

This then produces a W-domain representation of the Uncompensated system:

```
Gw =

  4.033e-07 s^3 - 0.0003602 s^2 - 0.93 s + 822
  --------------------------------------------
          s^3 + 14 s^2 + 63 s + 1.951e-08
```

Where its appearance is simmilar to the the the original S-domain Transfer function especially with the characteristic polynomial but with aditional zeros.

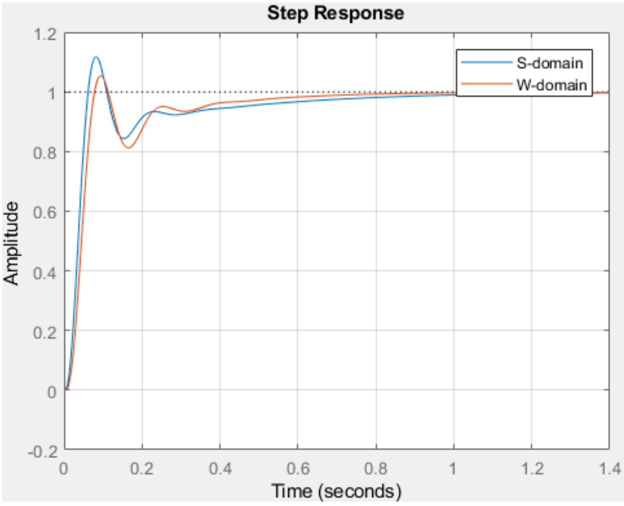
W-Domain Compensator Design

```
%% W-DOMAIN LEAD COMPENSATOR POLE AND ZERO CALCULATIONS
% At Rwcw of 12.2704, Phase angle = 153
% additional Phase angle is:
AdPaw = CPm + 153 - 180;

CPmlw = deg2rad(AdPaw+5); % added 5 degrees
% sqrt(Z*P) = Rwc
alphaw = ((1 + sin(CPmlw))/(1 - sin(CPmlw)));

zw = Rwc/sqrt(alphaw);
pw = alphaw*zw;

% lead compensator is
Dw = tf([1/zw 1],[1/pw 1]);
```

```
Dw =

  0.1258 s + 1
  -------------
  0.02346 s + 1

Continuous-time transfer function.
```

Due to the similarities between the S-domain and W-domain the compensator can be designed in the same manner. Through experimentation I found a **300 milliseconds** settling time to be a good target for the W-domain design.

W-Domain Response Plots



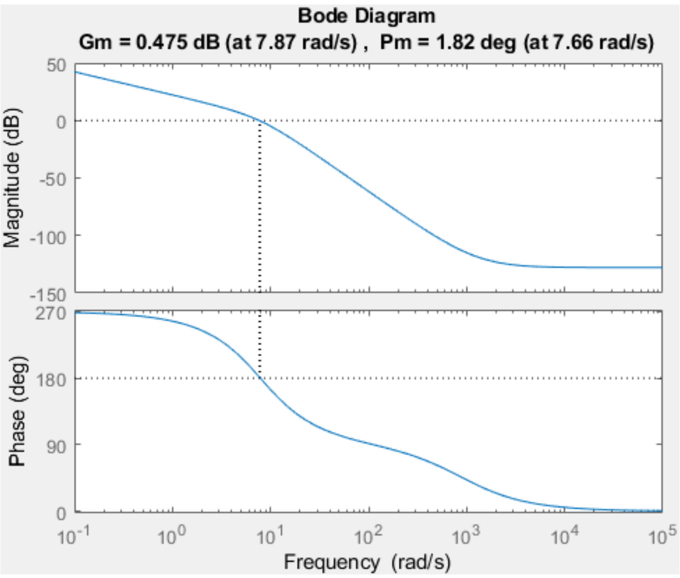

```
stepdataC =

  struct with fields:

        RiseTime: 0.046562346797533
    SettlingTime: 0.358068526324956
     SettlingMin: 0.812500879931765
     SettlingMax: 1.054001258000135
       Overshoot: 5.400125802514921
      Undershoot: 0.015519436660863
            Peak: 1.054001258000135
        PeakTime: 0.093531612630757
```

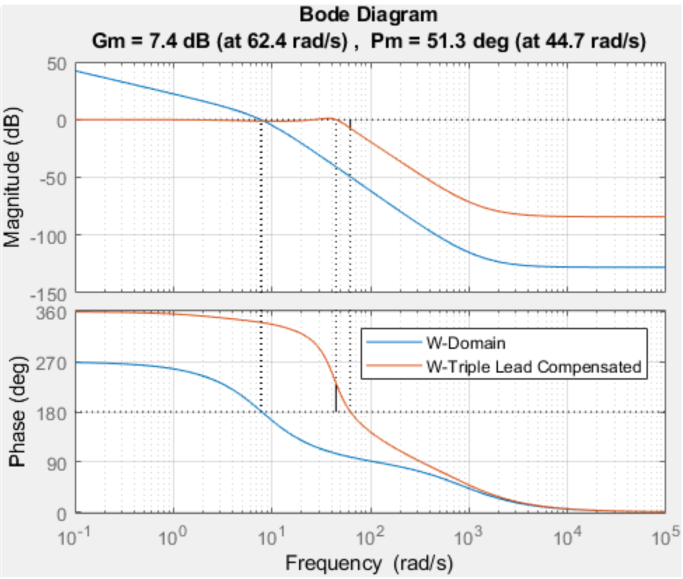
The plot shows the responses to a step of the S-domain double Lead compensated system vs triple Lead compensated system. A triple Lead is used here as extra attentiation and loss of informationduring conversion needed to be legislated for.

Requirements	S-domain double Lead	W -domain triple Lead
Percentage Overshoot (%)	11.77	5.4
Rise Time (s)	0.038	0.0466
Settling Time (s)	0.448	0.358

It seems that I have managed to achieve better performance in the W-domain in 2 out of the 3 requirement specifications with overshoot being half that of the S-domain plot and Settling time being 100milliseconds faster. This can be attributed to the extra lead present.

W-Domain Bode and Margins





This Bode diagrams compares the Uncompensated W- domain system to that compensated with a triple Lead compensator This shows an increase in the Phase Margin of the system from its original 1.82 degrees to 51.3 degrees.

Z-Domain Compensator Implementation

The W-domain compensator is translated by into the Z domain via:

$$\boxed{w = \frac{2}{T}\left[\frac{z-1}{z+1}\right]}$$

However, I seem to be having issues with implementing this in MATLAB and as a result I failed to generate a Z-domain representation of the compensator. However, due to the fact that the compensator works in the W domain, according to the literature, it should work as well in the Z domain with only minor attenuation.

Conclusion

The controller design was successful as the system has been recovered from instability and all the time domain design specifications have been met by the W domain compensators.