|  |
| --- |
| ACS214 |
| **Computer Aided Control Systems Analysis and Design Using MATLAB (section 4)** |
| Registration Number: 140154045 |

# Introduction

Figure 1

Y(s)

X(s)

­­\_\_

+

Figure 1

The aim of the task is to digitise the system shown in figure 1. Digitising a system will change is from the analogue continuous s-domain to the discrete digital z-domain. Samplers are crucial in achieving digitising a system as we need to take values at regular intervals. Therefore it is also crucial that we choose correct settling times to avoid aliasing which is displayed in figure 2

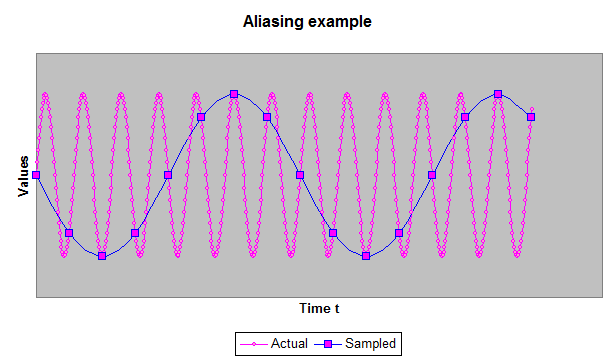


Figure 2

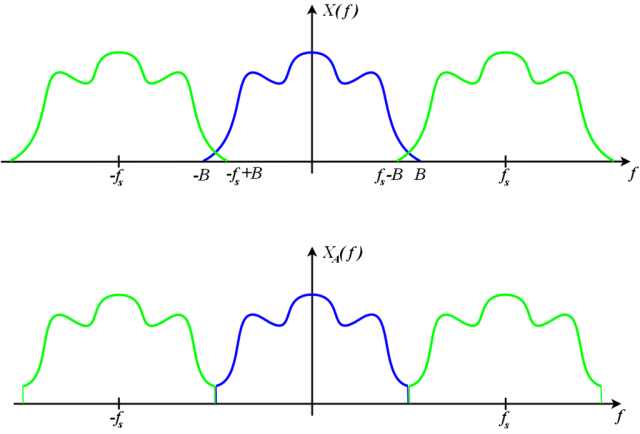
Figure 2 clearly shows how important the sampling time is as it can cause the digital output becoming entirely different from the original analogue signal.

Figure 3

Moreover picking a sampling time that is too small can cause overlapping which causes interference. This is the phenomenon also known as folding back shown in Figure 3. This interference causes loss of information as is undesirable in the design.

Noise

To avoid this, the Shannon Nyquist theorem must be applied to avoid the loss of information. The Shannon Nyquist theorem is as follows:

To be able to reconstruct F( jw ) from F\*( jw ), the sampling frequency “fs “ must be chosen to be at least twice as high as the highest frequency pertaining to the signal F( jw )

In practice however the sampling frequency must be at approximately between 10 -20 x Bandwidth. This is what I will be using in my design to convert my current system into the discrete domain.

## Discrete Design of plant

In order to proceed with my design, it is vital that the sampling time is chosen carefully. In order to calculate the appropriate sampling frequency I must first calculate the bandwidth using the equation

Our desired settling time is 0.5 seconds and our damping ratio ( ) which is calculated from the previous analogue report is equal to 0.59 therefore the bandwidth can be calculated by substituting these values into the equation to give:

It is vital for a good response to use Shannon Nyquist theorem however in practicality I will use 20 times the bandwidth frequency to find the sampling frequency:

Therefore the sampling time can be attained using the following equation:

Now that I have obtained my sampling time I can now convert my original transfer function into the digital Z (Gz) domain using the following MATLAB command:

>> Num[684]

>> Den [ 1, 12, 57, 0 ]

>> [numz,denz] = c2dm( num , den , ts , ‘tustin’ )

This gives us the numerator and denominator of the Gz

>> Gz = tf( numz , denz , Tsamp )

This will give us the open loop transfer function in the z domain

>> Gzc = feedback( Gz,1 )

This will produce the closed loop digital transfer function in the Z domain

This method using MATLAB has allowed me to obtain the open loop transfer function in the Z domain of

Therefore the closed loop transfer function is

This can be represented in the following block diagram in figure 4 produced via Simulink

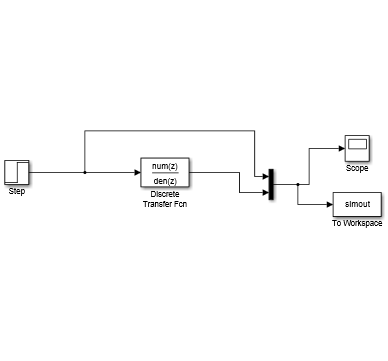
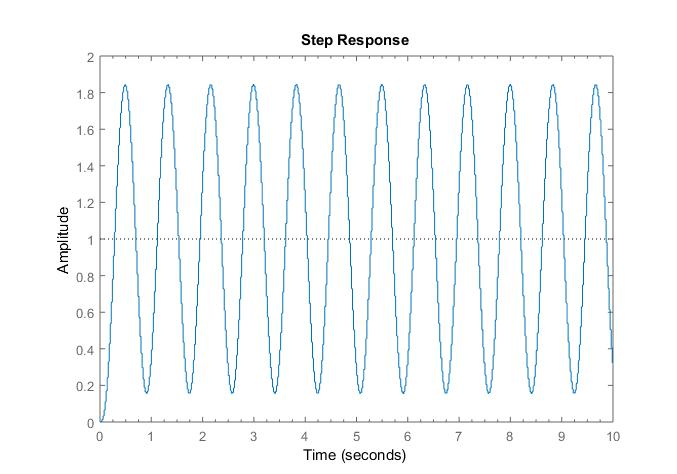


Figure 4

This produces the following step response:

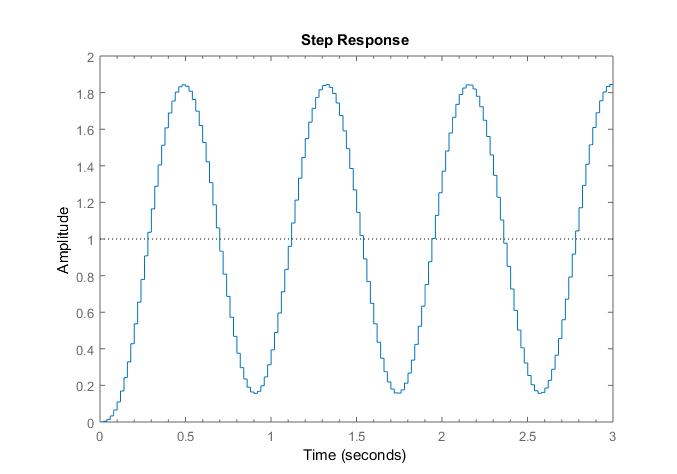


Figure 5: Discrete

Zooming into figure 5 shows clearly that the system is now discrete and that it is the discrete Z transform of the original continuous step response by comparison of Figure 5 and Figure 6.

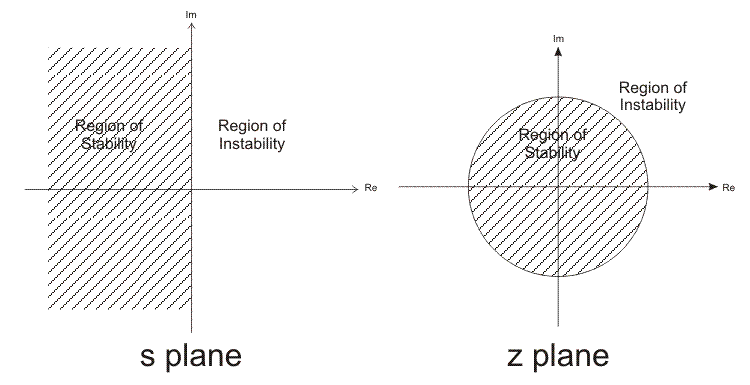
## Introducing the w domain

Figure 6:

Continuous

Now that the plant is discrete the oscillatory behaviour is not desired therefore I must design a compensator. Unfortunately it is difficult to design compensators in the Z domain as this can be represented in the s and z plane as the unstable and stable regions are represented differently in Figure 7.

Figure 7

Reference: http://learn.mikroe.com/ebooks/digitalfilterdesign/chapter/bilinear-transformation/

Therefore the same method cannot be used to design in the Z domain. This will be avoided by transforming the transfer function from the digital Z domain to the digital w domain as the following W plane is similar to the s plane and so the same methods can be utilised. This is achieved through the bilinear transformation method.

## Transforming into the W domain

Figure 8

To convert from the transfer function above from the Z domain to the W domain the first step to take is to replace z by –z in the equation therefore:

The bilinear transformation is represented when where T is the sampling time interval and

W is the w domain. When using the MATLAB command bilinear it will transform the transfer functions into a function of X as shown:

>> num=[-0.0006085 0.001825 -0.001825 0.0006085]

>> den = [ -1 -2.766 -2.553 -0.7867 ]

>> [ numx, denx ]= bilinear( num, den, 5 )

Using this command I have obtained the transfer function:

Such that:

Preceding with my value of sampling time to be 0.02001 I can substitute this into so that therefore

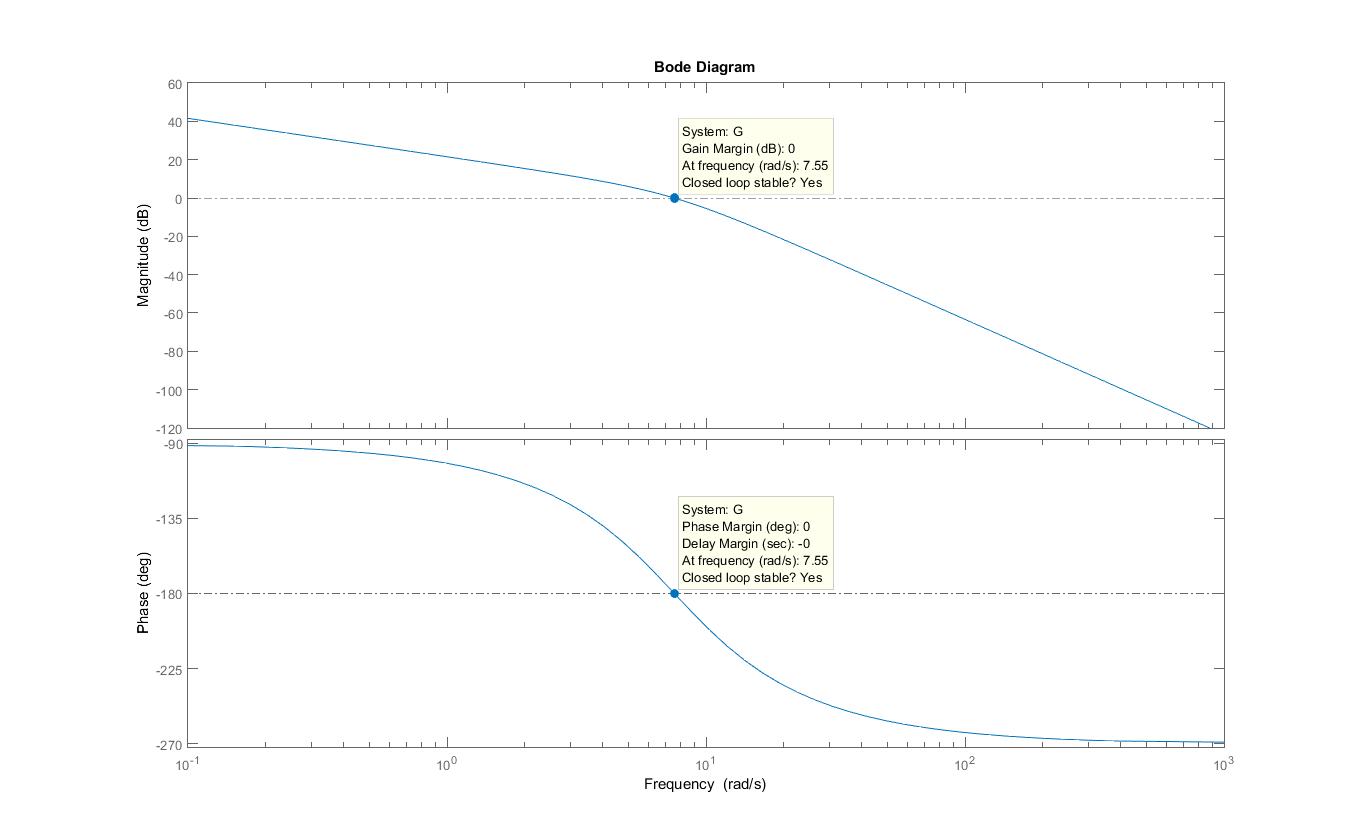
Now we have obtained the equation which relates X to the W domain, the transfer function in the W domain can be found by substitution. I will achieve this in the following MATLAB command:

>> numw = [ numx ].\*[ (-0.01)^3, (-0.01)^2, (-0.01), 1 ]

>> denw = [ denx ].\* [ (-0.01)^3, (-0.01)^2, (-0.01), 1 ]

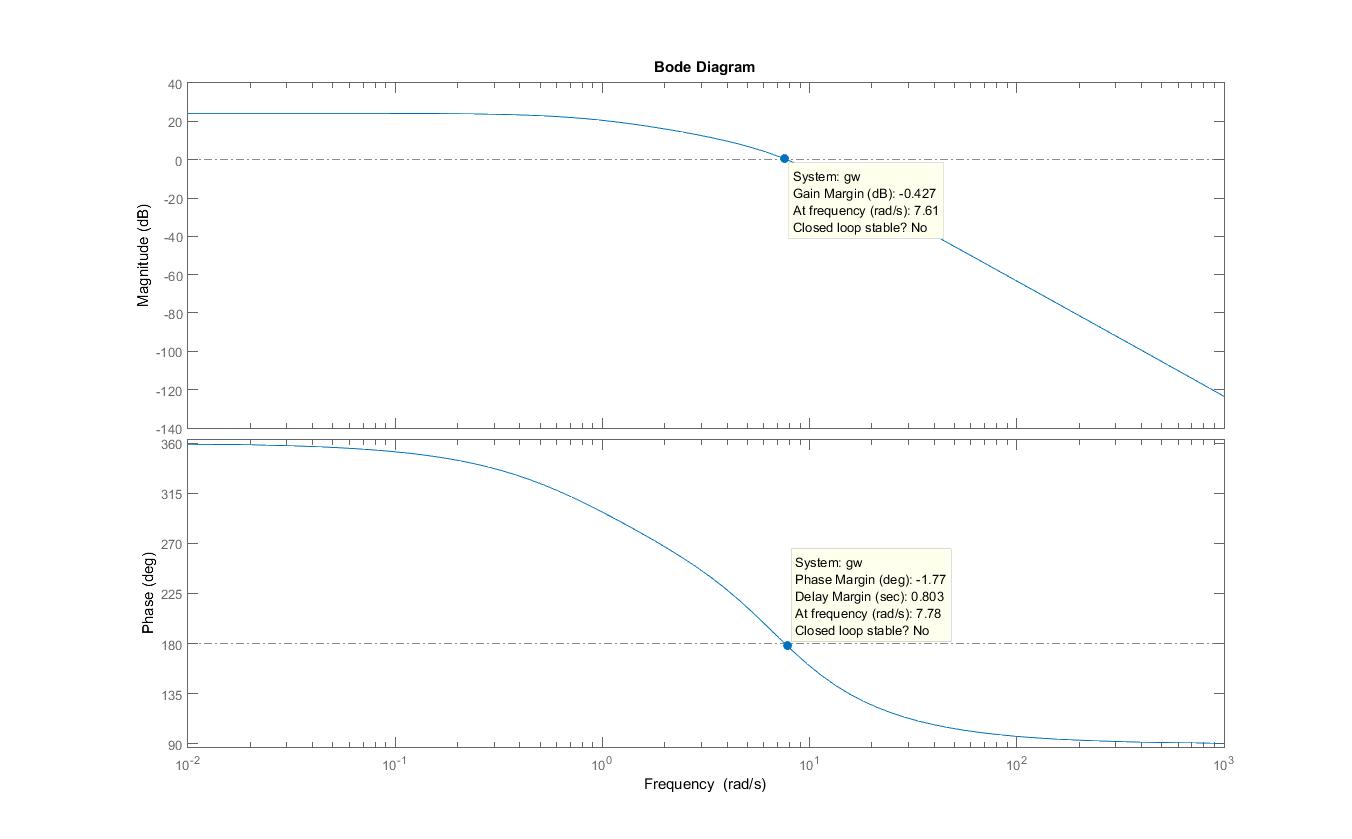
Having obtained my plant in the w domain i can now test the properties against the properties of the continuous plant in the s domain to compare how true to the real value it is. I will compare the bode plots for this as follows:

Bode response for G( S ) Figure 9



## 

## Bode response for G(w) Figure 10

  
These bode plots of the plant in the s domain and w domain has different properties. The summery of the properties are in the following table:

|  |  |  |
| --- | --- | --- |
| Properties | Bode plot figure 9 | Bode plot figure 10 |
| Phase cross over frequency | 7.55 | 7.78 |
| Phase margin | 0 | -1.77 |
| Gain cross over frequency | 7.55 | 7.61 |
| Gain margin | 0 | -0.427 |

From the summarised table in figure 11 it is clear that there are only small differences between the s domain performance and w domain performance, hence it can be concluded that the transfer function in the w domain has been transformed successfully.

## Design procedure of the digital compensator

The aim is to design a digital compensator so that the closed loop feedback response of the plant in parallel with the compensator to have properties of:

* Percentage overshoot <15%
* Rise-time < 100 msec =0.1 second
* Settling time <500 msec = 0.5 second
* Zero steady state error to a step

To achieve an percentage overshoot of 15% I will use the value of 10% as in the design of the compensator in the s domain to achieve a good response and increase reliability.

Therefore to calculate my phase margin required to add I will proceed with the following calculations:

The damping ratio will allow me to calculate the required phase margin for 10% overshoot using the following calculations:

Substituting I can get the phase margin:

From the bode diagram in figure 10 the digital G(w) has a phase margin of therefore I must add that phase on to my required phase