M270A Linear Dynamic Systems
Final Project

Samuel Chien and Sanskar Anil Nalkande

15 December 2023

Task 1: Empirical Frequency Response Estimates

Using '`cpsd`', an inbuilt MATLAB function, we calculated the auto and cross spectra for the corresponding channels. Below are the six spectra plotted in logarithmic scale.

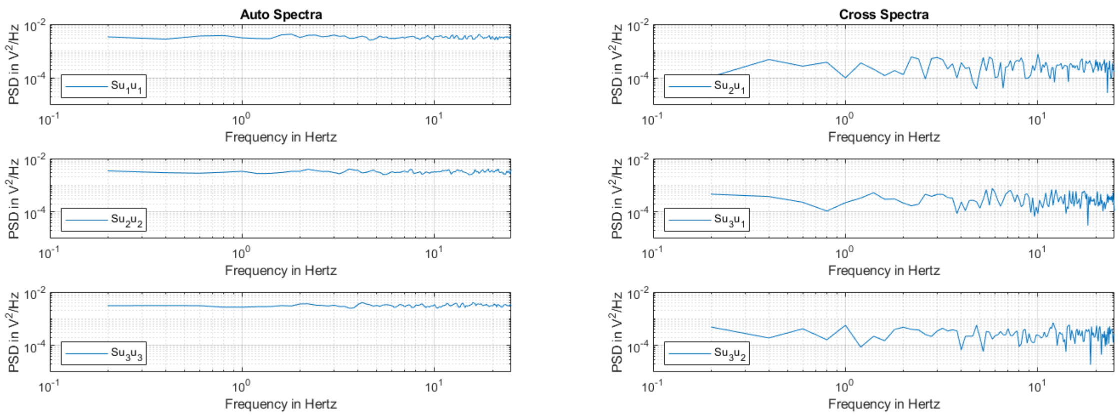

Figure 1: Auto and Cross Spectra for the input channels

Question 1

In the graphs above, we see that the Auto Spectra for each channels and more and less constant over the whole frequency range. Since every frequency is present in the input in the same constant amplitude, we can say that the input given to us are 'White Noise'.

Question 2

Once again referring to the graphs, we see that the magnitudes of the auto spectra are much higher than that of the cross spectra. In fact, let's look at the average magnitudes for all these spectra and compare them.

Auto Spectra	Average Magnitude	Cross Spectra	Average Magnitude
$S_{u_1u_1}$	0.003280	$S_{u_2u_1}$	0.000285
$S_{u_2u_2}$	0.003187	$S_{u_3u_2}$	0.000284
$S_{u_3u_3}$	0.003170	$S_{u_3u_1}$	0.000309

In the table above, we can clearly see that the cross spectra are, on average, a magnitude smaller than the auto-spectra. This shows that there is little to no correlation between the input signals.

Question 3

Using the inbuilt '`var`' function we calculated the variances of the input signals. Since we have already calculated the mean values of the auto spectra, we can just multiply them with the sampling frequency = 50 Hz. The comparison is given in the table below. We can see that these results are very close to each other.

Input Channel	Variance of the signal	Auto Spectra	Average Magnitude $\times$ Sampling frequency
u_1	0.1636	$S_{u_1u_1}$	0.1640
u_2	0.1594	$S_{u_2u_2}$	0.1593
u_3	0.1582	$S_{u_3u_3}$	0.1585

Question 4

To calculate the frequency response of the individual SISO systems, we used the following formula:

$$H_{ij} = \frac{S_{u_i u_j}}{S_{u_j u_j}} \tag{1}$$

Then using the inbuilt '`abs`' and '`angle`' functions, we plotted the frequency response magnitudes and phase plots for each input channels.

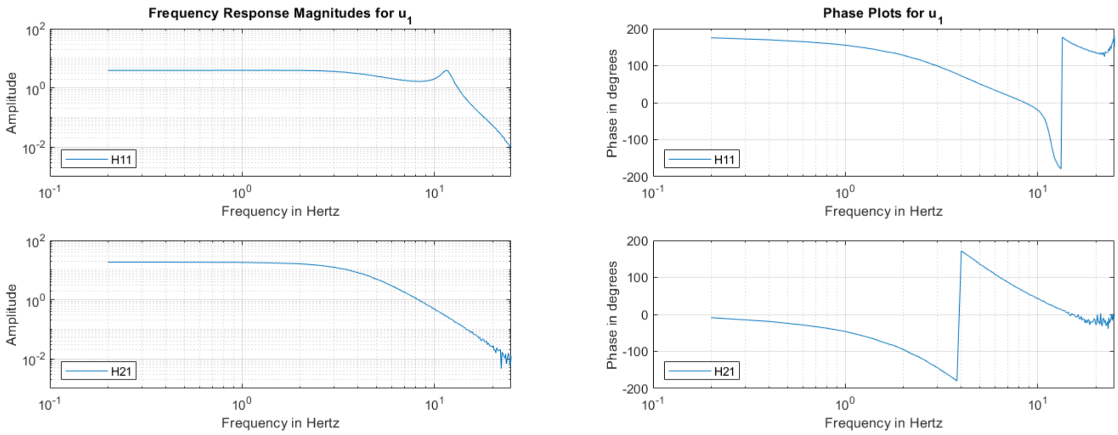

Figure 2: Frequency Response Magnitude and Phase Plots for u_1

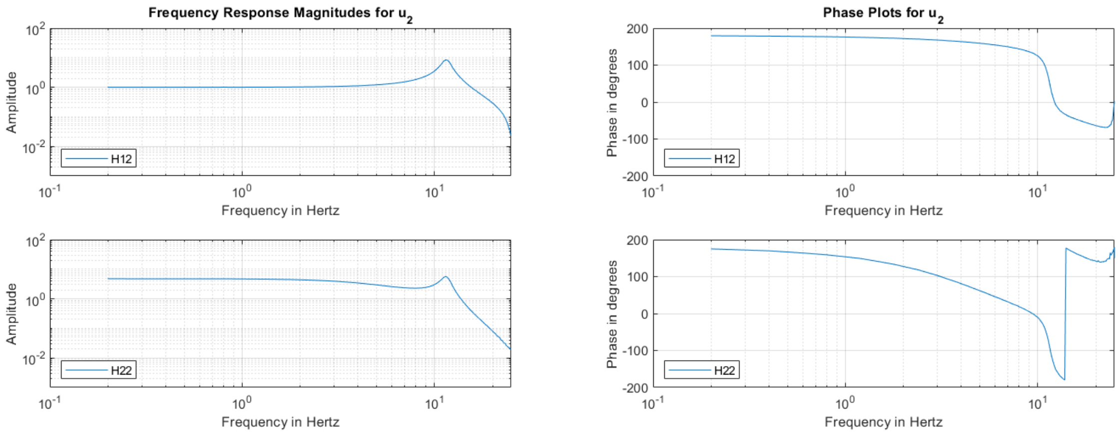

Figure 3: Frequency Response Magnitude and Phase Plots for u_2

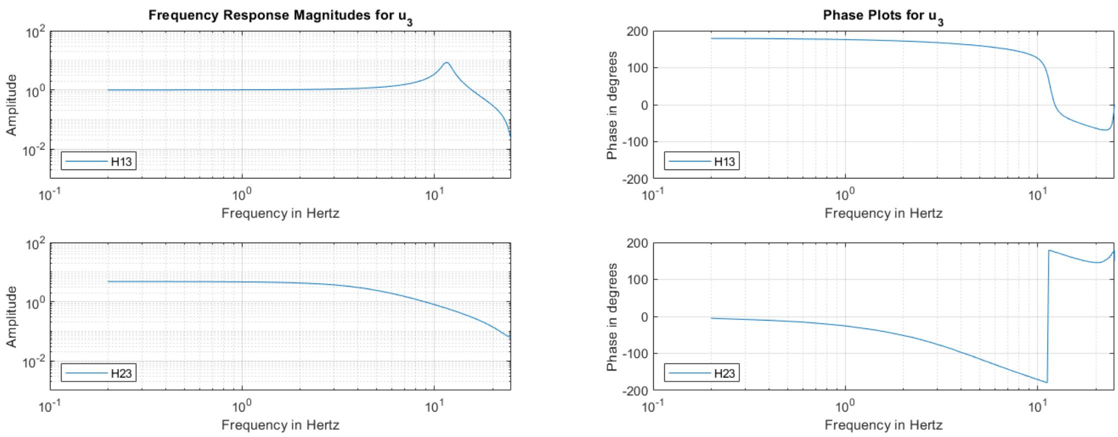

Figure 4: Frequency Response Magnitude and Phase Plots for u_3

Task 2: Pulse Response Estimates

The '`ifft`' function in MATLAB allows us to perform an inverse fast Fourier transform on the frequency response data we got from Task 1. After doing this, we get the pulse response estimates. The inverse fast Fourier transform for a discrete data set works on the following formula:

$$x(t) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j(\frac{2\pi}{N})tk} \tag{2}$$

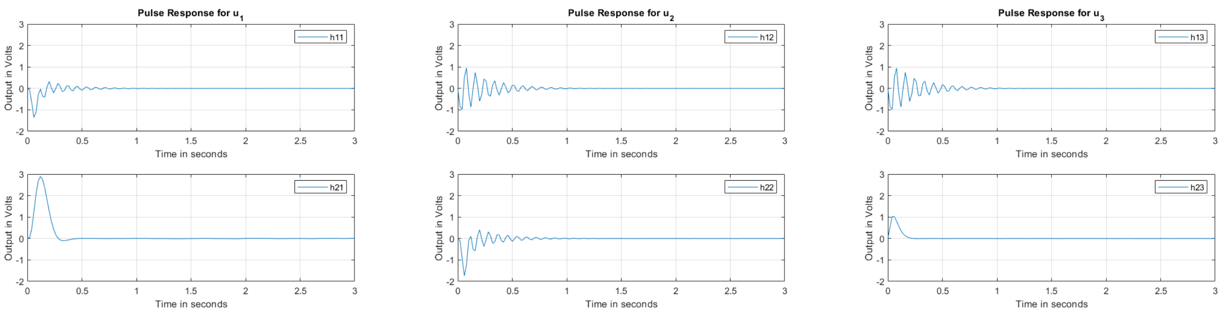

Figure 5: Impulse Response for $n_s = 7$

Task 3: Hankel matrix analysis and parametric model

Question 1

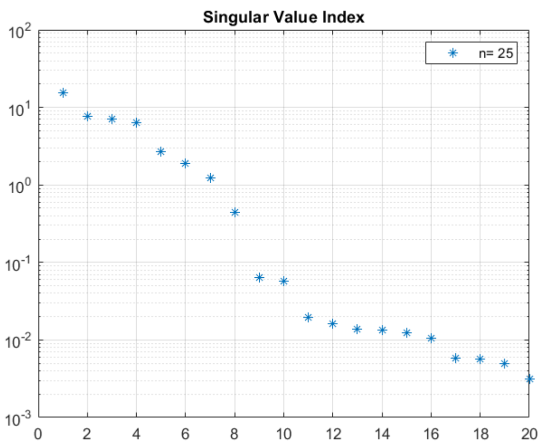

Figure 6: Singular Value Index for M_{25}

In the table below, we see that none of the eigenvalues for $n_s = 7$, 8 and 10 have a magnitude greater than one. Hence these systems are asymptotically stable. On the other hand, for $n_s = 16$, has at least one eigenvalue whose magnitude exceeds one. Hence this system is asymptotically unstable.

	$n_s = 7$	$n_s = 8$	$n_s = 10$	$n_s = 16$
`max(abs(eig(A)))`	0.9244	0.9249	0.9249	1.0317

Question 2

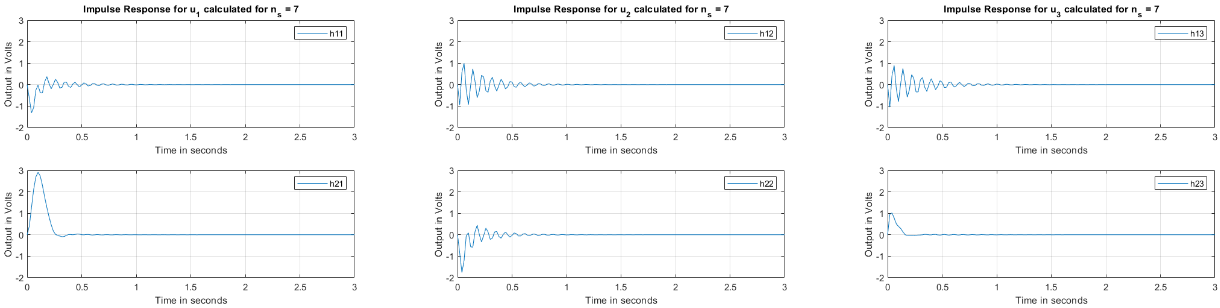

Figure 7: Impulse Response for $n_s = 7$

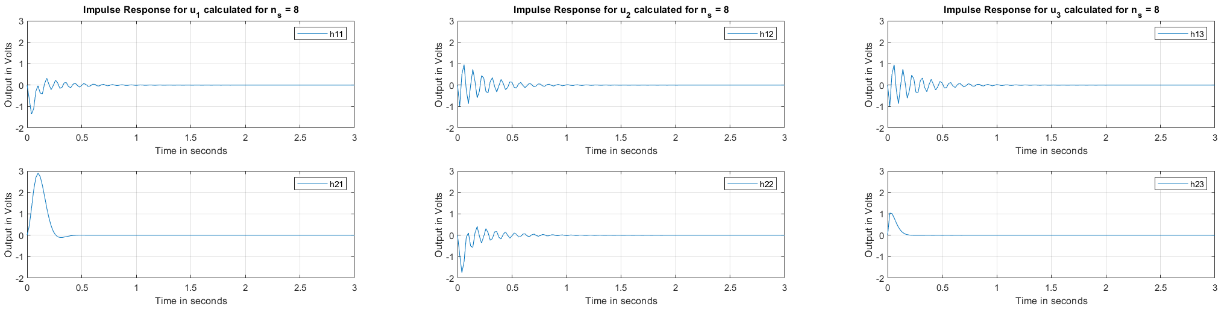

Figure 8: Impulse Response for $n_s = 8$

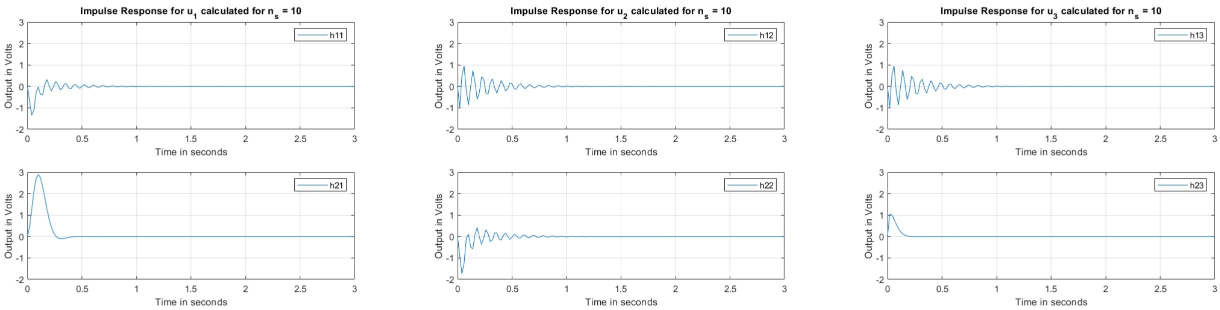

Figure 9: Impulse Response for $n_s = 10$

Although it is not clearly visible in the pulse response, we can see that the impulse response for $n_s = 7$ differs slightly from the empirical data given. This difference will be more visible in the frequency response. Hence this model cannot be used as a valid model for the system. Whereas the impulse response for $n_s = 8$ and 10 are essentially indistinguishable. Hence these models may be used as a valid model for the system.

Question 3

Frequency Response for $n_s = 7$

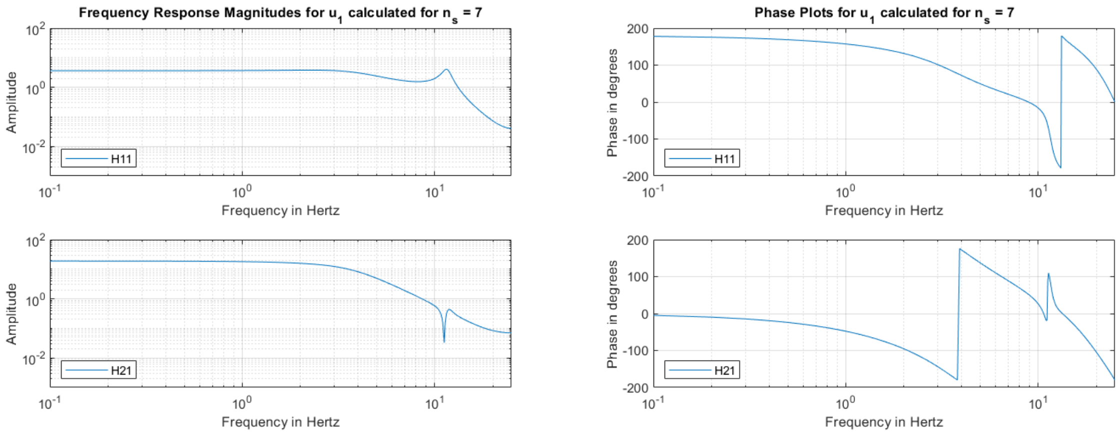

Figure 10: Frequency Response Magnitude and Phase Plots for u_1 calculated for $n_s = 7$

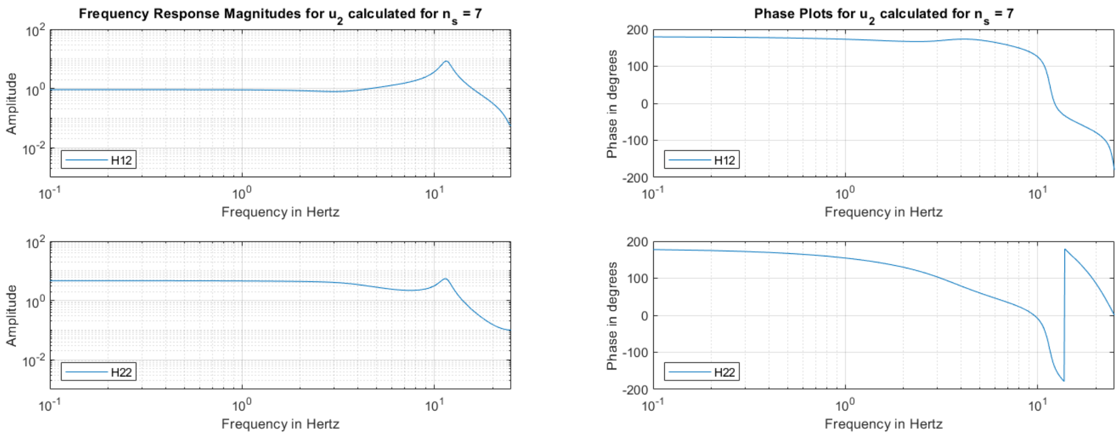

Figure 11: Frequency Response Magnitude and Phase Plots for u_2 calculated for $n_s = 7$

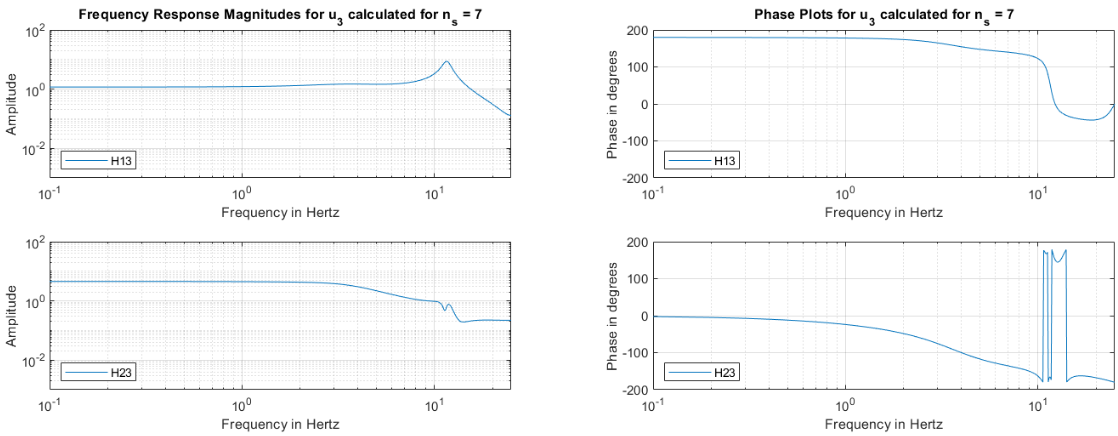

Figure 12: Frequency Response Magnitude and Phase Plots for u_3 calculated for $n_s = 7$

Frequency Response for $n_s = 8$

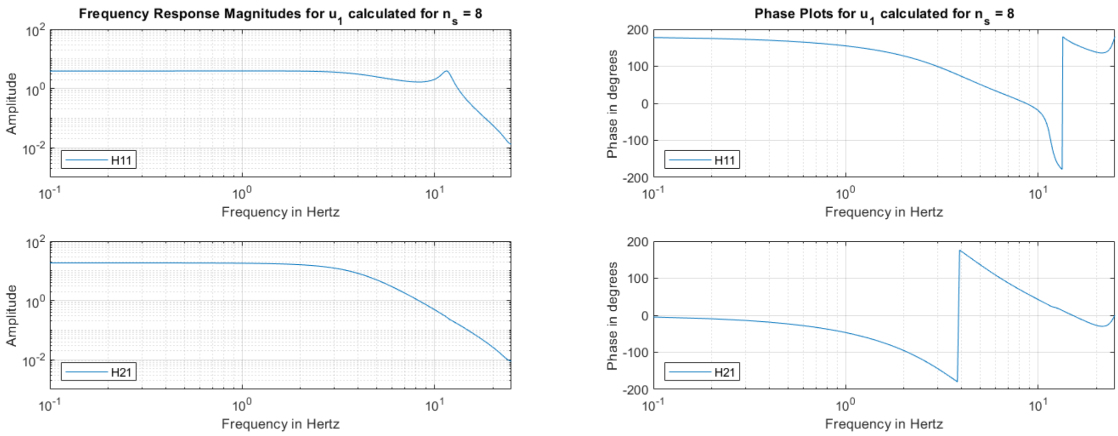

Figure 13: Frequency Response Magnitude and Phase Plots for u_1 calculated for $n_s = 8$

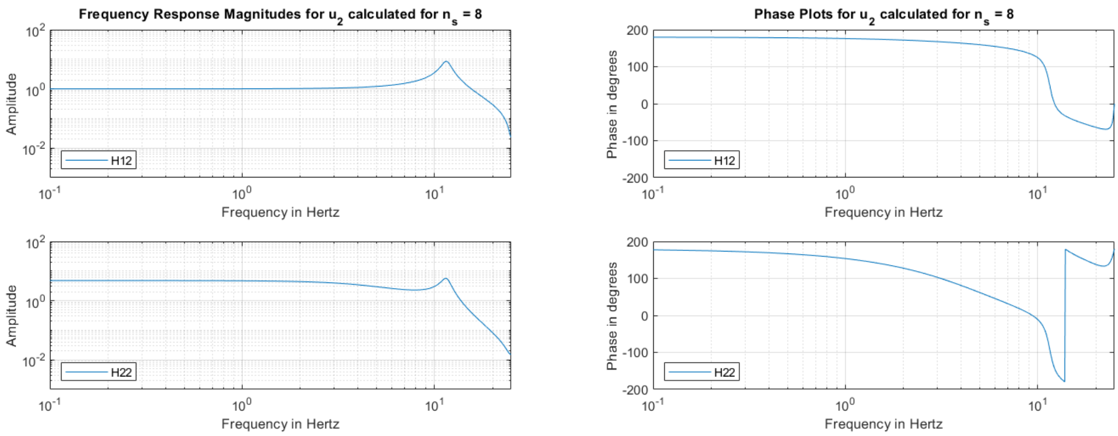

Figure 14: Frequency Response Magnitude and Phase Plots for u_2 calculated for $n_s = 8$

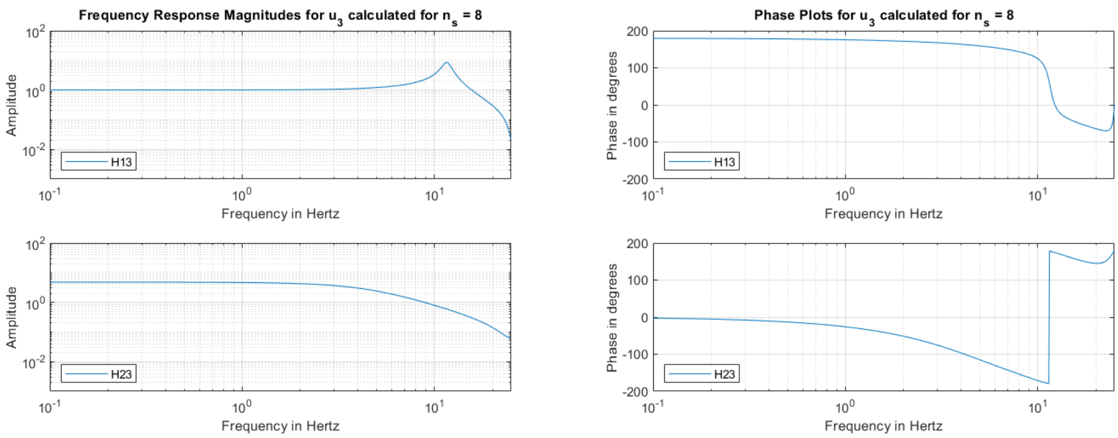

Figure 15: Frequency Response Magnitude and Phase Plots for u_3 calculated for $n_s = 8$

Frequency Response for $n_s = 10$

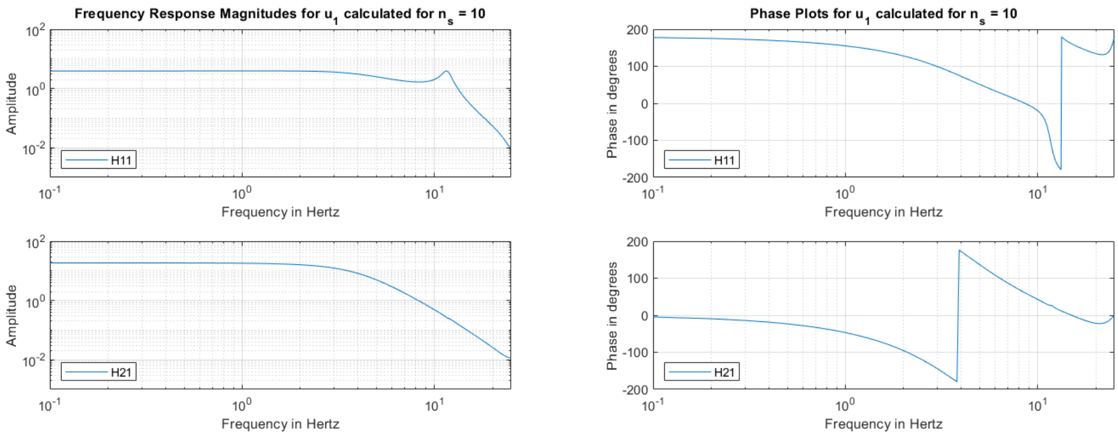

Figure 16: Frequency Response Magnitude and Phase Plots for u_1 calculated for $n_s = 10$

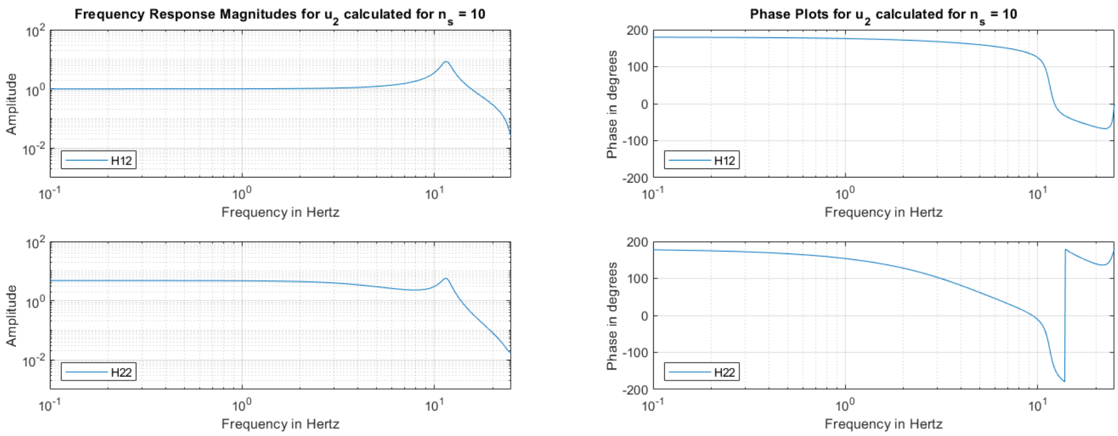

Figure 17: Frequency Response Magnitude and Phase Plots for u_2 calculated for $n_s = 10$

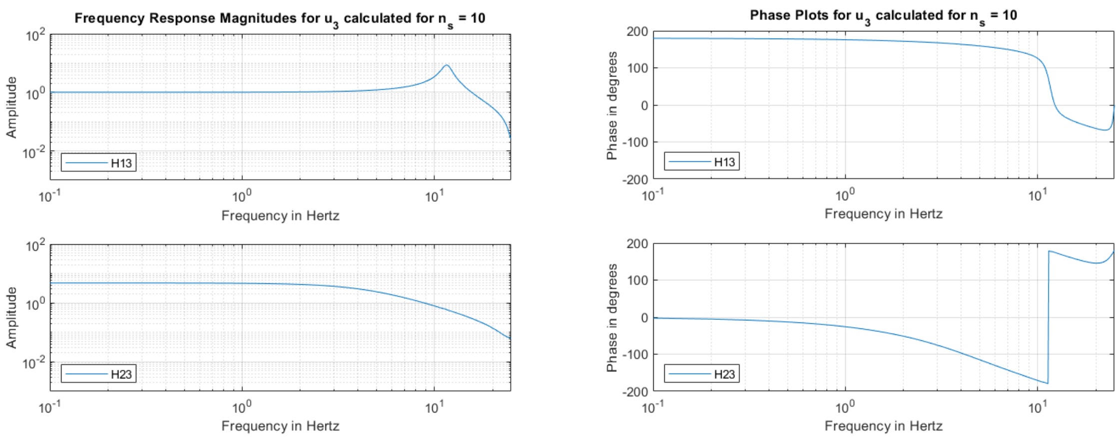

Figure 18: Frequency Response Magnitude and Phase Plots for u_3 calculated for $n_s = 10$

From the frequency response perspective, the $n_s = 7$ model is again the worst performing of all. We can see a visible difference from the empirical data. The frequency response for $n_s = 8$ and 10 yield almost indistinguishable frequency responses compared to the empirical data.

Task 4: Transmission zeros of individual I/O channels

Question 1

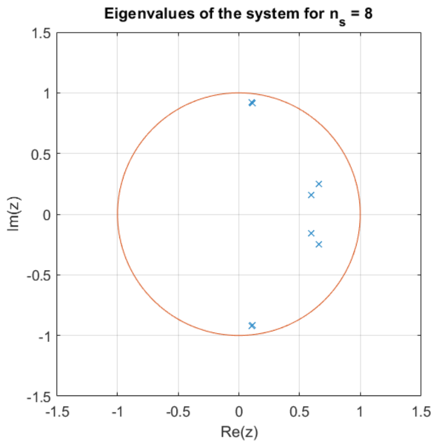

Figure 19: Eigenvalues of the $n_s = 8$ model in the complex plane

Question 2

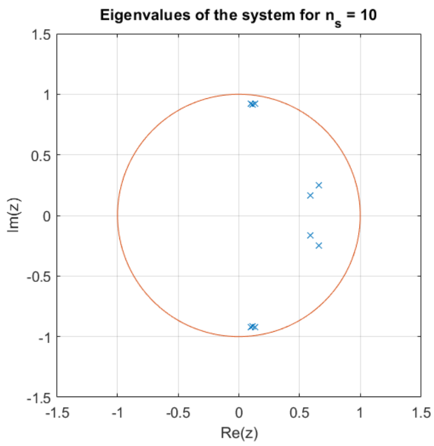

Figure 20: Eigenvalues of the $n_s = 10$ model in the complex plane

In the $n_s = 10$ model, there are two extra eigenvalues present from the $n_s = 8$ model. And we can see that this is indeed the case from the eigenvalue plots above. We see that the two extra eigenvalues in $n_s = 10$ model occur very near the two eigenvalues in the $n_s = 8$ model. We can deduce further if the model is over-parameterized from the transmission zero plots below.

Question 3

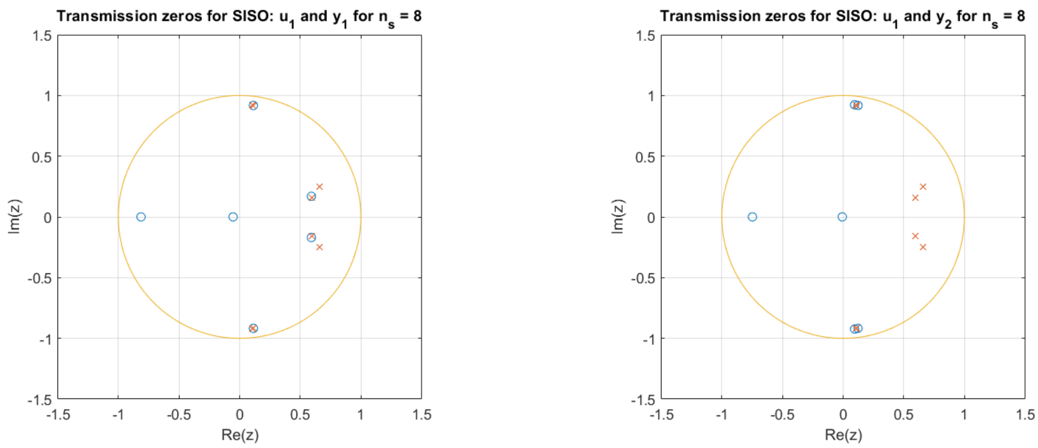

Figure 21: Transmission zeros of the $n_s = 8$ for u_1

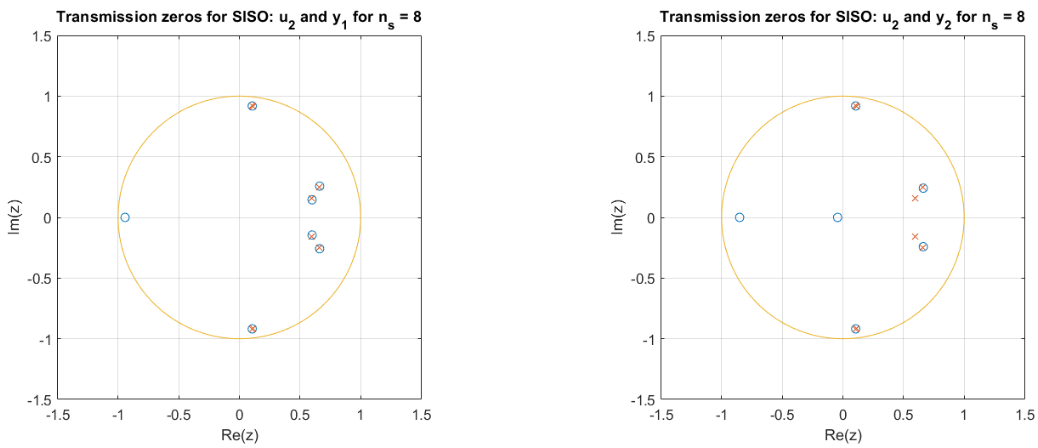

Figure 22: Transmission zeros of the $n_s = 8$ for u_2

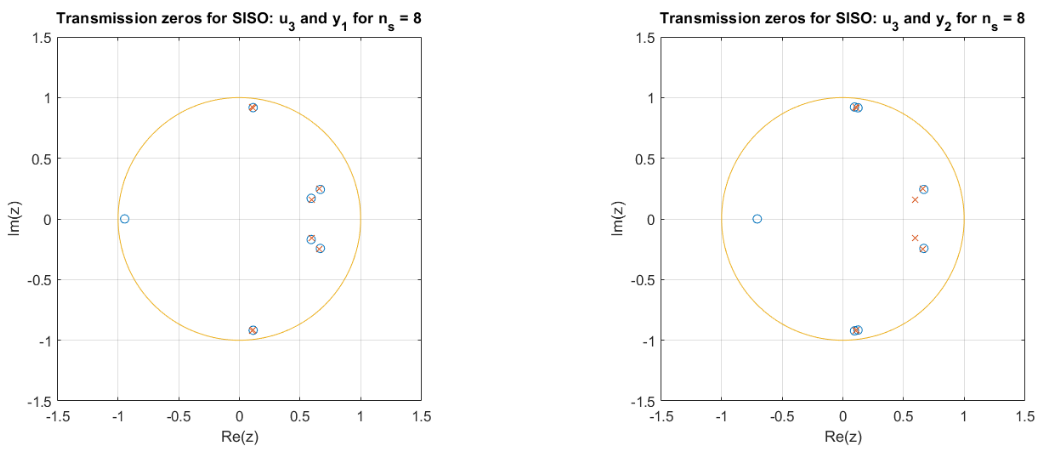

Figure 23: Transmission zeros of the $n_s = 8$ for u_3

Question 4

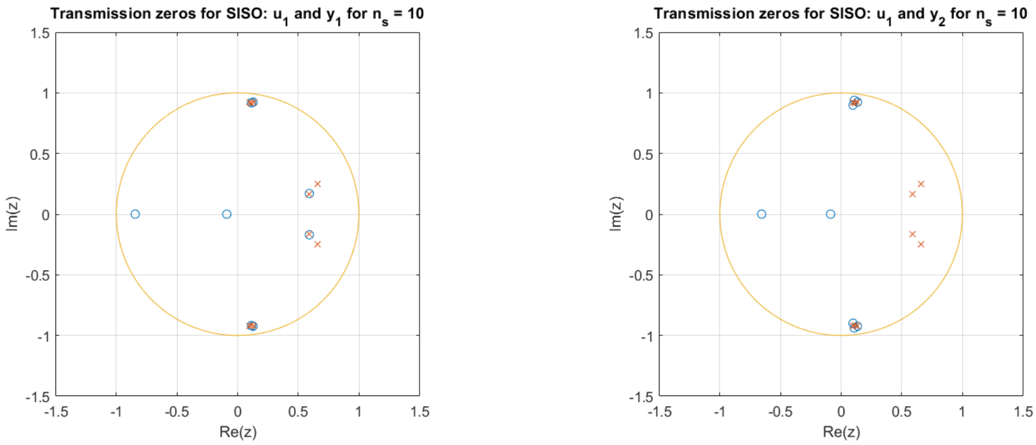

Figure 24: Transmission zeros of the $n_s = 10$ for u_1

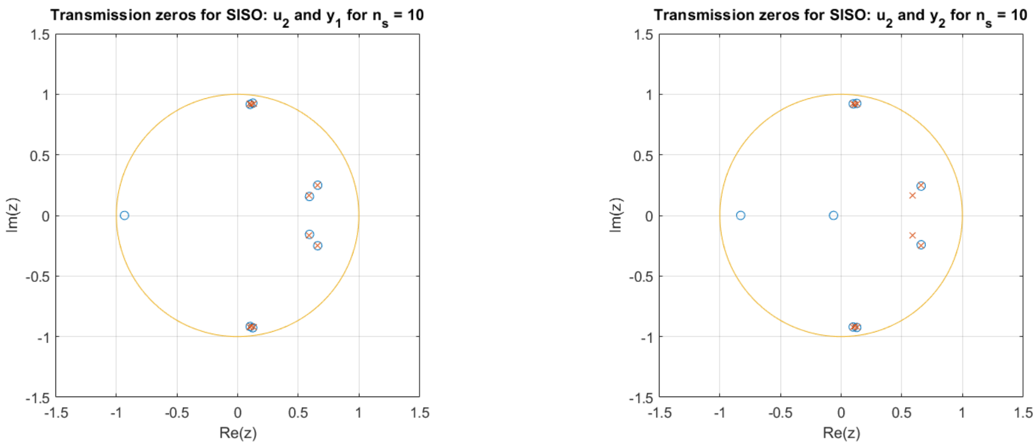

Figure 25: Transmission zeros of the $n_s = 10$ for u_2

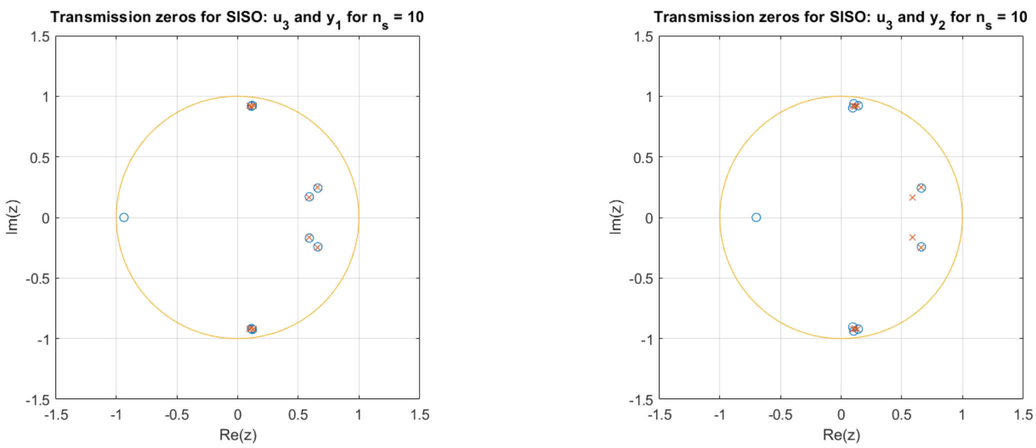

Figure 26: Transmission zeros of the $n_s = 10$ for u_3

In the $n_s = 10$ model, there should be two extra eigenvalues present from the $n_s = 8$ model. From the above plots, we can see that the added eigenvalues are always accompanied by zeros in close proximity and this occurs for all six channels. hence we can say that the $n_s = 10$ model is over-parameterized and the extra two eigenvalues do not have any impact on the input-output properties of the system.

Task 5: Block diagram derived from analysis of SISO channels

Below is a table showing the eigenvalues of the discrete-time model and their corresponding continuous time equivalents.

λ_d	λ_c
$0.6621 + 0.2502j$	$-17.2769 + 18.0636j$
$0.6621 - 0.2502j$	$-17.2769 - 18.0636j$
$0.5959 + 0.1565j$	$-24.2147 + 12.8392j$
$0.5959 - 0.1565j$	$-24.2147 - 12.8392j$
$0.1140 + 0.9174j$	$-3.9268 + 72.3605j$
$0.1140 - 0.9174j$	$-3.9268 - 72.3605j$
$0.1058 + 0.9188j$	$-3.9051 + 72.8084j$
$0.1058 - 0.9188j$	$-3.9051 - 72.8084j$

Resonators

In the table above there are two conjugate pairs for λ_c that have a large imaginary parts. These two pairs represent the lightly damped resonators since their real parts are small compared to the other eigenvalues and also the imaginary parts. These resonators are represented in the table below:

Resonator	Eigenvalues	Natural Frequency	Resonator	Eigenvalues	Natural Frequency
RES 1	$-3.9051 + 72.8084j$ $-3.9051 - 72.8084j$	72.8084 Hz	RES 2	$-3.9268 + 72.3605j$ $-3.9268 - 72.3605j$	72.3605 Hz

Low Pass Filters

In the table above there are two conjugate pairs for λ_c that have a large real parts. These two pairs represent the low pass filters since their real parts are large compared to the other eigenvalues and also the imaginary parts. These low pass filters are represented in the table below:

Low Pass Filter	Eigenvalues	Low Pass Filter	Eigenvalues
LP 1	$-24.2147 + 12.8392j$ $-24.2147 - 12.8392j$	LP 2	$-17.2769 + 18.0636j$ $-17.2769 - 18.0636j$

Constructing Block Diagram

By analysing the transmission zero plots with the eigenvalues of the system, we can figure out which subsystems are present in the path from an input to an output. This analysis is presented in a table format below:

Path	Subsystems Present
u_1 to y_1	RES 1 and LP 2
u_1 to y_2	LP1 1 and LP 2
u_2 to y_1	RES 2
u_2 to y_2	RES 2 and LP 1
u_3 to y_1	RES 1
u_3 to y_2	LP 1

We are given:
Low-frequency asymptote of RES 1 = 1
Low-frequency asymptote of RES 2 = 1

This given data can be verified by looking at the frequency response magnitudes for H_{12} and H_{13}. Since the paths u_1 to y_2 and u_3 to y_1 only contain RES 2 and RES 1 respectively, their frequency response magnitudes should also have low-frequency asymptote equal to 1. And this is indeed the case.

Now looking at the frequency response magnitude for H_{23}, it has a low frequency asymptote approximately equal to 5. And since the path u_3 to y_2 only contains LP 1:

Low-frequency asymptote of LP 1 = 5.

Now we analyse the frequency response magnitude for H_{22}. It has a low frequency asymptote approximately equal to 5. And since the path u_2 to y_2 contains RES 2 and LP 1, the low frequency asymptotes are multiplied together for these subsystems. This implies:

RES 2 and LP 1 are in series between u_2 to y_2.

Now we analyse the frequency response magnitude for H_{11}. It has a low frequency asymptote approximately equal to 4. And the path u_2 to y_2 contains RES 2 and LP 1. Also looking at the frequency response magnitude for H_{21}. It has a low frequency asymptote approximately equal to 20. And the path u_2 to y_2 contains RES 2 and LP 1. Looking at the two requirements above, the only conditions that make these possible are:

Low-frequency asymptote of LP 2 = 4.

RES 1 and LP 2 are in series between u_1 to y_1. Since the low-frequency asymptote for this is 4, signifying the multiplication of low frequency asymptotes for RES 1 and LP 2.

LP 1 and LP 2 are in series between u_1 to y_2. Since the low-frequency asymptote for this is 20, signifying the multiplication of low frequency asymptotes for LP 1 and LP 2.

Finally we have the following information:
RES 1 and LP 2 are in series between u_1 to y_1.
LP 1 and LP 2 are in series between u_1 to y_2.
RES 2 is the only subsystem between u_2 to y_1.
RES 2 and LP 1 are in series between u_2 to y_2.
RES 1 is the only subsystem between u_3 to y_1.
LP 1 is the only subsystem between u_3 to y_2.

Using the above info, we can draw the following block diagram:

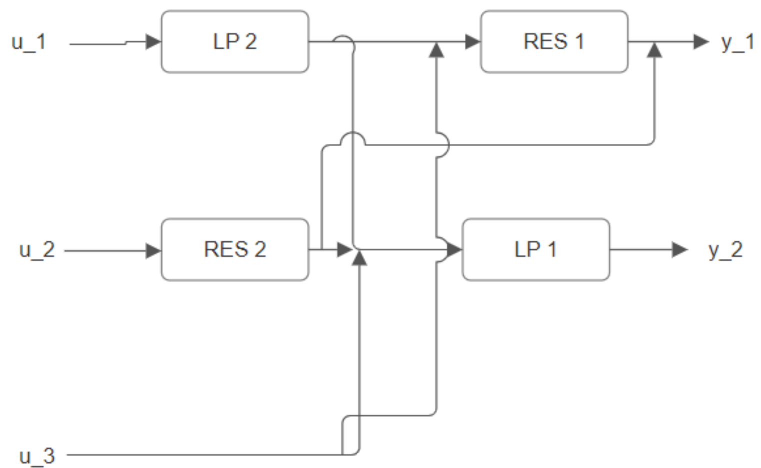

Figure 27: Block Diagram of the system

Appendix

MATLAB Code for the Project

```
ts=1/50;                    % sampling time in seconds
fs = 1/ts;                  % sample rate in hertz
nfft = 250;                 % sub-record length nfft/fs = 5 seconds
win = hamming(nfft);        % use Hamming data tapering window
fnyq = fs/2;                % Nyquist frequency
t = [0:nfft-1]*ts;          % Time array

% load response data
load random_u1.mat
y11 = y1;
y21 = y2;
u11 = u1;
load random_u2.mat
y12 = y1;
y22 = y2;
u22 = u2;
load random_u3.mat
y13 = y1;
y23 = y2;
u33 = u3;

%% Task 1

[Su1u1,f] = cpsd(u11,u11,win,[],nfft,fs,'twosided');    % auto-spectral density of u1
[Su2u2,~] = cpsd(u22,u22,win,[],nfft,fs,'twosided');    % auto-spectral density of u2
[Su3u3,~] = cpsd(u33,u33,win,[],nfft,fs,'twosided');    % auto-spectral density of u3

[Su2u1,~] = cpsd(u22,u11,win,[],nfft,fs,'twosided');    % cross-spectral density of y and u
[Su3u1,~] = cpsd(u33,u11,win,[],nfft,fs,'twosided');    % cross-spectral density of y and u
[Su3u2,~] = cpsd(u33,u22,win,[],nfft,fs,'twosided');    % cross-spectral density of y and u

% Task 1 Q1
% Plotting the Auto Spectra
figure(1)
subplot(311)
loglog(f,Su1u1)
title('Auto Spectra')
grid on; axis([0.1 fnyq 1e-5 1e-2]); legend('Su_1u_1'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('PSD in V^2/Hz');
subplot(312)
loglog(f,Su2u2)
grid on; axis([0.1 fnyq 1e-5 1e-2]); legend('Su_2u_2'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('PSD in V^2/Hz');
subplot(313)
loglog(f,Su3u3)
grid on; axis([0.1 fnyq 1e-5 1e-2]); legend('Su_3u_3'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('PSD in V^2/Hz');

% Plotting the Cross Spectra
figure(2)
subplot(311)
loglog(f,abs(Su2u1))
title('Cross Spectra')
grid on; axis([0.1 fnyq 1e-5 1e-2]); legend('Su_2u_1'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('PSD in V^2/Hz');
subplot(312)
loglog(f,abs(Su3u1))
```

```
grid on; axis([0.1 fnyq 1e-5 1e-2]); legend('Su_3u_1'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('PSD in V^2/Hz');
subplot(313)
loglog(f,abs(Su3u2))
grid on; axis([0.1 fnyq 1e-5 1e-2]); legend('Su_3u_2'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('PSD in V^2/Hz');

% Task 1 Q2
% Calculating average magnitudes of spectra
avg_mag_Su1u1 = mean(abs(Su1u1));
avg_mag_Su2u2 = mean(abs(Su2u2));
avg_mag_Su3u3 = mean(abs(Su3u3));
avg_mag_Su2u1 = mean(abs(Su2u1));
avg_mag_Su3u1 = mean(abs(Su3u1));
avg_mag_Su3u2 = mean(abs(Su3u2));

% Display average magnitudes
fprintf('Average Magnitude Su1u1: %f\n', avg_mag_Su1u1);
fprintf('Average Magnitude Su2u2: %f\n', avg_mag_Su2u2);
fprintf('Average Magnitude Su3u3: %f\n', avg_mag_Su3u3);
fprintf('Average Magnitude Su2u1: %f\n', avg_mag_Su2u1);
fprintf('Average Magnitude Su3u1: %f\n', avg_mag_Su3u1);
fprintf('Average Magnitude Su3u2: %f\n', avg_mag_Su3u2);

% Comparison
fprintf('\nComparison of Average Magnitudes:\n');
fprintf('Su2u1 is, on average, smaller than Su1u1: %s\n', ...
    string(avg_mag_Su2u1 < avg_mag_Su1u1));
fprintf('Su3u1 is, on average, smaller than Su1u1: %s\n', ...
    string(avg_mag_Su3u1 < avg_mag_Su1u1));
fprintf('Su3u2 is, on average, smaller than Su2u2: %s\n', ...
    string(avg_mag_Su3u2 < avg_mag_Su2u2));

% Task 1 Q3
Mu_1 = var(u11);        % Varience of the signal u1
Mu_2 = var(u22);        % Varience of the signal u2
Mu_3 = var(u33);        % Varience of the signal u3

MSu_1u_1 = mean(Su1u1)*50;      % Mean values of auto spectra multiplied by the sampling rate
MSu_2u_2 = mean(Su2u2)*50;      % Mean values of auto spectra multiplied by the sampling rate
MSu_3u_3 = mean(Su3u3)*50;      % Mean values of auto spectra multiplied by the sampling rate

% Task 1 Q4
[Sy1u1,~] = cpsd(y11,u11,win,[],nfft,fs,'twosided');    % cross-spectral density of y1 and u1
[Sy2u1,~] = cpsd(y21,u11,win,[],nfft,fs,'twosided');    % cross-spectral density of y2 and u1

[Sy1u2,~] = cpsd(y12,u22,win,[],nfft,fs,'twosided');    % cross-spectral density of y1 and u2
[Sy2u2,~] = cpsd(y22,u22,win,[],nfft,fs,'twosided');    % cross-spectral density of y2 and u2

[Sy1u3,~] = cpsd(y13,u33,win,[],nfft,fs,'twosided');    % cross-spectral density of y1 and u3
[Sy2u3,~] = cpsd(y23,u33,win,[],nfft,fs,'twosided');    % cross-spectral density of y2 and u3

% Calculating the frequency response
H11 = Sy1u1./Su1u1;
H21 = Sy2u1./Su1u1;
H12 = Sy1u2./Su2u2;
H22 = Sy2u2./Su2u2;
H13 = Sy1u3./Su3u3;
H23 = Sy2u3./Su3u3;

% Plotting the Frequency Response Magnitudes
```

```
figure(3)
subplot(211)
loglog(f,abs(H11))
title('Frequency Response Magnitudes for u_1')
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H11'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
subplot(212)
loglog(f,abs(H21))
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H21'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');

figure(4)
subplot(211)
loglog(f,abs(H12))
title('Frequency Response Magnitudes for u_2')
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H12'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
subplot(212)
loglog(f,abs(H22))
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H22'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');

figure(5)
subplot(211)
loglog(f,abs(H13))
title('Frequency Response Magnitudes for u_3')
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H13'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
subplot(212)
loglog(f,abs(H23))
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H23'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');

% Plotting the Phase Plots
figure(6)
subplot(211)
semilogx(f,rad2deg(angle(H11)))
title('Phase Plots for u_1')
grid on; axis([0.1 fnyq -200 200]); legend('H11'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');
subplot(212)
semilogx(f,rad2deg(angle(H21)))
grid on; axis([0.1 fnyq -200 200]); legend('H21'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');

figure(7)
subplot(211)
semilogx(f,rad2deg(angle(H12)))
title('Phase Plots for u_2')
grid on; axis([0.1 fnyq -200 200]); legend('H12'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');
subplot(212)
semilogx(f,rad2deg(angle(H22)))
grid on; axis([0.1 fnyq -200 200]); legend('H22'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');

figure(8)
subplot(211)
semilogx(f,rad2deg(angle(H13)))
title('Phase Plots for u_3')
grid on; axis([0.1 fnyq -200 200]); legend('H13'); legend('Location', 'southwest');
```

```
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');
subplot(212)
semilogx(f,rad2deg(angle(H23)))
grid on; axis([0.1 fnyq -200 200]); legend('H23'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');

%% Task 2

% Calculating the Pulse response
h11 = ifft(H11);
h21 = ifft(H21);
h12 = ifft(H12);
h22 = ifft(H22);
h13 = ifft(H13);
h23 = ifft(H23);

h = [h11 h12 h13; h21 h22 h23];

% Plotting the Pulse Response
figure(9)
subplot(211)
plot(t,h11)
title('Pulse Response for u_1')
grid on; axis([0 3 -2 3]); legend('h11'); xlabel('Time in seconds'); ylabel('Output in Volts');
subplot(212)
plot(t,h21)
grid on; axis([0 3 -2 3]); legend('h21'); xlabel('Time in seconds'); ylabel('Output in Volts');

figure(10)
subplot(211)
plot(t,h12)
title('Pulse Response for u_2')
grid on; axis([0 3 -2 3]); legend('h12'); xlabel('Time in seconds'); ylabel('Output in Volts');
subplot(212)
plot(t,h22)
grid on; axis([0 3 -2 3]); legend('h22'); xlabel('Time in seconds'); ylabel('Output in Volts');

figure(11)
subplot(211)
plot(t,h13)
title('Pulse Response for u_3')
grid on; axis([0 3 -2 3]); legend('h13'); xlabel('Time in seconds'); ylabel('Output in Volts');
subplot(212)
plot(t,h23)
grid on; axis([0 3 -2 3]); legend('h23'); xlabel('Time in seconds'); ylabel('Output in Volts');


%% Task 3

% Task 3 Q1
n = 25;
M_n = [];

% Constructing the Hankel Matrix
for i = 1:n
    tmp=[];
    for j = 1:n
        hk = [h11(j+i) h12(j+i) h13(j+i); h21(j+i) h22(j+i) h23(j+i)];
        tmp=[tmp hk];
    end
    M_n = [M_n;tmp];
```

```
end

% Plotting the Singular Values of M_n
figure(12)
semilogy(svd(M_n), '*')
title('Singular Value Index')
grid on; axis([0 20 1e-3 1e2]); legend(compose('n= %d',n));

% Singular Value Decomposition of M_n
[U,S,V] = svd(M_n);

% Reducing the State Dimensions
% Modify n_s below to get the plots for different n_s
n_s = 7;

S_n = S(1:n_s, 1:n_s);
U_1 = U(:,1:n_s);
V_1 = V(:,1:n_s);
M_w = U_1*S_n*V_1';
L = U_1;
R = S_n*V_1';

Mbar = [];
for i = 1:n
    tmp=[];
    for j = 1:n
        tmp=[tmp [h11(j+i+1) h12(j+i+1) h13(j+i+1); h21(j+i+1) h22(j+i+1) h23(j+i+1)]];
    end
    Mbar = [Mbar;tmp];
end

% Constructing A, B, C and D matrices
A = U_1.'*Mbar*V_1/S_n;
B = R(:,1:3);
C = L(1:2,:);
D = [0 0 0; 0 0 0];

% Checking if the system is asymptotically stable or not
% If the below value is greater than 1 the system is asymptotically
% unstable. And if it is less than 1 then it is asymptotically stable.
max(abs(eig(A)))

% Task 3 Q2
% Simulating the impulse response for reduced model
hI = zeros(2,3*3/ts+1);
hI11(1) = 0;
hI21(1) = 0;
hI12(1) = 0;
hI22(1) = 0;
hI13(1) = 0;
hI23(1) = 0;
for i = 2:3/ts+1
    hk = C*A^(i-1)*B;
    hI(1,3*i-2) = hk (1,1);
    hI(2,3*i-2) = hk(2,1);
    hI(1,3*i-1) = hk(1,2);
    hI(1,3*i-1) = hk(2,2);
    hI(1,3*i) = hk(1,3);
    hI(1,3*i) = hk(2,3);

    hI11(i) = hk(1,1);
```

```
    hI21(i) = hk(2,1);
    hI12(i) = hk(1,2);
    hI22(i) = hk(2,2);
    hI13(i) = hk(1,3);
    hI23(i) = hk(2,3);
end

% Plotting the impulse response for reduced models
figure(13)
subplot(211)
plot(0:ts:3,hI11)
title(['Impulse Response for u_1 calculated for n_s = ',num2str(n_s)])
grid on; axis([0 3 -2 3]); legend('h11'); xlabel('Time in seconds'); ylabel('Output in Volts');
subplot(212)
plot(0:ts:3,hI21)
grid on; axis([0 3 -2 3]); legend('h21'); xlabel('Time in seconds'); ylabel('Output in Volts');

figure(14)
subplot(211)
plot(0:ts:3,hI12)
title(['Impulse Response for u_2 calculated for n_s = ',num2str(n_s)])
grid on; axis([0 3 -2 3]); legend('h12'); xlabel('Time in seconds'); ylabel('Output in Volts');
subplot(212)
plot(0:ts:3,hI22)
grid on; axis([0 3 -2 3]); legend('h22'); xlabel('Time in seconds'); ylabel('Output in Volts');

figure(15)
subplot(211)
plot(0:ts:3,hI13)
title(['Impulse Response for u_3 calculated for n_s = ',num2str(n_s)])
grid on; axis([0 3 -2 3]); legend('h13'); xlabel('Time in seconds'); ylabel('Output in Volts');
subplot(212)
plot(0:ts:3,hI23)
grid on; axis([0 3 -2 3]); legend('h23'); xlabel('Time in seconds'); ylabel('Output in Volts');

% Task 3 Q3
w = [0.1:0.1:25];        %Frequency Matrix
HI11 = zeros(250,1);
HI21 = zeros(250,1);
HI12 = zeros(250,1);
HI22 = zeros(250,1);
HI13 = zeros(250,1);
HI23 = zeros(250,1);

% Calculating the frequency response for the reduced model
for i = 1:250
    H = C/(exp(sqrt(-1)*ts*w(i)*(2*pi))*eye(n_s)-A)*B+D;
    HI11(i) = H(1,1);
    HI21(i) = H(2,1);
    HI12(i) = H(1,2);
    HI22(i) = H(2,2);
    HI13(i) = H(1,3);
    HI23(i) = H(2,3);
end

% Plotting the Frequency Response
figure(16)
subplot(211)
semilogx(w,rad2deg(angle(HI11)))
title(['Phase Plots for u_1 calculated for n_s = ',num2str(n_s)])
grid on; axis([0.1 fnyq -200 200]); legend('H11'); legend('Location', 'southwest');
```

```
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');
subplot(212)
semilogx(w,rad2deg(angle(HI21)))
grid on; axis([0.1 fnyq -200 200]); legend('H21'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');

figure(17)
subplot(211)
loglog(w,abs(HI11))
title(['Frequency Response Magnitudes for u_1 calculated for n_s = ',num2str(n_s)])
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H11'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
subplot(212)
loglog(w,abs(HI21))
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H21'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');

figure(18)
subplot(211)
semilogx(w,rad2deg(angle(HI12)))
title(['Phase Plots for u_2 calculated for n_s = ',num2str(n_s)])
grid on; axis([0.1 fnyq -200 200]); legend('H12'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');
subplot(212)
semilogx(w,rad2deg(angle(HI22)))
grid on; axis([0.1 fnyq -200 200]); legend('H22'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');

figure(19)
subplot(211)
loglog(w,abs(HI12))
title(['Frequency Response Magnitudes for u_2 calculated for n_s = ',num2str(n_s)])
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H12'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
subplot(212)
loglog(w,abs(HI22))
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H22'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');

figure(20)
subplot(211)
semilogx(w,rad2deg(angle(HI13)))
title(['Phase Plots for u_3 calculated for n_s = ',num2str(n_s)])
grid on; axis([0.1 fnyq -200 200]); legend('H13'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');
subplot(212)
semilogx(w,rad2deg(angle(HI23)))
grid on; axis([0.1 fnyq -200 200]); legend('H23'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Phase in degrees');

figure(21)
subplot(211)
loglog(w,abs(HI13))
title(['Frequency Response Magnitudes for u_3 calculated for n_s = ',num2str(n_s)])
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H13'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
subplot(212)
loglog(w,abs(HI23))
grid on; axis([0.1 fnyq 1e-3 1e2]); legend('H23'); legend('Location', 'southwest');
xlabel('Frequency in Hertz'); ylabel('Amplitude');
```

```matlab
%% Task 4

% Task 4 Q1 and Q2 (Change the value of 'n_s' in task 3)
%Calculating the Eigenvalues of our system
lambda_d = eig(A);

%Plotting the eigenvalues
figure(22)
plot(real(lambda_d),imag(lambda_d),'x')
hold on
fplot(@(t) sin(t), @(t) cos(t));
title(['Eigenvalues of the system for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');

% Task 4 Q3 and Q4 (Change the value of 'n_s' in task 3)
% Constructing the 6 SISO systems
B11 = B(:,1);
C11 = C(1,:);

B21 = B(:,1);
C21 = C(2,:);

B12 = B(:,2);
C12 = C(1,:);

B22 = B(:,2);
C22 = C(2,:);

B13 = B(:,3);
C13 = C(1,:);

B23 = B(:,3);
C23 = C(2,:);

% Calculating the Transmission zeros for SISO systems
transmission_zeros11 = tzero(A,B11,-C11,0);
transmission_zeros21 = tzero(A,B21,-C21,0);
transmission_zeros12 = tzero(A,B12,-C12,0);
transmission_zeros22 = tzero(A,B22,-C22,0);
transmission_zeros13 = tzero(A,B13,-C13,0);
transmission_zeros23 = tzero(A,B23,-C23,0);

% Plotting the transmission zeros for each SISO system
figure(23)
plot(real(transmission_zeros11), imag(transmission_zeros11),'o')
hold on
plot(real(lambda_d),imag(lambda_d),'x')
fplot(@(t) sin(t), @(t) cos(t));
title(['Transmission zeros for SISO: u_1 and y_1 for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');
hold off

figure(24)
plot(real(transmission_zeros21), imag(transmission_zeros21),'o')
hold on
plot(real(lambda_d),imag(lambda_d),'x')
fplot(@(t) sin(t), @(t) cos(t));
title(['Transmission zeros for SISO: u_1 and y_2 for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');
hold off
```

```
figure(25)
plot(real(transmission_zeros12), imag(transmission_zeros12),'o')
hold on
plot(real(lambda_d),imag(lambda_d),'x')
fplot(@(t) sin(t), @(t) cos(t));
title(['Transmission zeros for SISO: u_2 and y_1 for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');
hold off

figure(26)
plot(real(transmission_zeros22), imag(transmission_zeros22),'o')
hold on
plot(real(lambda_d),imag(lambda_d),'x')
fplot(@(t) sin(t), @(t) cos(t));
title(['Transmission zeros for SISO: u_2 and y_2 for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');
hold off

figure(27)
plot(real(transmission_zeros13), imag(transmission_zeros13),'o')
hold on
plot(real(lambda_d),imag(lambda_d),'x')
fplot(@(t) sin(t), @(t) cos(t));
title(['Transmission zeros for SISO: u_3 and y_1 for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');
hold off

figure(28)
plot(real(transmission_zeros23), imag(transmission_zeros23),'o')
hold on
plot(real(lambda_d),imag(lambda_d),'x')
fplot(@(t) sin(t), @(t) cos(t));
title(['Transmission zeros for SISO: u_3 and y_2 for n_s = ',num2str(n_s)])
grid on; axis square; axis([-1.5 1.5 -1.5 1.5]); xlabel('Re(z)'); ylabel('Im(z)');
hold off

%% Task 5

% Calculating the eigenvalues for continuous time system
lambda_c = log(lambda_d)/ts;
```