
Table of Contents

MAE 283A: Homework 3, Question 3	1
Question 3.1	1
1. Load and organize data into Hankel matrices	1
2. Perform singular value decomposition	1
3. Approximate R with $R_1 R_2$	2
4. Compute inverses	2
5. Compute state space model	2
Simulate and plot results	3
Question 3.2	4
Least squares estimation in frequency domain (curve fitting)	4
Plot results	4
Multiple iterations with adjusted weighting	6
Plot results of iteration	7

MAE 283A: Homework 3, Question 3

% By David Lim

clear

Question 3.1

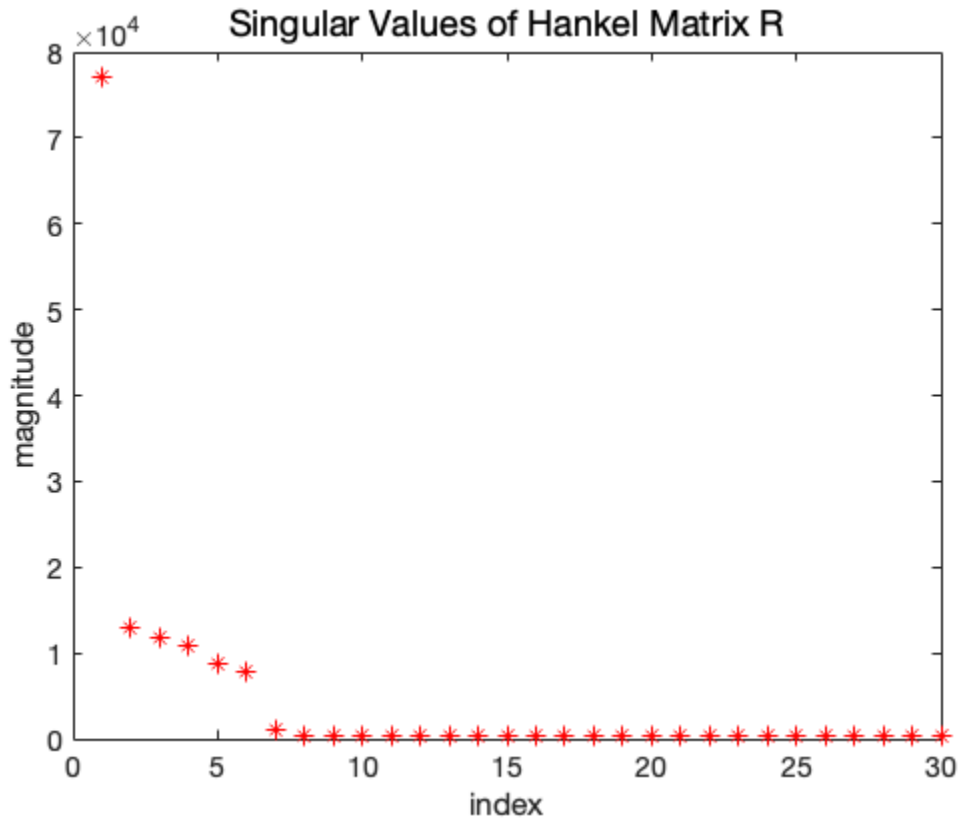
1. Load and organize data into Hankel matrices

```
load('R333_STEP_data.mat');

N = length(y);
N1 = N/2;
R = hankel(y(2:N1),y(N1:N-1)) - kron(y(1:N1-1),ones(1,N1));
Rbar = hankel(y(3:N1+1),y(N1+1:N)) - kron(y(2:N1),ones(1,N1));
```

2. Perform singular value decomposition

```
figure(1)
[U,S,V] = svd(R);
sigma = diag(S);
plot(sigma(1:30),'*',LineWidth=1,Color='r');
xlabel('index')
ylabel('magnitude')
set(gca,'FontSize',14)
title('Singular Values of Hankel Matrix R','FontWeight','Normal','FontSize',18)
```



3. Approximate R with $R1 \cdot R2$

```
n = 6;
R1 = U(:,1:n)*sqrt(S(1:n,1:n));
R2 = sqrt(S(1:n,1:n))*V(:,1:n)';
```

4. Compute inverses

```
R1dagger = sqrt(S(1:n,1:n))\U(:,1:n)';
R2dagger = V(:,1:n)/sqrt(S(1:n,1:n));
```

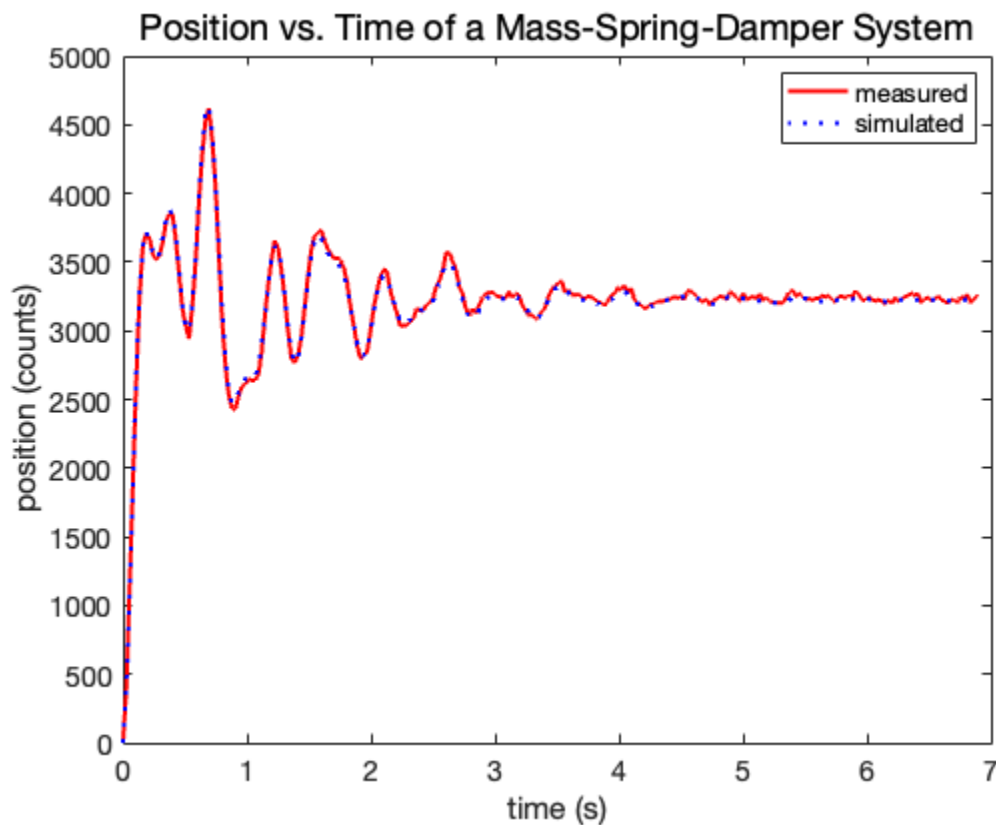
5. Compute state space model

```
A = R1dagger*Rbar*R2dagger;
B = R2(:,1);
C = R1(1,:);
D = y(1);

Ts = 0.0266;
G1 = ss(A,B,C,D,Ts);
[num1,den1] = tfdata(G1,'v');
```

Simulate and plot results

```
T = (N-1)*Ts;  
y_sim = step(G1,T);  
  
figure(2)  
p2 = plot(0:Ts:T,[y y_sim],LineWidth=2);  
p2(1).Color = 'r';  
p2(2).Color = 'b';  
p2(2).LineStyle = ':';  
xlabel('time (s)')  
ylabel('position (counts)')  
legend('measured','simulated')  
set(gca,'FontSize',14)  
title('Position vs. Time of a Mass-Spring-Damper  
System','FontWeight','Normal','FontSize',18)
```



Question 3.2

Least squares estimation in frequency domain (curve fitting)

```
load("R333_FRF_data.mat");

Ts = 1/f(end)/2;
w = 2*pi*f*Ts;

n = 6;
N = length(f);

X = ones(N,2*n+1);
for k = 1:n
    X(:,1+k) = exp(-k*1j*w);
end
for k = 1:n
    X(:,1+n+k) = -Gspa.*exp(-k*1j*w);
end

theta = [real(X);imag(X)]\[real(Gspa);imag(Gspa)];

G2 = tf(theta(1:n+1)',[1 theta(n+2:end)'],Ts);
```

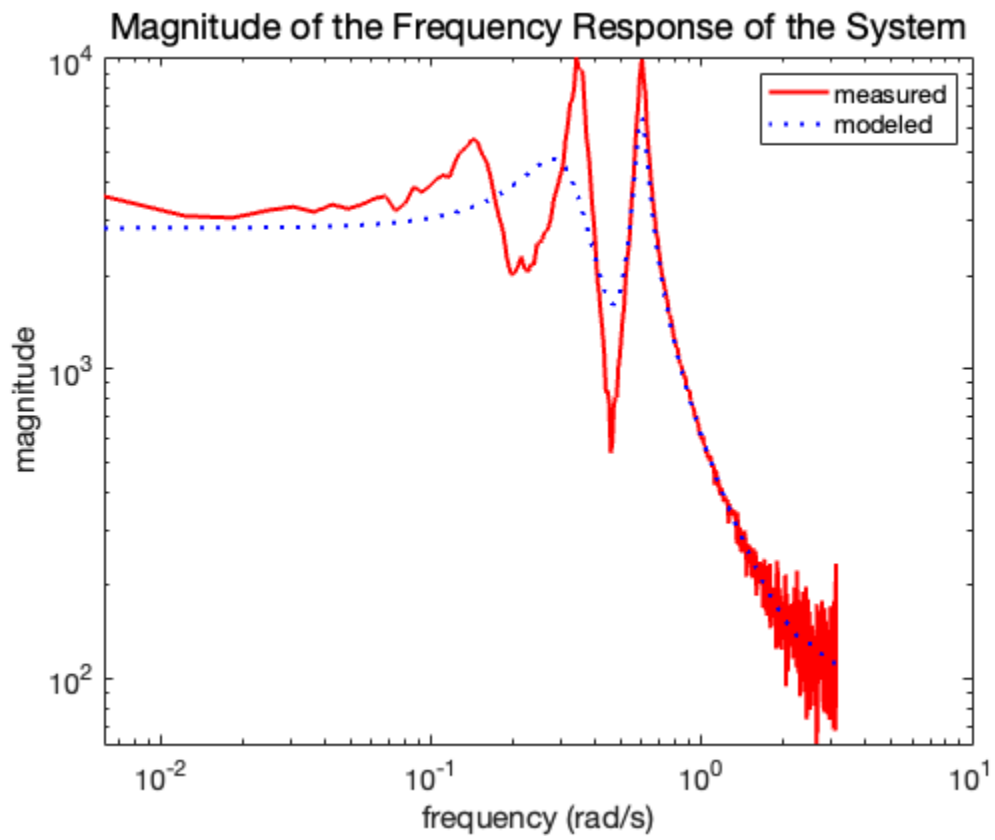
Plot results

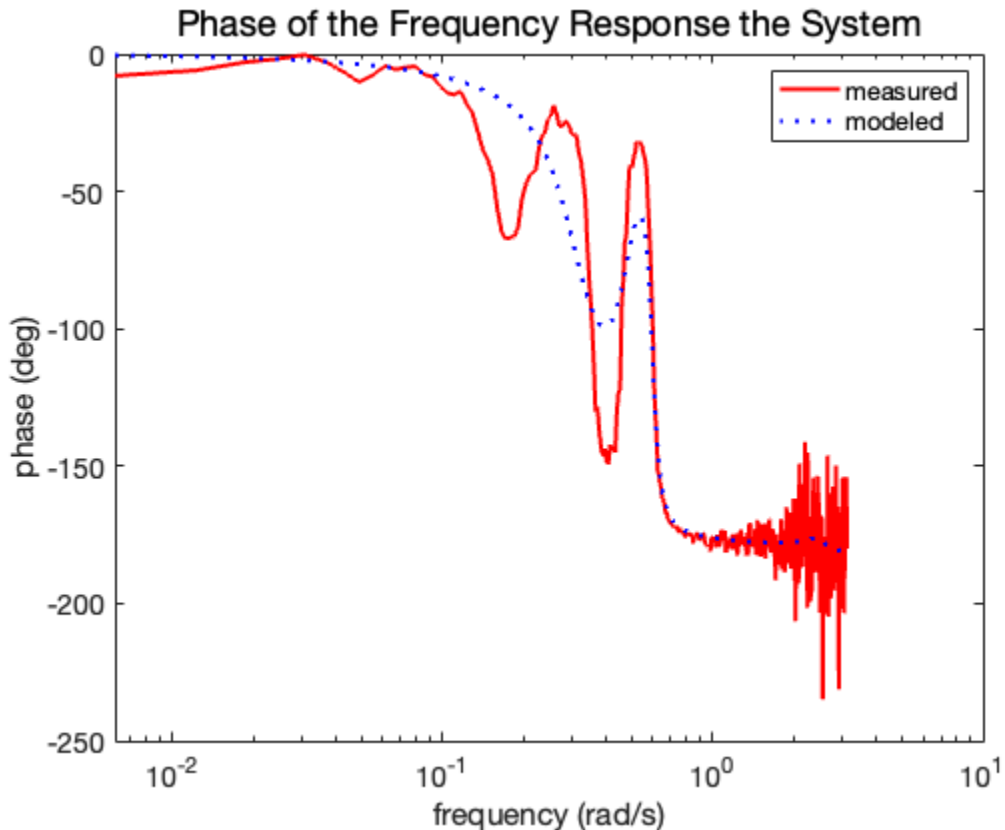
```
G2fr = reshape(freqresp(G2,exp(1j*w)),N,1);

figure(3)
p3 = loglog(w,[abs(Gspa) abs(G2fr)],LineWidth=2);
p3(1).Color = 'r';
p3(2).Color = 'b';
p3(2).LineStyle = ':';
xlabel('frequency (rad/s)')
ylabel('magnitude')
legend('measured','modeled')
set(gca,'FontSize',14)
title('Magnitude of the Frequency Response of the System','FontWeight','Normal','FontSize',18)

figure(4)
p4 = semilogx(w,[rad2deg(unwrap(angle(Gspa)))
rad2deg(unwrap(angle(G2fr)))],LineWidth=2);
p4(1).Color = 'r';
p4(2).Color = 'b';
p4(2).LineStyle = ':';
xlabel('frequency (rad/s)')
ylabel('phase (deg)')
legend('measured','modeled')
```

```
set(gca,'FontSize',14)
title('Phase of the Frequency Response the
System','FontWeight','Normal','FontSize',18)
```





Multiple iterations with adjusted weighting

```

max_par_diff = 1;
counter = 1;

while max_par_diff > 1e-8 && counter < 100
    X = ones(N,2*n+1);
    for k = 1:n
        X(:,1+k) = exp(-k*1j*w);
    end
    Weight = X(:,1:n+1)*[1;theta(n+2:end)];
    for k = 1:n
        X(:,1+n+k) = -Gspa.*exp(-k*1j*w);
    end
    theta_new = [real(Weight.\X);imag(Weight.\X)]\
[real(Weight.\Gspa);imag(Weight.\Gspa)];
    max_par_diff = max(abs(theta - theta_new));
    disp([num2str(counter) ' : max. par. difference = ' num2str(max_par_diff)
    '.']);
    theta = theta_new;
    counter = counter + 1;
end

G2 = tf(theta(1:n+1)',[1 theta(n+2:end)'],Ts);

```

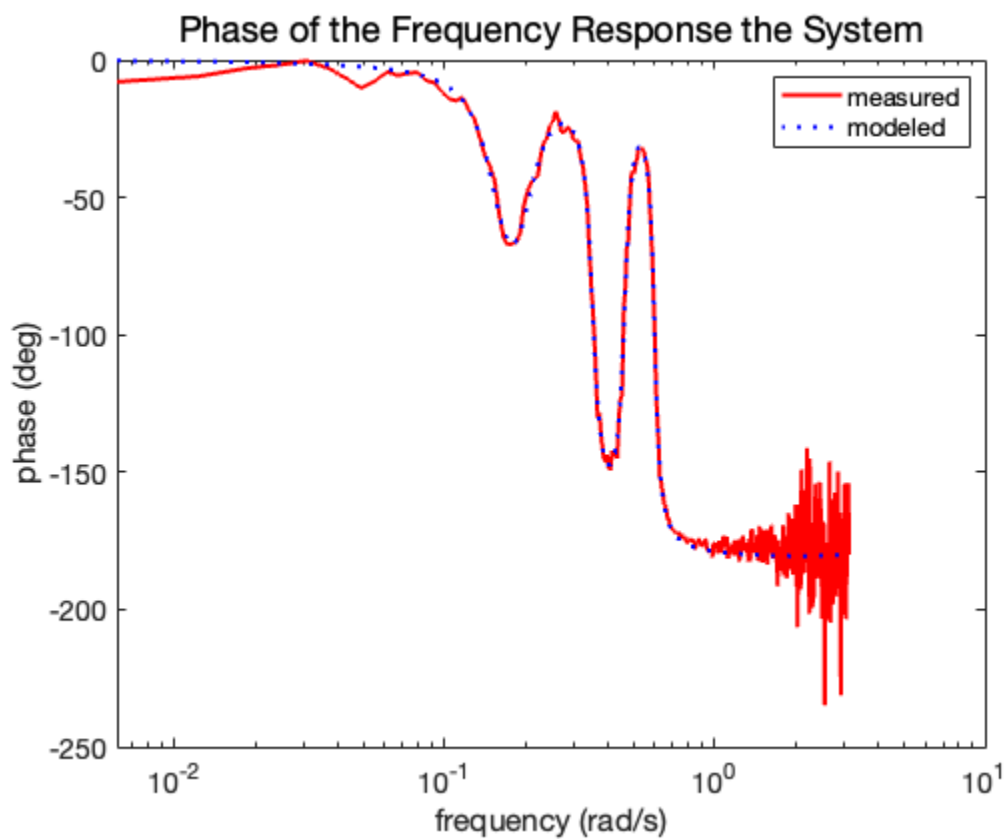
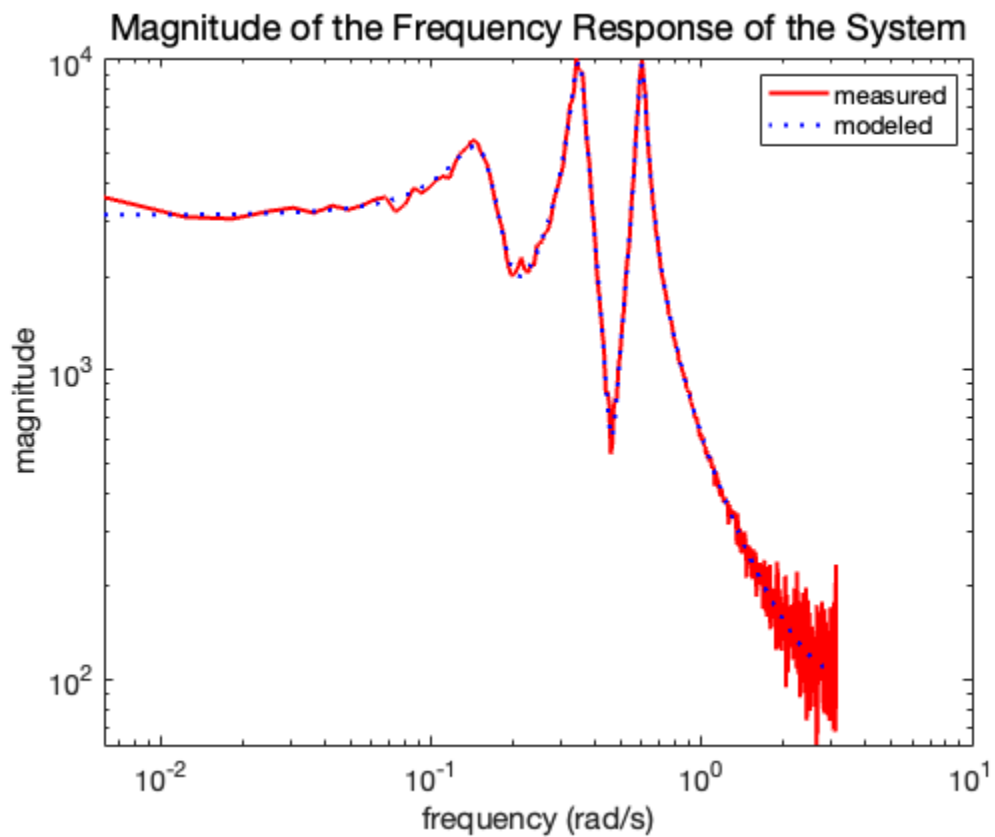
```
1: max. par. difference = 624.1538.
2: max. par. difference = 1858.6197.
3: max. par. difference = 73.8117.
4: max. par. difference = 26.1449.
5: max. par. difference = 1.0238.
6: max. par. difference = 0.051083.
7: max. par. difference = 0.001695.
8: max. par. difference = 7.0162e-05.
9: max. par. difference = 2.5196e-06.
10: max. par. difference = 1.9158e-07.
11: max. par. difference = 6.0918e-09.
```

Plot results of iteration

```
G2fr = reshape(freqresp(G2,exp(1j*w)),N,1);
```

```
figure(5)
p5 = loglog(w,[abs(Gspa) abs(G2fr)],LineWidth=2);
p5(1).Color = 'r';
p5(2).Color = 'b';
p5(2).LineStyle = ':';
xlabel('frequency (rad/s)')
ylabel('magnitude')
legend('measured','modeled')
set(gca,'FontSize',14)
title('Magnitude of the Frequency Response of the
System','FontWeight','Normal','FontSize',18)
```

```
figure(6)
p6 = semilogx(w,[rad2deg(unwrap(angle(Gspa)))
rad2deg(unwrap(angle(G2fr)))],LineWidth=2);
p6(1).Color = 'r';
p6(2).Color = 'b';
p6(2).LineStyle = ':';
xlabel('frequency (rad/s)')
ylabel('phase (deg)')
legend('measured','modeled')
set(gca,'FontSize',14)
title('Phase of the Frequency Response the
System','FontWeight','Normal','FontSize',18)
```



Published with MATLAB® R2024a