Final Take-Home Project

David Lim, A16398479

MAE 283A, Fall 2024

MAE UGCL mechanical rotational positioning system data:

- Input signal $u(t)$: voltage supply to the motor
- Output signal $y(t)$: encoder measurement of the lower disk

Figure 1: Input Auto-Correlation Estimate

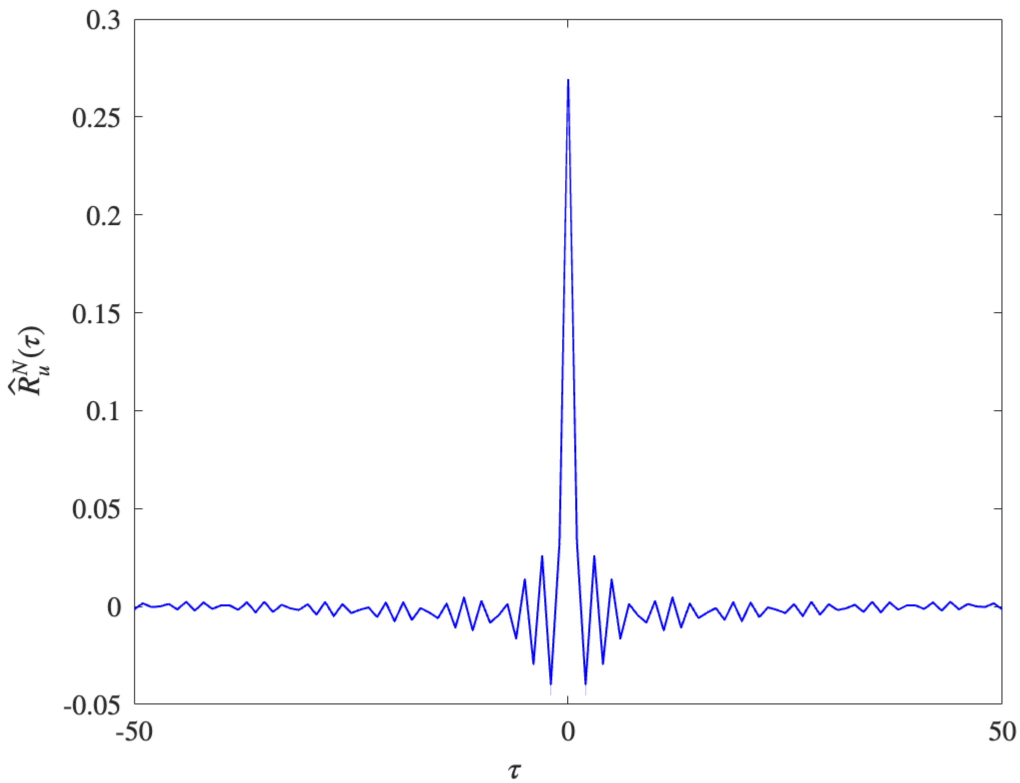

Fig. 1. Input auto-correlation estimate $\hat{R}_u^N(\tau)$ of the input signal $u(t)$. The input signal was a sinusoidal sweep with a linearly increasing frequency from 0.1 Hz to 50 Hz and an amplitude of 0.75 V. The input signal specified the voltage supply to a motor that rotated two disks coupled by a metal rod.

Figure 2: Input Spectrum Estimate

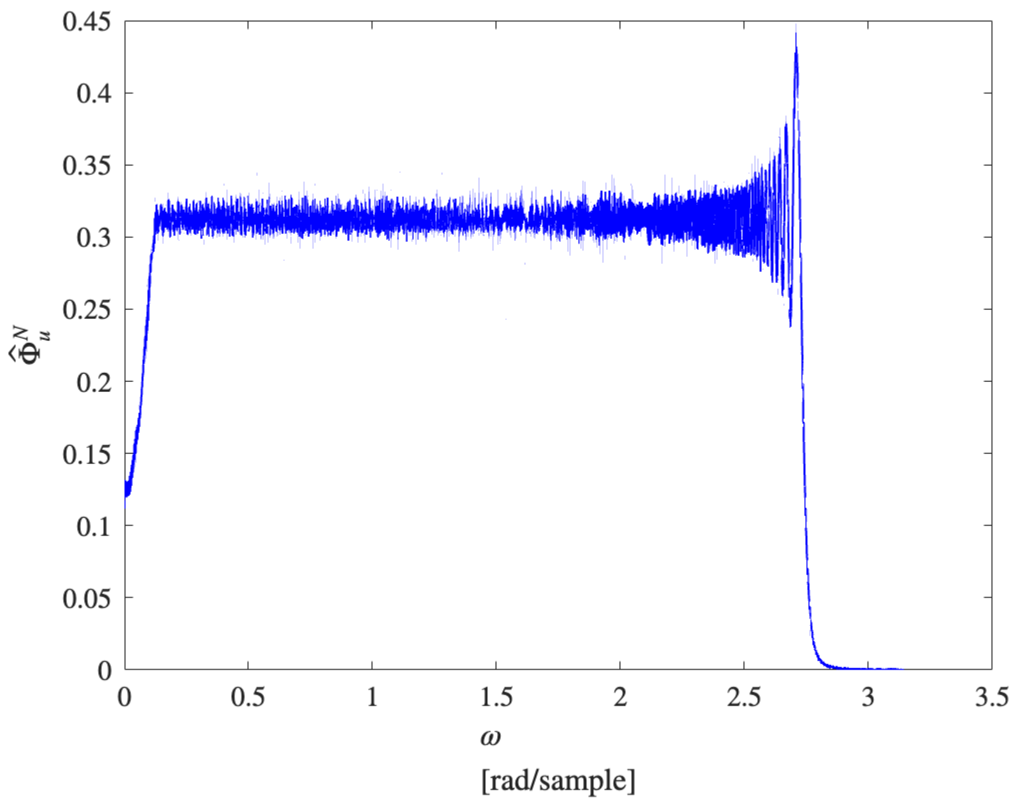

Fig. 2. Input spectrum estimate $\widehat{\Phi}_u^N(\omega)$. The input spectrum covers frequencies from 0 to 2.72 rad/sample (48.9 Hz). The spectral estimate of the system frequency response $G_0(e^{j\omega})$ is estimated reliably in the range where the input spectrum is approximately uniform, i.e. frequencies from 0.12 rad/sample (2.15 Hz) to 2.72 rad/sample (48.9 Hz).

Figure 3: SPA Estimate

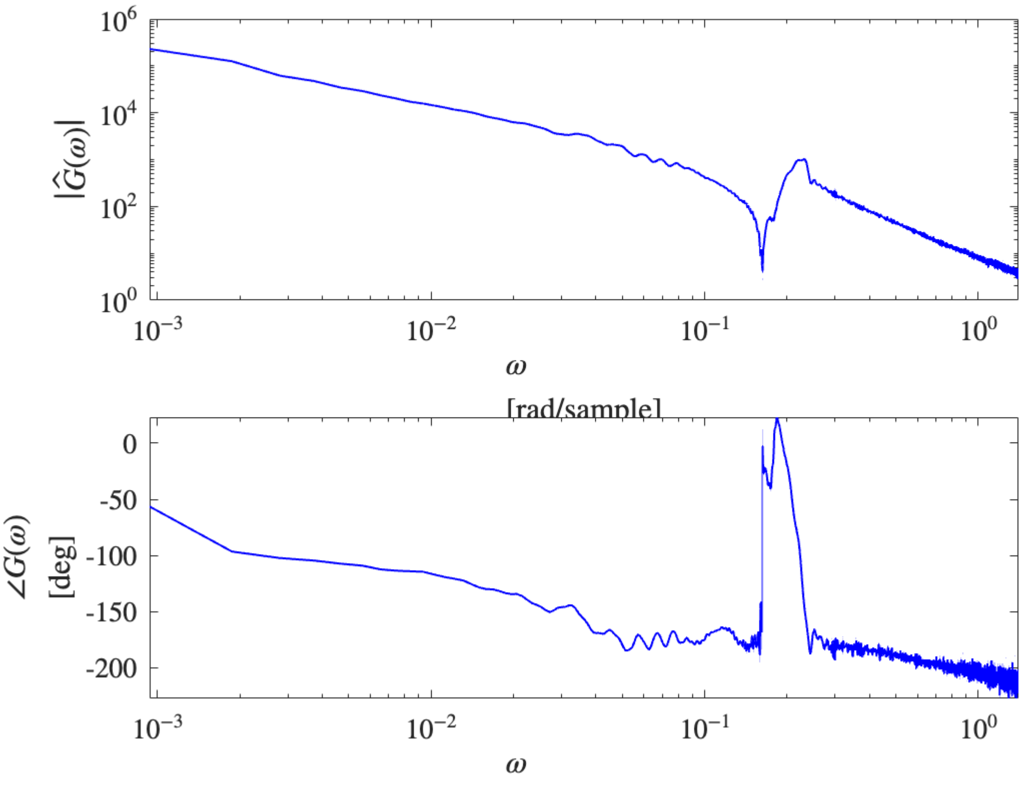

Fig. 3. Magnitude and phase Bode plot of the SPA estimate $\widehat{G}(\omega)$. A hanning window of length N was applied to the correlation estimates to improve the spectral estimate. The plot truncates the frequency response data, hiding an abrupt fluctuation occurs in the magnitude and phase.

Downsamping was not performed for the following analysis, as the samping rate was not too high.

Figure 4: FIR Model

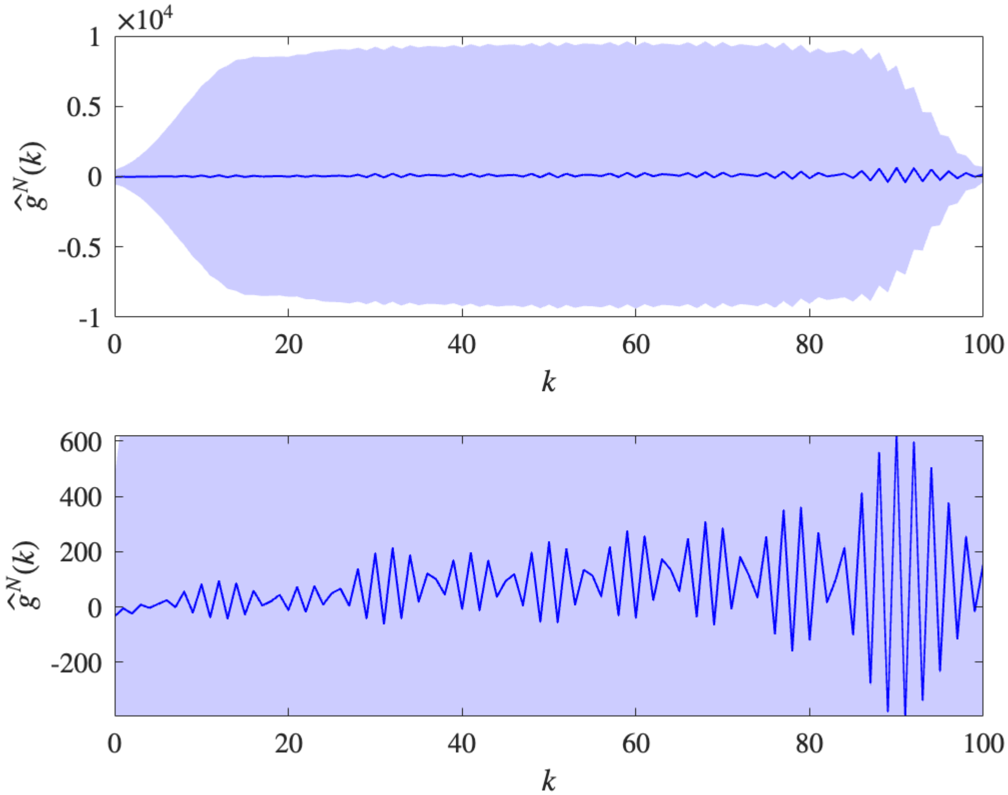

Fig. 4. FIR model parameters and 99% confidence bounds with $n = 100$ (the lower plot is zoomed in). The input signal is not white noise, so the FIR model does not estimate the impulse response of the system. The parameters trend upwards with an oscillation of increasing energy, which is not realistic for the impulse response of the actual system.

Figure 5: FIR Residuals vs. Past Inputs

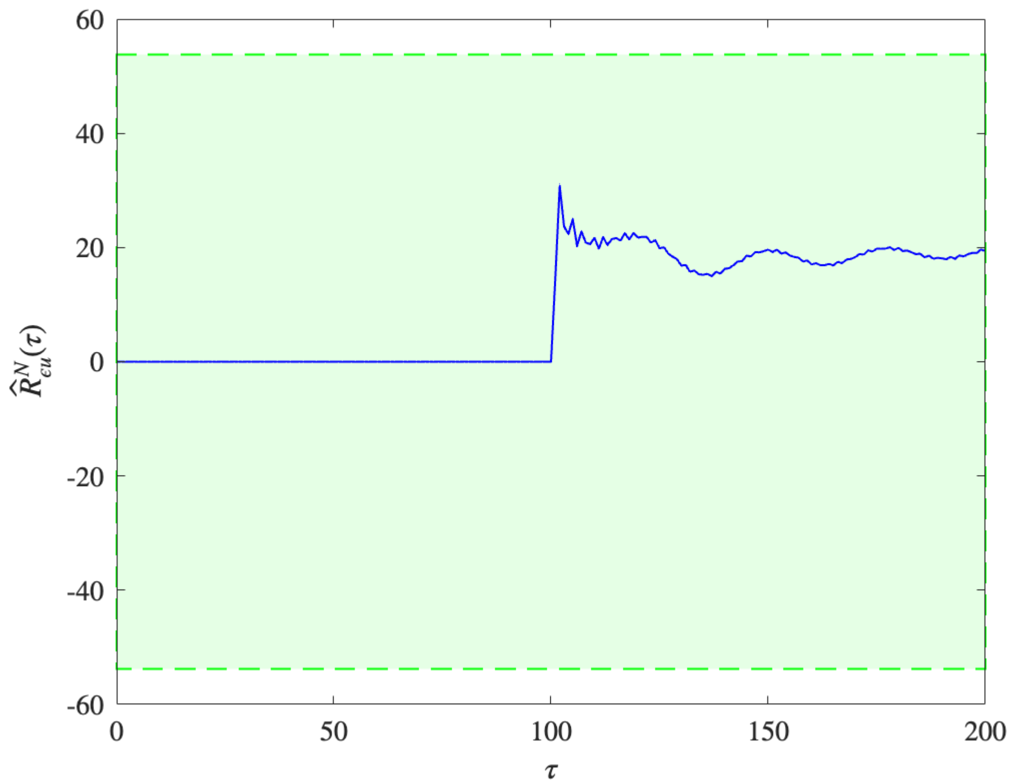

Fig. 5. Estimated cross-correlation $\widehat{R}^N_{\epsilon u}(\tau)$ between the input and the FIR prediction error with 99% confidence bounds for $0 \leq \tau \leq 200$ (the plot is not normalized by the variances of $u(t)$ and $\epsilon(t, \widehat{\theta})$ as is done by the resid() function). As expected, $\widehat{R}^N_{\epsilon u}(\tau)$ is zero for $0 \leq \tau \leq 100$ as a result of LS estimation. The subsequent values of $\widehat{R}^N_{\epsilon u}(\tau)$ remain within the confidence interval, so the model passes the validation test. However, this does not mean the FIR model is useful for simulation/ prediction.

The FIR model does not clearly indicate the delay of the system. Delay would have been evident if the first parameters of the FIR model were approximately 0. However, if the first negative parameters are considered, then this would possibly indicate about 3 time steps of delay.

Figure 6: Hankel Matrix Singular Values

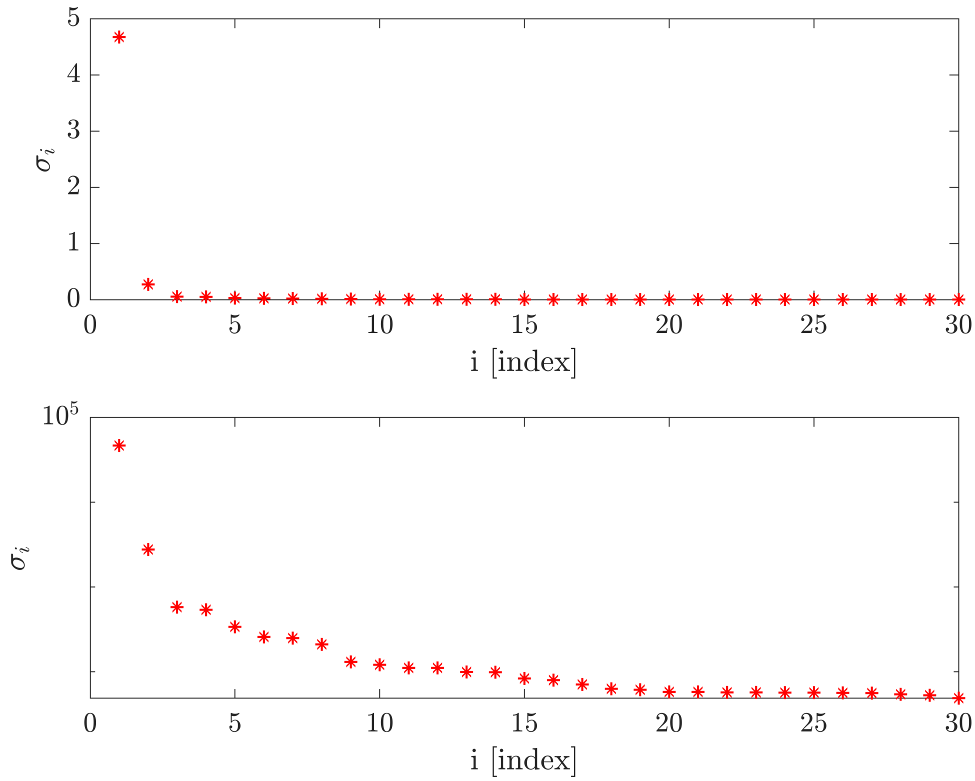

Fig. 6. Singular values of the Hankel matrix (the lower plot with logarithmic scaling). The impulse response coefficient estimates are obtained from the inverse Fourier transform of the frequency response estimate. The first two singular values are most prominent, but a second-order model would not be sufficient in capturing the high frequency resonant mode in the frequency response estimate. The next two singular values could be considered as contributing to the rank. A model order of $n = 4$ should be sufficient for modeling the two-mass system.

Figure 7: Realization Algorithmn

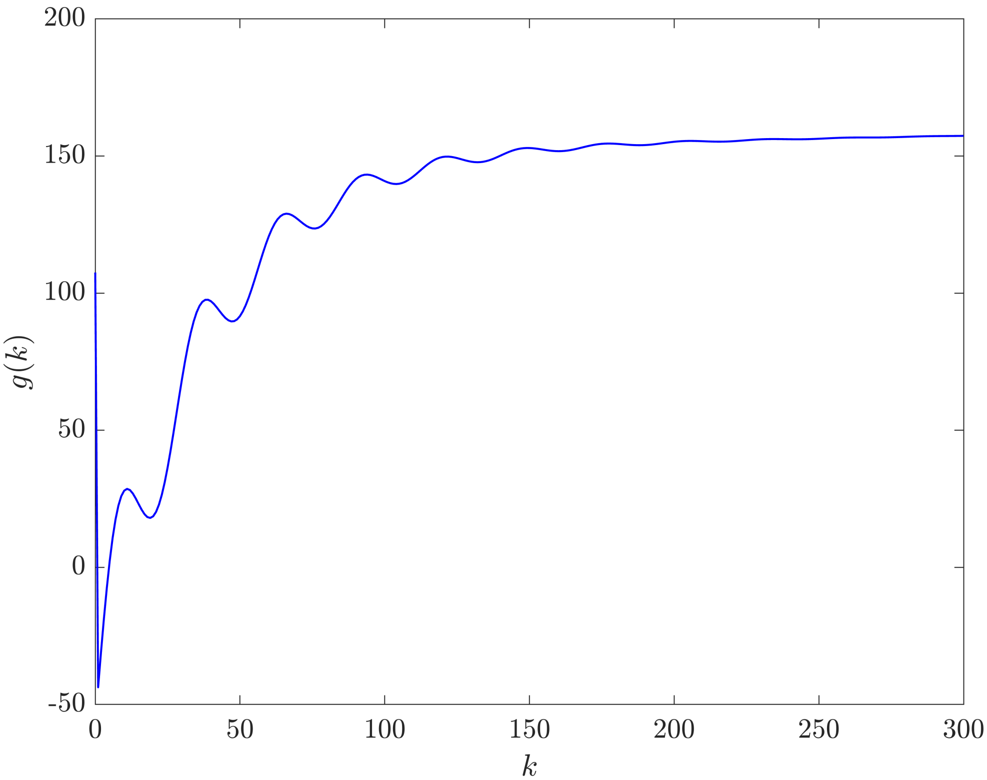

Fig. 7. Impulse response of the state space model obtained with the realization algorithm. This impulse response estimate is more accurate than the FIR model parameters. However, the impulse response of the model grows unbounded outside of the plotted time interval. The actual impulse response of the system is expected to settle to a non-zero value since the system contains an integrator relating position, the output signal, and velocity.

Figure 8: Realization Residuals vs. Past Inputs

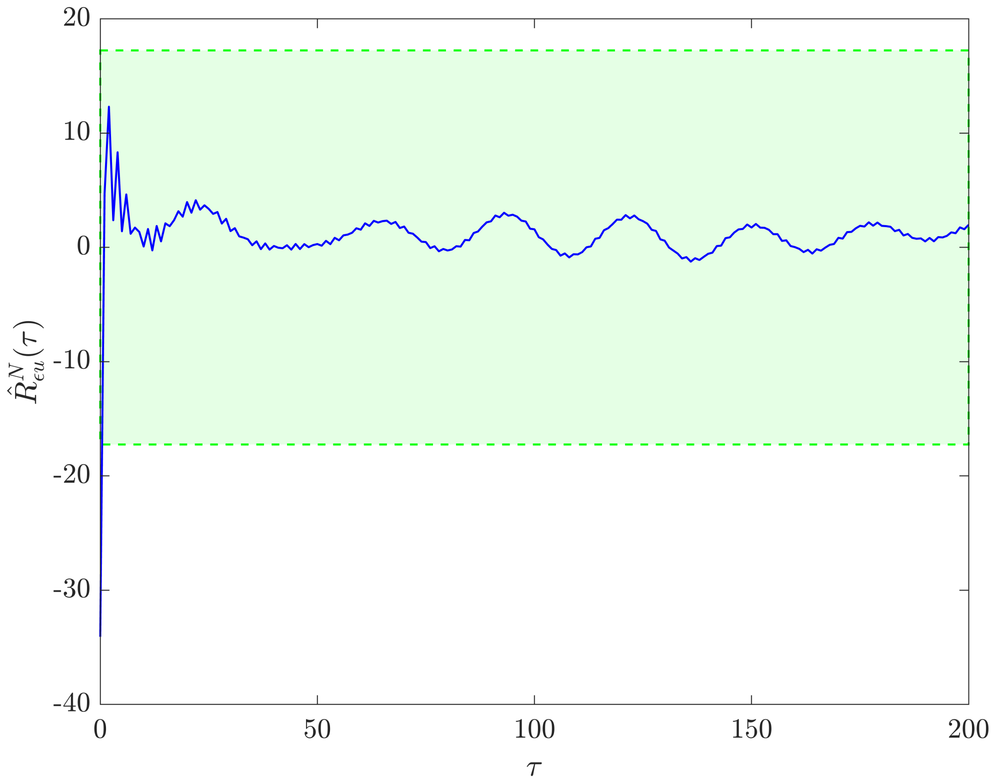

Fig. 8. Estimated cross-correlation $\hat{R}^N_{\epsilon u}(\tau)$ between the input and the state space model prediction error with 99% confidence bounds for $0 \leq \tau \leq 200$. The value $\hat{R}^N_{\epsilon u}(0)$ is outside of the confidence interval, but the subsequent values of $\hat{R}^N_{\epsilon u}(\tau)$ remain within the interval. This is the result of the model failing to match the high frequency attenuation behavior of the actual system (comparison of the frequency responses indicates this discrepancy). High frequency content of the input is reflected in the simulated output, and this contributes to the prediction error and the cross-correlation $\hat{R}^N_{\epsilon u}(\tau)$ at $\tau = 0$. However, if the high frequency behavior of the model is ignored, then this model technically passes the model validation test. However, this does not mean the model is useful for simulation/prediction, especially since the model has an unbounded impulse response.

Figure 9: OE Model Residuals vs. Past Inputs

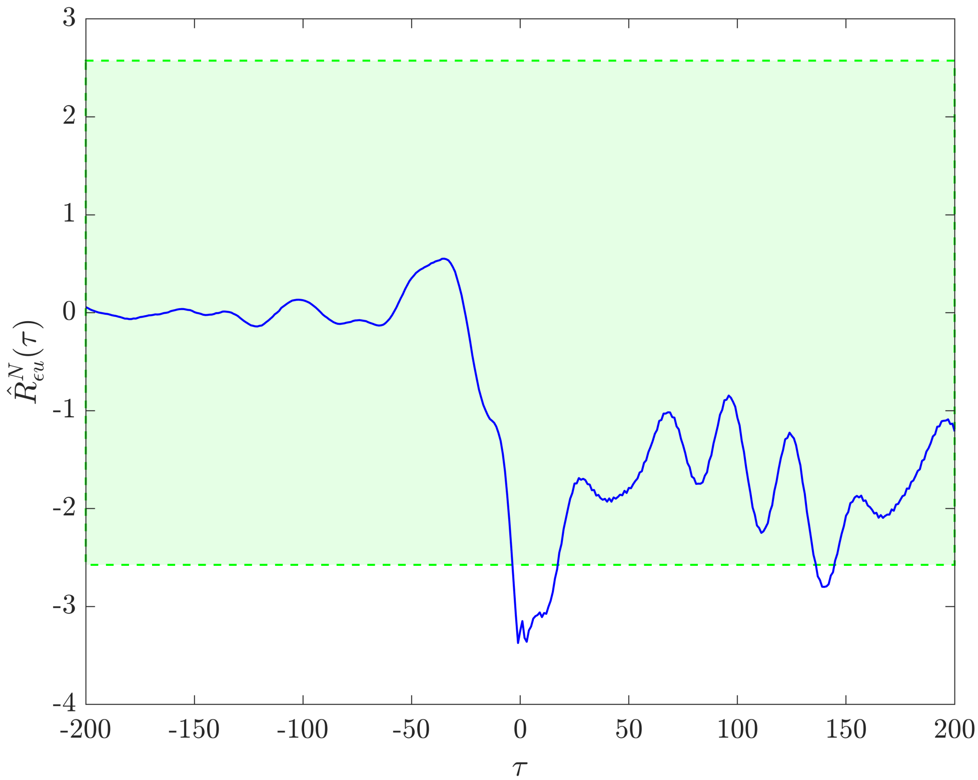

Fig. 9. Estimated cross-correlation $\widehat{R}^N_{\epsilon u}(\tau)$ between the input and an OE prediction error with 99% confidence bounds for $|\tau| \leq 200$. An OE model structure and a model order of $n = 4$ were chosen. Some values of $\widehat{R}^N_{\epsilon u}(\tau)$ are outside of the confidence interval, so the model does not pass the validation test. However, this does not necessarily mean the model should be rejected.

Figure 10: Measured Output vs. Simulation/Prediction

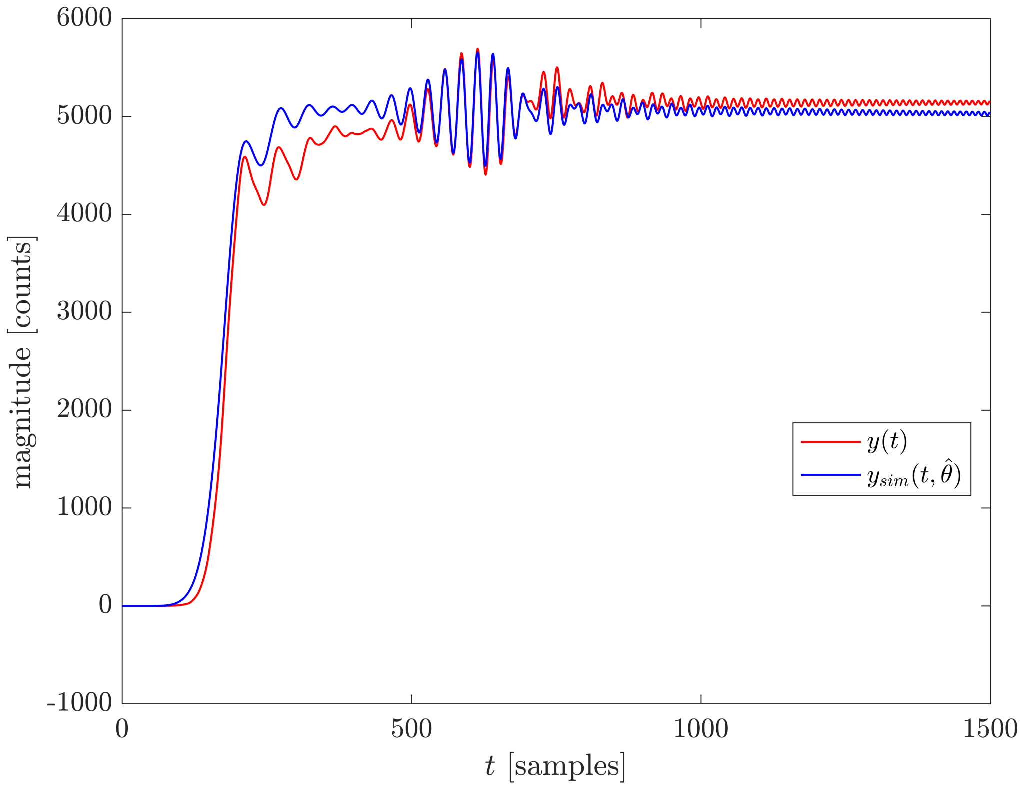

Fig. 10. Measured output $y(t)$ and simulated/predicted output $y_{sim}(t, \widehat{\theta})$. Since the model has an OE structure, the simulation and the prediction are equivalent. The model reproduces the overall behavior of the output and closely matches the amplitude, frequency, and offset of the main resonant mode.

Process:

I decided to use frequency domain identification and curve-fitting, so I only realized a system model $G(\cdot, \widehat{\theta})$ and assumed an OE structure with $H(\cdot, \widehat{\theta}) = 1$. I expected the system to have an integrator and thus a pole at $z = 1$. The following steps led to the best model:

- I removed the integrator by filtering the output data through the filter $L(q) = q - 1$, effectively differencing the output and cancelling the effect of the integrator.
- I obtained a frequency response estimate of the filtered data.

- I used constrained LS curve-fitting in the frequency domain with multiple iterations to obtain a parametric estimate. I constrained the model to match the DC gain.
- I applied the integrator to the parametric model by multiplying the model transfer function by the inverse transfer function of the filter, adding the pole at $z = 1$.

Conclusion:

I believe the input signal was a limitiation as the input spectrum did not adequately cover lower frequencies. I would go back and redo the experiment, paying special attention to the experiment design.

Thank you Professor de Callafon for teaching this course!

Table of Contents

All My Code

Proof of my own work

```
clear
load('data.mat')

linewidth = 1;
markersize = 15;
fontsize = 14;
```

Figure 1: Input Auto-Correlation Estimate

```
m = 50;
Ru = xcorr(u,m,'biased');  % 'biased' includes 1/N

figure(1)
clf
plot(-m:m,Ru,'b-','LineWidth',linewidth,'MarkerSize',markersize);
xlabel('$$\tau$$ [samples]','Interpreter','latex')
ylabel('$$\hat{R}_u^N(\tau)$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 2: Input Spectrum Estimate

```
U = fft(u);
PHIu = U.*conj(U)/N;
w = linspace(0,pi,N/2+1);

figure(2)
```

```
clf
plot(w,PHIu(1:N/2+1),'b-','LineWidth',linewidth,'MarkerSize',markersize);
xlabel('$$\omega$$ [rad/sample]','Interpreter','latex')
ylabel('$$\hat{\Phi}_u^N$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 3: SPA Estimate

```
Ruu = xcorr(u,u,N/2);
Ryu = xcorr(y,u,N/2);
gamma = N;
Ruu_weighted = Ruu.*[zeros((N-gamma)/2,1); hanning(gamma+1); zeros((N-gamma)/
2,1)];
Ryu_weighted = Ryu.*[zeros((N-gamma)/2,1); hanning(gamma+1); zeros((N-gamma)/
2,1)];
% ensure auto-spectrum is real-valued (since auto-correlation is symmetric)
Suu = 1/N*fft([Ruu_weighted(N/2+1:N); Ruu_weighted(1:N/2)]);
Syu = 1/N*fft([Ryu_weighted(N/2+1:N); Ryu_weighted(1:N/2)]);
P = Suu(1:N/2+1).\Syu(1:N/2+1);

m = 1500;  % truncated window to exclude artifact caused by input
figure(3)
clf
tiledlayout(2,1)
nexttile(1)
loglog(w(1:m),abs(P(1:m)),'b-','LineWidth',linewidth,'MarkerSize',markersize);
xlabel('$$\omega$$ [rad/sample]','Interpreter','latex')
ylabel('$$\left| \hat{G}(\omega) \right|$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
nexttile(2)
semilogx(w(1:m),rad2deg(unwrap(angle(P(1:m)))),'b-','LineWidth',linewidth,'Mar
kerSize',markersize);
xlabel('$$\omega$$ [rad/sample]','Interpreter','latex')
ylabel('$$\angle \hat{G}(\omega)$$ [deg]','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 4: FIR Model

yf = filter([1 -1],1,y);

```
yf = y;
n = 100;
p = n+1;
idx = (p:N);
PHI = zeros(N-p+1,p);
for k = 1:p
    PHI(:,k) = u(idx+1-k);
end
Y = yf(idx);
theta_FIR = PHI\Y;
```

```
mdl = fitlm(PHI,Y,'Intercept',false);
ci = coefCI(mdl,0.01);

k = 0:n;

figure(4)
clf
tiledlayout(2,1)
nexttile(1)
fill([k flip(k)],
[ci(:,1);flip(ci(:,2))],'b','FaceAlpha',0.2,'LineStyle','none')
hold on
plot(k,theta_FIR,'b-','LineWidth',linewidth,'MarkerSize',markersize);
hold off
xlabel('$$k$$','Interpreter','latex')
ylabel('$$\hat{g}^N(k)$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
nexttile(2)
fill([k flip(k)],
[ci(:,1);flip(ci(:,2))],'b','FaceAlpha',0.2,'LineStyle','none')
hold on
plot(k,theta_FIR,'b-','LineWidth',linewidth,'MarkerSize',markersize);
hold off
xlabel('$$k$$','Interpreter','latex')
ylabel('$$\hat{g}^N(k)$$','Interpreter','latex')
ax = axis;
axis([ax(1) ax(2) min(theta_FIR) max(theta_FIR)])
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 5: FIR Residuals vs. Past Inputs

```
eps = yf - filter(theta_FIR,1,u);
alpha = 0.01;
Nalpha = norminv(1-alpha/2,0,1);
P1 = sum(xcorr(eps,'biased').*xcorr(u,'biased'));
CIeps = sqrt(P1/N)*Nalpha;
Reu = xcorr(eps,u,2*n,'biased');

figure(5)
fill([0 2*n 2*n 0],[CIeps CIeps -CIeps
-CIeps],'g--','LineWidth',linewidth,'EdgeColor','g','FaceAlpha',0.1);
hold on
plot(0:2*n,Reu(2*n+1:end),'b','LineWidth',linewidth)
hold off
xlabel('$$\tau$$','Interpreter','latex')
ylabel('$$\hat{R}_{\epsilon u}^N(\tau)$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 6: Hankel Matrix Singular Values

ensure ifft() returns a real-valued, time domain signal (make two-sided spectrum)

```
P2 = [P(1); P(2:end); flip(conj(P(2:end)))];
g = ifft(P2);
N1 = 300;  % impulse response starts to decay after 300 steps
H = hankel(g(2:N1+1),g(N1+1:2*N1-1));
[U,S,V] = svd(H);
sigma = diag(S);

m = 30;
figure(6)
clf
tiledlayout(2,1)
nexttile(1)
plot(sigma(1:m),'*',LineWidth=1,Color='r');
xlabel('i [index]','Interpreter','latex')
ylabel('$$\sigma_i$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
nexttile(2)
semilogy(sigma(1:m),'*',LineWidth=1,Color='r');
xlabel('i [index]','Interpreter','latex')
ylabel('$$\sigma_i$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 7: Realization Algorithmn

```
n = 4;  % best result
H1 = U(:,1:n)*sqrt(S(1:n,1:n));
H2 = sqrt(S(1:n,1:n))*V(:,1:n)';
H1dagger = sqrt(S(1:n,1:n))\U(:,1:n)';
H2dagger = V(:,1:n)/sqrt(S(1:n,1:n));
Hbar = hankel(g(3:N1+2),g(N1+2:2*N1));
A = H1dagger*Hbar*H2dagger;
B = H2(:,1);
C = H1(1,:);
D = g(1);  % need to improve estimate (g estimate initially noisy)
G_SS1 = ss(A,B,C,D,1);
% [num,den] = tfdata(G_ss1,'v');

% state = ss(A,B,eye(n),0,1);  % internal state x(t)
% x = lsim(state,u);  % simulate state x(t) with input u(t)
% % LS estimate of C and D
% PHI = [x u];
% Y = y;
% theta_r = PHI\Y;
% C_LS = theta_r(1:n)';
% D_LS = theta_r(n+1);
% G_SS2 = ss(A,B,C_LS,D_LS,1);
% % [num,den] = tfdata(G_ss2,'v');
```

```
m = 300;
figure(7)
clf
plot(0:m,impulse(G_SS1,m),'b-','LineWidth',linewidth,'MarkerSize',markersize);
xlabel('$$k$$','Interpreter','latex')
ylabel('$$g(k)$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 8: Realization Residuals vs. Past Inputs

```
m = 100;
eps = y - lsim(G_SS1,u);
alpha = 0.01;
Nalpha = norminv(1-alpha/2,0,1);
P1 = sum(xcorr(eps,'biased').*xcorr(u,'biased'));
CIeps = sqrt(P1/N)*Nalpha;
Reu = xcorr(eps,u,2*m,'biased');

figure(8)
clf
fill([0 2*m 2*m 0],[CIeps CIeps -CIeps
-CIeps],'g--','LineWidth',linewidth,'EdgeColor','g','FaceAlpha',0.1);
hold on
plot(0:2*m,Reu(2*m+1:end),'b','LineWidth',linewidth)
hold off
xlabel('$$\tau$$','Interpreter','latex')
ylabel('$$\hat{R}_{\epsilon u}^N(\tau)$$','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 9: OE Model Residuals vs. Past Inputs

```
load("G1a3b5d0i100CLS.mat")
m = 100;
eps = y - lsim(G1,u);
alpha = 0.01;
Nalpha = norminv(1-alpha/2,0,1);
P1 = sum(xcorr(eps,'biased').*xcorr(u,'biased'));
CIeps = sqrt(P1/N)*Nalpha;
Reu = xcorr(eps,u,2*m,'biased');

figure(9)
fill([-2*m 2*m 2*m -2*m],[CIeps CIeps -CIeps
-CIeps],'g--','LineWidth',linewidth,'EdgeColor','g','FaceAlpha',0.1);
hold on
plot(-2*m:2*m,Reu,'b','LineWidth',linewidth)
hold off
xlabel('$$\tau$$','Interpreter','latex')
ylabel('$$\hat{R}_{\epsilon u}^N(\tau)$$','Interpreter','latex')
```

```
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Figure 10: Measured Output vs. Simulation/Prediction

```
ysim = lsim(G1,u);

m = 1500;
figure(10)
clf
plot(0:m-1,y(1:m),'r-','LineWidth',linewidth,'MarkerSize',markersize);
hold on
plot(0:m-1,ysim(1:m),'b-','LineWidth',linewidth,'MarkerSize',markersize);
hold off
xlabel('$$t$$ [samples]','Interpreter','latex')
ylabel('magnitude [counts]','Interpreter','latex')
legend('$$y(t)$$','$$y_{sim}(t,\hat{\theta})$
$','Location','best','Interpreter','latex')
set(gca,'LineWidth',linewidth/
2,'FontSize',fontsize,'TickLabelInterpreter','latex')
```

Frequency Domain Realization

```
clear
M = readmatrix("lin_sweep.txt",'Whitespace',[';','[',']']);
t = M(:,3);
y = M(:,5); % x1
u = M(:,8);

N = length(t);
Ts = mean(diff(t));

uf = u;
yf = y;

[G, w] = tfestimate(uf,yf,rectwin(N),0,N);

Y = fft(yf);
U = fft(uf);
P = U(1:N/2+1).\Y(1:N/2+1);
```

Plot

```
figure(4)
clf
loglog(w,abs(P),'o',w,abs(G),'.','LineWidth',1);

xlabel("Frequency (rad/s)")
ylabel("Magnitude")
title('Amplitude Bode plot')
```

```
ax = gca;
ax.TitleHorizontalAlignment = 'left';
set(ax,'FontSize',18)
legend('G_{etfe}','tfestimate')

% save("Getfe.mat","P")
```

Constrained LS ?!?!

```
clear
load("Getfe.mat")
w = linspace(0,pi,length(P))';

m = 2000;
Gest = P(1:m);
w = w(1:m);

N = length(Gest);

nb = 5;
na = 3;
nd = 0;
ntotal = nb+na;

gamma = 1.78e2;
A = [ones(1,nb) -gamma*ones(1,na)];

X = zeros(N,ntotal);
for k = nd:nd+nb-1
    X(:,1+k) = exp(-k*1j*w);
end
for k = 1:na
    X(:,nb+k) = -Gest.*exp(-k*1j*w);
end

PHI = [real(X);imag(X)];
Y = [real(Gest);imag(Gest)];

theta = [PHI'*PHI, A'; A, zeros(size(A,1))] \ [PHI' * Y; gamma];

G = tf(theta(1:nb)',conv([1 zeros(1,nd)],[1 theta(nb+1:ntotal)']),1);

counter = 0;
max_par_diff = 1;

% Plot results

Gfr = reshape(freqresp(G,exp(1j*w)),N,1);

figure(3)
p3 = loglog(w,[abs(Gest) abs(Gfr)],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)
subtitle(['nb = ' num2str(nb) ', na = ' num2str(na) ', nd = ' num2str(nd) ',
iteration: ' num2str(counter)])

figure(4)
p4 = semilogx(w,[rad2deg(unwrap(angle(Gest)))
rad2deg(unwrap(angle(Gfr)))],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)
subtitle(['nb = ' num2str(nb) ', na = ' num2str(na) ', nd = ' num2str(nd) ',
iteration: ' num2str(counter)])
```

Multiple iterations with adjusted weighting

```
while max_par_diff > 1e-8 && counter < 100
    X = zeros(N,na+1);
    for k = 0:na
        X(:,1+k) = exp(-k*1j*w);
    end
    Weight = X*[1;theta(nb+1:ntotal)];
    X = zeros(N,ntotal);
    for k = nd:nd+nb-1
        X(:,1+k) = exp(-k*1j*w);
    end
    for k = 1:na
        X(:,nb+k) = -Gest.*exp(-k*1j*w);
    end
    PHI = [real(Weight.\X);imag(Weight.\X)];
    Y = [real(Weight.\Gest);imag(Weight.\Gest)];
    theta_new = [PHI'*PHI, A'; A, zeros(size(A,1))] \ [PHI' * Y; gamma];
    max_par_diff = max(abs(theta - theta_new));
    theta = theta_new;
    counter = counter + 1;
    disp([num2str(counter) ': max. par. difference = ' num2str(max_par_diff)
'.']);
end

G = tf(theta(1:nb)',conv([1 zeros(1,nd)],[1 theta(nb+1:ntotal)']),1);

% Plot results of iteration
```

```
Gfr = reshape(freqresp(G,exp(1j*w)),N,1);

figure(5)
p5 = loglog(w,[abs(Gest) abs(Gfr)],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)
subtitle(['nb = ' num2str(nb) ', na = ' num2str(na) ', nd = ' num2str(nd) ',
iteration: ' num2str(counter)])

figure(6)
p6 = semilogx(w,[rad2deg(unwrap(angle(Gest)))
rad2deg(unwrap(angle(Gfr)))],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)
subtitle(['nb = ' num2str(nb) ', na = ' num2str(na) ', nd = ' num2str(nd) ',
iteration: ' num2str(counter)])

zero(G)
pole(G)

num = cell2mat(G.Numerator);
den = cell2mat(G.Denominator);

G1 = tf(num,den,1)*tf(1,[1 -1],1);

%
save(strcat("G","a",num2str(na),"b",num2str(nb),"d",num2str(nd),"i",num2str(co
unter),"CLS.mat"),"G")
%
save(strcat("G1","a",num2str(na),"b",num2str(nb),"d",num2str(nd),"i",num2str(c
ounter),"CLS.mat"),"G1")
```

Published with MATLAB® R2024a