

24-774 Lab 3 Report

- Roshan Pradhan (roshanpr)

A - Frequency Domain System Identification

Question 1: Closed Loop Transfer Function

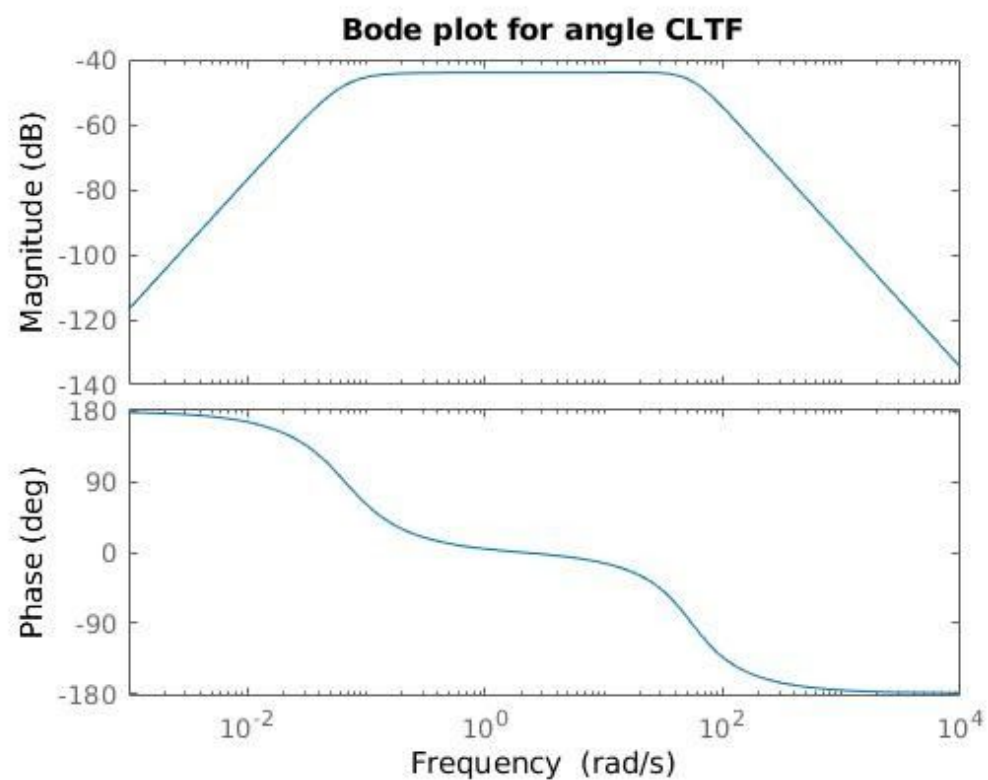
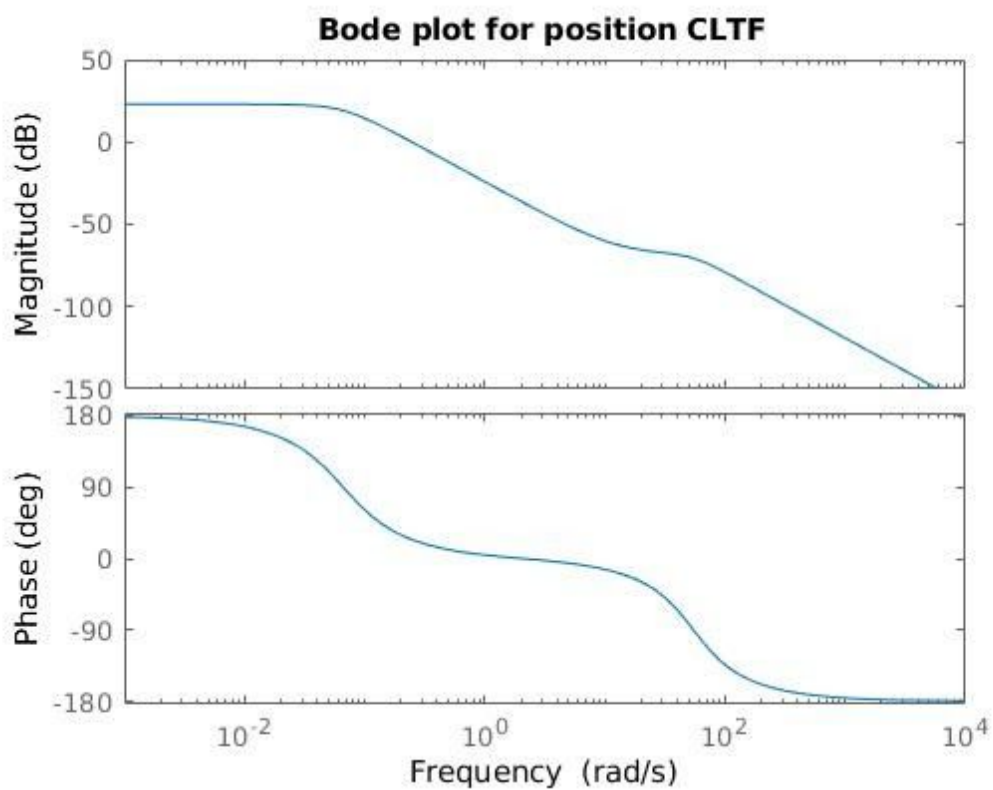
In the last lab, we obtained a feedback gain matrix K using infinite-horizon LQR. To calculate the closed state space system, we replace ' A ' by ' $A - B*K$ ', with everything else remaining the same. Then to find transfer functions we use the `ss2tf` command to derive the transfer functions for angle and position by setting $C = [1,0,0,0; 0,0,1,0]$

`cltf_x =`

$$\frac{1.105 s^2 - 183}{s^4 + 72.78 s^3 + 2968 s^2 + 305.9 s + 12.94}$$

`cltf_theta =`

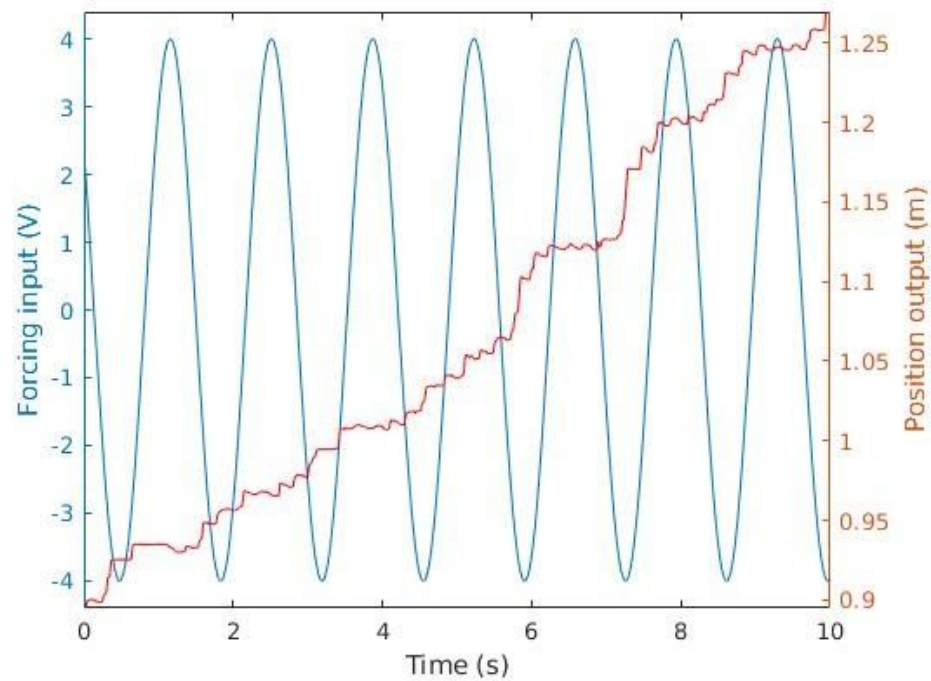
$$\frac{18.66 s^2 + 8.285e-15 s + 2.071e-15}{s^4 + 72.78 s^3 + 2968 s^2 + 305.9 s + 12.94}$$

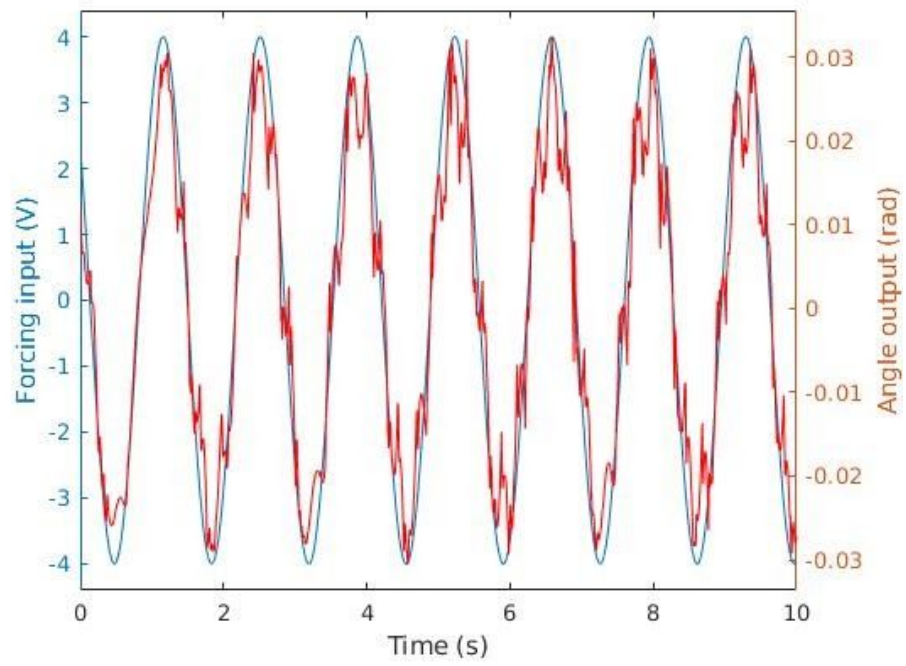


Question 2: Sinusoidal Forcing plots

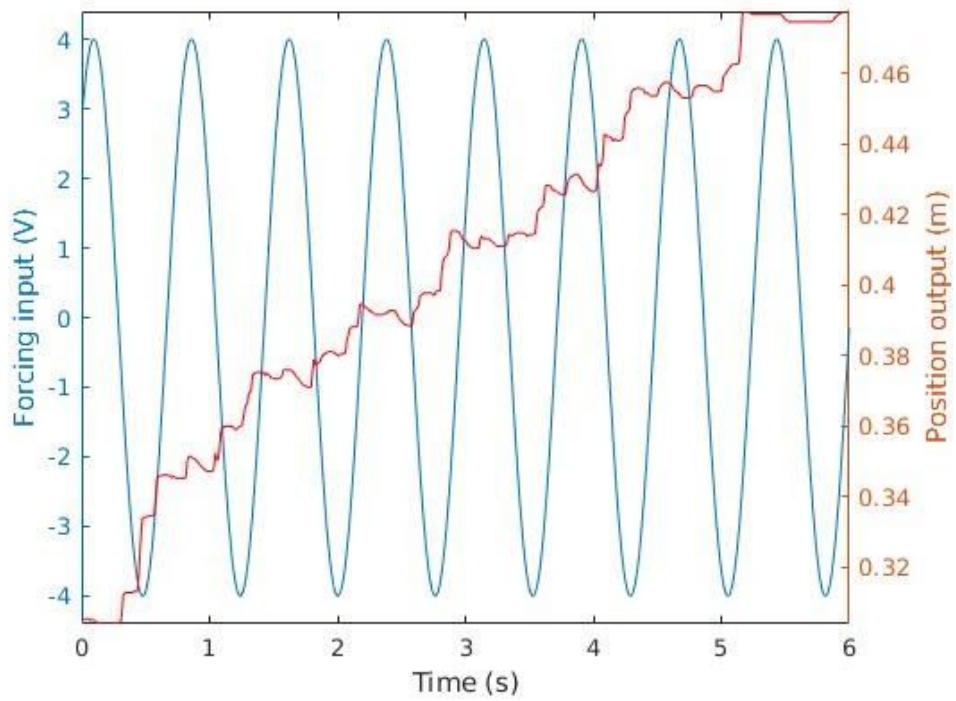
While some sinusoidal behavior was apparent on hardware for the angle at an amplitude $> 2V$, in order to observe sinusoidal motion in position, amplitude was set to $4V$.

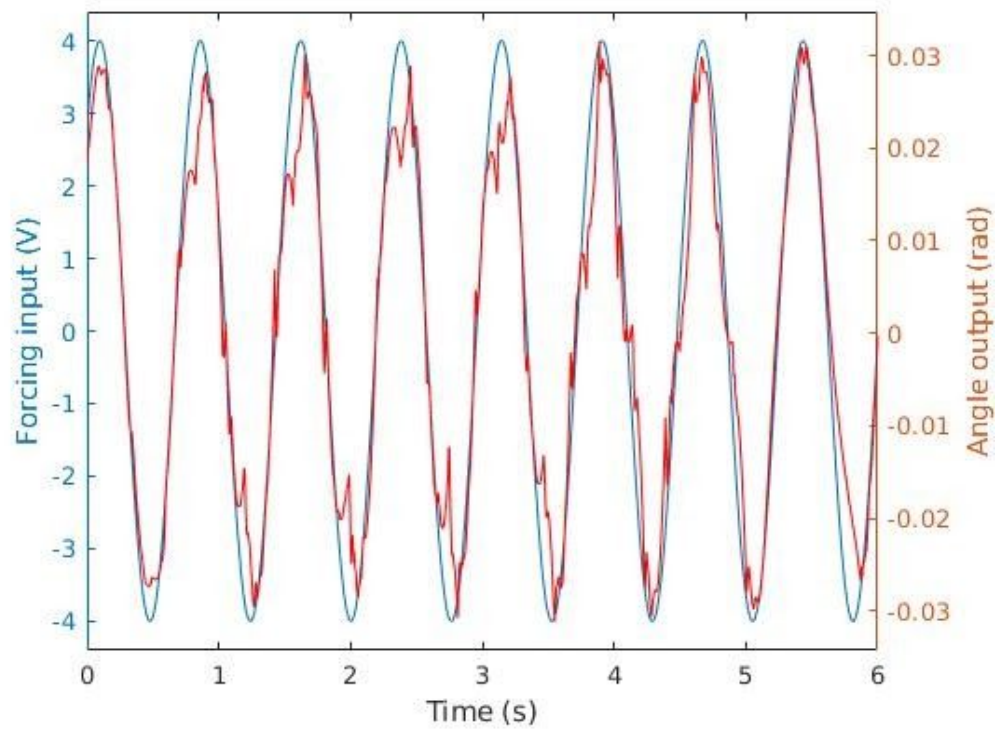
1) Freq = 0.3 Hz



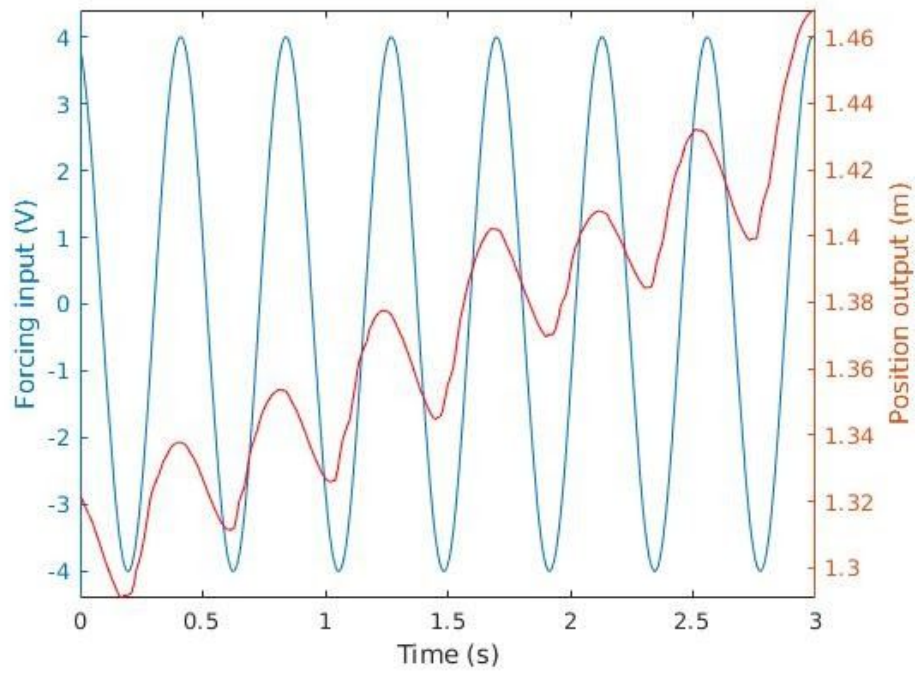


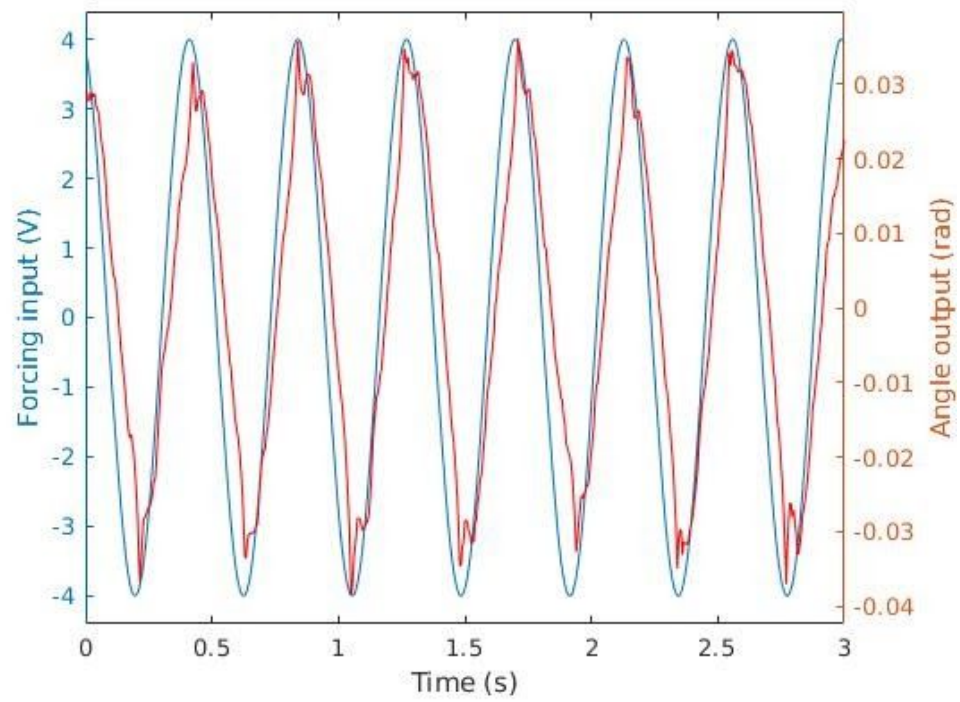
2) Freq = 0.5335 Hz



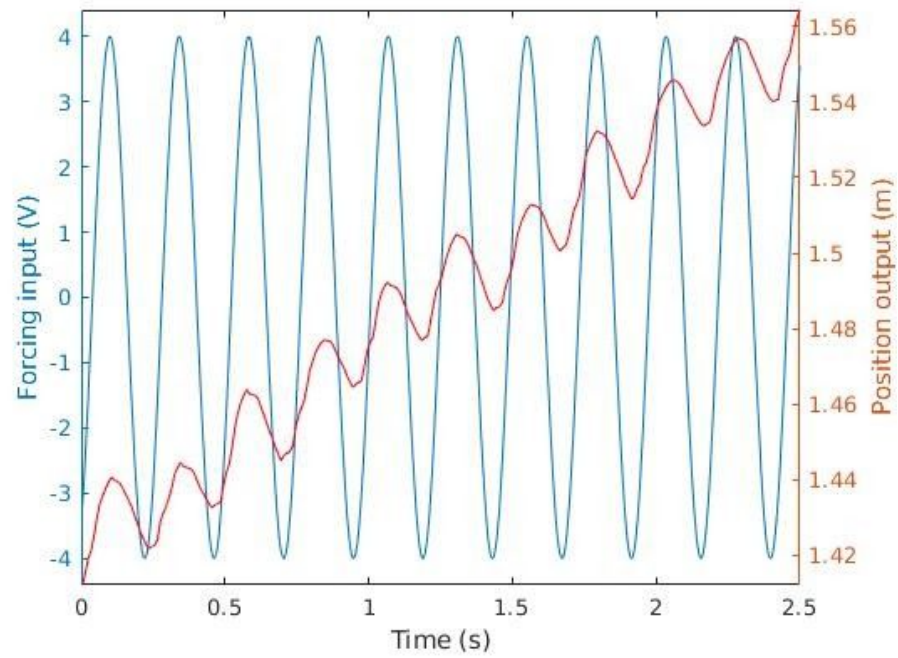


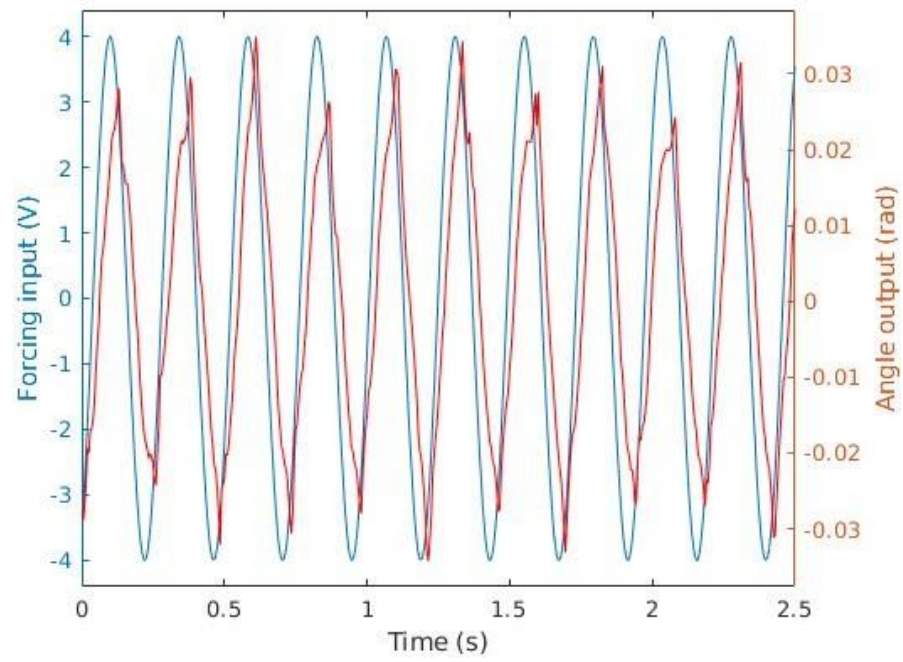
3) Freq = 0.9487 Hz



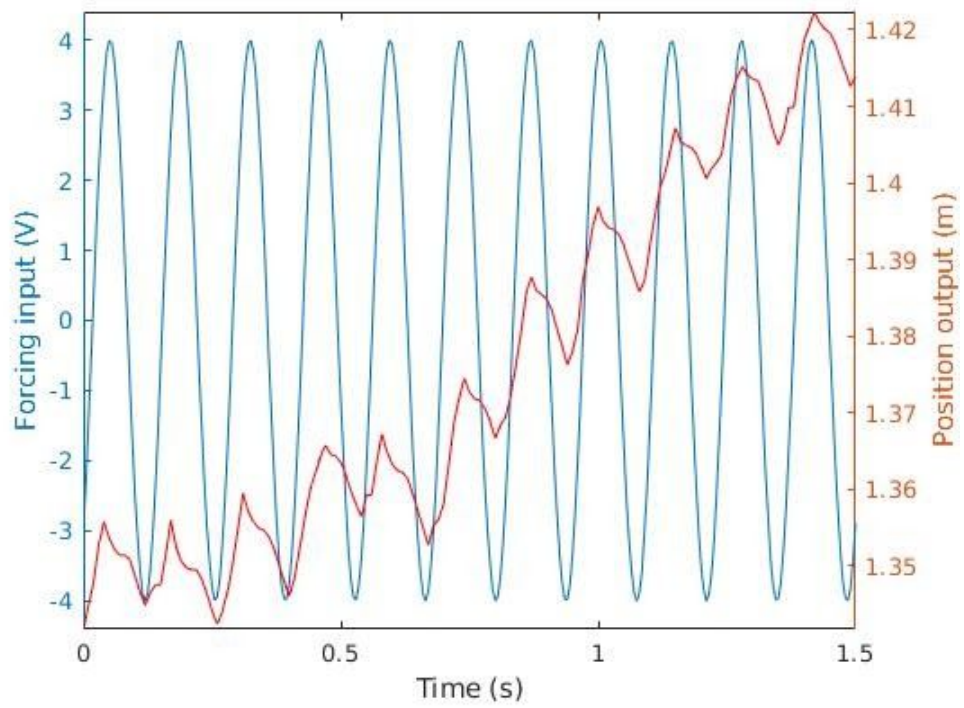


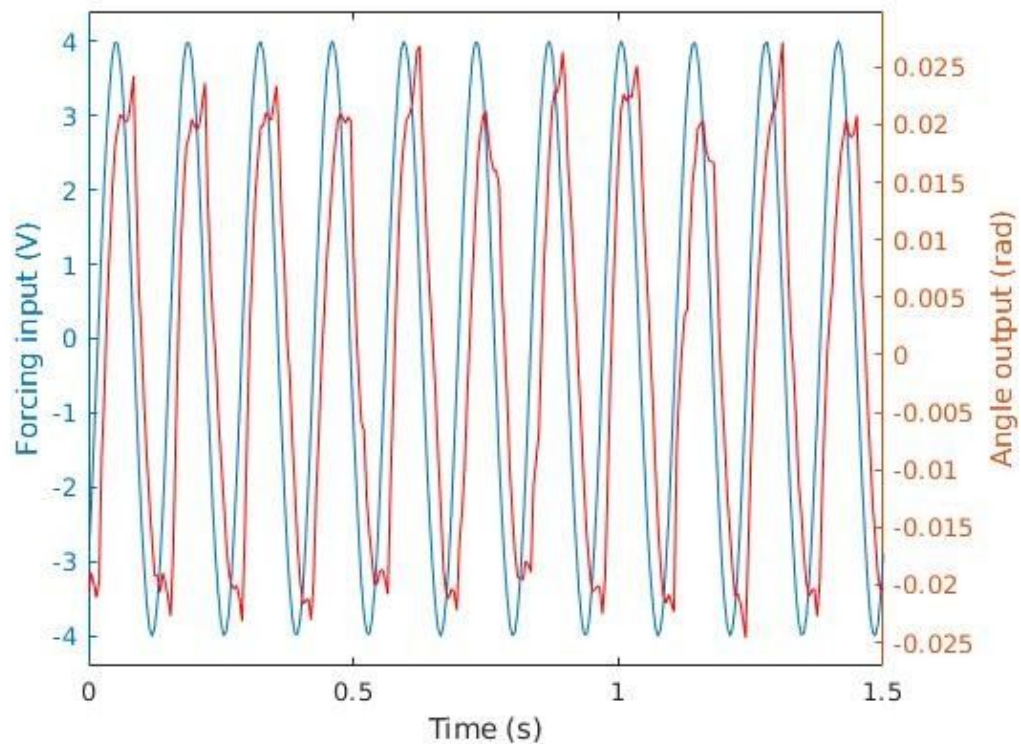
4) Freq = 1.687 Hz





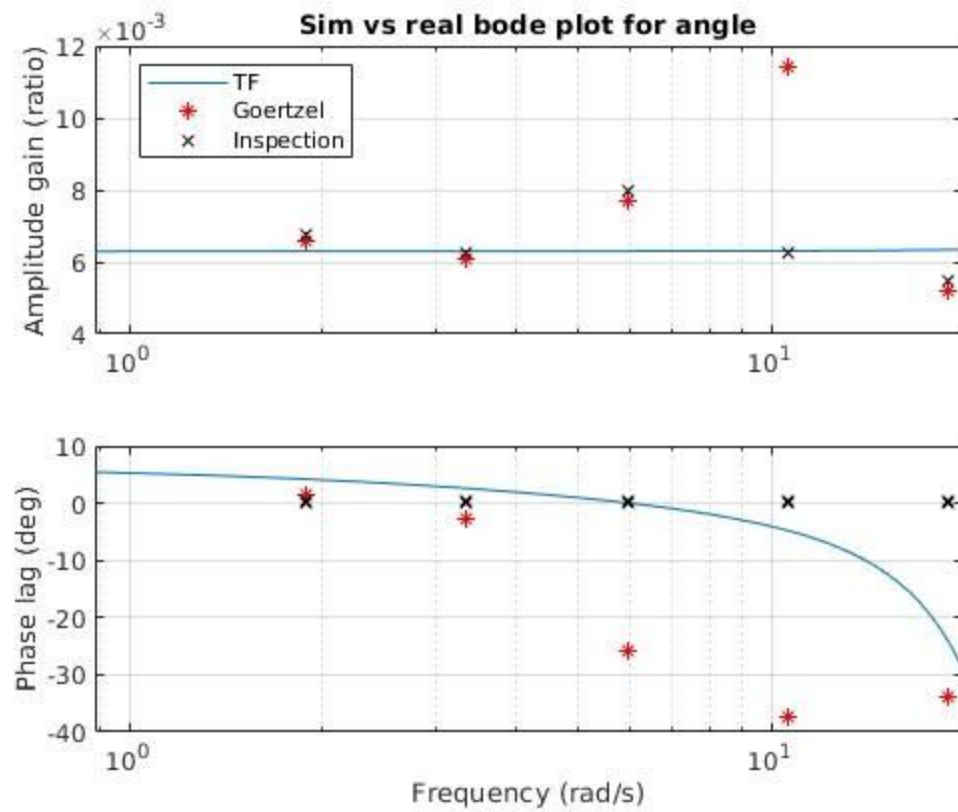
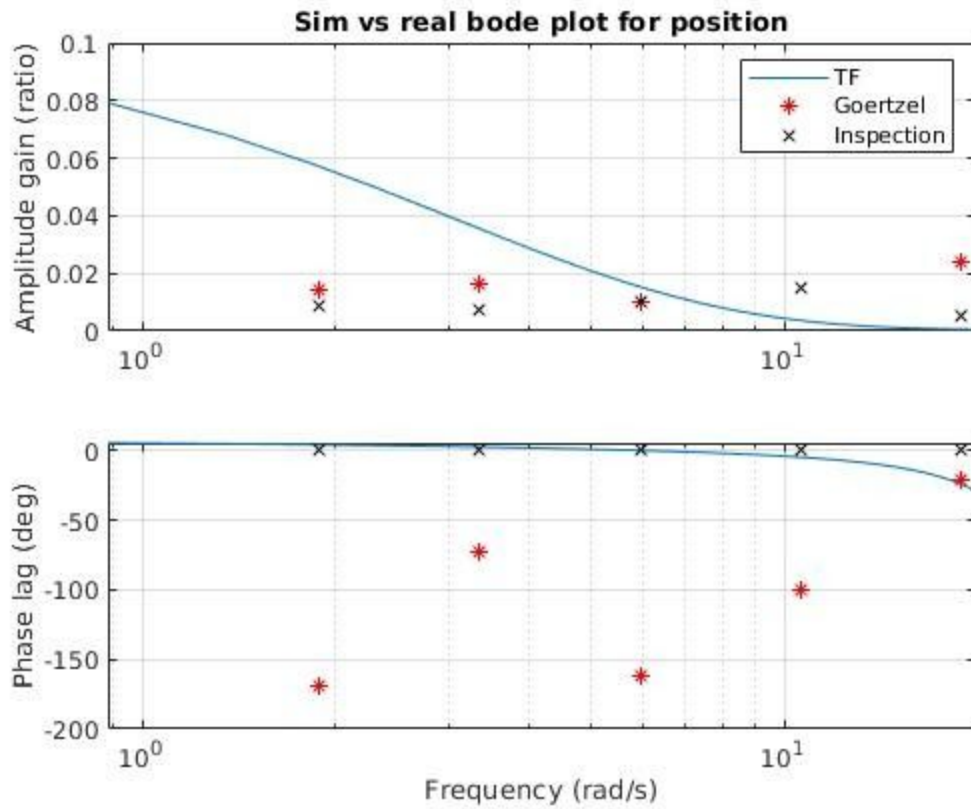
5) Freq = 3 Hz





Question 3 & 4: Comparison with closed loop TF

Based on the plots shown above, we can eyeball the approximate steady state amplitude gain and phase lag between the forcing signal and outputs. Additionally, instead of tedious visual inspection, we can use Goertzel command to find the frequency response of the closed loop system at the forcing frequency. For each forcing frequency f , the DFT is computed and the amplitude and phase are queried at f using Goertzel's algorithm. Code is given in appendix. The figures below show the comparison b/w the theoretical transfer function and the hardware experiments.



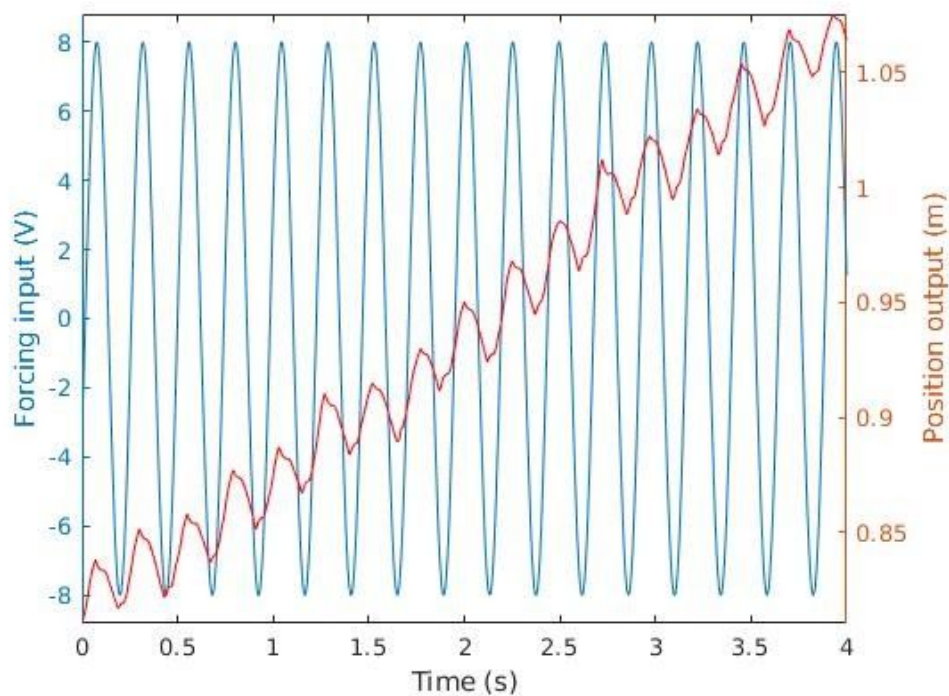
We can observe that in the case of angle, there is a better match between the Goertzel values and visual inspection, mainly because the sinusoids are more visually apparent. Additionally, the position measurement data isn't very reliable since the encoders are single channel and thus measuring direction has to be approximated using the pwm signals.

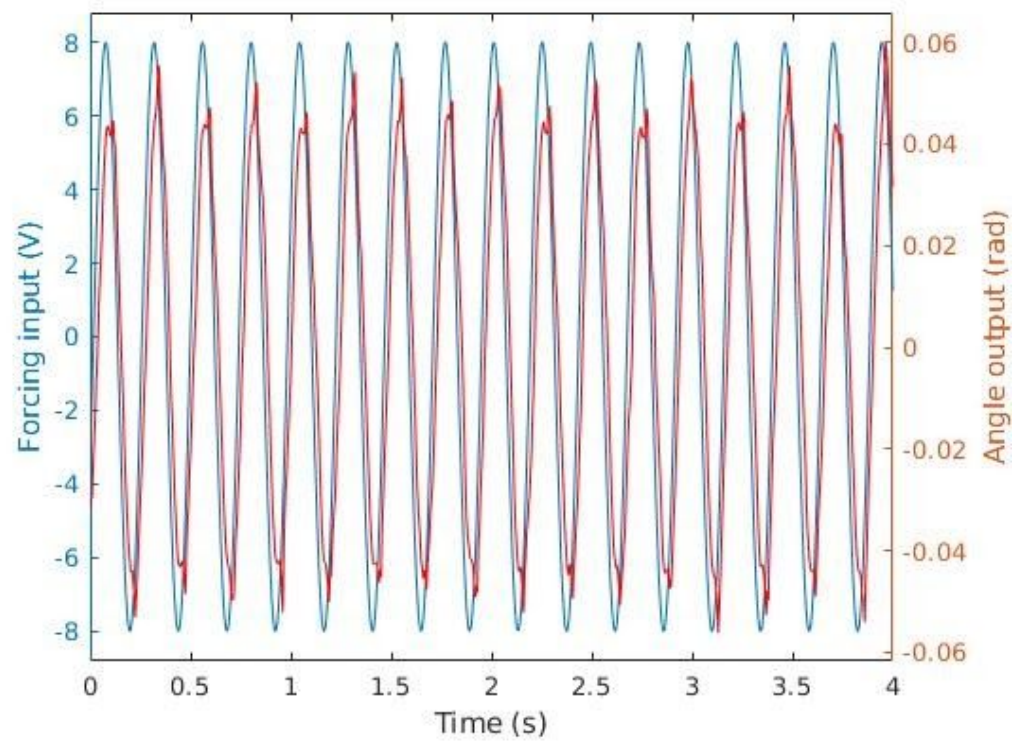
As for how the nominal TF model compares, there are certain mismatches due to imperfect model information, state estimation noise, etc. but overall the values are in the same ballpark and exhibit similar trends.

Question 5: Double the forcing frequency

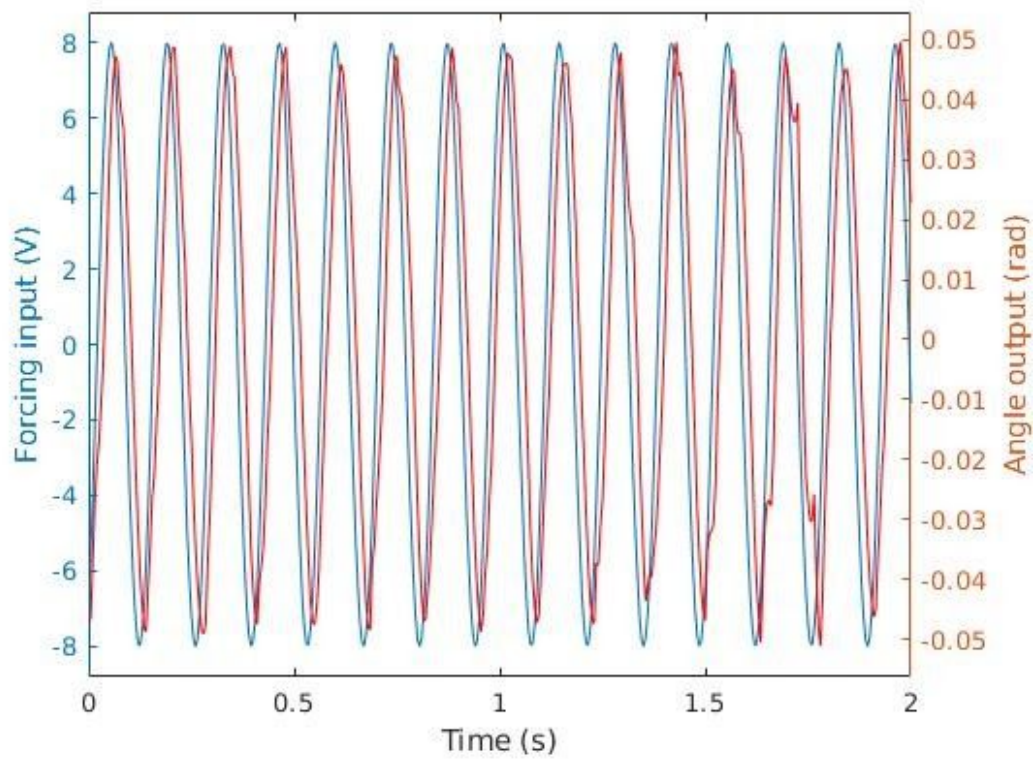
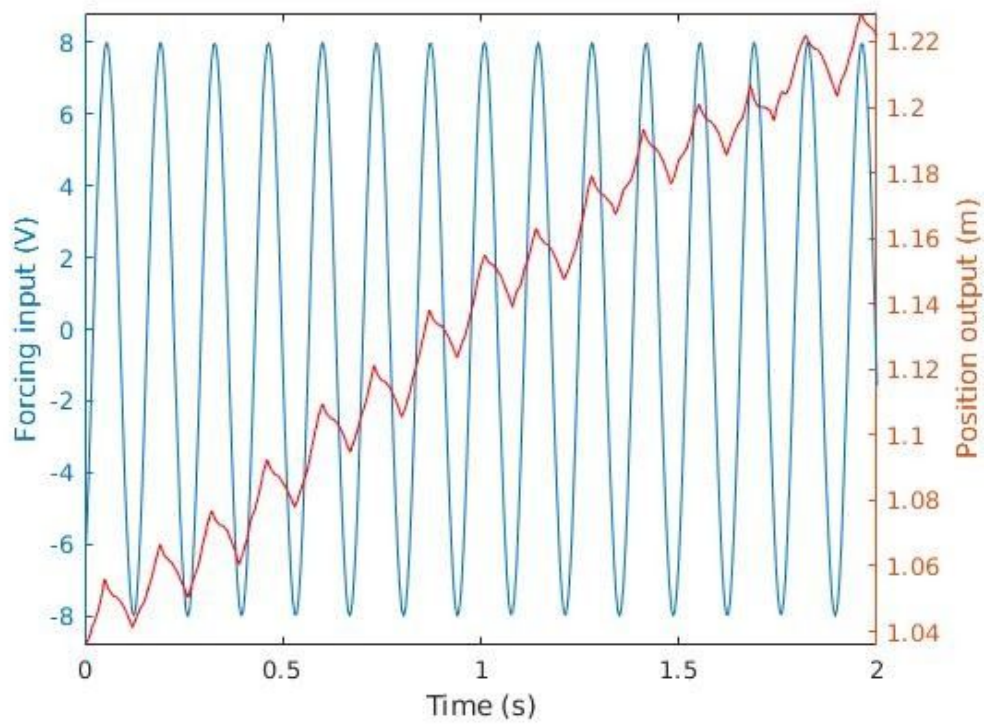
A forcing input of amplitude 8 V was applied at two different frequencies. Plots are shown below -

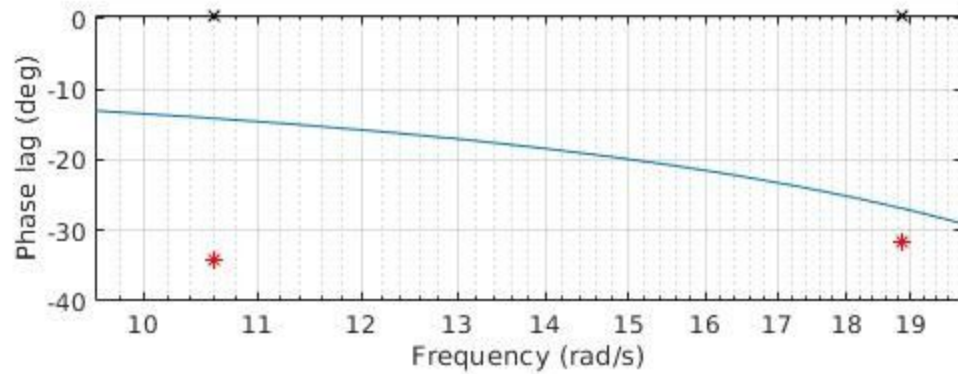
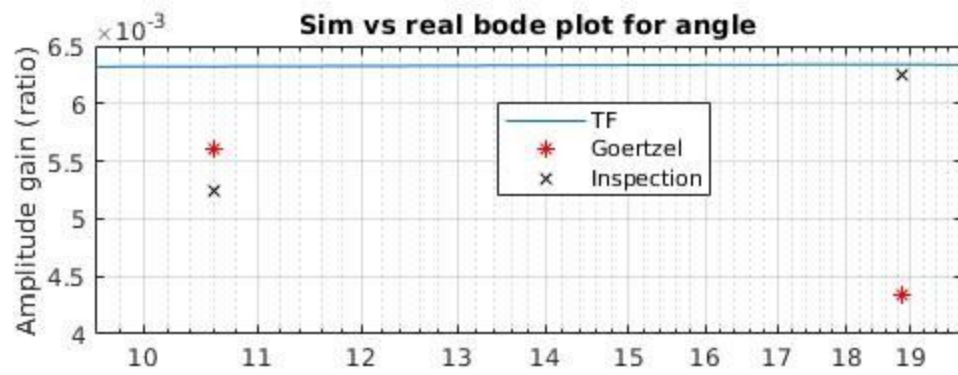
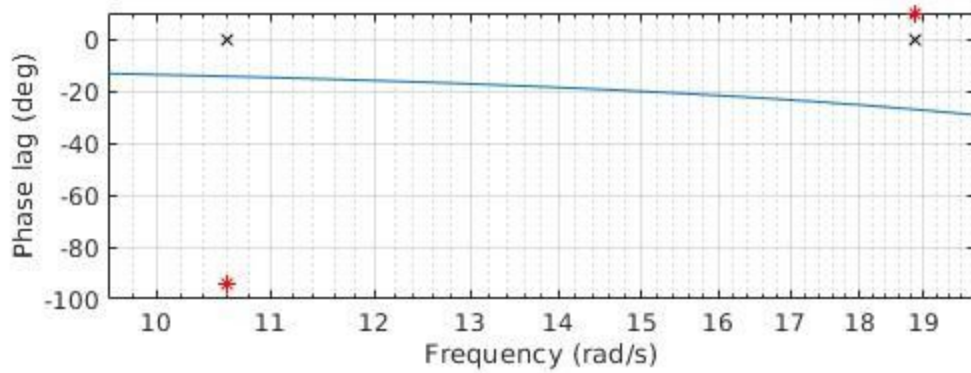
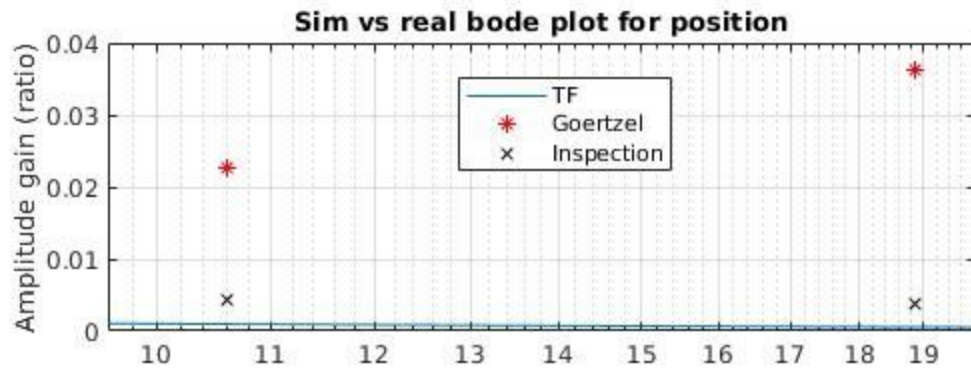
1) Freq = 1.687 Hz





2) Freq = 3 Hz





For the system to be linear, the amplitude gains would have to be the same as in the 4V case. For both position and angle, we can observe that the gains are fairly different, but not drastically changed from the 4V case. We can infer that there are some nonlinear effects at play at this frequency range, but our approximations may still serve the purpose.

B - System ID Toolbox

A white noise `random()` input was added to the control with uniform distribution between -8 and 8 V. The output angle and position readings were then processed in the system ID toolbox in MATLAB. After tuning the order of the state space / poles and zeros of TF, the following models were estimated.

State space model:

ssFin =

$$dx/dt = A x(t) + B u(t) + K e(t)$$

$$y(t) = C x(t) + D u(t) + e(t)$$

A =

| | x1 | x2 | x3 | x4 | x5 |
|----|----------|-----------|----------|---------|---------|
| x1 | 0.009201 | -0.009187 | 0.02669 | 0.03212 | 0.04337 |
| x2 | 0.007541 | -0.1984 | 15.6 | -0.1054 | 3.398 |
| x3 | 0.01687 | -15.39 | -0.02917 | -1.419 | -5.389 |
| x4 | 0.1795 | 3.351 | -2.059 | -187.7 | -286.5 |
| x5 | -0.03883 | 1.179 | 0.6943 | 21.36 | -6.743 |

B =

| | u1 |
|----|------------|
| x1 | -1.705e-06 |
| x2 | -0.0002275 |
| x3 | 0.0001204 |
| x4 | 0.01069 |
| x5 | -0.001084 |

C =

| | x1 | x2 | x3 | x4 | x5 |
|----|----------|----------|---------|----------|-----------|
| y1 | -239.2 | -0.01321 | 0.01395 | -0.03834 | -0.009549 |
| y2 | -0.09176 | -2.188 | 0.5973 | -0.02464 | 0.07378 |

D =

| | u1 |
|--|----|
|--|----|

y1 0
y2 0

K =

| | y1 | y2 |
|----|---------|--------|
| x1 | -0.8502 | 0.2618 |
| x2 | 5.888 | -60.03 |
| x3 | 16.34 | 12.5 |
| x4 | 1859 | -370.1 |
| x5 | -477.2 | 501.6 |

Fit to estimation data: [99.88;92.73]% (prediction focus)
FPE: 3.717e-12, MSE: 6.144e-06

Transfer Function for position:

tfPosFin () =

$$\frac{-0.005575 s^4 + 0.002014 s^3 - 0.001516 s^2 + 0.0005495 s - 9.743e-05}{s^5 + 0.04829 s^4 + 0.1699 s^3 + 0.005299 s^2 + 0.0004878 s + 1.447e-14}$$

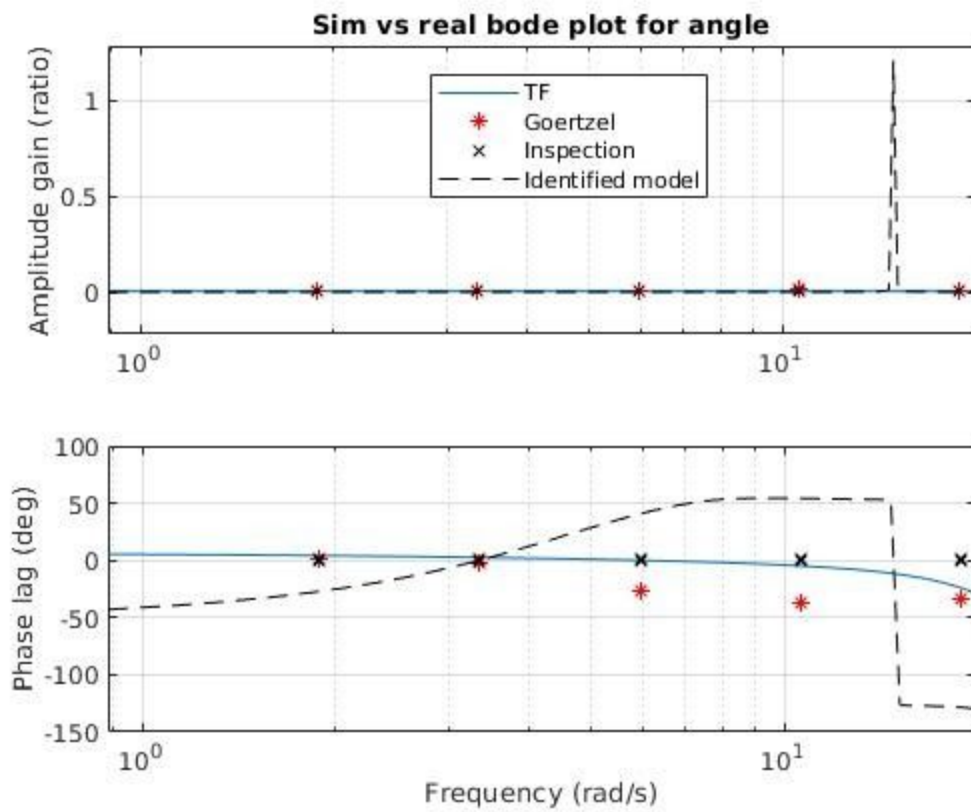
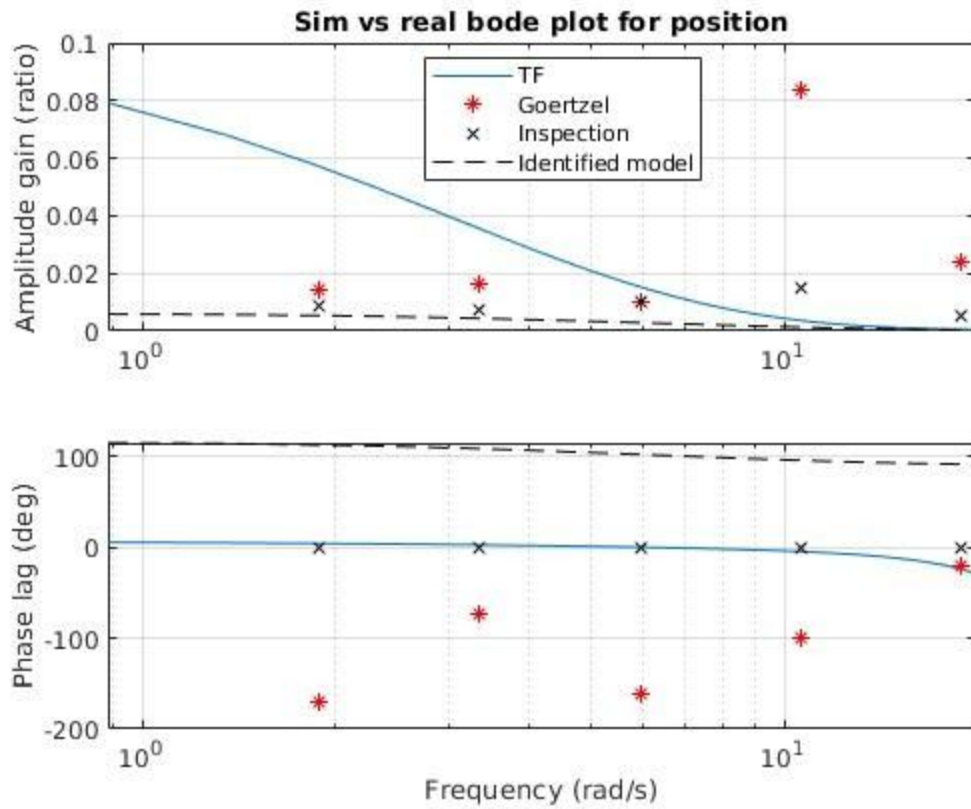
Fit to estimation data: 88.78% (stability enforced)
FPE: 0.006029, MSE: 0.005992

Transfer Function for angle:

tfAngleFin =

$$\frac{0.048 s^2 + 0.1676 s + 0.1709}{s^4 + 34.81 s^3 + 240.6 s^2 + 8365 s + 62.87}$$

Fit to estimation data: 88.81% (stability enforced)
FPE: 1.3e-05, MSE: 1.294e-05



We can observe from the Bode plots that the estimated TFs using white noise don't exactly match the trends that were expected as per the nominal model on which we designed our controller.

There is an unexpected frequency peak in the estimated model for angle, which does not correspond to anything in the nominal model. Additionally, the frequency domain data points are too sparse to judge whether this peak actually exists in the model, or is a consequence of inaccurate curve-fitting. Ideally we would validate white noise system ID using a chirp system ID, or vice versa.

In the case of position, there is an offset in the estimated model from the nominal model. However, it is unclear how much to trust this result given the encoder limitations, which may be generating spurious data.

It is difficult to conclude which method appears to give more accurate results, clearly the controller is easier to design and interpret on the nominal model, which would not be the case with the black-box estimated model. However, more refined system ID with a validation dataset can help us gain more confidence in using estimated system models for designing and updating controllers.

C - Discussion

My main takeaway from this lab was how much scope for improvement there is for improving controller design when we start to get more and more refined measurements about our plant. I was at times caught off guard by how different my measurements were looking compared with predictions from the nominal model. This clearly shows that for high performance and high precision applications, taking a system ID approach is essential, whether online or offline.

That being said, it was also apparent that the controller that was designed on the nominal model performs so well even with significant mismatch with the real system. This is a testament to how control theory lets us do so much while knowing so little about the real world. This motivates me to learn more about adaptive control, model predictive control, etc. - methods that can shield us from our ignorance of the plant.

Contents

- [Init variables](#)
- [Nominal state space model](#)
- [LQR design](#)
- [Closed loop TF](#)
- [Process freq data](#)
- [Frequency domain - Visual inspection vs Goertzel](#)
- [Plots for Bode comparison](#)

```
%Author - Roshan Pradhan
% This script is written for lab3 of 24774 course at CMU

clear
clc
close all

load('lab3_ws.mat')
```

Init variables

```
m_c = 0.493; m_p = 0.312; I_p = 0.00024;
l = 0.04; f = 0.01; k_t = 0.11; R = 10; r = 0.0335; g=9.81;
sample_time = 0.005;
```

Nominal state space model

```
X_ref = [0,0,0.0,0]';

A = [0,1,0,0;
     0, -(I_p + m_p*l^2)*f/( I_p*(m_c+m_p) + m_c*m_p*l^2 ), ...
     m_p^2 * g* l^2/(I_p*(m_c+m_p) + m_c*m_p*l^2), 0;
     0,0,0,1;
     0, -m_p*l*f/( I_p*(m_c+m_p) + m_c*m_p*l^2 ), ...
     m_p*g*l*(m_c + m_p)/( I_p*(m_c+m_p) + m_c*m_p*l^2 ), 0];
B = [0;
     (I_p + m_p*l^2)/( I_p*(m_c+m_p) + m_c*m_p*l^2 );
     0;
     m_p*l/( I_p*(m_c+m_p) + m_c*m_p*l^2 ) * 2*k_t/(R*r);
C = eye(4);
D = zeros(size(C,1), size(B,2));

[n,d] = ss2tf(A,B,C,D);
```

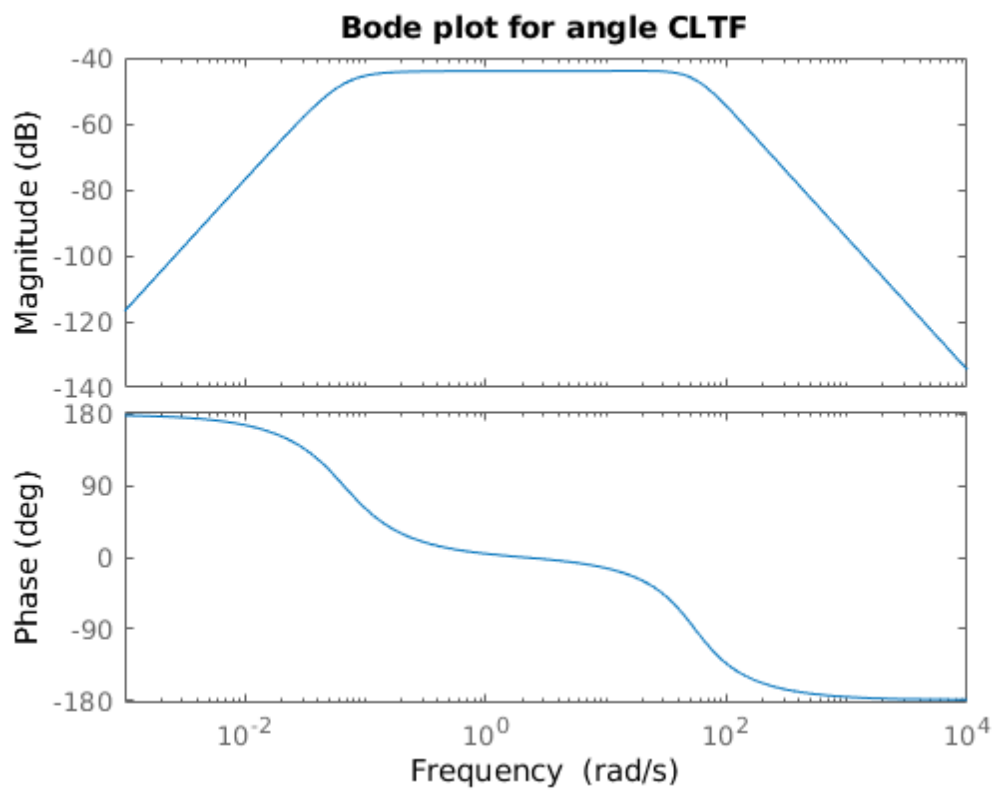
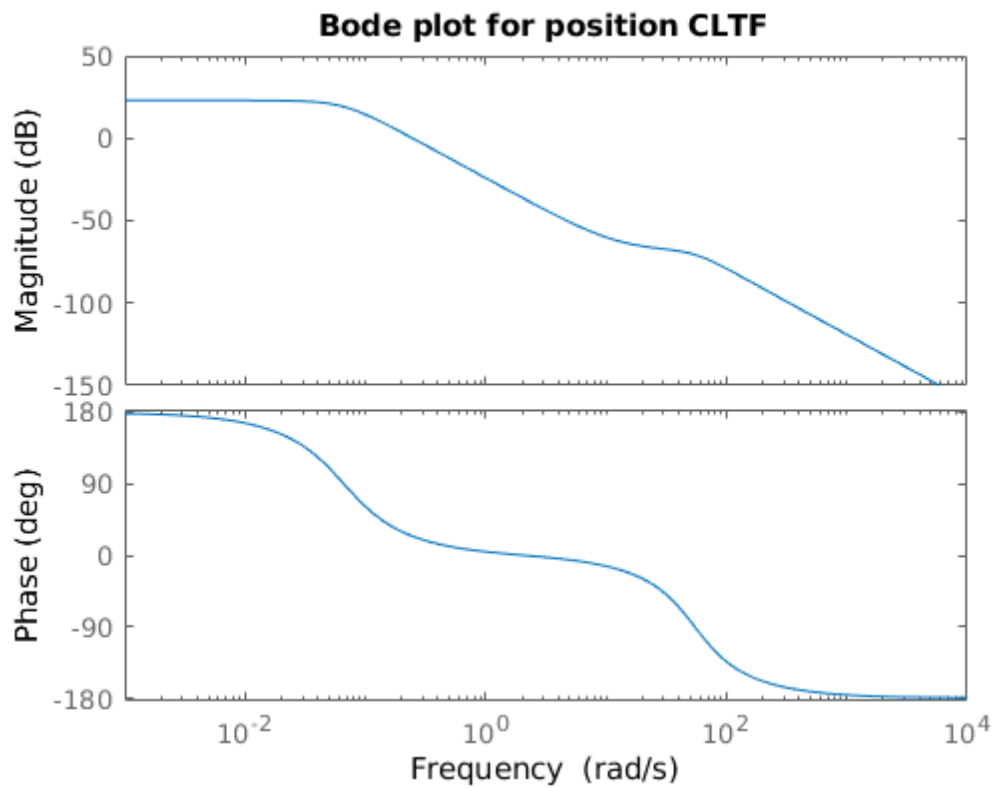
LQR design

```
Q = diag([0.001, 0.1, 5000, 1]);
R = 0.2;

[k,S,CLP] = lqr(A,B,Q,R);
k(4) = 4.0; % reduce gains slightly to account for gyro noise
```

Closed loop TF

```
[n_closed, d_closed] = ss2tf(A - B*k, B, C, D);  
  
cltf_X = tf(n_closed(1,:), d_closed);  
cltf_theta = tf(n_closed(3,:), d_closed);  
  
figure(1)  
bode(cltf_X)  
title('Bode plot for position CLTF')  
  
figure(2)  
bode(cltf_theta)  
title('Bode plot for angle CLTF')
```



Process freq data

```
time_real = 0:0.005:2;  
figure(3)  
xlabel('Time (s)')
```

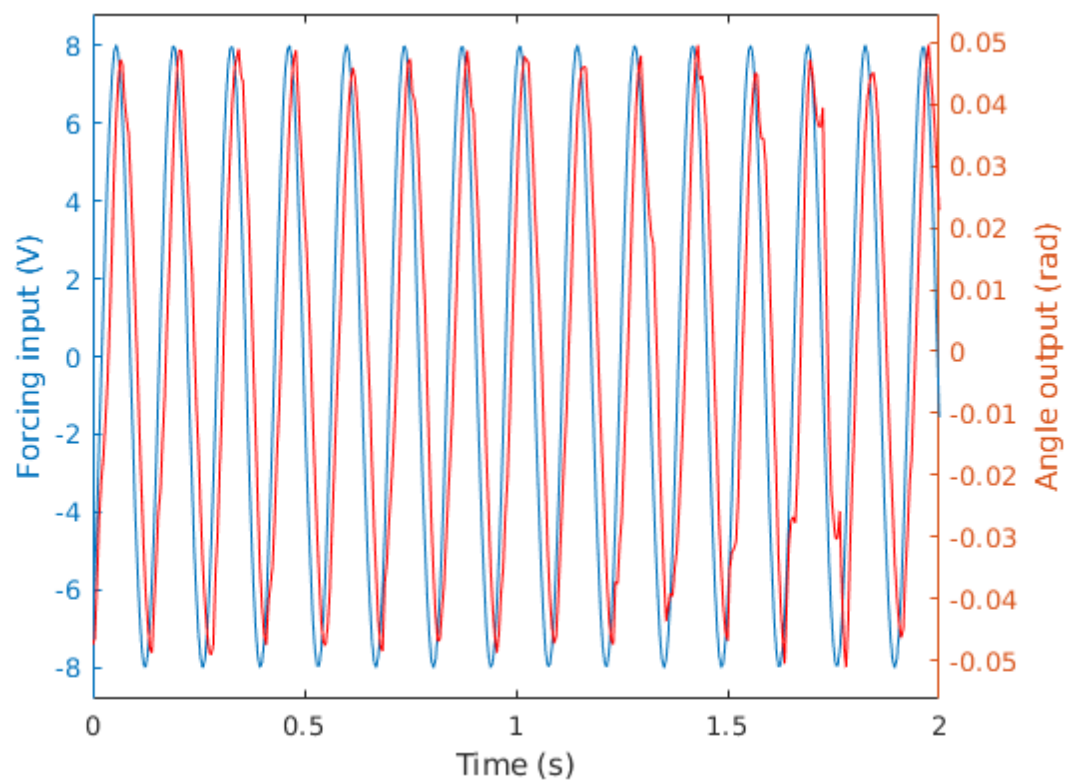
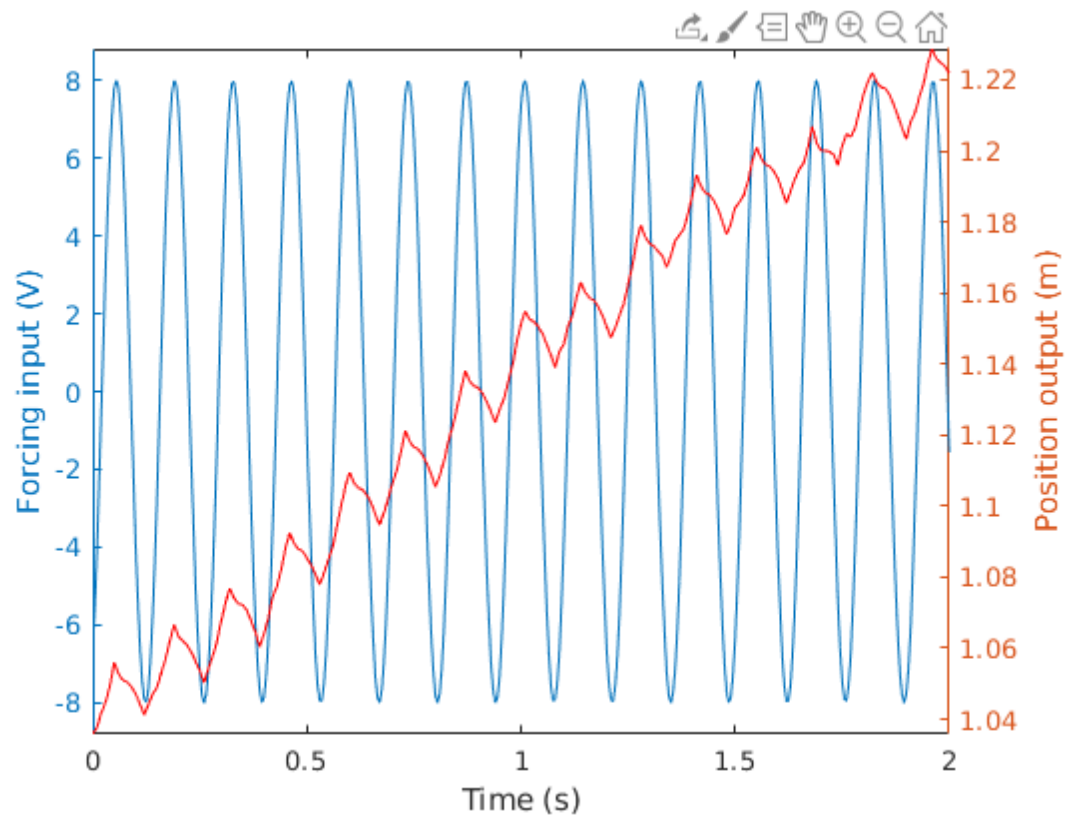


```
yyaxis left
plot(time_real, -freq4twice(1:401,1))
set(gca, 'YLim', [min(freq4twice(1:401,1))*1.1, max(freq4twice(1:401,1))*1.1]);
ylabel(gca, 'Forcing input (V)')

yyaxis right
plot(time_real, freq4twice(1:401, 3), 'r-')
set(gca, 'YLim', [min(freq4twice(1:401,3)), max(freq4twice(1:401,3))]);
ylabel(gca, 'Position output (m)')

figure(4)
xlabel('Time (s)')
yyaxis left
plot(time_real, -freq4twice(1:401,1))
set(gca, 'YLim', [min(freq4twice(1:401,1))*1.1, max(freq4twice(1:401,1))*1.1]);
ylabel(gca, 'Forcing input (V)')

yyaxis right
plot(time_real, freq4twice(1:401, 4), 'r-')
set(gca, 'YLim', [min(freq4twice(1:401,4))*1.1, max(freq4twice(1:401,4))*1.1]);
ylabel(gca, 'Angle output (rad)')
```



Frequency domain - Visual inspection vs Goertzel

```
freqs = 2*pi* logspace(log10(0.3), log10(3), 5); % in rad/s
% visual inspected values
```

```

amp_gain_angle = [0.027/4, 0.025/4, 0.032/4, 0.025/4, 0.022/4];
amp_gain_pos = [0.035/4, 0.028/4, 0.04/4, 0.03/2, 0.01/2];
phase_angle = [pi/2, pi/2, pi/2+0.1745, pi/2+0.35, pi/2+0.5] - pi/2*ones(5,1);
phase_pos = [0, 0, -5, 0, 5]*pi/180;

% Goerztel script
Fs = 66.67;
L = 1401;
logs = zeros(L, 4, length(freqs));
logs(:,:,1) = freq0(1:L,:);
logs(:,:,2) = freq1(1:L,:);
logs(:,:,3) = freq2(1:L,:);
logs(:,:,4) = freq3(1:L,:);
logs(:,:,5) = freq4(1:L,:);

fin_gainX = zeros(1,length(freqs));
fin_gainTheta = zeros(1,length(freqs));
fin_phaseTheta = zeros(1,length(freqs));
fin_phaseX = zeros(1,length(freqs));

for i = 1:length(freqs)
    log_temp = logs(:,:,i);
    fid = round( freqs(i)*L/(2*pi*Fs))+1;
    U = goertzel(-log_temp(1:L,1)', fid);
    pos = goertzel(log_temp(1:L,3)', fid);
    angleTemp = goertzel(log_temp(1:L,4)', fid);

    fin_gainX(i) = abs(pos/U);
    fin_gainTheta(i) = abs(angleTemp/U);
    fin_phaseX(i) = angle(pos/U);
    fin_phaseTheta(i) = angle(angleTemp/U);
end

```

Plots for Bode comparison

```

% nominal model
[magX,phaseX] = bode(c1tf_X, {freqs(1)-1, freqs(end)+1});
magX = squeeze(magX); phaseX = squeeze(phaseX);
wradX = linspace(freqs(1)-1, freqs(end)+1, length(magX));

%identified model
[sys_magX,sys_phaseX] = bode(tfPosFin, {freqs(1)-1, freqs(end)+1});
sys_magX = squeeze(sys_magX); sys_phaseX = squeeze(sys_phaseX);
sys_wradX = linspace(freqs(1)-1, freqs(end)+1, length(sys_magX));

figure(5)
subplot(2,1,1)
semilogx(wradX, magX)
ylabel('Amplitude gain (ratio)')
title('Sim vs real bode plot for position')
hold on
plot(freqs(1:end),fin_gainX, 'r*', freqs, amp_gain_pos, 'kx')
semilogx(sys_wradX, sys_magX, 'k--')
legend('TF', 'Goertzel', 'Inspection', 'Identified model')
hold off
grid
subplot(2,1,2)
semilogx(wradX, phaseX)

```

```
xlabel('Frequency (rad/s)')
ylabel('Phase lag (deg)')
hold on
plot(freqs(1:end), 180/pi*fin_phaseX, 'r*', freqs, phase_pos, 'kx')
semilogx(sys_wradX, sys_phaseX, 'k--')
hold off
grid

% nominal model
[magTheta, phaseTheta] = bode(cltf_theta, {freqs(1)-1, freqs(end)+1});
magTheta = squeeze(magTheta); phaseTheta = squeeze(phaseTheta);
wradTheta = linspace(freqs(1)-1, freqs(end)+1, length(magTheta));

% identified model
[sys_magTheta, sys_phaseTheta] = bode(tfAngleFin, {freqs(1)-1, freqs(end)+1});
sys_magTheta = squeeze(sys_magTheta); sys_phaseTheta = squeeze(sys_phaseTheta);
sys_wradTheta = linspace(freqs(1)-1, freqs(end)+1, length(sys_magTheta));

figure(6)
subplot(2,1,1)
semilogx(wradTheta, magTheta)
ylabel('Amplitude gain (ratio)')
title('Sim vs real bode plot for angle')
hold on
plot(freqs, fin_gainTheta, 'r*', freqs, amp_gain_angle, 'kx')
semilogx(sys_wradTheta, sys_magTheta, 'k--')
legend('TF', 'Goertzel', 'Inspection', 'Identified model')
hold off
grid
subplot(2,1,2)
semilogx(wradTheta, phaseTheta)
xlabel('Frequency (rad/s)')
ylabel('Phase lag (deg)')
hold on
plot(freqs, 180/pi*fin_phaseTheta, 'r*', freqs, phase_angle, 'kx')
semilogx(sys_wradTheta, sys_phaseTheta, 'k--')
hold off
grid
```

