

MECH 6325 - Computer Assignment 1

Jonas Wagner

2020, September 28

Abstract

In the assignment, a single-input single-output LTI system Transfer Function is identified given the frequency response of the system. An n-dimensional transfer function is defined and data of the the input-output frequency response of the system is used to estimate the parameters by minimizing a cost function defined by the least-squares error. The process is first derived analytically and then implemented and simulated within MATLAB.

Problem Outline

Let the transfer function of an LTI Transfer Function be defined as:

$$G(s) = \frac{B(s)}{A(s)} \quad (1)$$

with $B(s)$ and $A(s)$ defined as:

$$\begin{aligned} B(s) &= \sum_{i=0}^{n-1} b_i s^i = b_0 + b_1 s + b_2 s^2 + \dots + b_{n-1} s^{n-1} \\ A(s) &= 1 + \sum_{i=1}^n a_i s^i = 1 + a_1 s + a_2 s^2 + \dots + a_n s^n \end{aligned} \quad (2)$$

The input-output frequency response, $\hat{G}(j\omega)$ is calculated from the response of the system from sine waves applied to the system at different frequencies:

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix}$$

These measurements can then be used to estimate the actual system Transfer Function, $G(s)$.

1 Problem 1

In order to determine the optimal values for the coefficients of $A(s)$ and $B(s)$, the cost function can be defined as the sum of the squared error of the Transfer Function frequency response. Given that the frequency response of the output can be calculated as

$$A(s)G(s) = B(s) \quad (3)$$

The error between the measured response and the estimate can then be calculated as:

$$\epsilon_i = A(j\omega_i)\hat{G}(j\omega_i) - B(j\omega_i) \quad (4)$$

By substituting in the polynomials from (2) and defining the measured $\hat{G}_i = \hat{G}(j\omega_i)$, the error can be expanded into:

$$\epsilon_i = \hat{G}(j\omega_i) \left(1 + \sum_{i=1}^n a_i s^i \right) - \sum_{i=0}^{n-1} b_i s^i \quad (5)$$

Next, the error can be rewritten in the least squares matrix form as follows:

$$\mathcal{E} = y - H\hat{x} \quad (6)$$

where,

$$y = \hat{G} = [\hat{G}_1 \quad \hat{G}_2 \quad \cdots \quad \hat{G}_N]^T \quad (7)$$

$$\hat{x} = [a_1 \quad a_2 \quad \cdots \quad a_n \quad b_0 \quad b_1 \quad b_2 \quad \cdots \quad b_{n-1}]^T \quad (8)$$

$$H = \begin{bmatrix} -\hat{G}_1 (j\omega_1) & -\hat{G}_1 (j\omega_1)^2 & \cdots & -\hat{G}_1 (j\omega_1)^n & 1 & (j\omega_1) & \cdots & (j\omega_1)^{n-1} \\ -\hat{G}_2 (j\omega_2) & -\hat{G}_2 (j\omega_2)^2 & \cdots & -\hat{G}_2 (j\omega_2)^n & 1 & (j\omega_2) & \cdots & (j\omega_2)^{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\hat{G}_N (j\omega_N) & -\hat{G}_N (j\omega_N)^2 & \cdots & -\hat{G}_N (j\omega_N)^n & 1 & (j\omega_N) & \cdots & (j\omega_N)^{n-1} \end{bmatrix} \quad (9)$$

The cost function used to estimate the coefficients is then defined as:

$$J = \sum_{i=1}^N \epsilon_i^* \epsilon_i = \mathcal{E}^* \mathcal{E} \quad (10)$$

This can then be expanded and simplified as follows:

$$\begin{aligned} J = \mathcal{E}^* \mathcal{E} &= (y - H\hat{x})^* (y - H\hat{x}) \\ &= y^* y - \hat{x}^* H^* y - y^* H \hat{x} + \hat{x}^* H^* H \hat{x} \end{aligned} \quad (11)$$

The minimum conditions for the cost function can then be found by setting $\frac{\partial J}{\partial \hat{x}} = 0$, as shown below:

$$\begin{aligned} 0 &= \frac{\partial J}{\partial \hat{x}} = 0 - y^* H - y^* H + 2\hat{x}^* H^* H \\ 2y^* H &= 2\hat{x}^* H^* H \\ y^* H (H^* H)^{-1} &= \hat{x}^* (H^* H) (H^* H)^{-1} \\ \hat{x} &= (y^* H (H^* H)^{-1})^* \end{aligned} \quad (12)$$

The optimal solution, \hat{x} , is therefore defined as:

$$\hat{x} = (H^* H)^{-1} H^* y \quad (13)$$

2 Problem 2 - Nanopositioner Frequency Response Analysis

The implementation of system identification was done in MATLAB by defining an `lsf()` function that takes in the input and output data along with the filter order. The function then generates an H matrix and uses (13) to calculate the optimal solution. A figure is then created with a bode plot comparing the measured frequency response with the optimal solution. The `lsf()` function and MATLAB script can be found in Appendix A.

2.1 Guessed System Order

Looking at the Bode Plot of the frequency response it seemed as though the system order was 2, given the single peak and -180 deg steady-state phase response. This was implemented and the results for the second order system can be seen in Fig.1.

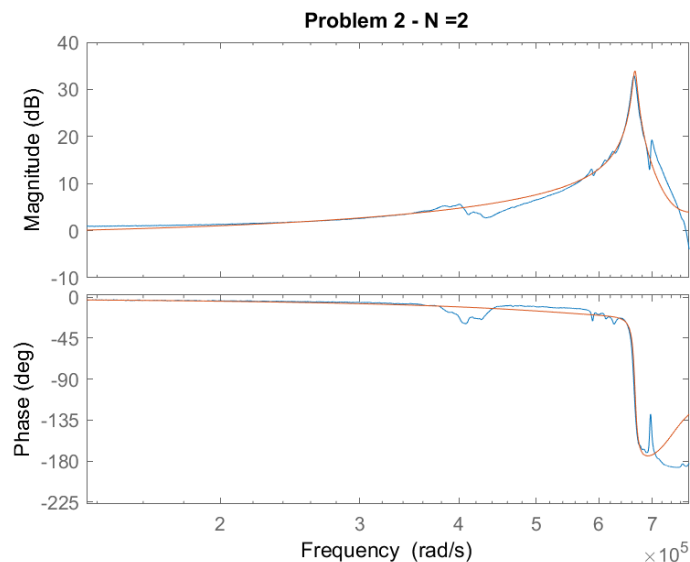


Fig. 1: Bode Plot of the measured response of the nanopositioner versus the second order least-squares estimated system.

2.2 Higher Order System Comparison

The initial estimate appears to be effectively close to the measured data, but doesn't appear to capture all of the features of the response. Higher order systems were then also computed and plotted. Fig.2 shows the response for the third, fifth, and seventh order systems. As can be seen, the third and fifth order estimates appear to align well with the measured data and are generally close fits then the second order systems; however, the seventh order system (and even some other higher order) system responses appear to diverge from the measured data at higher frequencies. Additional orders of estimates can be seen in Appendix B.

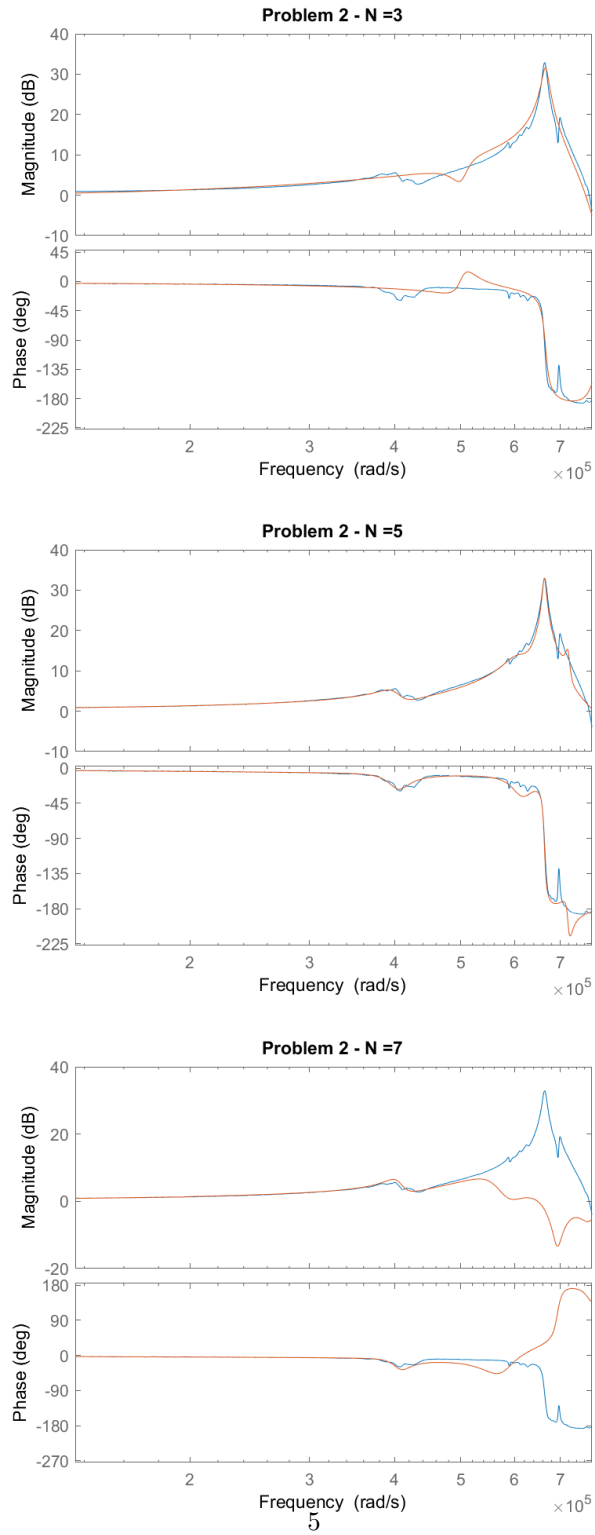


Fig. 2: Bode Plot of the measured response of the nanopositioner versus least-squares estimated systems of higher order.

3 Problem 3: Cantilever Beam Mode Analysis

For analyzing the second set of data, a similar process was used. The same `lsf()` function was used and similar bode plots were used to compare with the measured response.

3.1 Guessed System Order

The bode plot of the measured frequency response appeared to have 3 peaks, 2 troughs, and maintained a negative 20 db/decade at the end, leading me to guess a system order of 7. The seventh order system response can be seen in Fig. 3. This appears to be a good fit to the measured system response (although the phase does get plotted off by 2π for some reason).

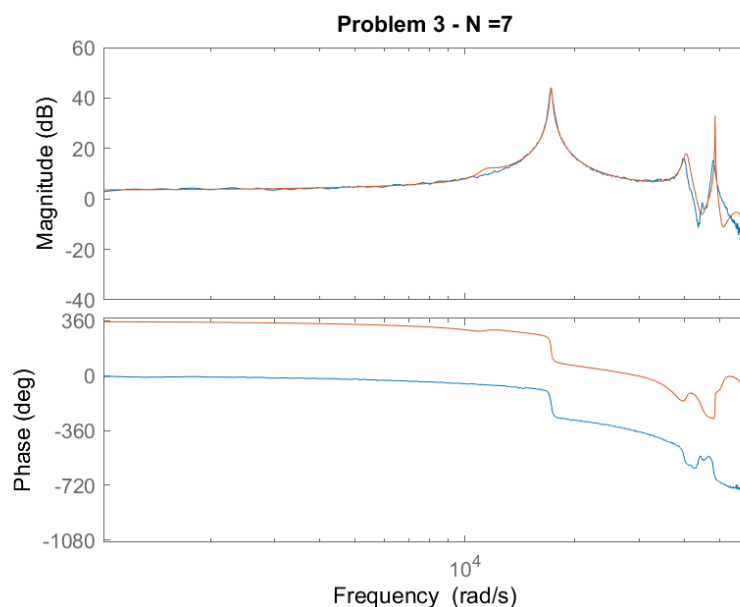


Fig. 3: Bode Plot of the measured response for the cantilever beam versus the seventh order least-squares estimated system.

3.2 Higher Order System Comparison

Similarly to how the higher order systems of the nanopositioners estimates, the fit improved for slightly higher-order systems, but untimely diverged at higher frequencies when the order continued increasing. As shown in the bode plots within Fig. 4, the 8th order system was a better fit than the 7th order, (especially noticeable at higher frequencies), but the 10th and 12th order systems clearly diverge at the higher frequencies. Additional plots can be seen in Appendix B.

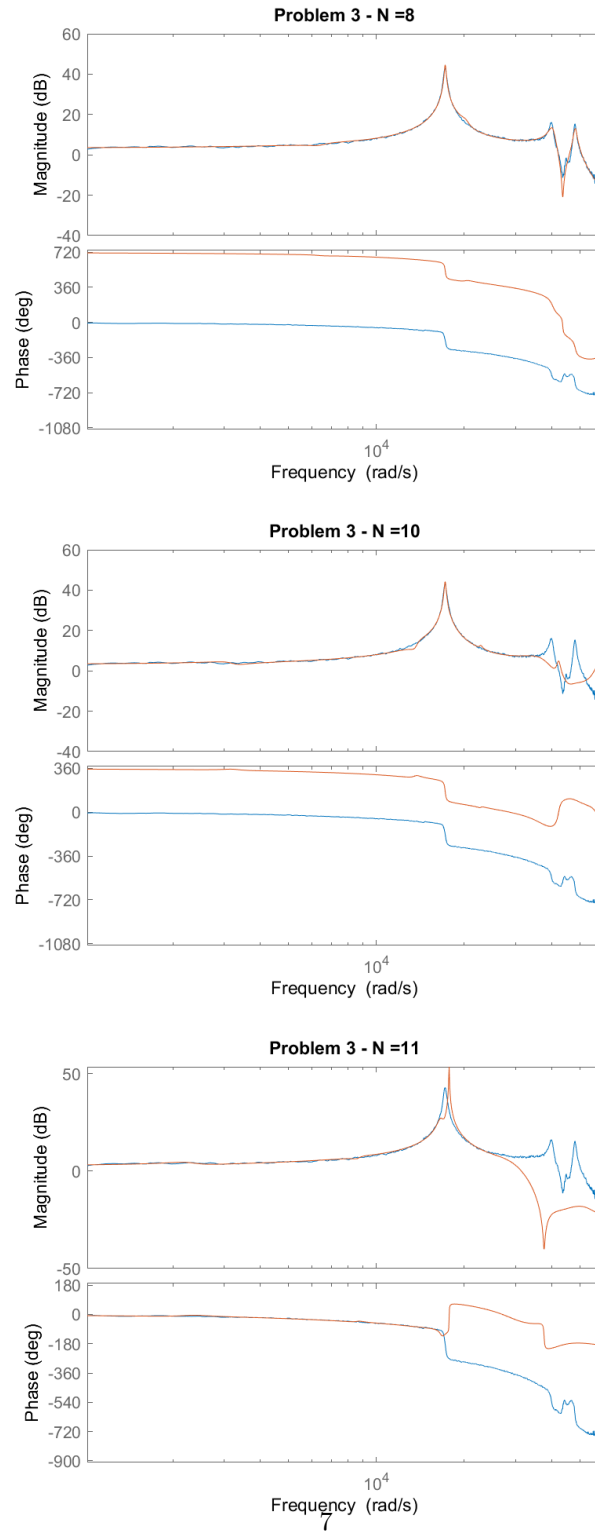


Fig. 4: Bode Plot of the measured response of the cantilever beam versus least-squares estimated systems of higher order.

3.3 Weighting Function Implementation

In order to improve the fit of the least-squares estimate to the actual system response, a method that attempts to eliminate the noise is needed. A weighting function is easy to implement when the uncertainty of each measurement is explicitly known; however, the uncertainty of measurements for this data is not known. It is possible to make the assumption that uncertainty is a function of frequency, however if it is white noise is assumed, this does not exist. One potential method of generating uncertainty is to compare each data point to a windowed mean, however this may actually have the reverse effect and cause the higher frequencies to have a worse fit. Either way, the process of implementing weighted least-squares algorithm remains the same: First, in addition to the definitions from (6), (7), (8), and (9), let the following be defined as:

$$v = [v_1 \quad v_2 \quad \cdots \quad v_N]^T \quad (14)$$

$$R = E[vv^T] = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_N^2 \end{bmatrix} \quad (15)$$

$$\begin{aligned} J &= \mathcal{E}^* R^{-1} \mathcal{E} \\ &= (y - H\hat{x})^* R^{-1} (y - H\hat{x}) \\ &= y^* R^{-1} y - \hat{x}^T H^* R^{-1} y - y^* R^{-1} H \hat{x} + \hat{x}^T H^* R^{-1} H \hat{x} \end{aligned} \quad (16)$$

The optimal solution can then be derived by minimizing the cost function:

$$\begin{aligned} \frac{dJ}{d\hat{x}} &= 0 = -y^* R^{-1} H + \hat{x}^T H^* R^{-1} H \\ H^* R^{-1} y &= H^* R^{-1} H \hat{x} \end{aligned} \quad (17)$$

This results in the optimal solution defined as:

$$\hat{x} = (H^* R^{-1} H)^{-1} H^* R^{-1} y \quad (18)$$

Using this optimal solution, this can then be implemented by modifying the `lsf()` function to include an R matrix that will result in a weighted fit.

3.4 Limited Data Least-Squares

Selecting a subset of data will provide less information for a least-squares estimate to use to create a good fit. A smart selection may make it possible to have an acceptable estimate. In order to test this, the same `lsf()` function was used and provided with various subsets of the data. In general, the fits were not very great and untimely this proved that the best way to fit the data was to ensure that there is enough data provided for regions that are of interest. Two fits that appeared to be fairly acceptable are shown in Fig.5. Depending on the range of operation, these fits may be perfectly acceptable. Each subset contains about 75% of the original response data. That attached

MATLAB script will produce a wide range of bode plots for different subsets of data, but as each subset gets smaller the frequency range it is an acceptable fit for decreases.

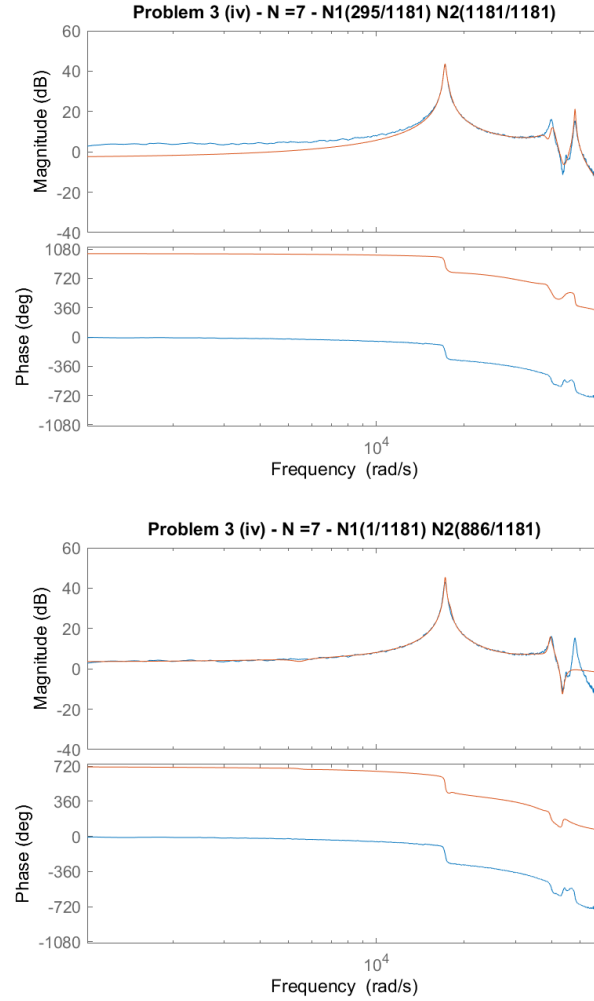


Fig. 5: Frequency response of a least-squares estimate using subsets of data.

A MATLAB Code

A.1 lsf() Function

```
function [H, x, num, den] = lsf(n,N,u,y)
    H = zeros(N,n);
    for i = 1:n
        for k = 1:N
            H(k,i) = -y(k)* (1i * u(k))^i;
        end
    end

    for i = 0:(n-1)
        for k = 1:N
            H(k,i+n+1) = (1i * u(k))^i;
        end
    end

    x = inv(H' * H) * H' * y;

    num = fliplr(x(n+1:end).');
    den = fliplr([1, x(1:n).']);
end
```

A.2 MATLAB Script

```
clear
close all
clc

%-----
%Problem 2

% vector fresp contains (complex-valued) frequency responses
% vector w, contains frequencies at which measurements were taken

load G11QA1.mat
fresp=Yo2i(1:end);

%frequency vector (in Hz)
freq=Yo2ix(1:end);

%frequency vector (in rad/sec)
w = 2*pi*freq;

%non-parametric model
```

```

G_h = frd(fresp,w);

N = length(w);

y = fresp(1:N);
u = w(1:N);

% (i) Order Guess
n = 2;

[~, ~, num, den] = lsf(n,N,u,y);
G = tf(num,den);

figure()
bode(G_h,w)
hold on
bode(G,w)
title("Problem 2 - N =" + string(n))
hold off

saveas(gcf,"fig/MECH6325_CA1_pblm2_n=" + string(n) + ".png")

% (ii) Higher Orders

for i = (n+1):(n+5)
    n = i;
    [~, ~, num, den] = lsf(n,N,u,y);
    G = tf(num,den);

    figure()
    bode(G_h,w)
    hold on
    bode(G,w)
    title("Problem 2 - N =" + string(n))
    hold off
    saveas(gcf,"fig/MECH6325_CA1_pblm2_n=" + string(n) + ".png")
end

close all

%-----
%Problem 3

% vector fresp contains (complex-valued) frequency responses
% vector w, contains frequencies at which measurements were taken

```

```

load G11QA2.mat
fresp=o2i1(20:1200);%frequency response vector

%frequency vector (in Hz)
freq=o2i1x(20:1200);

%frequency vector (in rad/sec)
w = 2*pi*freq;

%non-parametric model
G_h = frd(fresp,w);

N = length(w);

y = fresp(1:N);
u = w(1:N);

% (i) Order Guess
n = 7;

[~, ~, num, den] = lsf(n,N,u,y);
G = tf(num,den);

figure()
bode(G_h,w)
hold on
bode(G,w)
title("Problem 3 - N =" + string(n))
hold off

saveas(gcf,"fig/MECH6325_CA1_pblm3_n=" + string(n) + ".png")

% (ii) Higher Orders

for i = (n+1):(n+5)
    n = i;
    [~, ~, num, den] = lsf(n,N,u,y);
    G = tf(num,den);

    figure()
    bode(G_h,w)
    hold on
    bode(G,w)
    title("Problem 3 - N =" + string(n))
    hold off

```

```

        saveas(gcf,"fig/MECH6325_CA1_pblm3_n=" + string(n) + ".png")
    end

    close all

    % (iv) Subset of data tests

    Nmax = N;

    for i = 0:4
        for j = 0:4
            if i > 0
                N1 = int32(Nmax/i);
            else
                N1 = 1;
            end
            if j > 0
                N2 = Nmax - int32(Nmax/j);
            else
                N2 = Nmax;
            end

            if N2 > N1
                N = N2-N1 +1;

                y = fresp(N1:N2);
                u = w(N1:N2);

                n = 7;
                [~, ~, num, den] = lsf(n,N,u,y);
                G = tf(num,den);

                figure()
                bode(G_h,w)
                hold on
                bode(G,w)
                title("Problem 3 (iv) - N =" + string(n) + " - N1(" + string(N1) + " - N2(" + string(N2) + ")")
                hold off
                saveas(gcf,"fig/MECH6325_CA1_pblm3_iv_n=" + string(n) + "N1(" + string(N1) + "N2(" + string(N2) + ")")
            end
        end
    end
end

```

B Additional Figures

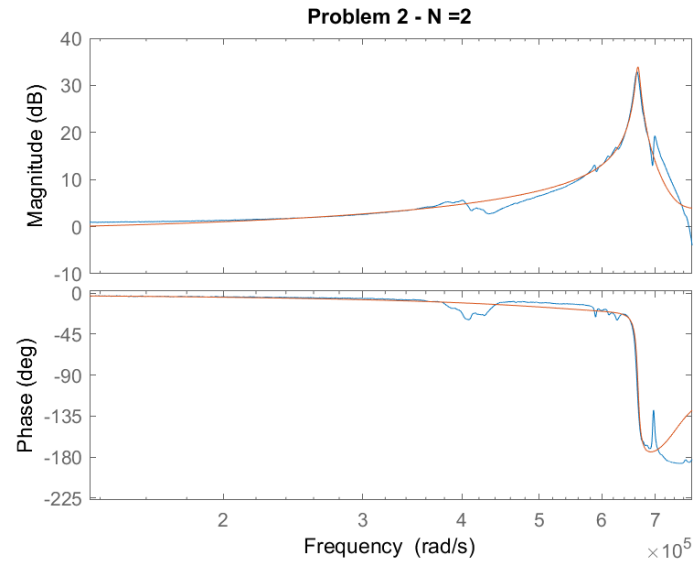


Fig. 6: Bode Plot of the measured response versus the second order least-squares estimated system.

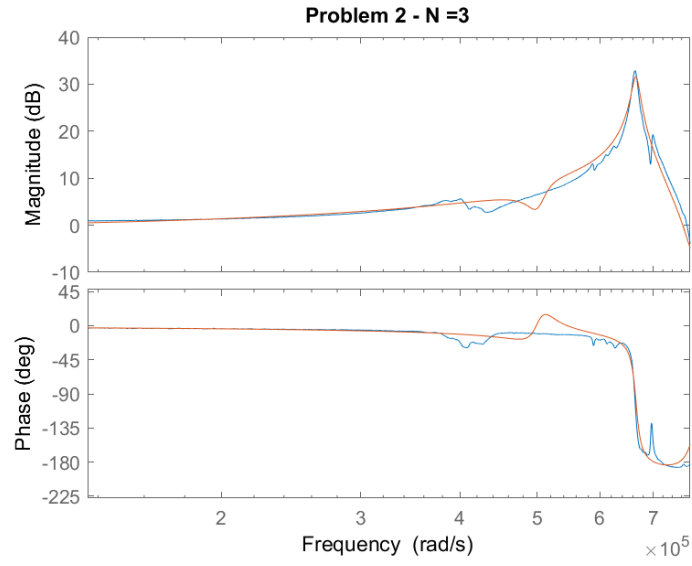


Fig. 7: Bode Plot of the measured response versus the 3rd order least-squares estimated system.

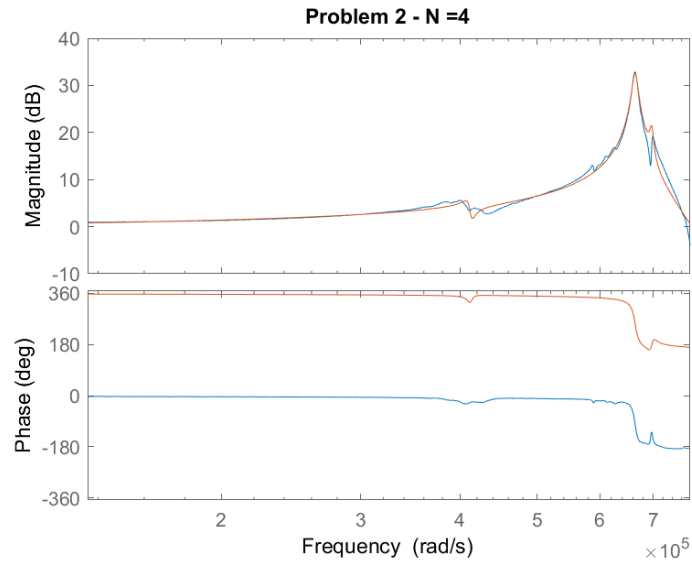


Fig. 8: Bode Plot of the measured response versus the 4th order least-squares estimated system.

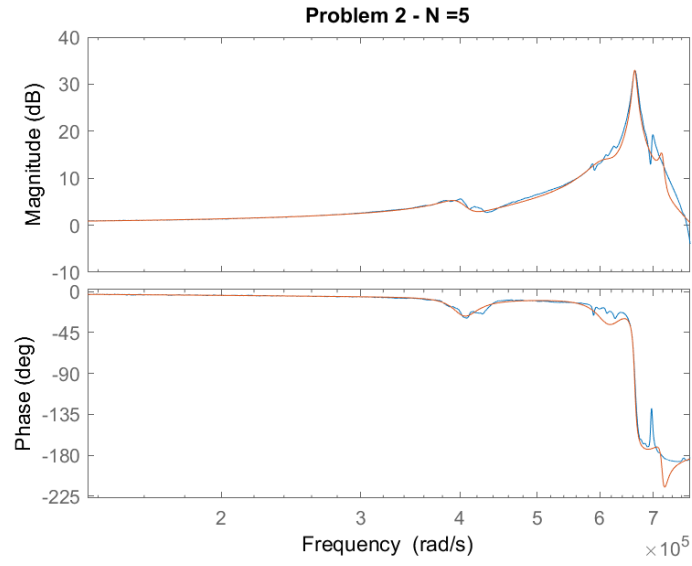


Fig. 9: Bode Plot of the measured response versus the 5th order least-squares estimated system.

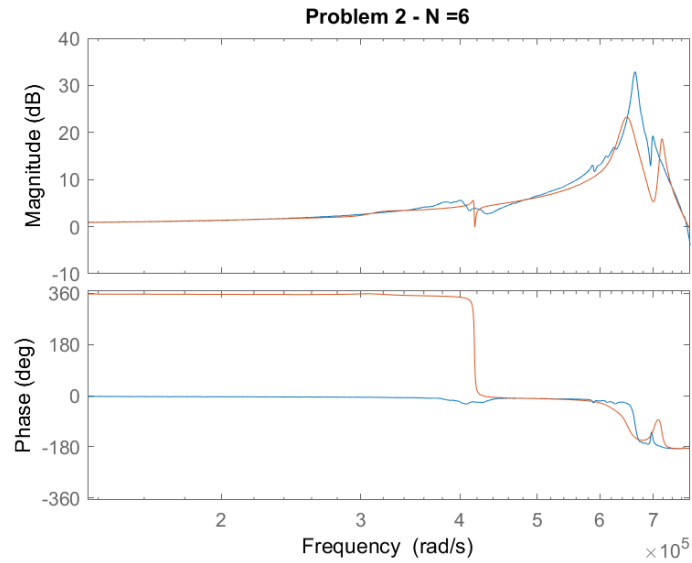


Fig. 10: Bode Plot of the measured response versus the 6th order least-squares estimated system.

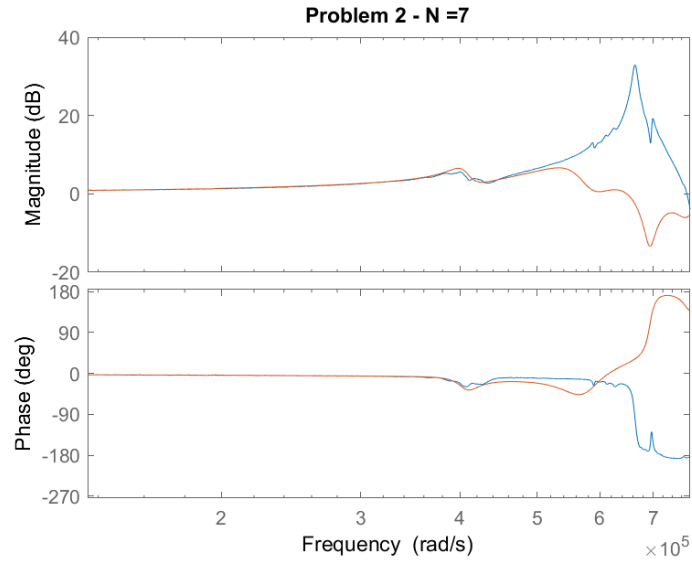


Fig. 11: Bode Plot of the measured response versus the 7th order least-squares estimated system.

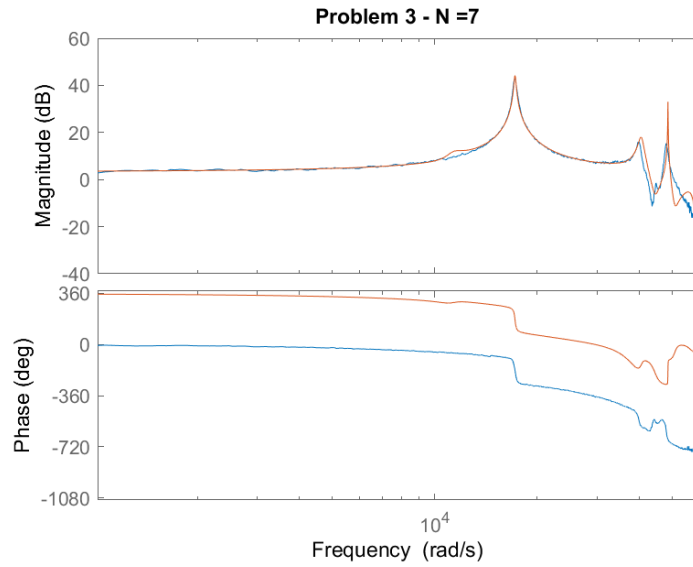


Fig. 12: Bode Plot of the measured response versus the 7th order least-squares estimated system.

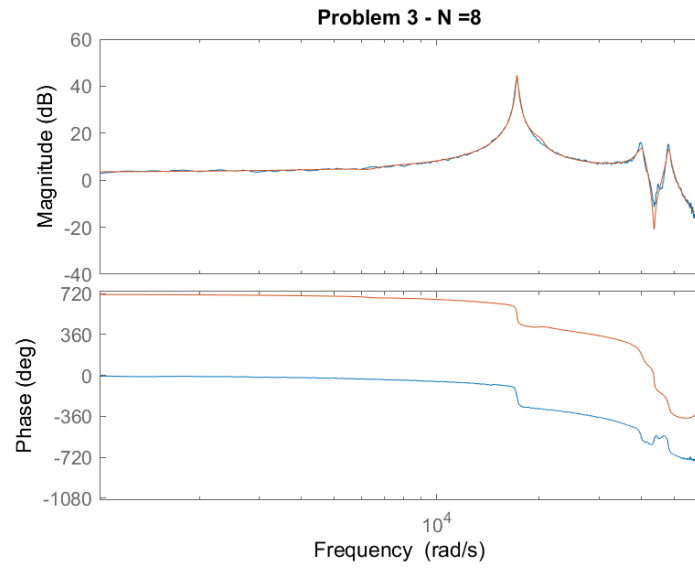


Fig. 13: Bode Plot of the measured response versus the 8th order least-squares estimated system.

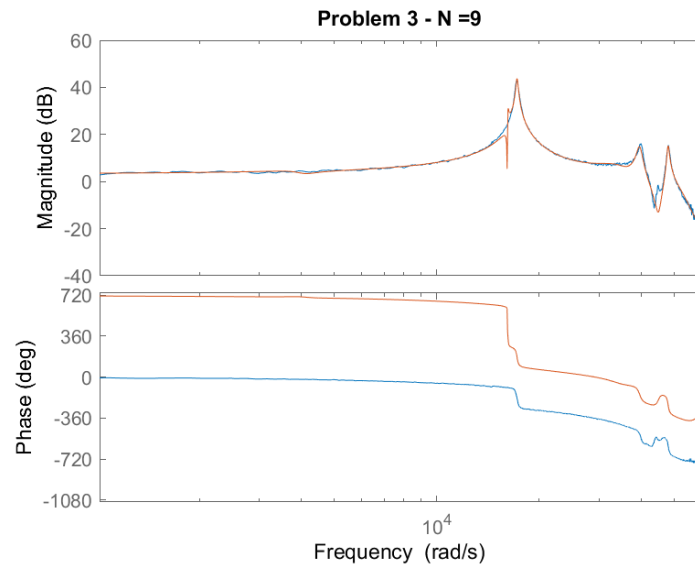


Fig. 14: Bode Plot of the measured response versus the 9th order least-squares estimated system.

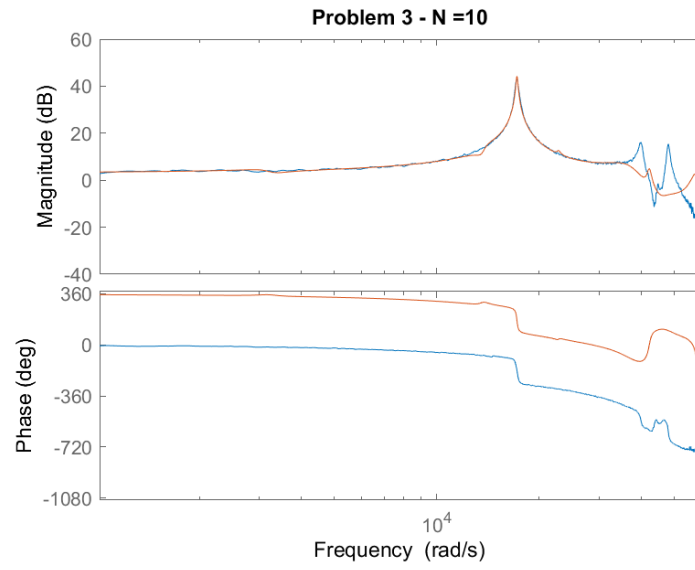


Fig. 15: Bode Plot of the measured response versus the 10th order least-squares estimated system.

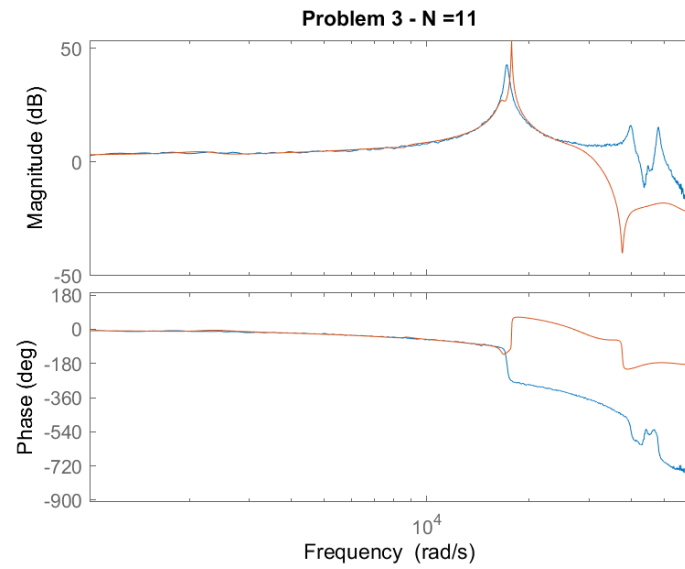


Fig. 16: Bode Plot of the measured response versus the 11th order least-squares estimated system.

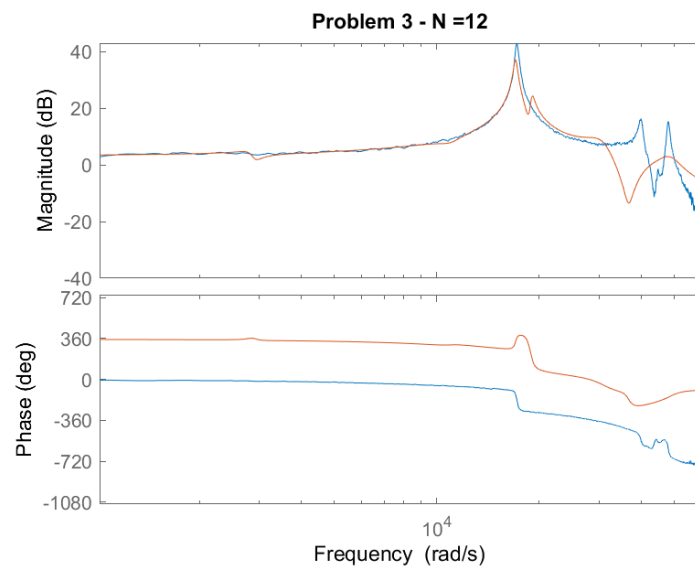


Fig. 17: Bode Plot of the measured response versus the 12th order least-squares estimated system.