System Identification
Final Project: Identification of a Torsional System

Camilla Carta
Michael Spieler

January 11, 2018

1 Introduction

For this project, the goal was to identify a linear parametric model for the system showed in Figure 1. This system consists in a torsional device, that was described in the project assignment as follows:

> *The plant consists of three disks supported by a torsionally flexible shaft which is suspended vertically on anti-friction ball bearings. The shaft is driven by a brushless servo motor connected via a rigid belt (negligible tensile flexibility) and pulley system with a 3 : 1 speed reduction ration. An encoder located on the base of the shaft measures the angular displacement of the first disk. The second disk is connected to its encoder by a rigid belt/pulley with a 1 : 1 speed ratio. The input of the system is the voltage of the servo motor and the output is the position of the third disk.*
>
> *An unknown digital controller is used to control the position of the third disk. A closed-loop experiment has been carried out in which a PRBS signal, r(k), is applied as the reference position and the input to the plant, u(k), as well as the output of the plant, y(k), are measured and saved in FPdata.mat, which is available in Moodle. The sampling period is $T_s = 0.04s$. The objective is to identify a parametric model for the torsional plant (between u and y) and validate it using different methods.*

The plant input signal u(k) consisted in a PRBS of size 6138, with 6 periods and a n-bit shift register of 10 bits. With a sampling period of $T_s = 0.04s$, the system was excited for a period of over 4 minutes.

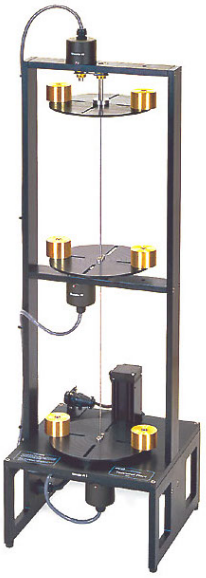
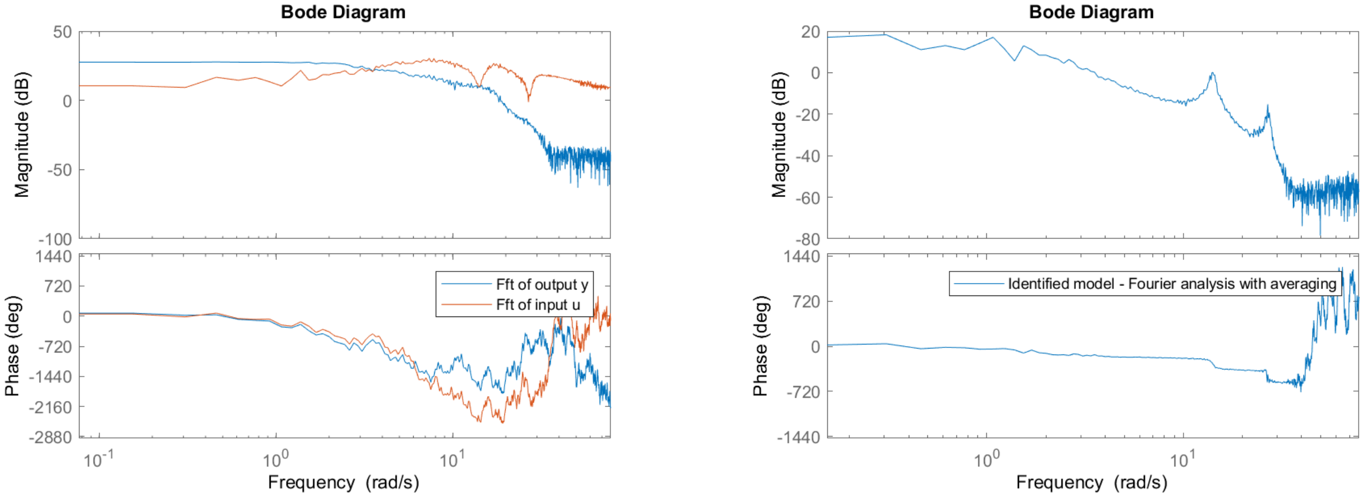


Figure 1: The torsional device to be identified

2 Frequency domain model identification

2.1 Fourier analysis

In order to identify the frequency-domain model of the system, the Fourier analysis was first applied. An averaging was applied to the Fourier Transforms of input and output, as it improves the results' accuracy. The corresponding bode diagram is shown in Figure 2b. Figure 2a shows the bode diagram of the Fourier Transforms for the input and the output respectively. It can be seen that the two resonance frequencies in the model correspond to the resonance frequencies of the input u, while the lowpass effect can be found in the Fourier Transform of the output.



(a) Bode diagram for input and output signals

(b) Bode diagram for the identified model using Fourier analysis and averaging

Figure 2: Fourier Analysis

2.2 Spectral analysis

Next, the spectral analysis was performed. In order to improve the accuracy of the results, averaging was applied as in the course notes at page 56. The estimation for a closed-loop system with excitation at the reference signal r needs to be applied, as shown on page 60 in the coruse notes.

$$G(e^{j\omega_n}) = \frac{\bar{\Phi}_{yr}(\omega_n)}{\bar{\Phi}_{ur}(\omega_n))} \tag{1}$$

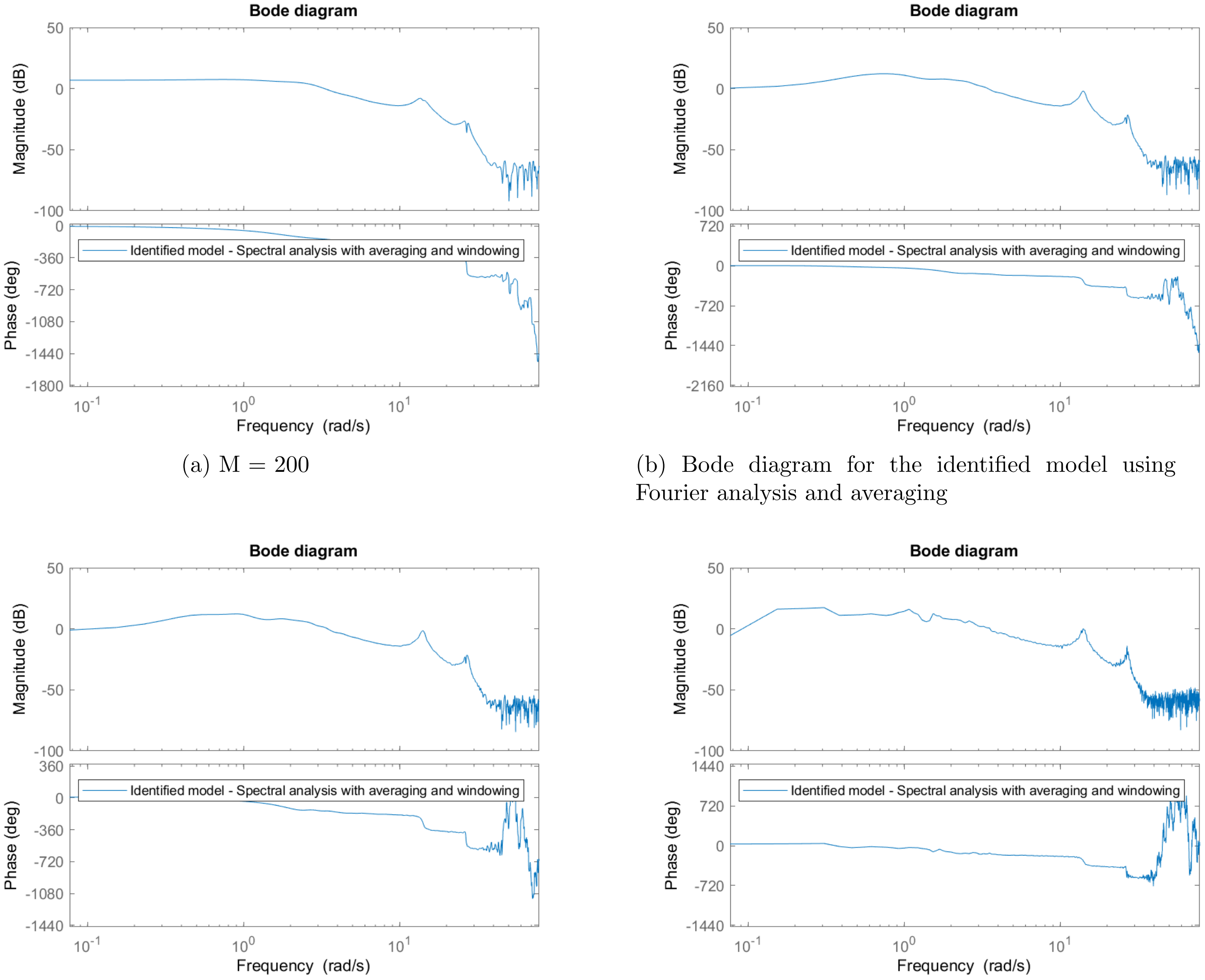
where

$$\bar{\Phi}_{ur}(\omega_n) = \frac{1}{m}\sum_{i=1}^{m}\phi_{ur,i}(\omega_n) \qquad \bar{\Phi}_{yr}(\omega_n) = \frac{1}{m}\sum_{i=1}^{m}\phi_{yr,i}(\omega_n) \tag{2}$$

for m equal to the number of periods in the signal.

Moreover, a windowing was applied on every period of the Fourier Transform in order to remove disturbances. Both Hann and Hamming windows were tested, but the results given were very similar. As such, it was chosen to show only the results for Hann windowing.

The window size M must be chosen carefully as it is important to keep all the significant part of the signal: in Figure 3 the effects for different window sizes can be observed. For a window too small (as in Figure 3a) the curve is smooth but the peaks at the resonance frequencies are lost. On the other hand, for a window too big (as in Figures 3c and 3d) the peaks are maintained but the noise is not attenuated sufficiently. In particular, Figure 3d shows that a window with the same size as the period of the Fourier Transform to which it is applied gives results that are very similar to the ones obtained with the fourier analysis (see Figure 2).



(a) M = 200

(b) Bode diagram for the identified model using Fourier analysis and averaging

(c) Bode diagram for input and output signals

(d) Bode diagram for the identified model using Fourier analysis and averaging

Figure 3: Spectral Analysis results with averaging and different values of windowing. We observe two resonance frequencies $\omega_1 \approx 14.1s^{-1}$ and $\omega_2 \approx 26.6s^{-1}$

Observing the results, a window of size $M = 550$ can be chosen. The choice was validated using the correlation and the deconvolution approaches to reconstruct the impulse response, which can be seen in Figure 4.

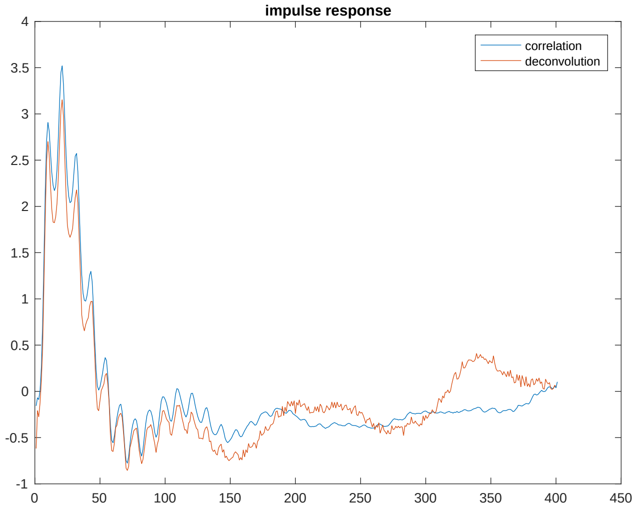


Figure 4: Reconstructed impulse response using correlation and deconvolution approach. The length of the settling time should correspond to the half size of the window $M/2$ which is about 250. This verifies window size chosen above.

3 System structure

In this section, different methods to select the order of the system were tested, as the First Principle modeling, the loss function method, the zero-pole cancellation method and the study of the first coefficients of the numerator polynomial.

3.1 Physical model

The physical model consists of 3 disks with different inertias connected in series by two torsionally flexible shafts acting as springs (k_1 and k_2). The first disk is driven by a electric motor with a torque $T = k_m u$ with u as input voltage. The output of the plant is the angle of the last disk θ_3. This model is illustrated in 5.

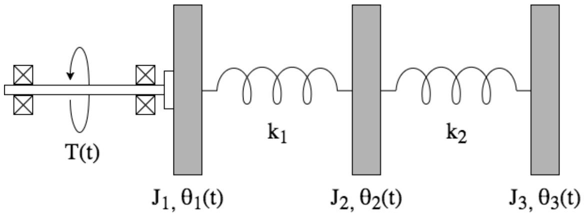


Figure 5: Physical plant model

Transfer function We can describe the dynamics of the above system as follows

$$T = k_m u \tag{3}$$

$$J_1\ddot{\theta}_1 + k_1(\theta_1 - \theta_2) = T \tag{4}$$

$$J_2\ddot{\theta}_2 + k_1(\theta_2 - \theta_1) + k_2(\theta_2 - \theta_3) = 0 \tag{5}$$

$$J_3\ddot{\theta}_3 + k_2(\theta_3 - \theta_1) = 0 \tag{6}$$

We solve for $G = \frac{\theta_3}{u}$

$$G = k_1 k_2 k_m \frac{1}{s^2} \frac{1}{J_1 J_2 J_3 s^4 + (J_1 J_2 k_2 + J_1 J_3 k_1 + J_1 J_3 k_2)s^2 + J_2 J_3 k_1 s^2 + J_1 k_1 k_2 + J_2 k_1 k_2 + J_3 k_1 k_2} \tag{7}$$

which we can reformulate as a system with double integrator. We can manually fit the values for $k \approx 10$ and the two resonance frequencies $\omega_1 \approx 14.1 s^{-1}$ and $\omega_2 \approx 26.6 s^{-1}$

$$G = \frac{k}{s^2(\frac{s^2}{\omega_1^2}+1)(\frac{s^2}{\omega_2^2}+1)} \tag{8}$$

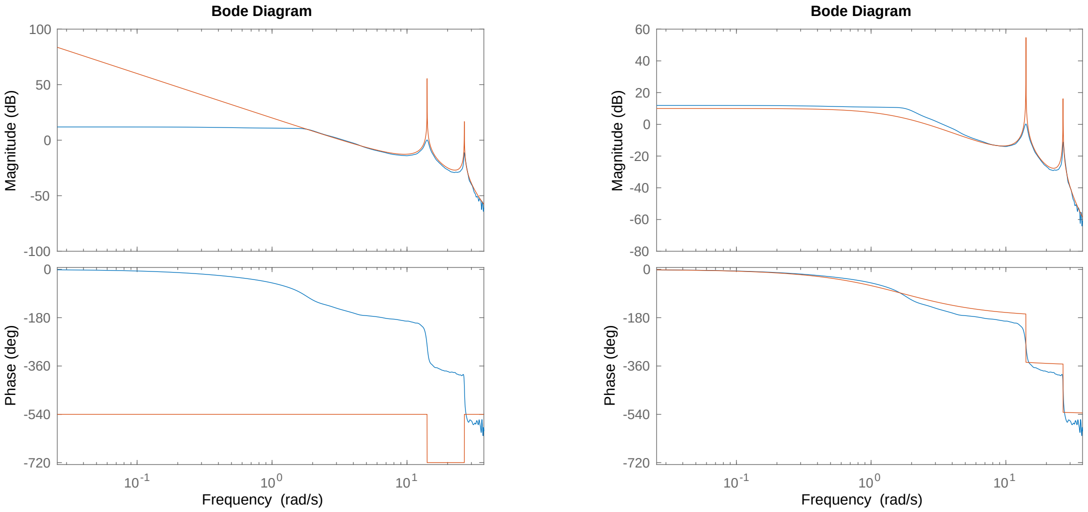


Figure 6: The bode plot to the left shows the above model manually fit to the spectral-analysis result. The one to the right is the same model but with a double high-pass filter with cutoff $\omega_c = 1.71 s^{-1}$ and $k = 3.16$. The right model has a better fit in lower frequencies and also correct phase.

Both models shown in Figure 6 are of degree 6, which serves as an indication of the degree of the real model.

Remark: We would expect a double integrator, which comes from the fact that the control input is a torque resulting in an acceleration while the output is a position. The two resonance frequencies come from the two spring-mass systems in series. One frequency is the two inertias in phase and the other in anti-phase.

State-space model We can also write the dynamical system above as a state-space model:

$$x = \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 & \dot\theta_1 & \dot\theta_2 & \dot\theta_3 \end{pmatrix}^\intercal \tag{9}$$

$$\dot{x} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k_1}{J_1} & \frac{k_1}{J_1} & 0 & 0 & 0 & 0 \\ \frac{k_1}{J_2} & -\frac{k_1+k_2}{J_2} & \frac{k_2}{J_2} & 0 & 0 & 0 \\ 0 & \frac{k_2}{J_3} & -\frac{k_2}{J_3} & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{k_m}{J_1} \\ 0 \\ 0 \end{pmatrix} u \tag{10}$$

$$y = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} x = \theta_3 \tag{11}$$

This means that we can expect to identify a state-space model for the plant with an order of about 6.

Note: We assume that friction is negligible. In the real system there is viscous friction, which would result in non zero values in the diagonal of the lower right part of the matrix A. Though the hypothesis of zero viscous friction is plausible, since we are operating in air at small amplitudes around $\dot\theta_3 = 0$. We also neglect dry friction, which would add some non-linearities. This is reasonable, since the axis are fixed with low friction ball bearings.

3.2 Parametric identification methods

Estimation of n by the loss function First, the loss function method for an ARX was applied as explained in the course book at page 109-110. The order of delay n_k was fixed to 1 and $n_a = n, n_b = n$ were increased in a range $1 : 25$. A global criterion was established with a penalty term of type BIC which is given by

$$BIC = L_f(n, N) + \frac{n \log N}{N} \tag{12}$$

The results can be seen in Figure 7. The minimum of the global function gives an order of $n = 5$.

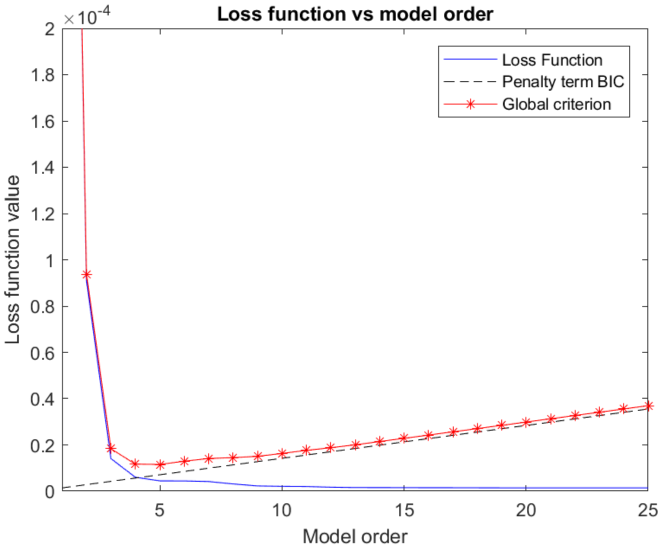


Figure 7: Loss function for the ARX model. Estimated order is $n = 5$

Zero-Pole cancellation method The estimation made above had to be validated. To do this, the zero-pole cancellation method was applied for $n_k = 1$ and $n_a = n_b = n_c = n$ for $n = 1 : n_{max}$, where n_{max} was chosen equal to 9 and then the orders were applied to an ARMAX structure. The results can be seen in Figure 11. As the system is subjected to noise, an exact zero-pole cancellation is impossible; as such, the 95% confidence intervals corresponding to $\pm 2\sigma$ was plotted as well. It was noticed that a zero-pole cancellation occurs for $n = 7$, so the order of the model can be estimated to be 6. The underestimation of the loss function method could be due to a too strong penalty term, as such the zero-pole cancellation estimation is preferred.

Delay estimation Once the order of the system was fixed, the delay n_k was estimated. To do this, the first coefficients of $B(q^{-1})$ for a FIR structure have to be evaluated. Then, the `arx` command was applied for $n_a = 0$, $n_b = n$, in order to create a FIR structure. As explained in the course notes at page 111-112, one has to enumerate the first coefficients equal to 0 or that can be considered 0, as to say if

$$0 \in [b_k - 2\sigma_k, b_k + 2\sigma_k] \tag{13}$$

As it can be seen in Figure 8, it can be stated that the delay n_k can be estimated equal to 0.

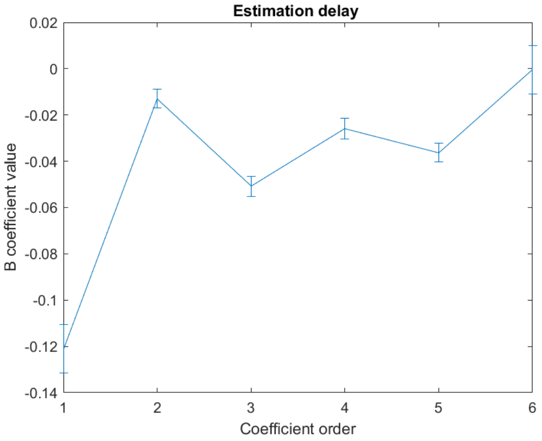


Figure 8: Estimated delay is $n_k = 0$

Estimation of n_b by loss function method In order to estimate the best n_b for the system, the loss function method was repeated, this time with $n_a = 6$ and $n_k = 0$ thanks to the results found above. The parameter n_b was tested for values $1 : 10$ and the results can be observed in Figure 9. As before, the minimum for the global criterion (as in Eq. 12) was evaluated and gave an estimation of $n_b = 6$. It is worth noticing that the result respects the condition $1 \leq n_b \leq n - n_k + 1$.

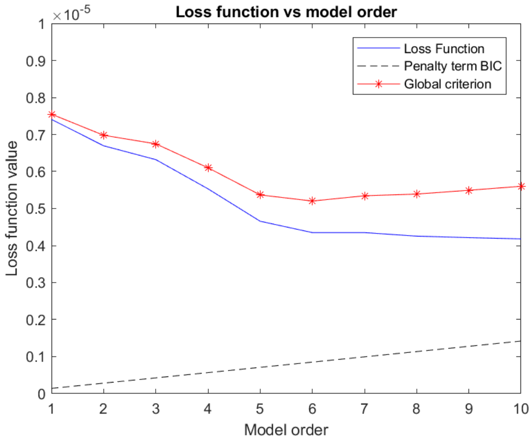
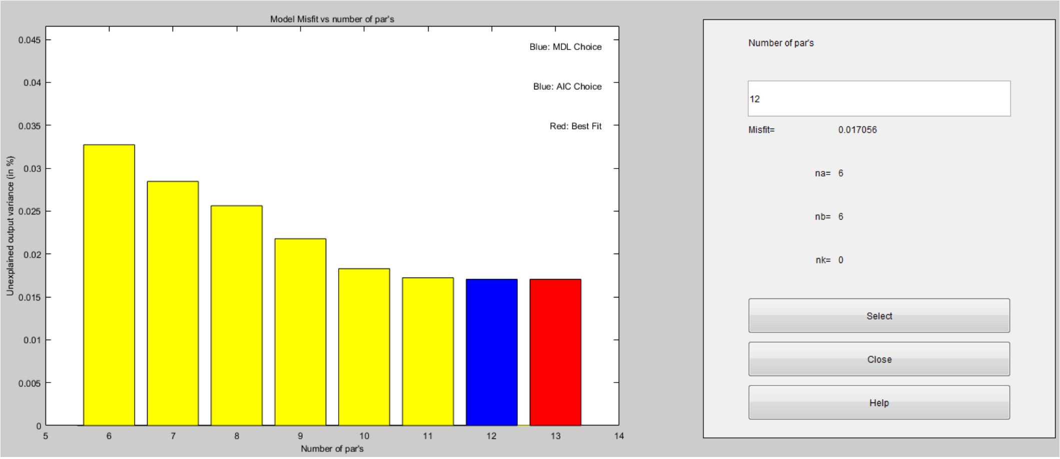


Figure 9: Loss function for estimating n_b, with $n_a = 6$ and different n_b. Estimated $n_b = 6$

Matlab selstruc Finally, the command `selstruc` was used to validate the choices made. The command was run for $n_a = 5 : 6$, $n_b = 1 : 7$ and $n_k = 0 : 4$. In Figure 10, the red bar represent the best fit for the Matlab command and corresponds to the choice $n_a = 6$, $n_b = 7$ and $n_k = 2$, for a total of 13 parameters to be estimated and a misfit of 0.0169%. The blue bar represents the best AIC and best MDL and corresponds to $n_a = 6$, $n_b = 6$ and $n_k = 0$, which is the model the results above gave. The misfit is also very low (0.01705%) and it's lower than the misfit for the model $n_a = 5$, $n_b = 6$ and $n_k = 0$ (0.01722%). This is in favor of the second choice of n_a, made using the zero-pole cancellation method.



Number of par's

12

Misfit= 0.017056

na= 6

nb= 6

nk= 0

Figure 10: Matlab command **`selstruc`** results in $n_a = 6$, $n_b = 6$ and $n_k = 0$ for 12 parameters.

4 Identification and Validation

In order to identify and validate different structures, the data was split into two datasets of equal size: `train_data` and `test_data`. The first one was used to identify the structures and the second to validate them. The results found in the section above ($n_a = 6$, $n_b = 6$ and $n_k = 0$) were used for the identification of ARX and Instrumental Variables structures (*iv4*). Then $n_c = n_a$, $n_d = n_a$ and $n_f = n_a$ were declared as required by the assignment for the identification of ARMAX, Box-Jenkins and Output-Error structures. Finally, $n_x = n_a$ was declared for the State-Space structure (`n4sid`). The validation results are in Figure 12 and show that the best structures for validating unseen data are Output-Error (62.13%) and State-Space (60.48%).

As last step, the whiteness test and the uncorrelation test were studied for all the models and are shown in Figure 13 and 14.

The whiteness test gives good results for ARMAX, Box-Jenkins and State-Space. It's important to notice that the whiteness test is valid only for structures containing a noise model. As such, the whiteness test corresponding to the Output-Error structure is meaningless. Also the Output-Error structure can not be used in closed loop identification because of the correlation of input u(t) and noise, which would result in biased estimates.

The uncorrelation test shows that the best results are given by the Instrumental Variables model and acceptable results for ARMAX, Box-Jekins and State-space models.

The frequency-domain validation was performed for all 6 models as shown in Figure 15. We immediately see that the ARX, Instrumental Variables and Output-Error models fail to fit the second resonance frequency peak. Thus we limit our choice to the remaining models: ARMAX, Box-Jenkins and State-space.

ARMAX and State-space have comparable results in time-domain and frequency-domain validation and are always performing better than Box-Jenkins.

As a simpler model is preferred, the ARMAX structure was chosen as the optimal structure for the system.

5 Conclusion

This project showed us how to identify a system from its input and output signals using non-parametric and parametric techniques. The structure we identified corresponds to a ARMAX structure with $n_a = 6$, $n_b = 6$ and $n_k = 0$, which gave 59.95% validation accuracy.

Some difficulties were found in particular for the estimation of the order, especially for the estimation of the delay, which was theoretically found equal to 0. This gave results with a phase lead, while testing with a delay of $n_k = 3$ gave correct results. We can then suppose that the delay was not correctly estimated.

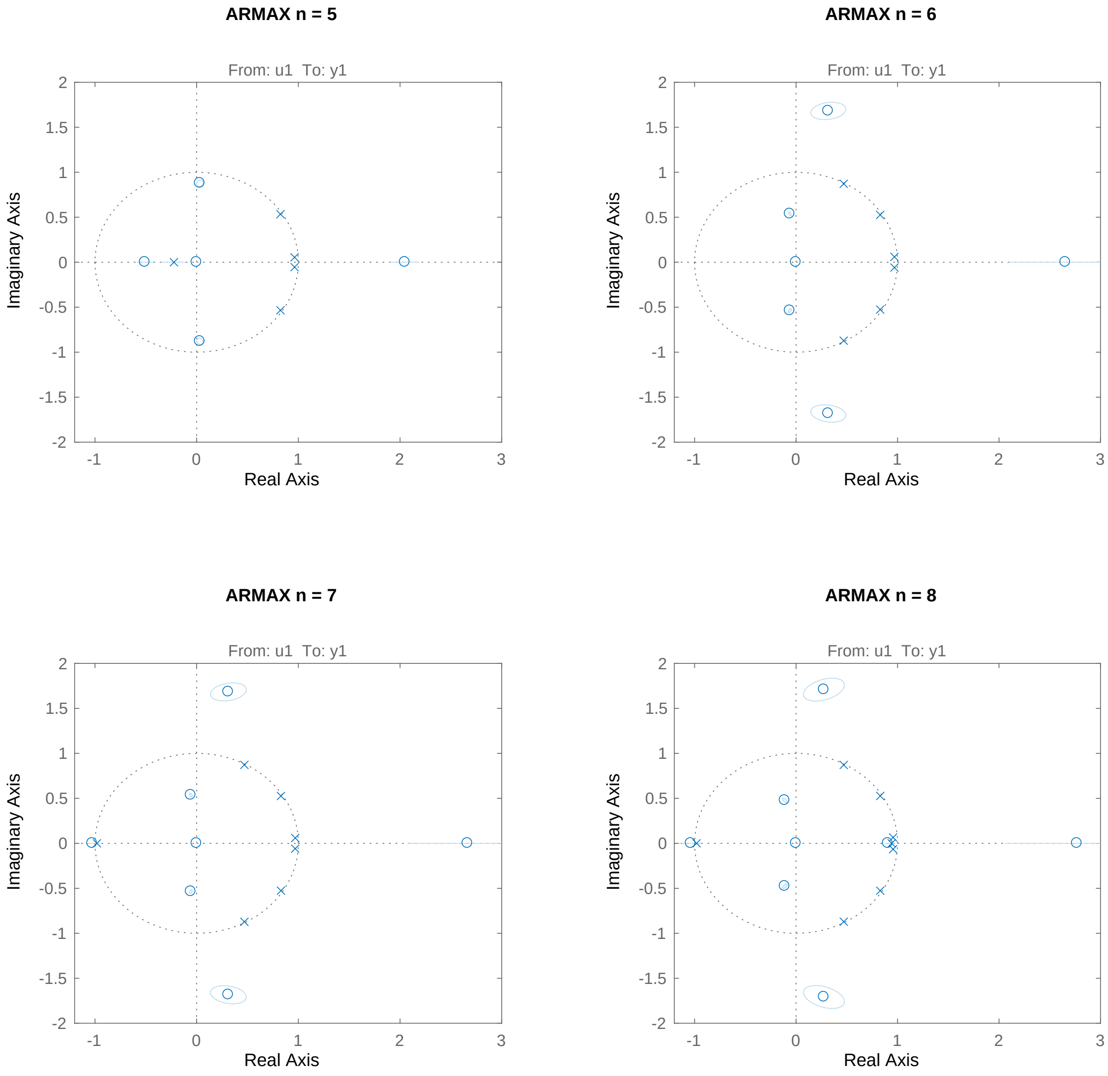


Figure 11: Zero-pole cancellation method for different orders. First zero-pole cancellation at $n = 7$

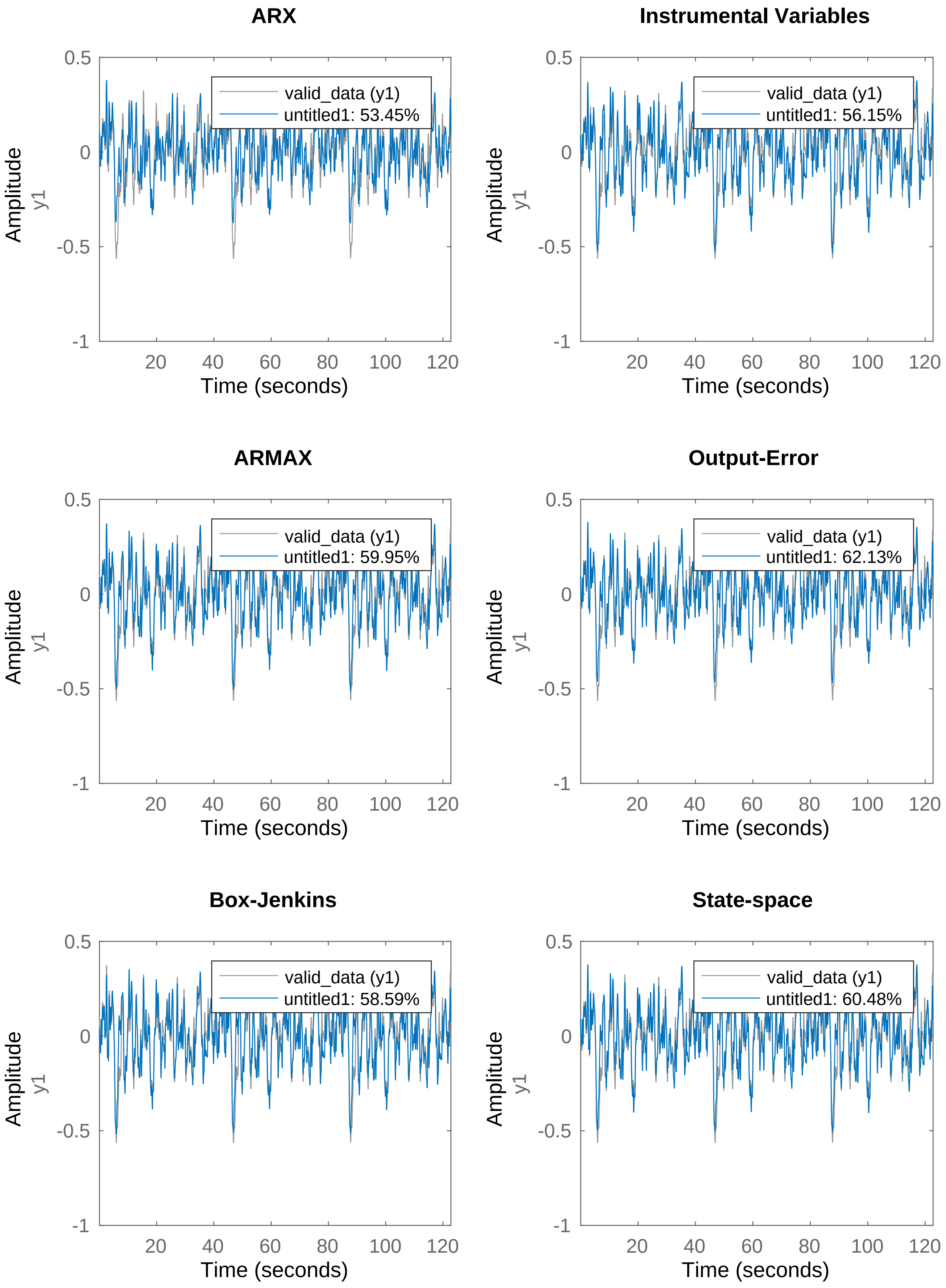


Figure 12: Identification and validation for different structures, while applying $n_a = 6$,$n_b = 4$,$n_c = 6$,$n_d = 6$,$n_f = 6$,$n_k = 0$ and $n_x = 6$: Surprisingly, the best result is given by the Output-Error structure with a reconstruction equal to 62.13% followed by the State-space model with 60.48% and ARMAX model with 59.95%

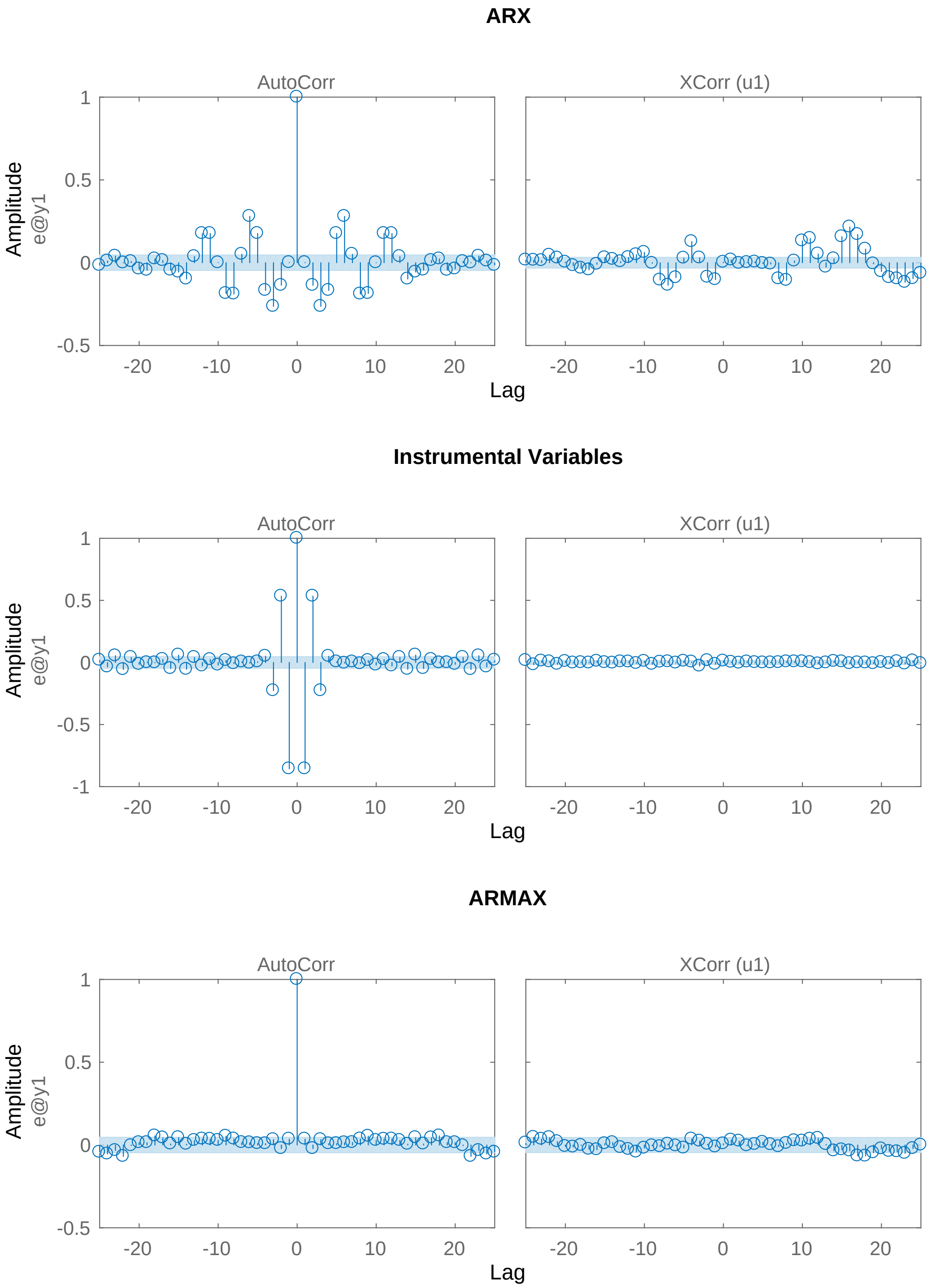


Figure 13: Whiteness test and uncorrelation test for the first three structures: the best result is given by the Instrumental Variables model

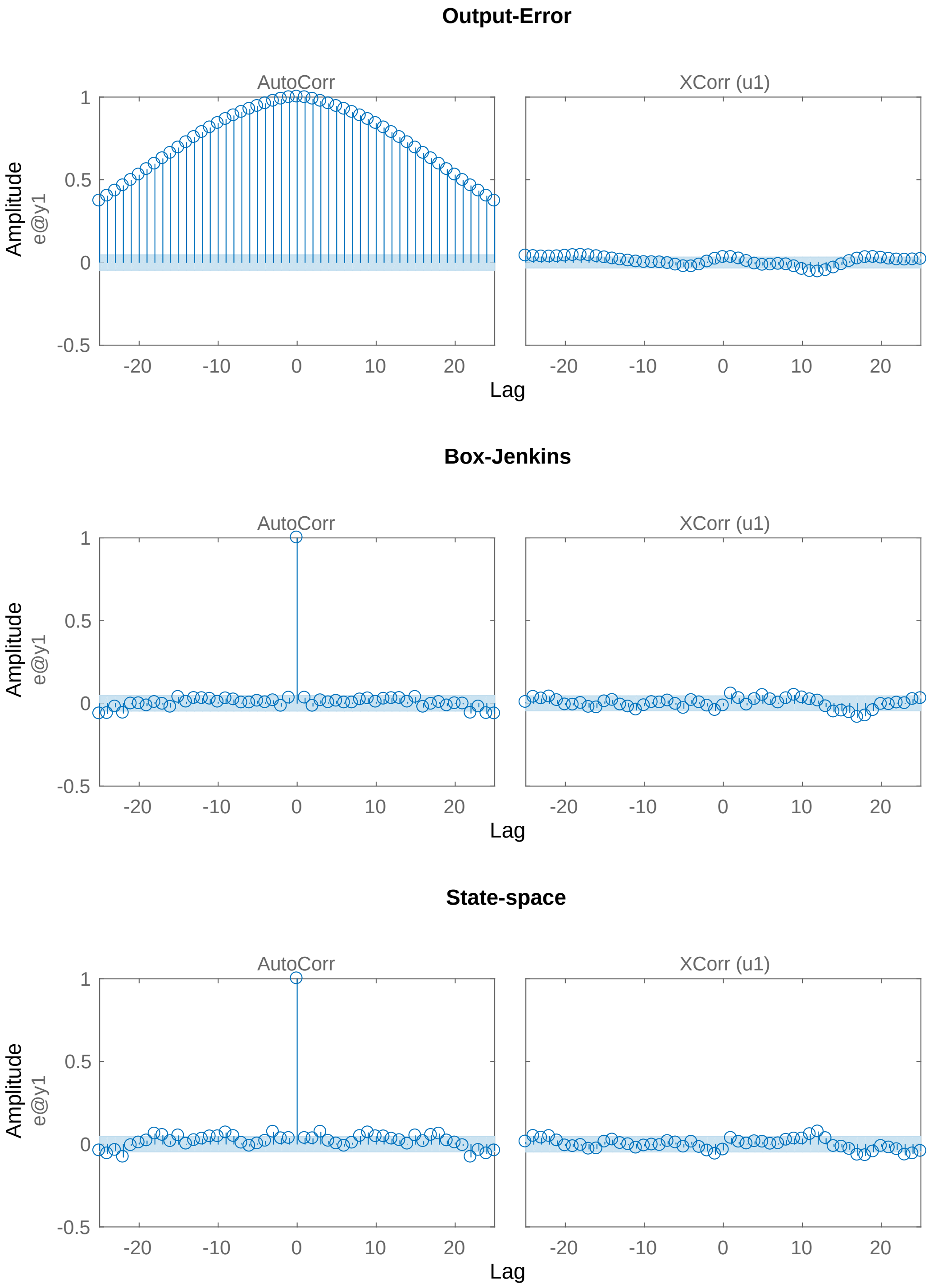


Figure 14: Whiteness test and uncorrelation test for the last three structures: the best result is given by the Box-Jenkins model, although the State-space model performs comparably well. We notice that for the Output-Error structure the whiteness test is irrelevant as this structure does not contain a noise model. Also its main assumption that u(t) and the noise are uncorrelated is not satisfied. Thus we exclude the OE model.

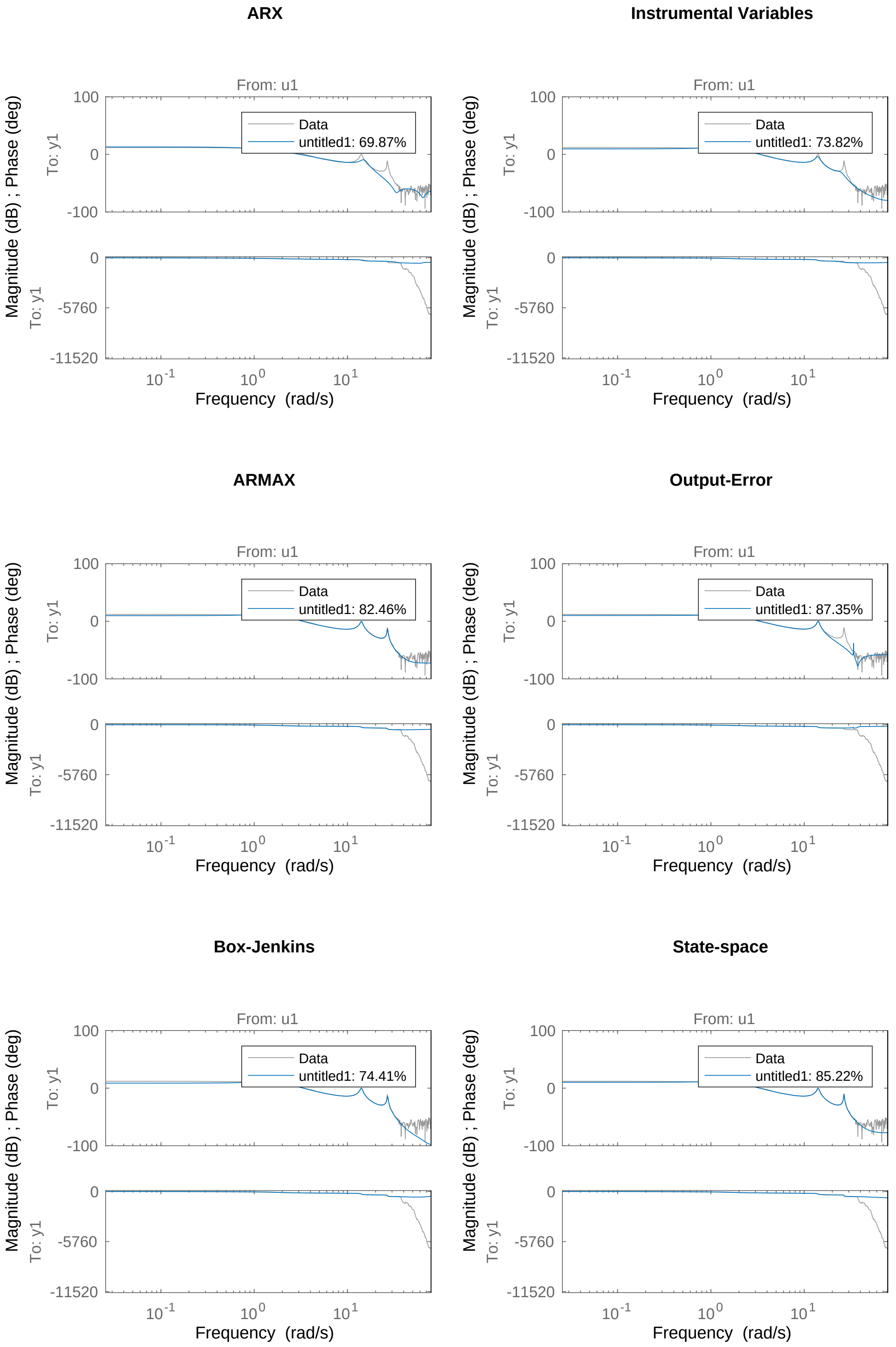


Figure 15: Frequency-domain validation: The models ARX, Output-Error and Instrumental Variables do not fit the second resonance frequency lobe. This is reason enough to discard them because it might be inside the operating frequency of the system.

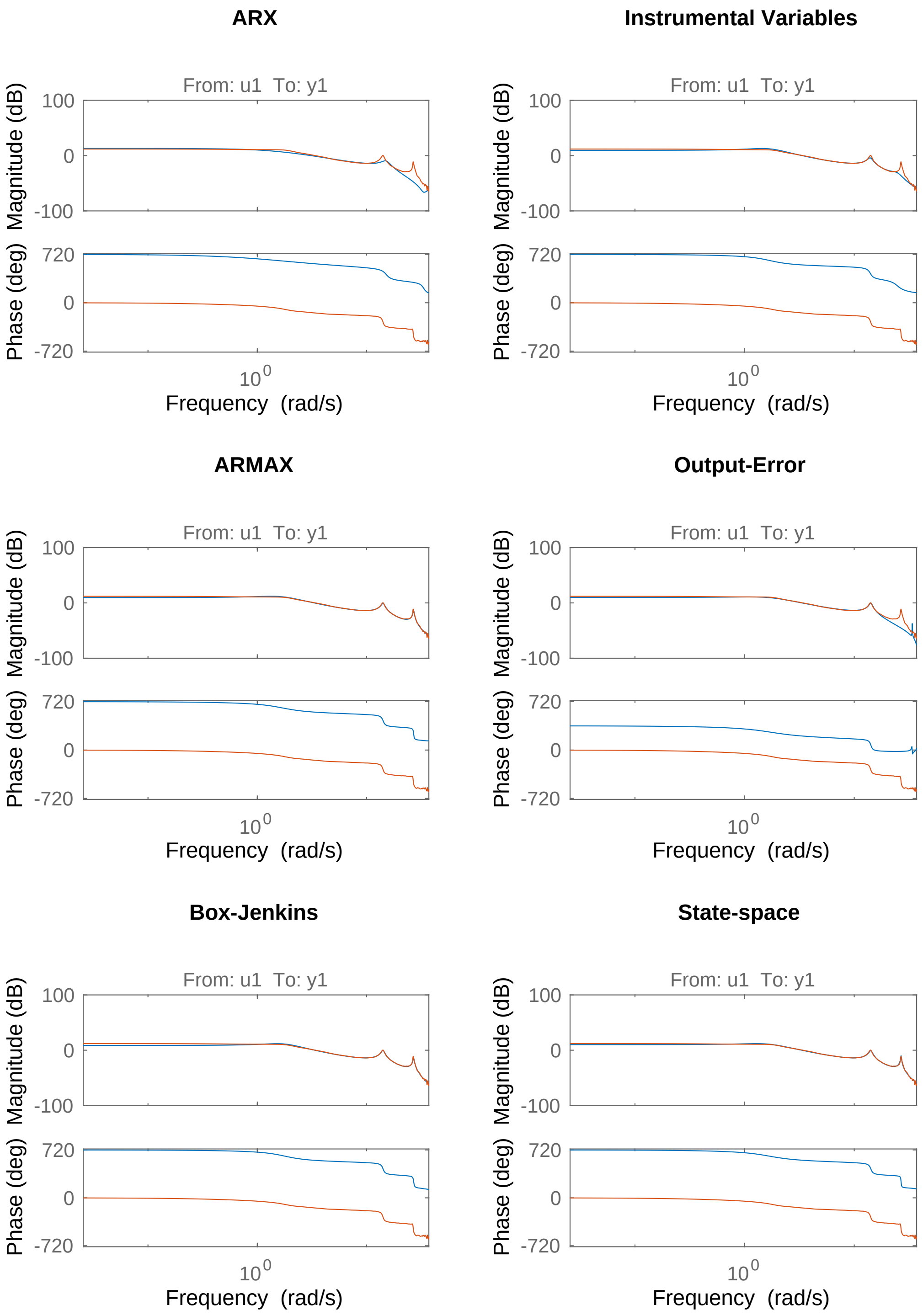


Figure 16: Closer look at frequency domain comparison of the different models (blue) with the spectral analysis model (red). We observe that most models have a 720° phase lead to the actual model.

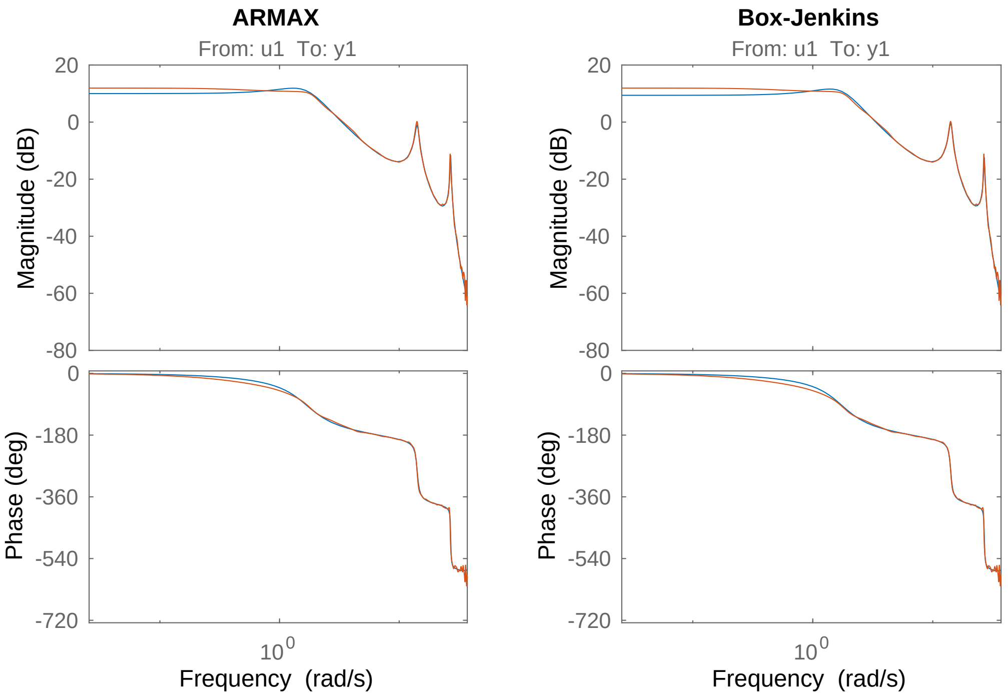


Figure 17: When adding a delay $n_k = 3$ the phase lead disappears from the ARMAX and Box-Jenkins models, giving a relatively good fit in magnitude and phase. We could presume $n_k = 3$ gives a better fit even though during the order selection process we could rule out $n_k > 0$.

Appendices

A final_project.m

```
clear all
close all
clc

FPdata = load('FPdata.mat');
Te =  0.04;

%%
run('frequency_analysis.m')

%%
run('physical_model.m')

%%
run('estimation.m')

%%
run('validation.m')
```

B frequency_analysis.m

```

% Windowing
hann_window = @(M) 0.5+0.5*cos(pi*[0:M-1]'/(M-1));

```

```

%% Impulse response using correlation and deconvolution approach
N = length(FPdata.u)/6;
k=400; % max length
u = FPdata.u(end-N+1:end);
y = FPdata.y(end-N+1:end);
Ruu = intcor(u, u); % use intcor because u periodic
Ryu = intcor(y, u);
U = toeplitz(Ruu(N:N+k));
theta = pinv(U’*U)*U’*Ryu(N:N+k);

figure
plot(1/Te*theta)
title(’impulse response’)
hold on

% deconvolution
U = toeplitz(u, [u(1);zeros(N-1,1)]);
Uk = U(:,1:k); % truncate U
theta = pinv(Uk)*y/Te;
plot(theta)
legend(’correlation’,’deconvolution’)
hold off

%% spectral analyis
u = FPdata.u;
y = FPdata.y;
r = FPdata.r;

N_AVG = 1;
N = floor(length(u)/N_AVG);

SCALEOPT = ’biased’;
%window = ones(N,1);
window = hann_window(250);
padding = zeros(N - length(window), 1);
window = [window; padding];

fft_Ryr = zeros(N,1);
fft_Rur = zeros(N,1);
for i = 0:N_AVG-1
    % split into chunks
    yc = y(i*N + 1:(i+1)*N);
    uc = u(i*N + 1:(i+1)*N);
    rc = r(i*N + 1:(i+1)*N);
    % correlation
    Ryr = xcorr(yc,rc, SCALEOPT);
    Rur = xcorr(uc,rc, SCALEOPT);
    Ryr = Ryr(N:end);
    Rur = Rur(N:end);
    % windowing
    Ryr = Ryr .* window;
    Rur = Rur .* window;
    % averaging
    fft_Ryr = fft_Ryr+fft(Ryr);
    fft_Rur = fft_Rur+fft(Rur);
end

% Reconstruction
Gr = fft_Ryr./fft_Rur;

omega_s = 2*pi/Te;
freq = (0:N-1).*(omega_s/N);

NYQUIST_INDEX = round(N/2);
Gr = Gr(1:NYQUIST_INDEX);
```

```
freq = freq(1:NYQUIST_INDEX);

spectral_analysis_model = frd(Gr, freq, Te);
figure
hold on;
bode(spectral_analysis_model); hold off;
title('spectral analysis')

%% fourier analyis
u = FPdata.u;
y = FPdata.y;

N_PERIODS = 6;
N = length(FPdata.u)/N_PERIODS;
omega_s = 2*pi/Te;
avg_y = zeros(N,1);
avg_u = zeros(N,1);
for i = 1:N:N*N_PERIODS
    in = y(i:i+N-1);
    avg_y = avg_y + fft(in);
    out = u(i:i+N-1);
    avg_u = avg_u + fft(out);
end
freq = [];
for i = 0:N-1
   freq = [freq; i*omega_s/N];
end
freq = (0:N-1).*(omega_s/N);

Y = avg_y / N_PERIODS;
U = avg_u / N_PERIODS;

% Reconstruction
Gr = Y ./ U;

NYQUIST_INDEX = round(N/2);
freq = freq(1:NYQUIST_INDEX);
Gr = Gr(1:NYQUIST_INDEX);

fourier_analysis_model = frd(Gr, freq, Te);
%figure
hold on
bode(fourier_analysis_model)
%title('fourier analysis')
title('frequency analysis')
legend('spectral analysis', 'fourier analysis')
hold off

% we see 2 resonance frequencies at 14.1 rad/s and 26.6 rad/s
```

C physical_model.m

```
%% double integrator model with two resonance frequencies
w1 = 14.1;
w2 = 26.6;
s = tf('s');
k = 10;
G = k * 1/s^2 * 1/(((s/w1)^2+1)*((s/w2)^2+1));
%c2d(G, Te)
freq = (0:1450).*(omega_s/N);
figure
subplot(1,2,1)
bode(spectral_analysis_model); hold on;
bode(G, freq); hold off;
```

```
%% ... model with lowpass instead of doule integrator
wc = 1.71;
k = 3.16;
G = k * 1/(s/wc + 1)^2 * 1/(((s/w1)^2+1)*((s/w2)^2+1));
%c2d(G, Te)
%figure
subplot(1,2,2)
bode(spectral_analysis_model)
hold on;
bode(G, freq)
hold off;
%% Analysis of physical system
syms x1 x2 x3 J1 J2 J3 k1 k2 k3 km u s
A=[
    (J1*s^2+k1), -k1, 0;
    -k1, (J2*s^2+k1+k2), -k2;
    0, -k2, (J3*s^2+k2)
];
B=[km*u;0;0];
x=[x1;x2;x3];
res = solve(A*x-B, x);
G = res.x3/(km*u)
```

D estimation.m

```
u = FPdata.u;
y = FPdata.y;
r = FPdata.r;
N = length(y);

data = detrend(iddata(y,u,Te));
valid_data = detrend(iddata(y(1:N/2),u(1:N/2),Te));
valid_data = detrend(iddata(y(N/2+1:end),u(N/2+1:end),Te));

% ARX models for different orders n
%% na estimate
order = 2:10;
loss_func = [];
for na = order
    SYS = arx(data, [na na 0]);
    loss_func = [loss_func, SYS.EstimationInfo.LossFcn];
end

scaling_factor = 0.001;
penalty_BIC = scaling_factor*order*log(N)/N;
global_crit = loss_func + penalty_BIC;
[~,i] = min(global_crit);
na_est = order(i);

figure; hold on;
plot(order, loss_func, ’b’);
plot(order, penalty_BIC, ’--k’)
plot(order, global_crit, ’-*r’)
hold off;
title(’ARX loss function’);
xlabel(’Model order na’);
ylabel(’Loss function’);
legend(’Loss Function’, ’Penalty term BIC’, ’Global criterion’);
na = na_est

%% nk estimate
nb = na; nk = 0;
SYS = arx(data, [na nb nk]);
figure;
```

```
errorbar(SYS.b, SYS.db*2) % plot B with a confidence interval of 2 sigma

lower = SYS.b + 2 * SYS.db;
higher = SYS.b - 2 * SYS.db;
nk_est = 0;
for i = 1:length(SYS.b)
    if(lower(i).*higher(i) <= 0)
        nk_est = nk_est + 1;
    else
        break; % if 0 in 2sigma confidence interval
    end
end
nk = nk_est

%% nb estimate
order = 1:10;
loss_func = [];
for nb = order
    orders = [na nb nk];
    SYS = arx(data, orders);
    loss_func = [loss_func, SYS.EstimationInfo.LossFcn];
end

scaling_factor = 1e-4;
penalty_BIC = scaling_factor*order*log(N)/N;
global_crit = loss_func + penalty_BIC;
[˜,i] = min(global_crit);
nb_est = order(i);

figure; hold on;
plot(order, loss_func, ’b’);
plot(order, penalty_BIC, ’--k’)
plot(order, global_crit, ’-*r’);
hold off;
title(’ARX loss function’);
xlabel(’Model order nb’); ylabel(’Loss function’);
legend(’Loss Function’, ’Penalty term BIC’, ’Global criterion’);
nb = nb_est
n = max(na, nb+nk)

%% Validate order with ARMAX model by observing pole-zero-cancellation
figure
order=5:8;
for n=order
    SYS = armax(data, [n n n 0]);
    subplot(2,2,n-min(order)+1);
    h = iopzplot(SYS);
    showConfidence(h, 2);
    title(’ARMAX n = ’+string(n));
    axis([-1.2 3 -2 2])
end
printpdf(gcf, ’final-report/images/3_zero_pole_cancel.pdf’, 1.5, 1.8)

%% compare with selstruc
%NN = struc(5:7,5:7,0:4);
%V = arxstruc(valid_data, valid_data, NN);
%order = selstruc(V)
```

E validation.m

```
u = FPdata.u;
y = FPdata.y;
N = length(y);
train_data = detrend(iddata(y(1:N/2),u(1:N/2),Te));
```

```
valid_data = detrend(iddata(y(N/2+1:end),u(N/2+1:end),Te));

n = max(na_est, nb_est)
na = n;
nb = n;
nk = nk_est;
nc = n; nd = n; nf = n; nx = n;

models = struct(’sys’,{},’name’,{});
models(1).sys = arx(train_data, [na nb nk]);
models(1).name = ’ARX’;
models(2).sys = iv4(train_data, [na nb nk]);
models(2).name = ’Instrumental Variables’;
models(3).sys = armax(train_data, [na nb nc nk]);
models(3).name = ’ARMAX’;
models(4).sys = oe(train_data, [nb nf nk]);
models(4).name = ’Output-Error’;
models(5).sys = bj(train_data, [nb nc nd nf nk]);
models(5).name = ’Box-Jenkins’;
models(6).sys = n4sid(train_data, nx);
models(6).name = ’State-space’;

%% Time domain validation
figure
for i = 1:length(models)
    subplot(3,2,i)
    compare(valid_data, models(i).sys)
    title(models(i).name)
end
printpdf(gcf, ’final-report/images/4_time_domain_valid.pdf’, 1, 1.8)
%% time domain compare
fitval = [];
names = {};
for i = 1:length(models)
    [YT, FIT, X0] = compare(valid_data, models(i).sys);
    fitval = [fitval; FIT];
    names{i} = models(i).name;
end
figure
bar(fitval)
ylim([0 Inf])
xticklabels(names);
xtickangle(45)
ylabel(’Fit (%)’)
title(’Time-domain comparison’)
saveas(gcf, ’final-report/images/4_time_domain_comp’, ’png’);
%% Statistical validation
figure
for i = 1:3
    subplot(3,1,i)
    resid(valid_data, models(i).sys)
    title(models(i).name)
end
printpdf(gcf, ’final-report/images/4_resid_1.pdf’, 1, 1.8)
figure
for i = 4:6
    subplot(3,1,i-3)
    resid(valid_data, models(i).sys)
    title(models(i).name)
end
printpdf(gcf, ’final-report/images/4_resid_2.pdf’, 1, 1.8)
%% Frequency-domain validation
figure
for i = 1:length(models)
    subplot(3,2,i)
    compare(spectral_analysis_model, models(i).sys)
    title(models(i).name)
end
printpdf(gcf, 'final-report/images/4_freq_domain_valid.pdf', 1.2, 2.4)
%% Frequency-domain compare
fitval = [];
names = {};
for i = 1:length(models)
    [YT, FIT, X0] = compare(spectral_analysis_model, models(i).sys);
    fitval = [fitval; FIT];
    names{i} = models(i).name;
end
figure
bar(fitval)
ylim([0 Inf])
xticklabels(names);
xtickangle(45)
ylabel('Fit (%)')
title('Frequency-domain comparison')
saveas(gcf, 'final-report/images/4_freq_domain_comp', 'png');
%% Frequency-domain manual comparsion
% define valid frequency spectrum, cut out noisy part where phase explodes.
freq = spectral_analysis_model.Frequency(2:1450);
figure
for i = 1:6
    subplot(3,2,i)
    bode(models(i).sys); hold on;
    bode(spectral_analysis_model, freq); hold off;
    %legend(models(i).name, 'SA');
    title(models(i).name)
end
printpdf(gcf, 'final-report/images/4_freq_visual_comp.pdf', 1, 1.8)
```

F intcor.m

```
function [R,h] = intcor(u,y)
% intercorrelation function
% supposes that u and y to be of same length

N = length(u);
R = [];
h = [];

for h0 = -N+1:N-1
    buff = 0;
    for k = 1:N
        buff = buff + 1/N * u(k)*y(mod(k-h0+N-1,N)+1);
    end
    R = [R;buff];
    h = [h;h0];
end
```