Grading template for laboratory exercise 3
EL2820, Modeling of Dynamical Systems
October 2022

Javier Lopez Iniesta Diaz del Campo

19990714-T692

Lukas Rapp

19970727-T558

	Pass		Fail	
		yes	no	
The report is handed in on time?		☐	☐	
		$\leq$ 2st	> 2	
Number of authors		☐	☐	
		yes	no	
Author names and personal identity number filled out?		☐	☐	
	yes	often	sometimes	no
The report is well structured? The language is understandable?	☐	☐	☐	☐
	yes	often	sometimes	no
The figures are clear? (Captions, high resolution, etc.)	☐	☐	☐	☐
		yes	no	
The preparation task is solved and motivated?		☐	☐	
		yes	no	
The working region is defined and motivated?		☐	☐	
		yes	no	
A detailed description of the input signal is given and the choice is motivated?		☐	☐	
		yes	no	
The amount of data used for estimation and validation is specified?		☐	☐	
		yes	no	
Models of more than one model structure have been estimated?		☐	☐	
		yes	no	
The model order of each model is motivated?		☐	☐	
		yes	no	
A ranking of the estimated models have been made?		☐	☐	
		yes	no	
The ranking is well motivated according to the requirements?		☐	☐	

	Pass	Fail	
First review	☐	☐	Sign:
. . . .		. . . .	
Second review (if failed in the first review)	☐	☐	Sign:

Pass

Signature:

1 Preparation task

1.1 Physical model of the magnetic levitator

We apply Newton's second law onto both discs for which the respective forces can be found in Fig. 2 of the laboratory exercise instruction [1]:

$$m\ddot{z} = E_{\mathrm{r}} - F_{\mathrm{lu}} - mg - \gamma\dot{z}, \qquad m\ddot{y} = -E_{\mathrm{a}} + F_{\mathrm{lu}} - mg - \gamma\dot{y}. \tag{1}$$

$F_{lu} = \alpha|y-z|^{-4}$ is the magnetic force acting from the lower magnet to the upper one, where α is an unknown proportional coefficient. E_i is the electromagnetic force applied from the coil onto the discs. According to Problem 1.4 of the exercise compendium [2], it is given by $E_i = \beta I^2/x^6$, where $x = z$ or $x = y$ is the distance between the coil and the respective disc and $\beta := \frac{A}{2\mu_0}\left(\mu_0 \frac{Nr^2}{2}\right)^2$ is a constant depending on parameters of the plant.

To derive the state space model, we introduce the velocities $v_{\mathrm{z}} = \dot{z}$ and $v_{\mathrm{y}} = \dot{y}$ as additional state variables besides z and y. Inserting all relations into (1) results in the state space model:

$$\begin{aligned}
\dot{z}(t) &= v_{\mathrm{z}}(t), & \dot{v_{\mathrm{z}}}(t) &= \frac{1}{m}\left(\beta\frac{I^2(t)}{z^6(t)} - \gamma v_{\mathrm{z}}(t) - mg - \alpha|y(t)-z(t)|^{-4}\right), \\
\dot{y}(t) &= v_{\mathrm{y}}(t), & \dot{v_{\mathrm{y}}}(t) &= \frac{1}{m}\left(-\beta\frac{I^2(t)}{y^6(t)} - \gamma v_{\mathrm{y}}(t) - mg + \alpha|y(t)-z(t)|^{-4}\right).
\end{aligned}$$

The model is not linear because the magnetic force F_{lu} between the upper and lower disc depends in an non-linear way on the difference between the state variables y and z and the electromagnetic force E_i, which the coil applies to both coils is non-linear with respect to the input signal I and y, z.

The state space model of the magnetic levitator is similar to a mass-spring-system, where two masses are vertically connected by a spring so that gravity is pulling the masses downwards. In addition, both masses are excited by external forces, which corresponds to the forces that the coil applies on the discs.

For the linearized model, we assume that the operating point $\vec{x}_0 = (v_z^0, v_y^0, z^0, y^0)$ and I^0 is an equilibrium point, i.e., $f(\vec{x}_0, I_0) = 0$. Then, the linearized model with respect to this operating point is given by:

$$\frac{d}{dt}\begin{pmatrix}\Delta v_z \\ \Delta v_y \\ \Delta z \\ \Delta y\end{pmatrix} = \begin{pmatrix} a_{11} & 0 & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}\begin{pmatrix}\Delta v_z \\ \Delta v_y \\ \Delta z \\ \Delta y\end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ 0 \\ 0\end{pmatrix}\Delta I,$$

where Δp denotes the difference between a variable p and its value in the operating point p^0 and a_{ij} and b_i are non negative constants.

1.2 Suggestion for a suitable model order for a linear model

The linearized state space model has order 4 assuming the state space variables are independent. Hence, the transfer function of the continuous linearized system has 4 poles. Therefore, the estimated discrete model should have at least $n_f = 4$ poles to capture the the dynamics of the model. In practice a higher order of 5 or 6 might be necessary to capture hidden dynamics, which don't appear in our theoretical model. We will continue these considerations in Sec. 5.1 when we choice the models structures for the estimation.

1.3 MATLAB codes for the required functions

```
function bar_v = getAverage(v, tail)
    % Convert from tail% to tail absolute
    tail_abs = floor(length(v)*tail/100);
    % Last tail samples of v
    v_tail = v(end-(tail_abs-1):end, 1);
    % Empirical mean
    bar_v = mean(v_tail);
end
```

MATLAB function 1: *getAverage(v, tail)*.

```
function bar_y = getStationaryAverages(y_step, Nwr, tail)
    bar_y = zeros(Nwr, 1);
    length_step = length(y_step)/Nwr;

    % For each one of the different levels
    for i = 1:Nwr
        % Obtain a segment of y_step of length_step samples
        y_part = y_step((i-1)*length_step+1:i*length_step);
        % Do the empirical mean of the last tail% samples of y_part
        bar_y(i) = getAverage(y_part, tail);
    end
end
```

MATLAB function 2: *getStationaryAverages(y_step, Nwr, tail)*.

2 Working region

In this case, the nonlinear system is estimated with a linear model. Therefore, we need the process to be close to our working point. Therefore, the working region is the largest range of values of our input signal, which guarantees a linear approximation.

To calculate the corresponding values, u_{min} and u_{max} of the input signal, we first observed in the provided file `working_region.mat` that the output signal contains 500 samples in each step. Of all of them, the first 5 samples corresponded to the delay. After that, there are approximately 125 samples of transient data and the rest of the signal is useful data. Therefore, 370 samples are useful, or 75% of the total, i.e, $tail = 75\%$. Then, the previously generated MATLAB function 2 has been applied to remove the transient phase and the noise. Where, y is the output signal of the `working_region.mat` file, *tail* is 75% and *Nwr* is 21 steps.

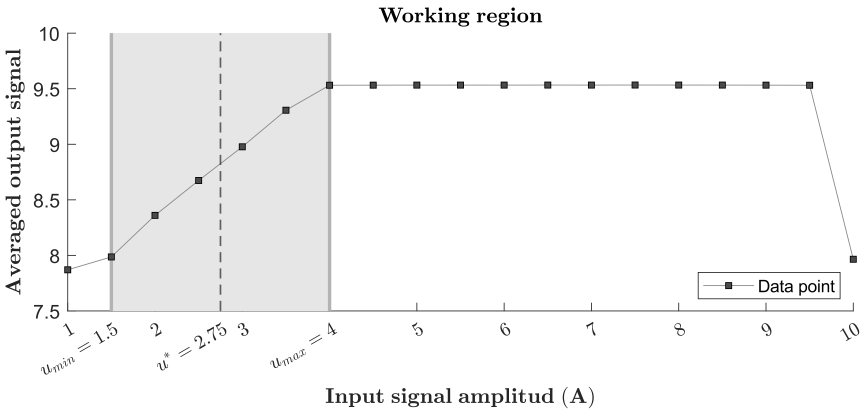


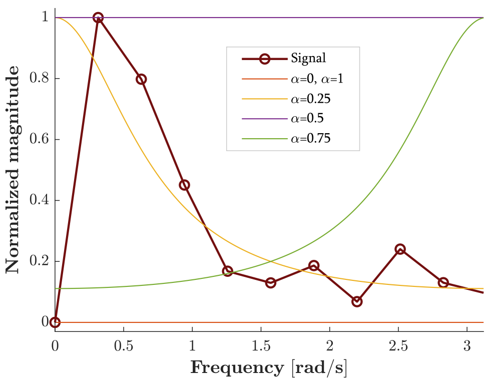
Figure 1: Working region.

Figure 1 shows the averaged output signal when several step response experiments are performed. As can be seen, the system is linear between the interval [1.5, 4]. In this interval the steady output amplitude should scale proportionally to the amplitude of the step amplitude. Thus, the working region is $[\mathbf{1.5}, \mathbf{4}]$ and the averaged value of the input signal is $u^* = 2.75$.

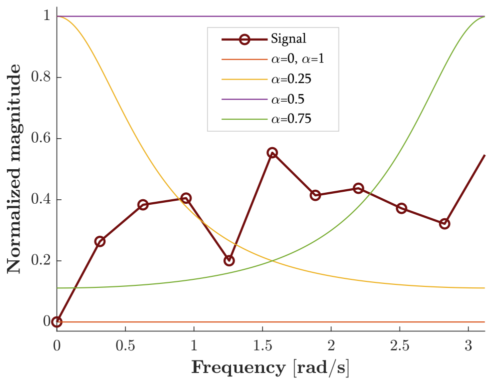
3 Input signals

For both uniformly distributed white noise input signals, our first choice have been the signals "white_noise_2" and "white_noise_4" because both signals have boundaries within the working regions and therefore are suitable for the identification process. In addition, these input signals have the smallest $(1.75 - 2.5)$ and the largest $(1.5 - 4)$ amplitude, so we can see later how the choice of amplitude affects the performance of the estimated models.

Afterwards, we followed the laboratory instructions to analyze the Fourier transform of the output signal. The white noise signals 1, 2 and 3 concentrate all the output energy at low frequencies, as shown in Fig. 2(a) of the output signal spectrum generated with "white_noise_1" where the energy is concentrated around $f = 0.5\,\mathrm{rad\,s^{-1}}$. However, the spectrum of the output signal of "white_noise_4" (Fig. 2(b)) is quite irregular. Hence, we assumed that it might be unsuitable for the estimation and choice white noise 3 instead of 4, which has the same amplitude as white noise 4.



(a) White noise 1.



(b) White noise 4.

Figure 2: Spectrum of output signal generated with "white_noise_1" as the input signal. The spectrum is compared by the spectrum of a binary random signal for several values of α.

Based on the previous reasoning, **white noise signals 2 and 3** have been chosen.

Regarding the binary signal, we have chosen one with $\alpha = 0.25$ as input signal because its energy is also concentrated in this region. Furthermore, for $\alpha = 0.25$, we select the available input signal with the largest possible amplitude inside the working region. This is "**binary_signal_2**" with the limits $1.5 - 4$.

4 Estimation and validation data

To estimate and validate the models of the plant, we first remove the transient, which is needed to reach the desired operating point, from the measured input and output signals. This is done in the Matlab code of the Appendix 3 by removing the first $t_{\text{min}} \cdot$ (length signal) samples of the signal.

After that, each input and output signal is divided into two parts of equal length, where the first part is used for parameter estimation and the second part for model validation. As each input has a different total number of samples, the amount of data used for identification and validation is:

- "binary_signal_2": 10000 samples.
- "white_noise_2": 7500 samples.
- "white_noise_3": 4000 samples.

5 Models

5.1 Model Structures

For the model estimation, we use an ARX (autoregressive with exogeneous input), an ARMAX (autoregressive moving average exogeneous input) and a BJ (Box Jenkins) model [3]:

$$\begin{aligned}
&\text{ARX:} && y[k] = \frac{B(q)}{A(q)}y[k] + \frac{1}{A(q)}u[k],\\
&\text{ARMAX:} && y[k] = \frac{B(q)}{A(q)}y[k] + \frac{C(q)}{A(q)}u[k],\\
&\text{BJ:} && y[k] = \frac{B(q)}{A(q)}y[k] + \frac{C(q)}{D(q)}u[k],
\end{aligned} \tag{2}$$

where $A(q), B(q), C(q)$ and $D(q)$ are polynomials.

To choose the order of each ARX model, the "order selection" tool was used. This tool evaluates the performance of different parameter combinations (n_a, n_b, n_k) and visualizes their accuracy graphically. We choice the parameter set with a high accuracy while keeping the order of the estimated parameters as small as possible. The reason for this is that beyond a certain point the size of the bars (in the plot given by the system identification tool: "Model Fit vs # of par's") does not decrease significantly.

For the cases of the ARMAX and BJ models, different parameters have been evaluated by their FIT coefficient. The initial model order was chosen according to our considerations in Sec. 1.2, i.e., $n_a = 4$. We estimated the delay $n_k = 4$ using the step responses in Sec. 2 and derived $n_b = n_a - 1 = 3$ following the model order selection rules of the lecture [3, slide 7 and 31]. The initial order of the noise model was chosen arbitrarily to $n_c = n_d = 4$ because no rules were given in the lecture. Starting from the initial parameter guesses, we optimized the parameters by hand until a model with a good FIT coefficient was found. During the optimization, we kept the model order n_a, n_b below 7, which should be sufficient to describe a physical model of order 4 and eventual hidden dynamics.

5.2 Estimated Models

Table 1 shows the accuracy, i.e. FIT, that was obtained for the 9 different models while they were used as a one-step ahead predictor on the validation data. In addition, for each dataset and model, the order of the model has been specified.

5.2.1 Comparison: Estimation with different input signals

For all three analyzed model structures, Table 1 shows that the model estimated using the binary signal 2 has the best accuracy followed by the model estimated by white noise 3 and then, white

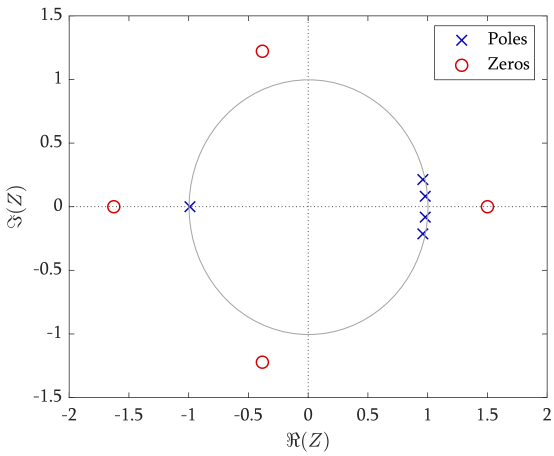
Input signal	Binary signal 2			White noise 2			White noise 3		
Model	ARX	ARMAX	BJ	ARX	ARMAX	BJ	ARX	ARMAX	BJ
FIT (%)	98.65	98.93	98.94	76.64	76.11	76.07	95.5	95.28	95.29
Order	4 1 8	4 4 4 1	5 5 5 5 1	1 1 2	6 6 6 4	4 4 4 4 1	1 5 9	5 4 4 1	4 4 4 4 8

Table 1: FIT and model order of the estimated models based on multiple input signals and model types. The row "Order" contains the degree of the polynomials in (2), where for ARX (n_a, n_b, n_k), for ARMAX (n_a, n_b, n_c, n_k) and for BJ $(n_a, n_b, n_c, n_d, n_k)$ is denoted.

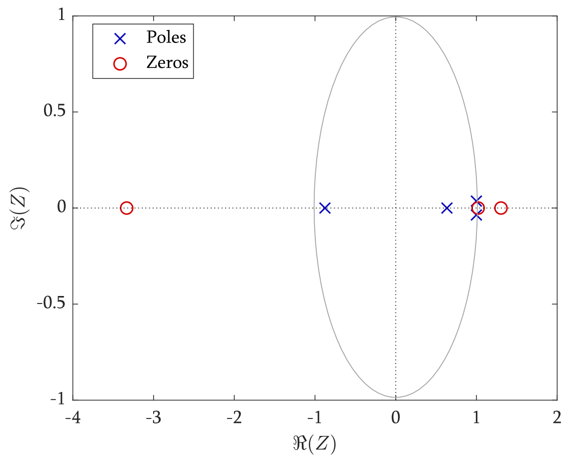
noise 2. Due to its construction, binary signal 2 has more power than the white noise signals at the frequencies where the energy of the frequency response of the system is concentrated. Hence, the accuracy of the estimated system will be better for these frequencies [3, slide 23], resulting in a better total accuracy on the validation signal. The performance of white noise 3 might be better than white noise 2 because the amplitude of white noise 2 might be to small to capture the relevant dynamics of the system.

5.2.2 Comparison: Best and worst estimated system

In the following, we compare the best model (BJ with binary signal 2) with the worst model (BJ white noise 2) that we have estimated. We will refer to these systems in in the following as "best system" and "worst system". To explain why one model performs better than the other, we analyzed the pole-zero plot (Figure 3), the residual plot (Figure 4) and the Bode plot (Figure 5) of both systems.



(a) Binary signal 2 (BJ).

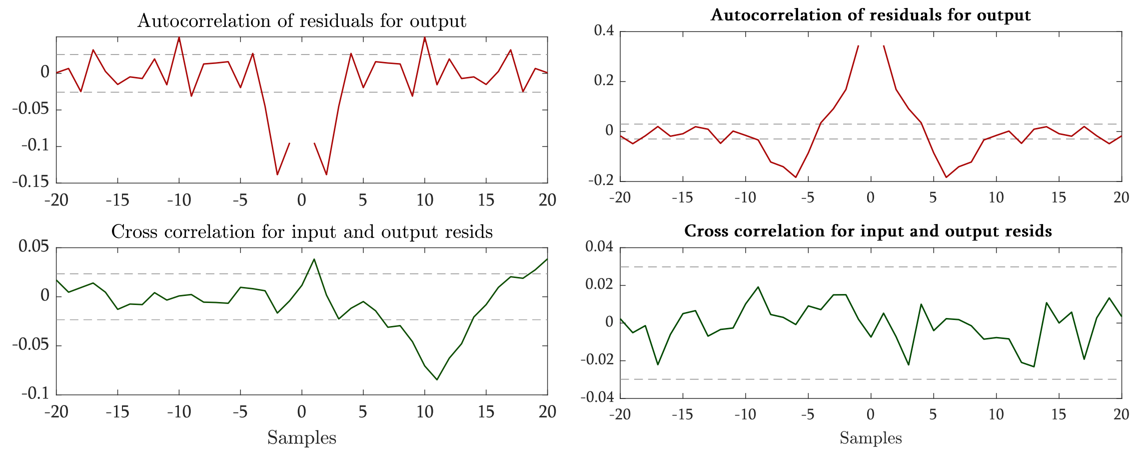


(b) White noise 2 (BJ).

Figure 3: Poles and zeros diagram of the best model and the worst.

For both systems, the poles in the pole-zero plots are inside the unit circle and hence, both systems are stable. The worst system has one zero and two poles near $z = 1$ resulting in a resonance at $f = 40\,\mathrm{rad\,s^{-1}}$ and low and high frequencies are damped. In contrast, the best system, doesn't damp the low frequencies and has only two small resonances.

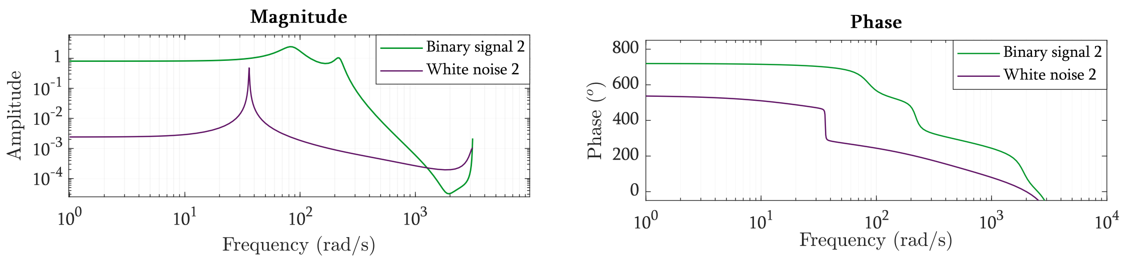
Furthermore, the autocorrelation plot of the residual in Figure 4 show a high autocorrelation for $1 \leq |\tau| \leq 8$ indicating unmodeled dynamics. The autocorrelation of the residual of the best system exceeds the the confidence interval for $1 \leq |\tau| \leq 2$ and it's cross correlation for input and output residual also exceeds the confidence interval for $\tau \in [7, 13]$. We can therefore assume that also the best model doesn't model all dynamics, which could be due to the nonlinearity of the physical system. A better accuracy might be possible by the use of nonlinear models. However, in total the autocorrelation of the output of the best system is smaller than the one of the worst signal, which motivates why the best system models the plant better.



(a) Binary signal 2 (BJ).

(b) White noise 2 (BJ).

Figure 4: Resids diagram of the best model and the worst.



(a) Magnitude.

(b) Phase.

Figure 5: BODE diagram. In green the BJ of the Binary signal 2 and in purple the BJ of the white noise 2.

According to Figure 2, the energy of the system's frequency response is concentrated at low frequencies. This explains the bad accuracy of the worst system because it's frequency response differs especially at the low frequencies from the best system.

5.2.3 Ranking of the 4 best models

Our ranking for the best four models is (from best to worst): 1) The BJ model, 2) the ARMAX model and 3) the ARX model, which are estimated using the binary signal 2 followed by 4) the ARX model estimated by white noise 3.

To rank the models, we first created to groups: Group 1 consists of the models based on the binary signal 2 and group 2 consists of the models based on white noise 3. We decided that the models of group 1 will be on place 1 to 3 of the ranking and one of the models of group 2 is on place 4 because the FIT of the models in group 1 is significantly better than the FIT of the ones in group 2.

Since the FIT for the models in each group is similar, we made a correlation analyse of the residual similar to the one in figure 4. For group 1, we saw that the ARX model performs worse than the other two with a maximum autocorrelation of 0.28 and a cross correlation of 0.1, hence, we ranked it on place 3. The other two models performed similar, therefore, we ranked them by FIT. In group 2, the autocorrelation of the ARX model is smaller than the autocorrelation of the other two, hence, the ARX model describes the dynamic of the system better and we ranked it on place 4.

References

[1] M. Müller, "Modeling of a magnetic levitator," 2018, [Online] Available: https://canvas.kth.se/courses/35049/files/5637379/download [Accessed: 12-10-2022].

[2] H. Hjalmarsson, B. Johansson, P. E. Valenzuela, R. A. Gonzalez, C. R. Rojas, and J. Fredriksson, "Exercise compendium," 2021, [Online] Available: https://canvas.kth.se/courses/35049/files/5720016?wrap=1 [Accessed: 11-10-2022].

[3] C. R. Rojas, "Parameter estimation in linear models," 2022, [Online] Available: https://canvas.kth.se/courses/35049/files/5787465/download?wrap=1 [Accessed: 14-10-2022].

Appendix

A Load data and prefiltering

```
% Load files
[ue, ye, uv, yv] = load_data("lab3_data\binary_signal_2.mat", "u", "y", 0);
[ue, ye, uv, yv] = load_data("lab3_data\white_noise_2.mat", "u2", "y2", 0.25);
[ue, ye, uv, yv] = load_data("lab3_data\white_noise_3.mat", "u", "y", 0.2);

Ts = 0.001; % Sampling period

function [u_est, y_est, u_val, y_val] = load_data(path, name_u, name_y, t_min)

file = load(path, name_u, name_y);

u = file.(name_u);
y = file.(name_y)';

u = reshape(u.',[], 1);
y = reshape(y.',[], 1);

start_idx = floor(length(u) * t_min) + 1;

u = u(start_idx:end);
y = y(start_idx:end);

% Remove mean
u_no_mean = u - mean(u);
y_no_mean = y - mean(y);

split_idx = floor(length(u) /2);

% Divide input and output into estimation and validation data
u_est = u_no_mean(1:split_idx, 1);
u_val = u_no_mean((split_idx+1):end, 1);
y_est = y_no_mean(1:split_idx, 1);
y_val = y_no_mean((split_idx+1):end, 1);

end
```

MATLAB code 3: Load data and prefiltering.