Grading template for laboratory exercise 3
EL2820, Modeling of Dynamical Systems
September 2018

Chang, Hongsheng

951228-2451

Jiang, Sifan

961220-8232

	Pass		Fail	
The report is handed in on time?		yes ☐	no ☐	
Number of authors		$\leq$ 2st ☐	>2 ☐	
Author names and personal identity number filled out?		yes ☐	no ☐	
The report is well structured? The language is understandable?	yes ☐	often ☐	sometimes ☐	no ☐
The figures are clear? (Captions, high resolution, etc.)	yes ☐	often ☐	sometimes ☐	no ☐
The preparation task is solved and motivated?		yes ☐	no ☐	
The working region is defined and motivated?		yes ☐	no ☐	
The sampling time 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 three differemt model structures have been estimated?		yes ☐	no ☐	
The model order of each model is motivated?		yes ☐	no ☐	
A top 4 ranking of the estimated models have been made?		yes ☐	no ☐	
The best and worst models have been compared using Bode plots, and analyze through pole/zeros and correlation(residuals) analysis using (compare) and correlation analysis(resid)?		yes ☐	no ☐	

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

Pass

Signature:

1 Preparation task

- Derivation of a physical model of the magnetic levitator in state-space form:
  From figure in the lab description, we can get:

$$m\ddot{z} = \gamma\dot{z} + E_r - F_{ul} - mg \tag{1}$$
$$m\ddot{y} = \gamma\dot{y} - E_a + F_{lu} - mg \tag{2}$$

Repulsive magnet force is proportional to $m|y-z|^{-4}$, and here, constant T is given to represent the proportion:

$$F_{ul} = F_{lu} = Tm\frac{1}{(y-z)^4} \tag{3}$$

From exercise 2.5, the electromagnetic force can be obtained, where K is a constant and A is the area of the coil:

$$B = \mu_0 \frac{N\gamma^2 I(t)}{2y^3} \tag{4}$$
$$E = \frac{1}{2}A\frac{B^2(t)}{\mu_0} \tag{5}$$
$$E = \frac{AN^2\mu_0\gamma^4}{8}\frac{I^2(t)}{y^6(t)} = Km\frac{I^2(t)}{y^6(t)} \tag{6}$$
$$\text{So:} \quad E_a = Km\frac{I^2(t)}{y^6(t)} \qquad E_r = Km\frac{I^2(t)}{z^6(t)} \tag{7}$$

Then we can obtain state space equations:

$$\dot{a} = \frac{\gamma}{m}\dot{z} + K\frac{I^2}{z^6} - T\frac{1}{(y-z)^4} - g \tag{8}$$
$$\dot{b} = \frac{\gamma}{m}\dot{y} + K\frac{I^2}{y^6} - T\frac{1}{(y-z)^4} - g \tag{9}$$
$$\dot{z} = a \tag{10}$$
$$\dot{y} = b \tag{11}$$

- Suggestion for a suitable model order for a linear model:
  In stationarity, we have:

$$x = \begin{bmatrix} a \\ b \\ z \\ y \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{a} \\ \dot{b} \\ \dot{z} \\ \dot{y} \end{bmatrix} = 0 = \begin{bmatrix} f_1(a,b,z,y,I_0) \\ \vdots \\ f_4(a,b,z,y,I_0) \end{bmatrix} \tag{12}$$

The linearized system is given by:

$$\dot{\mathbf{X}} = A\mathbf{X} + BI(t) \tag{13}$$
$$\mathbf{Y} = C\mathbf{X} + D \tag{14}$$

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial a} & \cdots & \frac{\partial f_1}{\partial y} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial a} & \cdots & \frac{\partial f_4}{\partial y} \end{bmatrix}\Bigg|_{a=a^0,b=b^0,z=z^0,y=y^0,I=I_0} = \begin{bmatrix} \frac{\gamma}{m} & 0 & -\frac{4T}{(y^0-z^0)^5} - \frac{6KI^2}{z^{07}} & \frac{4T}{(y^0-z^0)^5} \\ 0 & \frac{\gamma}{m} & \frac{4T}{(y^0-z^0)^5} & -\frac{4T}{(y^0-z^0)^5} + \frac{6KI^2}{y^{07}} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{15}$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial I} \\ \vdots \\ \frac{\partial f_4}{\partial I} \end{bmatrix}\Bigg|_{a=a^0,b=b^0,z=z^0,y=y^0,I=I_0} = \begin{bmatrix} \frac{2KI}{z^{06}} \\ -\frac{2KI}{z^{06}} \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \qquad D = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{17}$$

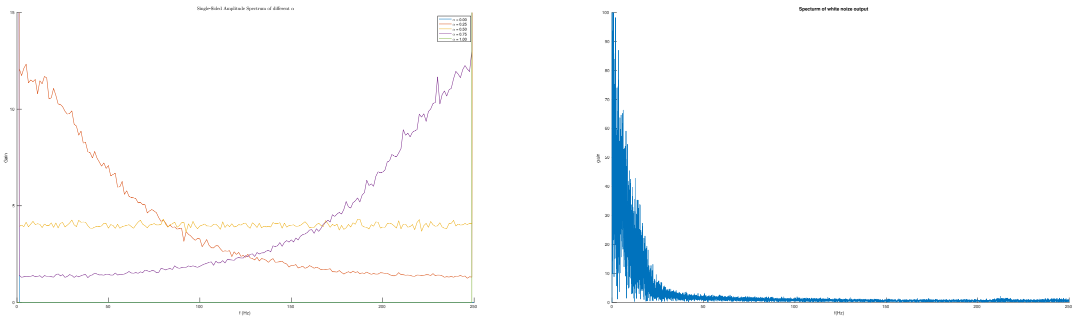
Then the transform function is given as, where $\mathbf{I}$ is the identity matrix and N_i^a is a polynomial of i degree with label a on nominator:

$$G(q) = C^T(q\mathbf{I} - A)^{-1}B + D \tag{18}$$

$$= \frac{N_3^a(q)}{D_4^a(q)} + \frac{N_0^b(q)}{D_4^b(q)} \tag{19}$$

Approximating $G(q)$ with $D_4^a(q) = D_4^b(q)$, the trasfer function would be $G(q) = \frac{N_3(q)}{D_4(q)}$. Plus, since our sampling time would be $2ms$ and the delay time would be $9ms$ in our experiment, base on the equation $n_k = \lfloor \frac{\tau}{h} \rfloor + 1$ from the lecture slides, we get $n_k = 5$. In all, our suggestion of the model order for ARX model could be ARX(4 3 5). Then the ARMAX and BJ models can be modified accordingly.

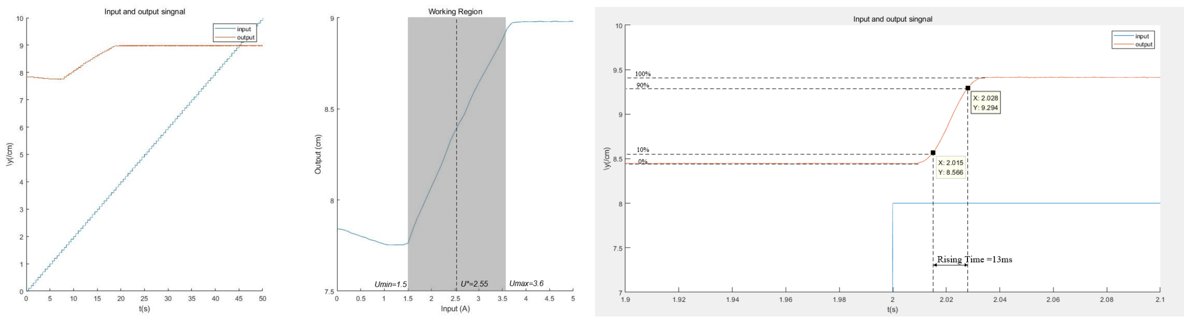
- MATLAB codes for the required functions are attached in the section 7.
- Plot for the spectrum of the binary random signal for the required values of α is shown in Fig. 1a.



(a) Spectrum of binary random signal.

(b) Spectrum of white noise.

Figure 1: Plot of spectrum.



(a) Working region.

(b) Step Response.

Figure 2: Working region and sampling time.

2 Working region

- After the experiment, we choose working region as 1.5A $\sim$ 3.6A.
- Motivation for working region:
  From Fig. 2a, we can see that when $u \in [1.5A, 3.6A]$, the plot have linearity.
- Plot for illustrating working region is shown in Fig. 2a.

3 Sampling time

- In the experiment, we chose $Ts = 2ms$ as sampling time.
- Motivation for sampling time:

  Fig. 2b is obtained with the sampling time of $1ms$, and we can see the rise time is about $13ms$, which means there are about 13 points in one rise time. According to the requirement of "the sampling time is set to give around four to ten samples per rise time" in the lab description, we choose $2ms$ as sampling time so that there will be 6 to 7 samples per rise time.
- Plot of step responses yielding sampling time is shown in Fig.2b.

4 Input signals

- The binary random signal we choose have the similar output spectrum as the one of the white noises, which can be seen from Fig. 1a and Fig. 1b.
- Motivation for chosen input signals:

  Comparing Fig. 1a and Fig. 1b, when $\alpha = 0.25$, the spectrum of binary random signal is simialr to the spectrum of white noise.

5 Estimation and validation data

Information about the amount of data used for identification and validation:

Table 1: Data used for identification and validation.

Input signal	Value (A)	Amount of data	Estimation (s)	Validation (s)
Uniform white noise	[2, 3]	10000	[0, 10]	[10, 20]
Binary random signal with $\alpha = 0.25$	[2, 3]	10000	[0, 10]	[10, 20]

6 Models

6.1 Model Description

- ARX model: $A(q)y[t] = B(q)u[k] + e[k]$, performs a filtering on both input and error with $\frac{1}{A}$ and adjust the transform function with B.
- ARMAX model: $A(q)y[t] = B(q)u[k] + C(q)e[k]$, is more generalized than ARX model with error being adjusted with C.
- BJ model: $y[k] = \frac{B(q)}{F(q)}u[k] + \frac{C(q)}{D(q)}e[k]$, includes all special cases.

6.2 Motivation of Model Orders

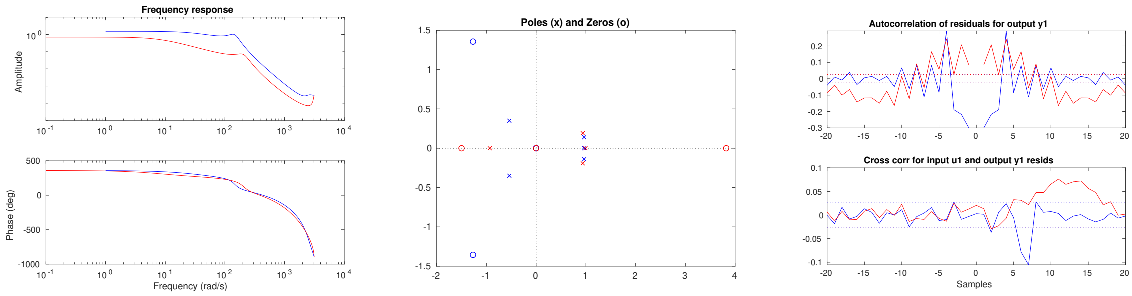
- ARX model: As mentioned in the "Preparation task" section, ARX(4 3 5) is first chosen for both white noise and binary random signal. In the case of white noise, it performs well enough. In the case of binary random signal, ARX(5 3 5) is chosen for obvious performance improvement.

- ARMAX model: Based on ARX model, we try different values on n_c and finally choose ARMAX(4 3 4 5) since performance drops when $n_c > 4$. For same reason, we choose ARMAX(5 3 2 5) for binary random signal.
- BJ model: Based on ARMAX model, we get BJ(3 4 4 4 5) for white noise and BJ(3 2 5 5 5) for binary random signal since we don't need to change the order for each polynomial in each transform function.

6.3 White Noise Signal Validation

In this part, we are going to use white noise signal 2 as the validation data since the process of white noise signal 1 is similar and the performance of white noise signal 2 is better.

The best and worst models in validating with binary random signal are compared in Fig. 3, where the red curves is the best model BJ(3 4 4 4 5) estimated from white noise 1 and the blue curves is the worst model ARX(5 3 5) estimated from binary random signal.



(a) Bode plot. (b) Poles and zeros. (c) Correlation.

Figure 3: Comparison between the best and worst models using white noise as validation data.

According to Bode plot in Fig. 3(a), there is a resonance at around frequency 100 rad/s. By comparison, BJ(3 4 4 4 5) has higher gains around resonance frequency than ARX(5 3 5). Also, according to the poles and zeros in Fig. 3(b), poles are in the unit circle and both models have effective pairs of conjugate poles that contribute to the peak in Bode plot.

Table 2: 1-step Prediction of Validation

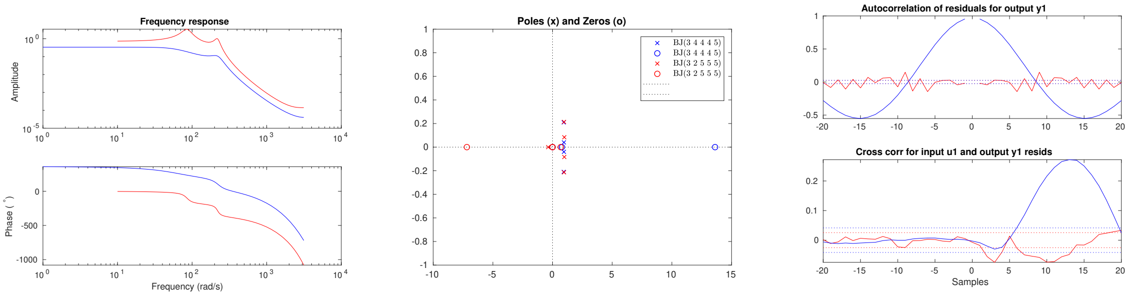
Model	Binary Signal Fit	**White Noise 2 Fit**
BJ(3 4 4 4 5) WN1	97.25%	92.15%
ARX(5 3 5) BR	98.36%	88.83%

Base on the 1-step prediction, both signal performs perfect on random binary validation data. But the ARX(5 3 5) model preforms not so good with white noise 2 data which can also be seen from Fig.3(c).

6.4 Binary Random Signal Validation

The best and worst models in validating with binary random signal are compared in Fig. 4, where the red curves is the best model BJ(3 2 5 5 5) estimated from binary random signal and the blue curves is the worst model BJ(3 4 4 4 5) estimated from WN2.

According to Bode plot in Fig. 4(a), there is a resonance at around frequency 100 rad/s. As the input binary random signal focuses on low frequencies, the variance is relatively small around the peak frequency and relatively big. By comparison, BJ(3 2 5 5 5) estimated from binary random signal has higher gains around resonance frequency than BJ(3 4 4 4 5) estimated from WN2.



(a) Bode plot. (b) Poles and zeros. (c) Correlation.

Figure 4: Comparison between the best and worst models using binary random signal as validation data.

According to poles and zeros in Fig. 4(b), the poles of both models are both in the unit circle. Plus, both models have effective pairs of conjugate poles that contribute to the peak in Bode plot.

Table 3: 1-step Prediction of Validation

Model	**Binary Signal Fit**	White Noise 2 Fit
BJ(3 2 5 5 5) BR	98.54%	90.38%
BJ(3 4 4 4 5) WN2	95.11%	91.73%

Base on the 1-step prediction, both signal performs perfect on random binary validation data and also good with on white noise validation data. However, it is clear according to correlation plots in Fig. 4(c), most correlation of BJ(3 2 5 5 5) is inside the confidence region while there is much outlier in the case of BJ(3 4 4 4 5). Therefore, BJ(3 4 4 4 5) estimated from WN2 is over fitted to binary random data signal.

6.5 Ranking and Comparison

According to validation result shown in Tab.2 and Tab.3, models estimated from binary random signal has relatively higher accuracy than with white noise.

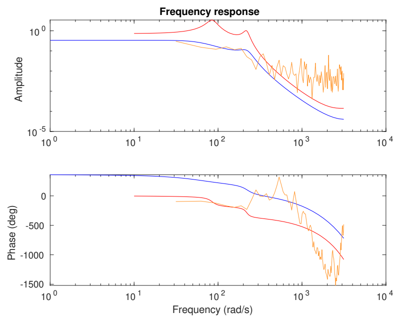
Table 4: Ranking of the Estimated Models with 1-step Prediction

	Validation Data		
Rank	White Noise 1	White Noise 2	Binary Random Signal
1	ARMAX(4 3 4 5) WN2: 89.33%	BJ(3 4 4 4 5) WN1: 92.15%	BJ(3 2 5 5 5) BR: 98.54%
2	BJ(3 4 4 4 5) WN2: 89.11%	ARMAX(4 3 4 5) WN1: 91.90%	ARMAX(5 3 2 5) BR: 98.53%
3	ARX(4 3 5) WN2: 88.91%	BJ(3 4 4 4 5) WN2: 91.73%	ARX(5 3 5) BR: 98.36%
4	BJ(3 4 4 4 5) WN1: 88.80%	ARMAX WN2: 91.71%	ARMAX(4 3 4 5) WN1: 97.93%

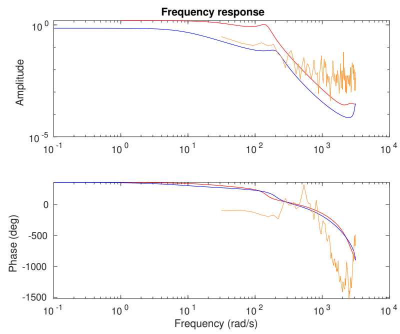
We also use

Table 5: Accuracy comparison of models with different input signals (binary random signal v.s. uniformly distributed white noise).

		Input Validation Signal	
Identification data	Model	White Noise 2 (%)	Binary Random Signal (%)
White Noise 1	ARX(4 3 5)	88.23	97.31
	ARMAX(4 3 4 5)	88.26	97.93
	BJ(3 4 4 4 5)	88.8	97.25
White Noise 2	ARX(4 3 5)	88.91	96.38
	ARMAX(4 3 4 5)	89.33	96.85
	BJ(3 4 4 4 5)	89.11	95.11
Binary Random Signal	ARX(5 3 5)	84.85	98.36
	ARMAX(5 3 2 5)	86.48	98.53
	BJ(3 2 5 5 5)	86.6	98.54



(a) Binary random signal.



(b) White noise.

Figure 5: Comparison the best and worst models with spectrum model (M=inf).

7 Attachment

generateBinarySignal.m

```
function u = generateBinarySignal(alpha, lims, N)
    %% Initialization
    w = 2 * (rand(N,1) > alpha) - 1;
    u = zeros(N,1);

    %% Initial value of u
    if w(1) == -1
        u(1) = lims(2);
    else
        u(1) = lims(1);
    end

    %% Rest values of u
    for i = 2 : N
        if w(i) == -1
            if u(i-1) == lims(1)
                u(i) = lims(2);
            else
                u(i) = lims(1);
            end
        else
            u(i) = u(i-1);
        end
    end
end
```

getAverage.m

```
function bar_v = getAverage(v, tail)
    bar_v = mean(v(length(v)*tail/100 + 1 : end));
end
```

getStationaryAverages.m

```
function bar_y = getStationaryAverages(y_step, Nwr, tail)
    wrLen = length(y_step)/Nwr;
    for i=1:Nwr
        bar_y(i) = getAverage(y_step((i-1)*wrLen+1:(i)*wrLen),tail);
    end
end
```

findWorkingRegion.m

```
close all; clear all; clc;
%% Initialization
Ts = 0.001;
t = 0;
i = 1;
maxCurrent = 10;
stairHeight = 0.1;
stairDuration = 0.2;

%% Create Stair Signal
stairSignal(i) = 0;
while true
    t = t + Ts;
    i = i + 1;
    if t < stairDuration
        stairSignal(i) = stairSignal(i-1);
    else
        if stairSignal(i-1) + stairHeight > maxCurrent
            break
        end
        stairSignal(i) = stairSignal(i-1) + stairHeight;
        t = 0;
    end
end
t = 0 : Ts : (length(stairSignal)-1) * Ts;
t = t.';
stairSignal = stairSignal.';

%% Plot Stair Signal and Time
figure
plot(t,stairSignal)
xlabel('Time (s)')
ylabel('Stair Input (A)')

%% Input Signal Amplitudes
averagedU = getStationaryAverages(stairSignal, (maxCurrent/stairHeight+1),
    0.6);

%% Conducting Experiment
y = getData(stairSignal, Ts);

%% Averaged Output Signal
```

```
averagedY = getStationaryAverages(y, (10/stairHeight+1), 0.6);

%% Plot to Find Working Region
figure
plot(averagedU,averagedY);
xlabel('Input (A)')
ylabel('Output (cm)')
```