Modeling of a Magnetic Levitator

Laboratory Exercise 3

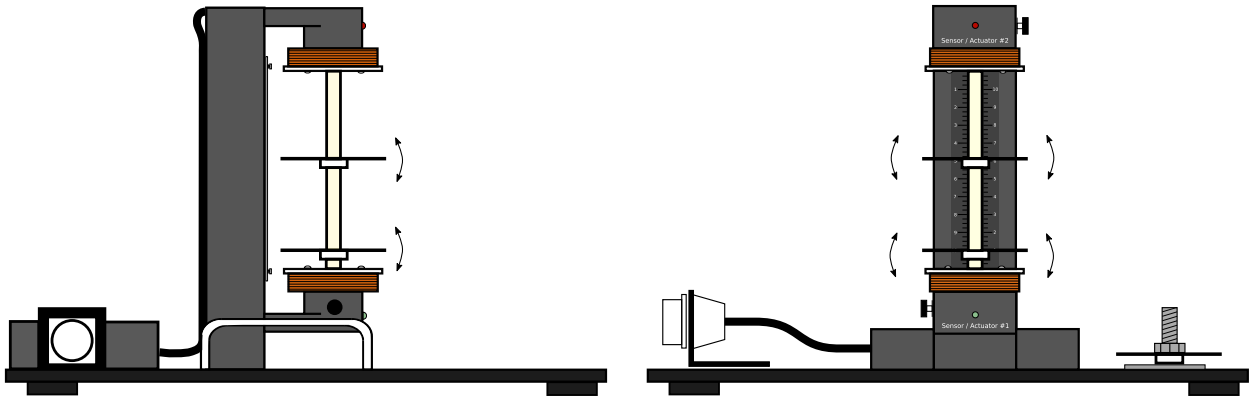
EL2820 – Modelling of Dynamical Systems

September, 2018

Matias Müller

Automatic Control, School of Electrical Engineering, KTH

Goal: To gain experience and insight in system identification techniques, by creating a linear model of a non-linear magnetic levitator. Performing this exercise will provide the student with tools to create linear models from input-output data of a real process.



Contents

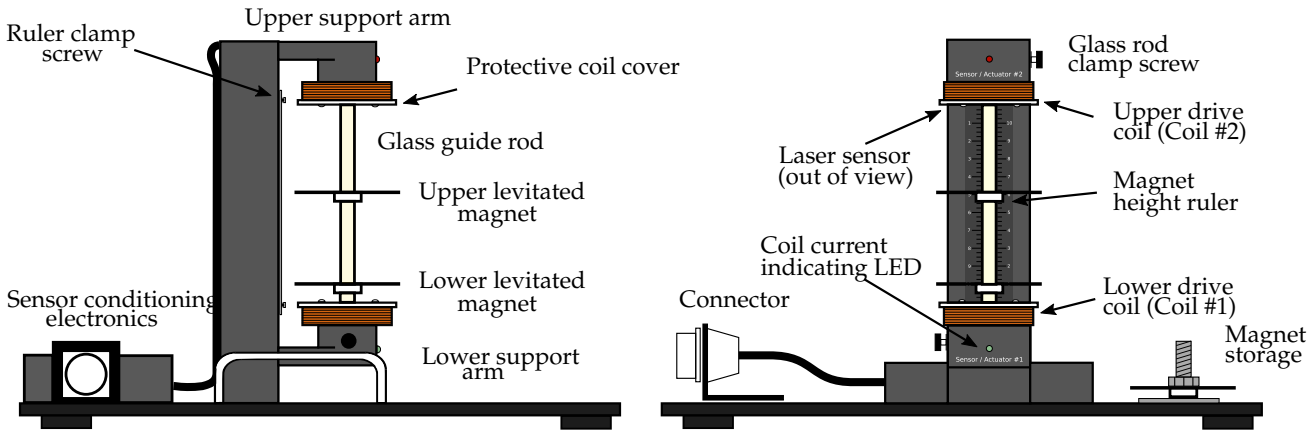


Figure 1: Plant overview

1 Introduction

System identification is an experimental method to construct models from input and output data. In this laboratory exercise, you will collect appropriate input and output data to construct a linear model of the Magnetic Levitator process. The model should be able to describe the behavior of the process around a working point. The model is constructed with the help of the System Identification Toolbox in MATLAB. The goal of the exercise is to give you practical experience in how to use system identification to devise models. The exercise intends to illustrate:

1. How to process data to be used for system identification.
2. How to choose an appropriate class of models.
3. How to construct models from acquired data.
4. How to validate and assess the quality of the constructed models.

Students, in groups not larger than two, must hand in a final report, containing the results of the exercise, as well as **an appropriate motivation** for every decision made along the modeling process.

Tip: Remember that there are no bad results, but only bad justifications.

2 System Overview

The complete experimental system is composed of three subsystems. The first of these is the electromechanical plant, described in Section 2.1. The second one corresponds to a real-time Data Acquisition system (DAQ) containing a digital signal processor. This subsystem transforms the analog upper magnet position (in volts), in Figure 1, into a digital signal which the computer can read. Similarly, the digital input signal is transformed to an analog current going directly into the system (using a current regulator!). Additionally, the DAQ works as a power supplier, providing the plant with enough current and voltage to work. The third subsystem is MATLAB, running in a provided computer, next to the plant.

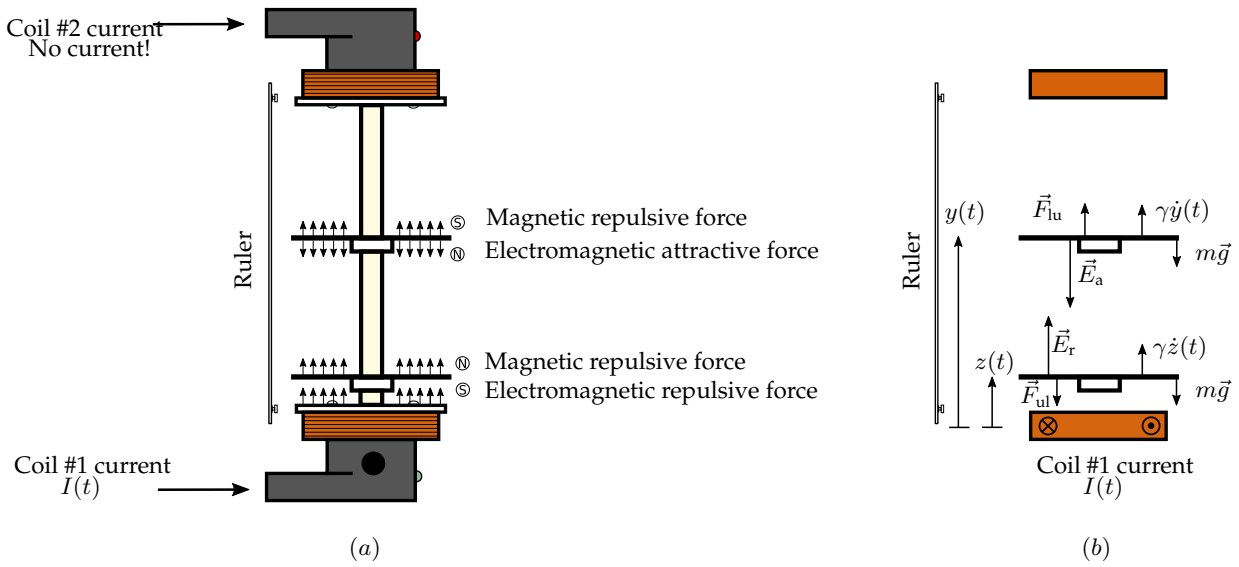


Figure 2: Plant forces interaction

2.1 Plant Description

The process consists of a drive coil producing a magnetic field in response to a DC current. In Fig. 1, a full view of the plant is provided, including the most important components regarding the laboratory exercise. By energizing the lower coil, the lower magnet is levitated through a repulsive magnetic force, moving along a glass guide rod. At the same time, the lower magnet is repelling the upper one due to the fact that both north poles are facing each other, as Fig. 2a shows. Along the experiment, the upper coil will not be excited with any input, so there will be no magnetic field attracting or repelling the magnets from above. A more detailed information about forces interaction is presented in Fig. 2.

The magnets are shaped as disks, and, as depicted in Fig. **??**, most of a magnet area is covered by a reflective laminate. The current height of each magnet is obtained by projecting a laser light over one of the sides (the upper side in the upper magnet), and by measuring the length of the illuminated line. In this sense, the higher the upper magnet is, the smaller the line becomes.

The input of the system is the current through the coil, while the output corresponds to the height at which the upper magnet levitates. Under this set-up, the system is open-loop stable.

3 Laboratory Exercise: Procedure

During input-output linear modeling, it is specially important to account for non-linearities of the plant. The obtained model should be valid for small deviations around a working point. In this way, system identification experiments are made by introducing small perturbations around this point. However, in order to maximize the information coming from the system output (and not from measurement noise), these deviations should be as large as possible.

The following procedure for this laboratory exercise is:

1. Find a working region from the given dataset.
2. Choose data you will use for identification from the given dataset.
3. Choose a model structure and model order.
4. Estimate a parametric model.
5. Validate your model.

System identification is an iterative process, where especially the three last steps have to be repeated until an acceptable model has been found.

Observation: Every experiment is performed with an input signal `u`, giving an output `y`. The set containing these to vectors (`u`,`y`) is called **data**. Data (`u`,`y`) can be divided in two halves: (`u(1:N/2)`,`y(1:N/2)`) for identification and (`u(N/2+1:end)`,`y(N/2+1:end)`) for validation. **Identification and validation data should not be mixed**. This process is also known as **cross validation**.

Tip: For each experiment performed, data as a `.mat` file is saved, containing at least the input signal `u`, the output signal `y`, and the sampling period `Ts`.

3.1 Finding a Working Region

When estimating a linear model of a nonlinear system, it is important that the process is kept sufficiently close to the working point. This is ensured by selecting a small enough range of the input signal such that a linear approximation of the nonlinear system is valid. On the other hand, to ensure that the system is sufficiently excited in the identification experiment, we want the range of the input signal to be as big as possible. Consequently, we want to find the largest range of the input signal that ensures that a linear approximation is valid. This range is called the working or linear region, and its center is called the working point.

One way to find the working region is to perform several step response experiments. If the system is linear, then the steady output amplitude should scale proportionally to the amplitude of the step amplitude, whenever the step amplitude is **contained in the working region**. This is one of the properties of a linear system. The working region is contained within the interval $[\check{u}, \hat{u}]$, where $\check{u}$ and $\hat{u}$ are the lower and upper bounds for the input signal (not necessarily equal to the lower/upper bound of the working region), provided by the manufacturer/company/university (whoever able to provide them!). The step response experiments are therefore performed within this interval.

3.1.1 Experimenting with the working region

Due to the plant specifications, the input signal must be bounded between 0 and 10 Amperes. Then, any working region you find must be completely contained in this interval.

Starting from amplitude equal to zero, we defined the amplitudes of the steps you will apply to the system and collect them in the set $\{\bar{u}_i\}_{i=1,\dots,N_{\text{wr}}} \in [\check{u}, \hat{u}]$, where N_{wr} is the number of different steps. We choose $N_{\text{wr}} = 20$. As Fig. 3 suggests, it is not necessary for elements in $\{\bar{u}_i\}$ to be equally spaced, but we will attain to this case for simplicity.

We denote the stationary value of the step response of amplitude $\bar{u}_i$ as $\bar{y}_i$. That is, $\bar{u}_i$ denotes the amplitude of the i-th step used as input signal, while $\bar{y}_i$ is the average stationary amplitude of the i-th step response (we consider the average because of noise). Make sure to wait for the transient to settle before switching to the next step in the input signal. After the step response experiments, plot $\{\bar{y}_i\}$ versus $\{\bar{u}_i\}$. The working region is where the relationship between $\{\bar{y}_i\}$ and $\{\bar{u}_i\}_i$ is *reasonably* linear, that is, where the plot $\{\bar{y}_i\}$ v/s $\{\bar{u}_i\}_i$ is approximately a straight line. We define the boundaries of this region as u_{min} and u_{max}. Figure 3 shows an example on how to choose a working region in an arbitrary system. Notice that the chosen working region in the example does not contain a straight lane, but something *close* to that.

Tip: The step response contains a transient which must not be included when computing the average stationary step response because, by definition, this first part is not steady. You must discard the output data belonging to the transient phase. In the other hand, the step response is noisy and it can be hard to find the stationary value $\bar{y}_i$. To tackle this problem, compute the empirical mean of the remaining output data.

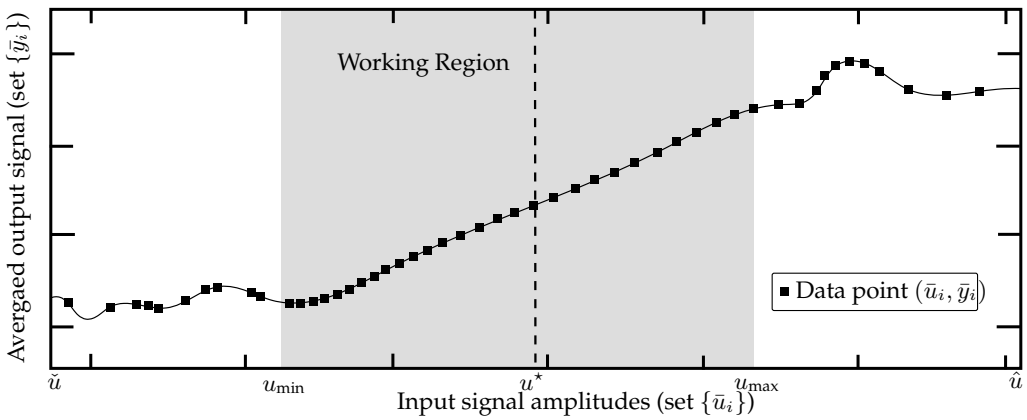


Figure 3: Working region for an arbitrary system, where 50 different amplitudes were experimented. Each point is indicated with a red square. Note that amplitudes are not equally distributed over $[\check{u}, \hat{u}]$. The underlying solid black line tries to fit the points, but it does not mean that there is such a line generating data. The black dotted line splits the working region in two halves.

In the following experiments, make sure that the input signal used in the identification experiments never exceeds the values on the boundary of the working region. Justify your choice for the working region by presenting a plot with the same information of the one in Fig. 3.

Tip: All step response experiments can be performed in one shot by using a stair signal as input. To obtain the average step responses, you will need to design a function

```
bar_y = getStationaryAverages(y_step, Nwr, tail)
```

detailed in the preparation tasks. The different values of the stair signal are the step amplitudes. You just need to make sure there is enough time between values in the stair signal, so that the transient effect disappears in every step response.

3.2 Sampling Period

Since we essentially want to capture as much dynamics as possible from the process, we would like to choose a sampling period of zero (equivalently, an infinite sampling frequency). Due to the limitations of the devices processing/storing data, this is intractable.

The sampling time is often chosen by first performing a step response experiment with a short sampling time. Then the sampling time is set to give around four to ten samples per rise time. The rise time can be obtained as Figure 4 suggests. From this initial step response experiment, it is also possible to get an estimation of the time delay of the process. We set the offset of the input signal to the value of the working point and choose an appropriate amplitude the input signal where the system is approximately linear. We make a step in the input signal with the chosen amplitude. We find the rise time and choose an appropriate sampling time for the next experiments. Note that the step response is noisy and it can be hard to find the rise time. In case you think it is necessary, perform several step response experiments and then determine the rise time of the averaged output signal of these experiments (or only one experiment with several binary steps). For the experiments in the given dataset, the sampling time is 0.001 [s]. You do not have to give motivation in the report.

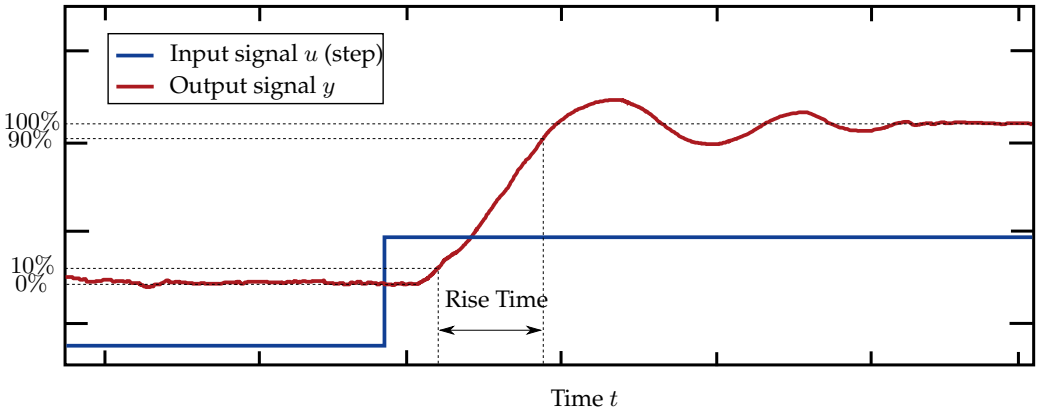


Figure 4: Let the stationary value of the step response be indexed as 100%, and the stationary initial value of the step response be indexed as 0%. The rise time is the time it takes for the step response to go from 10%, for the first time, to 90%, for the first time.

3.3 Input Signal

The input signals to be used in all identification experiments during this exercise are random signals. The goal of this task is to illustrate that different input signals derive different models, some of them more accurate. In this Lab we will restrict our attention to the following classes: **uniformly distributed white noise** and **binary random signals** (which are detailed in Section 3.3.3 and Section 3.3.4). The students' task is to choose signals of each class from the given dataset, and to discuss the accuracy of the different models derived from these experiments.

Summarizing, you will choose:

- 2 input being uniformly distributed white noise, and
- 1 input being a binary random signal.

In the following subsections you will find some guidelines to choose your experiments (*i.e.*, to choose your input signals).

3.3.1 Offset

The averaged value of the input signal should be in the middle of the working region, that is at

$$u^\star = \frac{u_{\max} + u_{\min}}{2}.$$

The offset of the input signal should be set to the value of the working point, u_{wp}.

3.3.2 Amplitude

The amplitude is set such that the input signal used in the identification experiments stays within the working region. Note that the same amplitude setting yields different variances of the generated and uniformly distributed white noise.

3.3.3 Uniformly Distributed White Noise

By using the command `u = rand(N,1)`, MATLAB will assign variable `u` to be a vector with N entries, where each of the entries is the realization of a random variable distributed uniformly between zero and one. Here, when choosing the input signal, it is necessary to consider the working region.

3.3.4 Binary Random Signal

Here, we described how a binary random signal is generated. You do not need to generate signals; instead, you have to choose an appropriate α. In the following text, we describe how to generate binary random signal. This is not necessary to read, but pay attention to Figure 5.

Let $w := (w_t)_{t\in\mathbb{Z}}$ be a discrete stochastic process, where

$$w_t = \begin{cases} -1, & \text{with probability } \alpha \\ 1, & \text{with probability } 1-\alpha \end{cases}$$

for $\alpha \in [0, 1]$. We define an auxiliary stochastic process $s = (s_t)_{t\in\mathbb{Z}}$ defined as

$$s_t = s_{t-1}w_t \tag{1}$$
$$s_0 = 1. \tag{2}$$

Clearly, at each time t, s_{t+1} switches to $-s_t$ with probability α. In words, the higher the value of α, the more likely the function is to switch at every time instant. Indeed, by choosing $\alpha = 1$, s becomes a non-random binary signal, switching from one extreme to the other at every t. Similarly, by assigning $\alpha = 0$, s also becomes non random signal that remains constant for any t.

Intuitively, α can be designed to characterize the spectral properties of the stochastic process s, which is exactly what we will exploit in this part of the lab. It is also possible to recover a closed expression for the spectrum of s as a function of α, given by

$$\Phi_s(\omega) = \frac{4\alpha(1-\alpha)}{|1-(1-2\alpha)\mathrm{e}^{j\omega}|^2}. \tag{3}$$

Notice that the process is asymptotically stationary, so its statistics do not depend on the initial state s_0 (we could have chosen $s_0 = -1$ as well).

Now signal u can be designed as a linear transformation of s: $u_t = As_t + B$, for some constants A and B in $\mathbb{R}$, so that the designed signals have the correct offset and amplitude.

To choose an appropriate α for a random binary signal, consider spectrum of a binary random signal for $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$, see Figure 5.

3.4 Identification Experiments

Remember to have enough data for both model estimation and validation. You should have **at least 10 seconds of estimation data and 7 seconds of validation data**.

First, choose the uniformly distributed input signal data from the given input set according to the chosen working region (the range should be as big as possible).

Process input `u` and output `y` by using command

```
Y = fft(y - mean(y)).
```

Here `Y` corresponds to the Fourier transform of the output (see `https://se.mathworks.com/help/matlab/ref/fft.html`). Plot the absolute value of vector `Y` (use command `abs`), and obtain, by inspection, the band where the output concentrates most of the spectral energy. This give us an insight on where the system itself allocates most of the energy.

Observation: As MATLAB's documentation shows, `Y` contains the Fourier transform at frequencies $\omega_n := 2\pi n/N$, $n = 0, \ldots, N-1$. Then, you should look at the first $N/2$ points.

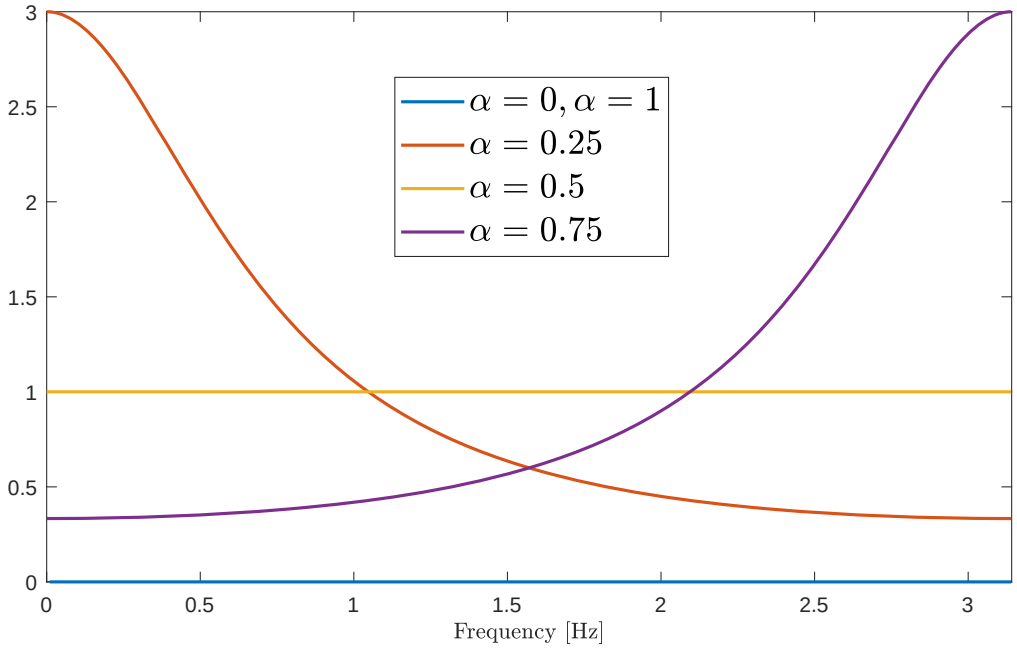


Figure 5: Spectrum of a binary random signal for $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$

We now use this information to design a random binary signal that allocates most of its power in the band where the system does. This is done by identifying a suitable value for α, say $\alpha^\star$. For this task, choose one value of $\alpha^\star$ in the set $\{0, 0.25, 0.5, 0.75, 1\}$, based on the first part.

We will use these datasets to compare the performance for one model structure when the input signal is chosen as a random binary signal versus the one designed with uniformly distributed white noise, according to Section 3.7. Can you tell which model performs better? Remember to use the same validation data in order to compare both methods (different from the identification data!).

3.5 Model Structure

To find a good model structure, always start with a simple structure, for example, ARX. If this is not good enough, try something a bit more complicated, for example, ARMAX. If this is still not good enough, increase the flexibility in the model even more, for example, BJ. Start with a low model order and increase gradually. The model structure can be suggested from a simple preliminary physical mode, like the one you have to derive in the preparation tasks.

Choosing a model order is an art rather than technical knowledge. The insight on how to choose a proper model order is given in the computer exercises.

Choose 3 different model structures (combine then as you wish: ARX, ARMAX, etc.) for estimating models.

3.6 Model Estimation

Now you are ready to estimate and validate the models as taught in the computer exercises. You can use the command-based version of the System Identification Toolbox if you do not want to use its GUI (graphical user interface). Since we have 3 datasets and 3 model structures, you will end up with 9 models for exactly the same plant.

3.7 Model Validation

Validate your model by comparing how similar the simulated output is to the real output, for each of your models.

> **Observation:** Do not forget to use the **validation data** in this section. Validation data is a set of data points used to evaluate the performance of a model, and **it should not have been used for identification**. By comparing models performance with **identification data**, the most complex model will always outperform others, since it overfits data.

Rank your **top 4 models** by comparing *how close* the true output (in the validation data!) is to the simulated output (by using the validation data!). The simulated output corresponds to a one-step ahead predictor:

$$\hat{y}_t = (\hat{H}(q) - 1)\hat{H}^{-1}(q)y_t + \hat{G}(q)\hat{H}^{-1}(q)u_t, \tag{4}$$

where, y is the output signal in the validation data, $\hat{y}$ is the estimated data, and $\hat{G}$ and $\hat{H}$ are models for G and H. Mathematically, the notion of *closeness* is defined by the FIT coefficient, given by

$$\text{FIT} = 100\ \frac{1 - \|\hat{y} - y\|_2}{\|y - \mathbb{E}\{y\}\|_2}. \tag{5}$$

A model completely explaining the validation data has a FIT coefficient of 100. Note also that, for output error (OE) models, it holds that $H(q) \equiv \hat{H}(q) \equiv 1$, and then the one-step ahead predictor equals to the simulation $\hat{G}(q)u(t)$.

> **Tip:** Command `compare(data, sys, prediction_horizon)` in MATLAB returns (among other arguments) the FIT coefficient of model `sys` using (validation) `data`. Variable `prediction_horizon` must be set to 1. For more information visit `http://se.mathworks.com/help/ident/ref/compare.html`.

Choose one model structure and compare the fit coefficients of models derived with one the binary random signal and with one of the uniformly distributed white noise signals you designed. Which one is higher? Discuss about this result.

Finally, pick your best and your worst model (in terms of fit) and perform a residual and a pole-zeros analysis. Explain why is one model better than the other one according to this information.

4 Preparation Tasks

Although not necessary, a good understanding of the underlying phenomena derived from physical modeling simplifies tasks in parametric modeling, such as, *e.g.*, the decision of the model structure. On the other hand, experimenting takes a lot of time, however, many aspects can be designed beforehand.

Following the sentences above, the following preparatory tasks are thought to help you in taking decisions for parametric modeling and to perform experiments efficiently. It is then crucial for you to thoroughly prepare before the laboratory exercises. Do the preparation tasks before the laboratory session!

1. Consider figure 2*b*, detailing the forces interacting in the experiment. Here, $\vec{F}_{\text{lu}}$ denotes the magnetic force from the lower magnet to the upper one, while $\vec{F}_{\text{ul}}$ is its reactive force, *i.e.*, from the upper magnet to the lower one. Both magnets repel each other since their N sides are facing each other's. These forces must be derived using Gilbert's model for two cylindrical magnets (see, *e.g.*, `https://en.wikipedia.org/wiki/Force_between_magnets`), when the approximation $L \ll |y - z|$, $\forall t$. In this case, the repulsive magnetic force is proportional to $|y-z|^{-4}$ and, for simplicity, the effective magnetic dipoles are m (equal to the mass of each disk). $\vec{E}_{\text{r}}$ and $\vec{E}_{\text{a}}$ denote the electromagnetic force

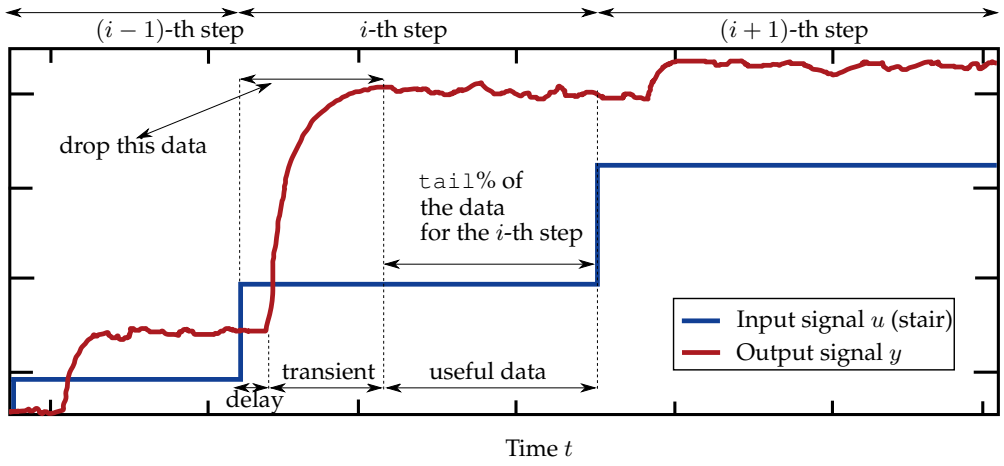


Figure 6: Stair signal as input signal, and its output response, for a stair signal with three different levels $\{\bar{u}_1, \bar{u}_2, \bar{u}_3\}$. The transient data and the delayed data must be dropped down in the computation of the average stationary step response.

repelling the lower magnet and attracting the upper one, respectively. Friction forces are denoted by $\gamma\dot{y}$ and $\gamma\dot{z}$, where γ is the friction coefficient of the air and the glass rod, and where $y(t)$ and $z(t)$ are the height of the upper and lower magnet, respectively. Additionally, by considering m to be the mass of each magnet (assume both of them equal), the gravity force affecting each magnet is $m\vec{g}$.

Derive a state space model describing the mathematical relationship between the current $I(t)$ going into the lower coil, and the height $y(t)$ of the upper magnet. You can base your procedure on Problem 1.4 of the exercise compendium.

- Is the model linear? If it is not, what originates non-linearities?
- Does your linear model (or a linearized version of it) suggest a model structure for the identification process? Which one?
- Is there any physical set-up similar to the one we are using in the magnetic levitator?

2. Design a function of the form

```
bar_v = getAverage(v, tail)
```

where `bar_v` equals to the empirical mean of the last `tail`% samples of v. Attach the code in the final report.

3. Using the last function, design a function of the form

```
bar_y = getStationaryAverages(y_step, Nwr, tail)
```

where `y_step` is the output signal obtained when the plant is excited with a stair signal with `Nwr` different levels. This function returns a vector `bar_y` with `Nwr` entries, where each component corresponds to the average of each stationary step response. The average is computed by using the function `getAverage` on every step response in the output. Assume that the length of the output signal is a multiple of `Nwr`, that it, the length of each step is given by `length(y)/Nwr`.

Figure 6 shows how the function should process the i-th step in the data, for the i-th step. The function should do the same for all $i = 1, \ldots,$ `Nrw`.

You must use this code to calculate the range of the working region.

5 Report

The results must be submitted as a report in Canvas, including the solutions for the preparation tasks. The report must be written in English, using the report template available on the course homepage (Latex or MS Word). The revised report will either get the grade *pass* or *fail*, according to the quality and the content of it, which will be notified to you via Canvas. In case you are graded with a *fail*, the report must be corrected and resubmitted (see the course homepage for deadlines). In case you fail twice, you have to redo the exercise next year.

Students are advised to collaborate in pairs.

Note that the following are the minimum requirements:

- The report should be concise.
- Maximum 6 pages, including figures, plus attached preparatory-tasks codes.
- One report per project group.
- A pdf-version of the report should be submitted to Canvas before the deadline specified on the homepage. **Do not send it by e-mail**.
- The report must include
  - the grading template of the last page
  - solutions for the preparatory tasks (see Section 4)
  - motivation for the your choice of the working region (it must include a plot)
  - detailed description of why the 3 chosen input signals were selected
  - information about the amount of data used for identification and validation
  - 3 model structures
  - a motivation for the choice of model order of each model structure used
  - a rank of the top 4 models in terms of the fit coefficient
  - for one model structure, a comparison between the accuracy of models obtained with different input signals (binary random signal `v/s` uniformly distributed white noise)
  - a comparison between the best and the worst model using plots of Bode diagrams, poles and zeros, and correlation analysis (resid)

You do not have to use the same validation data to compare 9 models. However, you have to separate identification and validation data and explain which validation data you have used for each model.

Grading template for laboratory exercise 3
EL1820, Modeling of Dynamical Systems
September 2018

Name 1:

Personal number 1:

Name 2:

Personal number 2:

	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 ☐
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 different 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 analyzed through pole/zeros and correlation (residuals) analysis using (`compare`) and correlation analysis (`resid`)?	yes ☐	no ☐
	Pass	**Fail**
First review	☐	☐
.....	.....	.....
	Pass	**Fail**
Second review (if failed in the first review)	☐	☐