System Identification
227-0689-00L
Midterm examination: instructions

Due: 10:00 on Wednesday, November 7^{th}, 2018

Overview

The midterm exam is composed of two problems. To answer these problem you need to write separate Matlab functions (the details will be provided in following pages).

Each function will analyze the data associated with the corresponding problem part.

This work must be done individually. You may not discuss these problems with anyone else.

Your functions will be submitted by email. In this email you must also include a scan of the signed declaration of originality form. In the form the title of work should be specified as `System Identification midterm exam HS2018` and the **scanned** document should be named as `HS2018_SysID_midterm_DO_LegiNumber.pdf`, where "LegiNumber" must be your legi-number, without any dash, slash, backslash, etc., e.g.,

`HS2018_SysID_midterm_DO_11235813.pdf`

If you do not have a legi-number, before the due date of exam, you should send an email to `sysid@ee.ethz.ch` and ask for a temporary number to be assigned.

Your submission must be emailed to `sysid@ee.ethz.ch` by the due date and time given above. The subject of the email must be `SystemID HS2018 Midterm Exam`. The email should contain given exact number of separate attachments, the requested Matlab m-files and a PDF (the details will be provided in the following pages).

Your grade for the midterm will be evaluated based on your performance in the two problems weighted in the following way: 50% for Problem 1 and 50% for Problem 2.

Downloadable data

On the Piazza course website

`https://piazza.com/ethz.ch/fall2018/227068900l/resources`

the data files required for the individual parts will be provided. The data files contain the following variables.

File	Problem	Variables
`midterm_2018_Prob1.mat`	Problem 1:	`p1_u, p1_y, p1_z`
`midterm_2018_Prob2.mat`	Problem 2:	`p2_G_mag, p2_omega, p2_y_step, p2_t_step`

Matlab function format

Write Matlab code as functions and call them `HS2018_SysID_midterm_p?_LegiNumber.m`, where the question mark corresponds to the problem number.

The description of the individual problems (provided in the following pages) will specify the variables that must be returned by your functions. The variables needed by the problem will be passed to you as arguments, in a given order which will be provided later versions of exam instructions.

Include your own functions in the file after the main function. This means that Matlab treats them as subfunctions and calls them instead of any other functions in the path that have the same name. This ensures that you will be calling your own functions and not someone else's function. See the Matlab documentation on functions and programming for examples of this.

The following are generic considerations for the behavior of your code. They apply to each of the problems.

- As your function runs it must print out in the command window (use the `disp` or `printf` functions) a description of what it is doing and what choices you are making (i.e. number of points, frequency range, DFT calculations, windowing, model order, model parameters, etc.). You will be graded on the quality of your explanation.
- For any figures requested in the problem use commands `figure(1)`, `figure(2)`, etc. to ensure that all figures remain visible on the screen after your code terminates.
- Each Matlab figure can contain at most two plots.
- Specify all axis limits via the Matlab command `axis(...)`. Auto-scaling must be avoided as it works differently on different machines.

- If your plot uses a legend use the `'Location'` flag in the `legend` command to make sure that relevant data is not covered.
- The script should run unattended and at the end the command window will describe what you have done, with any requested figures visible.
- Do not define any global variables in your functions or subfunctions.
- If you skip one part of a problem, a dummy variable (=0) of appropriate dimension must still be returned.

Before submitting your file, restart MATLAB and test it by running it in a folder/directory with no other files. Make sure that the workspace contains only the variables loaded from the appropriate data file. This will ensure that it can run in a stand-alone manner.

Problem 1 (Weight: 50%)

Consider an identification experiment conducted on the linear time-invariant, discrete-time system depicted in Figure 1 below. The input is $u(k)$, and two experiments are performed. One gives the output $y(k)$ and the other gives the output $z(k)$. Both measurements include noise $v(k)$, $k = 0 \ldots N-1$, where N is the data length.

The goal is to experimentally estimate and validate the unknown transfer function $G(e^{j\omega})$ using system identification methods. For this reason, two experiments were conducted using the setup depicted in Figure 1 for two different realizations of the noise: $v_1(k)$ and $v_2(k)$. Assume that the system is initially at rest – that is, for all $k < 0$, $u(k) = y(k) = v(k) = 0$.

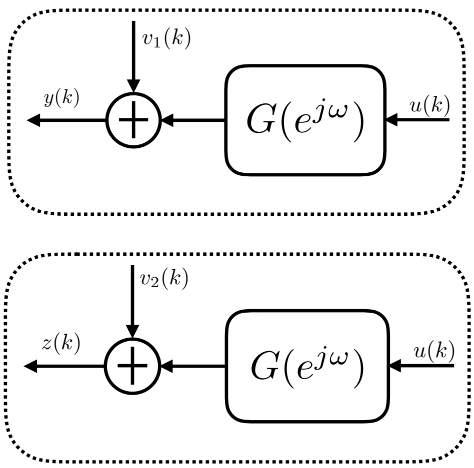

Figure 1: Experiment 1 and 2 under consideration

For this problem, you will write a MATLAB function of the following form:

```
[p1_G_y,p1_omega,p1_best_gamma, p1_G_y_best_hann] =
HS2018_SysID_midterm_p1_LegiNumber(p1_u, p1_y, p1_z)
```

You are given the following experiment data of length N, acquired with a sampling time $T_s = 2s$:

- `p1_u`: the given input signal $u(k)$.
- `p1_y`: the system output $y(k)$ in response to `p1_u` for one realization of noise $v(k) = v_1(k)$.
- `p1_z`: the system output $z(k)$ in response to `p1_u` for a second realization of noise $v(k) = v_2(k)$.

Your function should return the following variables in the order listed:

- `p1_G_y`: the unsmoothed plant estimate calculated from the output $y(k)$ at the frequencies `p1_omega`.
- `p1_omega`: at least 30 equally-spaced frequencies in the range $[0.01, 0.8]$ [rad/sec] where the plant estimates `p1_G_y` and `p1_G_y_best_hann` are provided.
- `p1_best_gamma`: the optimal choice of the Hann window width based on a cross-validation between the two experiments.
- `p1_G_y_best_hann`: the plant estimate corresponding to the output $y(k)$, smoothed using the optimal choice of Hann window width.

1. We conduct an experiment on the plant $G(e^{j\omega})$ using the input $u(k)$. Without using time-domain or frequency-domain windowing, we wish to determine the best possible estimate $\hat{G}_y(e^{j\omega})$ from $y(k)$.
   - Provide the transfer function estimate $\hat{G}_y(e^{j\omega})$ for at least 30 equally spaced frequencies in the range $[0.01, 0.8]$ [rad/sec], using solely $u(k)$ and $y(k)$. Choose a method that minimizes the mean square error defined by

     $$\mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}\left|\hat{G}_y(e^{j\omega_i}) - G(e^{j\omega_i})\right|^2$$

     where $\hat{G}_y(e^{j\omega})$ is your estimate of $G(e^{j\omega})$, and ω_i is the i-th component of the frequency vector ω of length M at which your estimate is defined. Return $\hat{G}_y(e^{j\omega})$ as `p1_G_y`, and ω as `p1_omega`.
   - Justify your choice of method and explain your reasoning in the command window of MATLAB.
2. Now, we use frequency-domain Hann windowing to smooth the transfer function estimate $\hat{G}_y(e^{j\omega})$ found in Experiment 1. Choose the optimal γ by cross-validation using the unwindowed $\hat{G}_z(e^{j\omega})$ which is estimated using $z(k)$ in the second experiment. To do this:
   - For each Hann window width $\gamma \in \{10, 20, 50, 100, 200\}$, compute the mean square error between the y data estimate and an estimate based on the z data. This is defined by

     $$\mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}\left|\tilde{G}_y^{H_\gamma}(e^{j\omega_i}) - \hat{G}_z(e^{j\omega_i})\right|^2$$

     where
     - $\tilde{G}_y^{H_\gamma}(e^{j\omega})$ is found by conducting frequency domain windowing on $\hat{G}_y(e^{j\omega})$ using a Hann window of width γ
     - $\hat{G}_z(e^{j\omega})$ is the unsmoothed transfer function estimate found using the same method you chose in Part 1, but evaluated on solely $u(k)$ and $z(k)$ instead.
     - M is the length of the frequency vector ω as in Part 1, with all ω_i equally spaced in $[0.01, 0.8]$ [rad/sec].
   - From the set of γ given above, return as `p1_best_gamma` the optimal window width γ^* that minimizes the above formula for the mean square error.

- Return as p1_G_y_best_hann the corresponding estimate of $\tilde{G}_y^{H_\gamma^*}(e^{j\omega})$.
- Explain your results and reasoning in the command window of MATLAB, using plots if helpful to illustrate your points.

Your grade for this problem will depend on:

- The correctness of the method you proposed.
- The accuracy of your estimates, which will be computed as the mean square error defined by

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{M}\left|G_{\text{est}}(e^{j\omega_i}) - G(e^{j\omega_i})\right|^2 .$$

  The MSE will be computed twice: once each for $G_{\text{est}}(e^{j\omega})$ equal to $\hat{G}_y(e^{j\omega})$ and $\tilde{G}_y^{H_\gamma^*}(e^{j\omega})$.

- The quality and clarity of your explanations including any figures you may choose to show.

Problem 2 (Weight: 50%)

Write a function of the following form.

```
[p2_T, p2_u] = HS2018_SysID_midterm_p2_LegiNumber(p2_G_mag, p2_omega, p2_y_step,
                                                   p2_t_step)
```

Make sure to keep the right order of outputs.

This problem is about experiment design. In particular the task is to design an input signal to be used in an identification experiment to give an estimate of an unknown plant $G(e^{j\omega})$. The following information about $G(e^{j\omega})$ is provided:

- A magnitude plot of $G_{\text{approx}}(e^{j\omega})$, which is similar to $G(e^{j\omega})$. More precisely, the vector `p2_G_mag` contains the magnitudes of $G_{\text{approx}}(e^{j\omega})$ corresponding to the frequencies `p2_omega` in [rad/sec].
- A step response plot of $G_{\text{approx}}(e^{j\omega})$, which is similar to the step response of $G(e^{j\omega})$. More precisely, the vector `p2_y_step` contains the step response values corresponding to the times in `p2_t_step` (times are in [sec]).

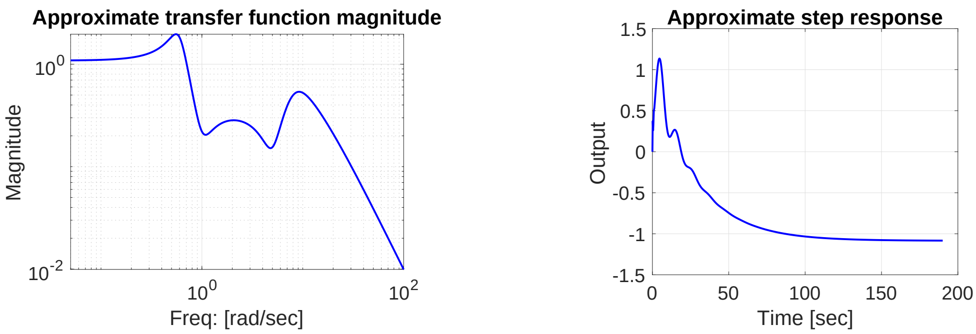

Figure 2: Provided data about G_{approx}, which is similar to the plant G.

Moreover, we know that our experiment will be corrupted by additive white noise v. That is, the measured outputs y satisfy

$$y = Gu + v,$$

with $v \sim \mathcal{N}(0, 0.5)$, where $\mathcal{N}(0, 0.5)$ denotes the normal distribution with mean 0 and variance 0.5. Finally, $G(e^{j\omega})$ has saturation input limits of ± 2.5. Therefore only inputs u with absolute value less than or equal to 2.5 can be applied.

You are asked to choose a sampling time `p2_T` (in [sec]) and an input signal `p2_u` so that an estimate $\hat{G}(e^{j\omega})$ of $G(e^{j\omega})$ can be obtained from input-output data. You will also have to summarize the identification steps you would follow to obtain $\hat{G}(e^{j\omega})$ from these data. Your choice is subject to the following restrictions.

The primary goal is to have a *minimum number of data-points*, i.e. the length of `p2_u` should be as small as possible. However, the following requirements also have to be satisfied.

The sampling time `p2_T` and the input signal `p2_u` have to be chosen such that:

1) The Nyquist frequency, and the highest frequency in your estimate, is greater than the frequency at which the plant has rolled off to less than 5% of its DC gain. (you can use the provided data `p2_G_magn`, `p2_omega` for $G_{\text{approx}}(e^{j\omega})$ to determine this frequency).
2) `p2_T` is an integer number of milliseconds.
3) $\hat{G}(e^{j\omega_n})$ is an unbiased estimate of $G(e^{j\omega_n})$.
4) The frequency resolution of $\hat{G}(e^{j\omega_n})$ is such that an estimate $\hat{G}(e^{j\omega_n})$ exists for at least 4 non-zero frequencies below the low-frequency peak of $G_{\text{approx}}(e^{j\omega})$. (you can use the provided data `p2_G_magn`, `p2_omega` for $G_{\text{approx}}(e^{j\omega})$ to determine the frequency of this peak).
5) The variance of the error $E(e^{j\omega_n}) = \hat{G}(e^{j\omega_n}) - G(e^{j\omega_n})$, at non-zero frequencies, is less than 0.025.

Your function should return the following variables in the order listed:

- `p2_T`: sampling time (in [sec]) chosen for the experiment, as scalar value,
- `p2_u`: chosen input signal as a $N \times 1$ vector.

Moreover, your function should print in the command window a *concise* answer to the following three questions. You should answer to each question in a separate paragraph.

(a) Which type of input signal did you choose? Name the characteristics of `p2_u` (maximum one sentence).

(b) How would you estimate $G(e^{j\omega})$ from the experimental data? Briefly describe the steps you would follow to obtain $\hat{G}(e^{j\omega_n})$ from the input-output data (no equations).

(c) How do your choices of `p2_T`, `p2_u` and the identification method you chose in question (b) satisfy the requirements 1)-5)? Treat each requirement separately, using a list of numbered points (max. 2 sentences each).

Your grade for this problem will depend on:

- The correctness of the sampling time `p2_T` and the input signal `p2_u` having minimum length while satisfying the specifications above.

- The correctness of the method you chose to estimate $G(e^{j\omega})$ so that conditions 1)-5) are satisfied.
- The quality, clarity and accuracy of your explanations.