System Identification

227-0689-00L

Final examination: instructions

Due: 12:00 (noon) on Friday, December 21$^{\text{st}}$, 2018

Overview

The final exam is composed of three problems. To answer these you will write three separate Matlab functions: one for Problem 1, one for Problem 2, and one for Problem 3. 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 final exam` and the **scanned** document should be named as `HS2018_SysID_final_DO_LegiNumber.pdf`, where “LegiNumber” must be your legi-number, without any dash, slash, backslash, etc., e.g.,

`HS2018_SysID_final_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 given above. The subject of the email must be `SystemID HS2018 Final Exam`. The email should contain exactly 4 separate attachments (3 Matlab m-files and a PDF).

Your grade for the final will be evaluated based on your performance in the three problems weighted in the following way: 30% for Problem 1, 30% for Problem 2, and 40% for Problem 3.

Downloadable data

On the Piazza course website

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

there are files that you must download for the individual parts. The files contain the followings

File	Problem	Variables
`HS2018_SysID_final_p1_data.mat`	Problem 1:	`p1_u, p1_y`
`HS2018_SysID_final_p2_data.mat`	Problem 2:	`p2_u, p2_y`
`HS2018_SysID_final_p3_system_sim.p`	Problem 3	

MATLAB function format

Write MATLAB code as functions and call them `HS2018_SysID_final_p?_LegiNumber.m`, where the question mark corresponds to the problem number.

The description of the individual problems (given below) 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 the order given in the table above.

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 the 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 30%)**

Write a MATLAB function of the following form:

```
[p1_pe, p1_a_est, p1_b_est, p1_b_var] =
            HS2018_SysID_final_p1_LegiNumber(p1_u, p1_y)
```

Consider an open-loop identification problem with the following configuration:

$$y(k) = \frac{B(z,\theta)}{A(z,\theta)} u(k) + \frac{1 - 2.25z^{-1} + 1.5625z^{-2}}{A(z,\theta)} e(k),$$

where $\{e(k)\}$ are i.i.d. random variables drawn from $\mathcal{N}(0, 0.03)$.
The polynomials are of the form:

$$A(z,\theta) = 1 + a_1 z^{-1} + a_2 z^{-2}\,,$$
$$B(z,\theta) = b_1 z^{-1}\,,$$

with $\theta = [a_1, a_2, b_1]^T$ being the unknown parameter vector.

For the questions below, you are given the following experimental data:

- `p1_u`: the input signal, $u(k)$, for $k = 0, 1, \dots, N-1$,
- `p1_y`: the corresponding output signal, $y(k)$, for $k = 0, 1, \dots, N-1$.

Answer the following questions:

a) Determine the order of the persistency of excitation (denoted by `p1_pe`) of the input signal, `p1_u`, and briefly explain how you derived it.

b) Compute an *unbiased* and *minimum variance* estimate $\hat{\theta} = [\hat{a}_1, \hat{a}_2, \hat{b}_1]^T$ of the parameters $\theta = [a_1, a_2, b_1]^T$. Explain the method that you used and the reasons for choosing it.

   For the identification of θ, start your regressor at $k = 2$ and print the first 5 rows of the regressor in the command window.

Let $\hat{b}_1$ be the estimate of b_1 according to your method in part b). Moreover, let $var_{\hat{b}_1}(K)$ be the variance of $\hat{b}_1$ where only the first K data $\{(u(k), y(k)), k = 0, 1, \ldots, K-1\}$ are used in the calculation of your estimate.

c) Compute $var_{\hat{b}_1}(K)$ for $K = 7, 8, \ldots, N$. Return those variances in a vector `p1_b_var` and plot them as a function of K in a separate figure. Moreover, print the formula you used to compute the variances in the command window.

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

- `p1_pe` (scalar number): the order of the persistency of excitation of the signal `p1_u`,
- `p1_a_est` (vector of dimension 2): the estimated coefficients $[\hat{a}_1, \hat{a}_2]^T$ of the polynomial $A(z, \theta)$,
- `p1_b_est` (scalar number): the estimated coefficient $\hat{b}_1$ of the polynomial $B(z, \theta)$,
- `p1_b_var` (vector of dimension 72): the variances of $\hat{b}_1$, for data lengths $K = 7, 8, \ldots, N$.

Your grade for this problem will depend on:

- the correctness of your methods both in explanation and in implementation,
- the accuracy of your parameter estimates,
- the quality of your explanations and figures.

Problem 2 (Weight 30%)

Consider an identification experiment conducted on a linear time-invariant, discrete-time system with input u, output y, and measurement noise v. The system dynamics are described by the following equation,

$$y(k) \; = \; u(k) \; + \; \theta u(k-1) \; + \; v(k), \quad k = 0, 1, \ldots, K-1,$$

where K is the data length, $v(k) = c_1\, e(k) + c_2\, e(k-1)$ with $c_1 = 0.043$ and $c_2 = -0.05$ and $e(k)$ is independent and identically normally distributed with $\mathcal{N}(0, 1)$. Assume that the system is initially at rest—that is, for all $k < 0$, $u(k) = y(k) = v(k) = 0$.

From our a prior knowledge about the model, the parameter θ is modeled as a random variable generated by the distribution $\mathcal{N}(-0.5, 0.5)$. Our goal in this problem is to experimentally estimate the transfer function from u to y by estimating the unknown parameter θ.

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

```
[p2_theta_ML,p2_theta_MAP] = HS2018_SysID_final_p2_LegiNumber(p2_u, p2_y)
```

You are given the following experiment data of length K:

- `p2_u`: the input signal $u(k)$.
- `p2_y`: the system output $y(k)$ in response to `p2_u` for one realization of the noise $v(k)$.

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

- `p2_theta_ML`: the Maximum Likelihood estimate, $\hat{\theta}_{\mathrm{ML}}$, for the given data set.
- `p2_theta_MAP`: the Maximum A Posterior estimate, $\hat{\theta}_{\mathrm{MAP}}$, for the given data set.

In addition you should answer the following questions.

a) Using the Maximum Likelihood method and the input and output data of length K, find an estimate $\hat{\theta}_{\mathrm{ML}}$. Display the Likelihood function and equation for computing $\hat{\theta}_{\mathrm{ML}}$ in the command window. Explain the derivation of your estimate the command window of MATLAB.

b) Calculate the Maximum A Posteriori estimate, $\hat{\theta}_{\mathrm{MAP}}$. Plot the corresponding density function and the equation for computing $\hat{\theta}_{\mathrm{MAP}}$ in the command window.

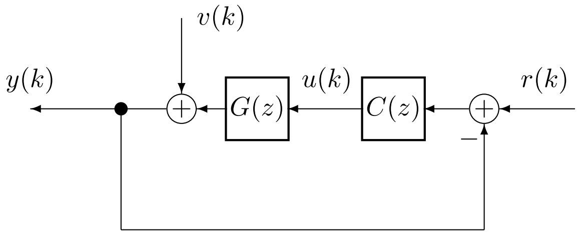
   Calculate and comment on the difference between the $\hat{\theta}_{\mathrm{ML}}$ and $\hat{\theta}_{\mathrm{MAP}}$ estimates.

Your grade for this problem will depend on:

- The correctness of the displayed likelihood and density functions and the equations for computing $\hat{\theta}_{\text{ML}}$ and $\hat{\theta}_{\text{MAP}}$.
- The accuracy of your estimates $\hat{\theta}_{\text{ML}}$ and $\hat{\theta}_{\text{MAP}}$.
- The quality and clarity of your explanations, including any figures you may choose to show.

Problem 3 (Weight 40%)

Consider a plant, $G(z)$, operating in closed-loop with a controller, $C(z)$, as illustrated in the following diagram.



Your task is to design a series of experiments to identify the plant $G(z)$. We have supplied the following function that simulates the above system.

```
p3_y = HS2018_SysID_final_p3_system_sim(legi_num,p3_r).
```

This function takes as input:

- `legi_num`: Your seven- or eight-digit legi number, entered as an integer, without dashes. For example, 12-345-678 should be entered as 12345678. This is used to create a unique plant for you to estimate.
- `p3_r`: Your desired experimental reference inputs. In the solution script you submit, you are allowed to run five experiments, each of maximum length 1024. Thus, this input is a matrix of maximum size 1024-by-5. Each column is the reference signal, $r(k)$, for one experiment. The values of the signal $r(k)$ must be in the range [-1 1].

The function returns `p3_y`, which is the output $y(k)$ of the closed-loop system, in response to the reference inputs that you supplied, one column per experiment.

The system controller $C(z)$ and discrete-time plant $G(z)$ run at a sample rate of $T_s = 0.01s$, which is thus the sample rate for `p3_r` and `p3_y` as well. Assume that the frequency of the fastest pole of $G(z)$ is less than 50π rad s^{-1}.

The noise signal, $v(k)$, is normally distributed $\sim \mathcal{N}(0, \sigma^2)$.

You can assume that prior to each experiment (i.e. each column of your `p3_r` input), the system is at rest:

$$y(k) = u(k) = r(k) = v(k) = 0 \text{ for all } k < 0\,.$$

You may only call `HS2018_SysID_final_p3_system_sim` once in the solution you submit (running up to five experiments of length ≤ 1024). However, you may call it as many times as you like when developing your solution.

Write a function of the following form:

```
[p3_cl_impulse_resp, p3_g_impulse_resp, p3_theta_iv, p3_r] =
                HS2018_SysID_final_p3_LegiNumber()
```

Note that in the following you can run five experiments in total, and all experiments must be run at the same time via a single call to the function `HS2018_SysID_final_p3_system_sim`. Data from any experiment can be used to answer all of the questions below.

(a) Design and conduct experiments to estimate the impulse response of the closed-loop transfer function from r to y. Return the impulse response over the first 2 seconds as `p3_cl_impulse_resp`.

Explain the inputs you chose and the calculations you did to arrive at the estimated impulse response.

(b) Now assume that the discrete-time controller $C(z)$ is given as

$$C(z) = \frac{10z - 9.01}{z - 0.9802}.$$

(i) Using indirect closed-loop identification methods and the estimate of the closed-loop system found in part (a), estimate the impulse response of $G(z)$. Return the impulse response over the first 2 seconds as `p3_g_impulse_resp`.

(ii) Explain whether the solution returned is biased or unbiased. Give your reasons.

(c) Use an instrumental variable (IV) method to estimate $\hat{\theta}_{\text{IV}} = [\hat{h}_1, \dots,\ \hat{h}_{200}]$, the first 200 samples (2 seconds) of the impulse response of $G(z)$. Assume that the impulse response of $G(z)$ is identically zero after the first two seconds.

(i) Describe your choice of instrumental variables, $\zeta(k)$, in the command window.

(ii) Estimate $\hat{\theta}_{\text{IV}}$ using your chosen $\zeta(k)$, and return this as `p3_theta_iv`. Explain in the command window how the estimate was determined.

Hint: Several suggestions include:

- Choose your instrumental variables from r at particular timesteps.
- Choose your regressors from u at particular timesteps. Note that u can be computed from the experimental y and r data, as well as the given $C(z)$ and the closed-loop equations.

Your function should return the following variables:

- `p3_cl_impulse_resp` (vector of length 200): the estimated impulse response of the closed-loop system, from r to y,
- `p3_g_impulse_resp` (vector of length 200): the estimated impulse response of the plant $G(z)$, using an indirect closed-loop identification method,
- `p3_theta_iv` (vector of length 200): the instrumental variable estimate of the first 200 samples of the impulse response of $G(z)$,
- `p3_r` (array of maximum dimension 1024-by-5): the experimental inputs fed into the system to produce the requested estimates.

Your grade for this problem will depend on:

- the quality of your explanations including any figures you choose to show,
- the experimental design, including the inputs used for identification,
- the correctness of the closed-loop estimation methods used to identify the impulse responses of the closed-loop system and $G(z)$,
- the choice of your instruments and correctness of the instrumental variables method,
- the accuracy of your estimates of the requested impulse responses over the first two seconds.

A final remark: your method will be tested on a similar system to the one you were given (roughly the same system timescales, noise levels, and general behavior). Note therefore that your code should try to be flexible and not anticipate exactly the same output of the system simulation script.