System Identification

227-0689-00L

Final examination: instructions

Due: 12:00 (noon) on Friday, December 20$^{\text{th}}$, 2019

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 `HS2019_SysID_final_DO_LegiNumber.pdf`, where "LegiNumber" must be your legi-number, without any dash, slash, backslash, etc., e.g.,

`HS2019_SysID_final_DO_12345678.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 HS2019 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: 35% for Problem 1, 30% for Problem 2, and 35% for Problem 3.

Downloadable data

On the Piazza course website

`https://piazza.com/ethz.ch/fall2019/227068900l/resources`

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

File	Problem	Summary description
`HS2019_SysID_final_p1_GenerateData.p`	Problem 1:	Generates the problem data.
`HS2019_SysID_final_p1_12345678.m`	Problem 1:	Solution template
`HS2019_SysID_final_p2_GenerateData.p`	Problem 2:	Generates the problem data.
`HS2019_SysID_final_p2_12345678.m`	Problem 2:	Solution template
`HS2019_SysID_final_p3_GenerateData.p`	Problem 3:	Generates the problem data.
`HS2019_SysID_final_p3_12345678.m`	Problem 3:	Solution template
`HS2019_SysID_final_check.p`	all problems	Code check function

Matlab function format

Each problem has a solution template (named `HS2019_SysID_final_p?_12345678.m`) which can be downloaded from Piazza. Replace the string `12345678` in the solution template with your Legi number (using only the digits) and use it as the starting point for your solution. You will also have to make the same change on the first line of the solution function. It is important that you do not change the name or the order of the output variables on the first line.

You will see that the solution template will call a function called

`HS2019_SysID_final_p?_GenerateData.p`

to generate the data for your analyses. There is a separate data generation function for each question and you will need to download all three from Piazza.

We have provided a code checking function, `HS2019_SysID_final_check.p`, which can be used to check that your function meets the minimal requirements for submission. Running this function, with your Legi number as the input argument, will perform this check on every function matching the solution naming format in your current directory. Do not submit any code that has not been successfully checked by this function.

The description of the individual problems (given on the following pages) will specify the variables that must be returned by your functions. You will also be asked to answer certain questions and explain your results. These explanations must be given in the command window of Matlab.

If you write any custom functions include them in the solution 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.

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

- You may only use functions from MATLAB and the Control Systems Toolbox. Functions from other toolboxes are not permitted. This is checked by the check function `HS2019_SysID_final_check.p`.
- Your function must perform the calculations requested on the data generated by the data generation function. If the solution of the problem requires that you make a choice of parameter then this may be hard-coded into your function. You should explain in the command window why and how you made any such choice. Examples of this include: sampling time, γ values for windowing, model order for SVD truncation, etc..
- 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 a single plot, except for Bode plots which can use the subplot function to display gain and phase separately.
- 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.
- Do not use the commands `pause`, `clear`, `clc`, `clf`, or `close` within your solution function.
- If you skip one part of a problem, please keep the dummy variable as in the solution template.

Before submitting your file, restart MATLAB and test it by running it in a folder/directory with no other files. Also test it by running the `HS2019_SysID_final_check.p` function in a directory containing only the solution file(s).

Problem 1 (Weight 35%)

To solve this problem, you should write a function of the following form

```
[p1_theta_est1, p1_Phi, p1_theta_est2, p1_y_pred] =
        HS2019_SysID_final_p1_LegiNumber()
```

using the provided solution template `HS2019_SysID_final_p1_12345678.m`. As visible from the solution template, the function `HS2019_SysID_final_p1_GenerateData()` is called and variables named `p1_u, p1_y, p1_theta_hat, p1_u_past, p1_y_past, p1_pred_err` are saved in your workspace. Note that this requires the file `HS2019_SysID_final_p1_GenerateData.p` to be in your working directory. These variables must be used to solve the problem tasks explained below. Make sure that your function returns the output variables `p1_theta_est1, p1_Phi, p1_theta_est2, p1_y_pred` in the right order.

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

$$y(k) = \frac{B(z)}{A(z)}u(k) + \frac{1 + c_1 z^{-1} + c_2 z^{-2}}{A(z)}e(k)\,, \tag{1}$$

where $\{e(k)\}$ are zero-mean random variables such that for all k and τ,

$$\mathbb{E}[e(k)e(\tau)] = \begin{cases} 0.9 & \text{if } \tau = k \\ 0.2 & \text{if } 0 < |\tau - k| \leq 3 \\ 0 & \text{otherwise} \end{cases}.$$

The polynomials are of the form

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}\,,$$
$$B(z) = b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}$$

where $a_1, a_2, a_3, b_1, b_2, b_3$ are unknown parameters to be identified. Let $\theta = [a_1, a_2, a_3, b_1, b_2, b_3]$.

You are given the following experimental data from the configuration above:

- `p1_u`: the input signal $u(k)$, for $k = 1, 2, \ldots, K$
- `p1_y`: the corresponding output signal $y(k)$, for $k = 1, 2, \ldots, K$.

You can assume that for $k \leq 0$ the system is at rest, i.e., $u(k) = y(k) = e(k) = 0$ for all $k \leq 0$.

1. Consider the case where the true parameters c_1 and c_2 are unknown, and obtain a linear estimate $\hat{\theta}_{\text{est1}}$ of the parameter vector θ assuming $c_1 = c_2 = 0$. Display the equation for the computation of $\hat{\theta}_{\text{est1}}$ in the command window. Summarize and motivate the statistical properties of the estimator $\hat{\theta}_{\text{est1}}$.

2. Let now $c_1 = 1, c_2 = \frac{\pi}{4}$, and use this knowledge to improve the quality of your estimate. Explain in the command window how the new estimate $\hat{\theta}_{\text{est2}}$ was determined and display the equations used for the computation. How have the statistical properties of this estimator changed compared to those of $\hat{\theta}_{\text{est1}}$?

To solve the last task, consider a new ARMAX model with the same structure of (1), and assume an estimate $\hat{\theta}$ of the system's parameters θ has been obtained and given to you in the 6×1 vector `p1_theta_hat`. As for c_1 and c_2, the same values used in task 2 should be considered. Let $\hat{y}(k|k-1, \hat{\theta})$ be the one-step ahead predictor of the signal $y(k)$ according to the estimate $\hat{\theta}$.

3. Consider conducting an experiment where at every timestep k you have access to: past applied inputs, corresponding output measurements, and the past prediction errors (i.e., the difference between measured and predicted outputs), and your goal is to predict the output of the system at the current timestep. Hence, at timestep k=40 you are given:
   - `p1_u_past`: 1×5 vector of inputs $u(39), \ldots, u(35)$.
   - `p1_y_past`: 1×5 vector of corresponding measured outputs $y(39), \ldots, y(35)$.
   - `p1_pred_err`: 1×5 vector of corresponding prediction errors.

   What is the one-step ahead prediction according to your model? That is, compute $\hat{y}(40|39, \hat{\theta})$. Display and motivate in the command window the equations used.

Your function should return the following variables:

- `p1_theta_est1`: the 6×1 vector $\hat{\theta}_{\text{est1}}$,
- `p1_Phi`: the regressor Φ used to obtain the estimate $\hat{\theta}_{\text{est1}}$. Note that `p1_Phi` must be a matrix of 6 columns.
- `p1_theta_est2`: the 6×1 vector $\hat{\theta}_{\text{est2}}$,
- `p1_y_pred`: the (scalar) one-step ahead prediction $\hat{y}(40|39, \hat{\theta})$.

Your grade for this problem will depend on:

- the quality of your explanations,
- the correctness of your methods,
- the accuracy of your parameter estimates.

Problem 2 **(Weight 30%)**

Write a MATLAB function of the following form:

```
[p2_a_ex1, p2_b_ex1, p2_a_ex2, p2_b_ex2, p2_mse_ex1, p2_mse_ex2] =
                HS2019_SysID_final_p2_LegiNumber()
```

Use the template file `HS2019_SysID_final_p2_12345678.m` for your solution. Do not change the order or the names of the output variables as written in the template.

As visible from the template, the function `HS2019_SysID_final_p2_GenerateData()` is called and the following data sets are saved to your workspace:

- `p2_u1`: the input signal of experiment 1, $u_1(k)$, for $k = 0, 1, \dots, N-1$,
- `p2_y1`: the corresponding output signal of experiment 1, $y_1(k)$, for $k = 0, 1, \dots, N-1$,
- `p2_u2`: the input signal of experiment 2, $u_2(k)$, for $k = 0, 1, \dots, N-1$,
- `p2_y2`: the corresponding output signal of experiment 2, $y_2(k)$, for $k = 0, 1, \dots, N-1$,
- `p2_u_cv`: the input signal of the cross-validation experiment, $u_{\text{cv}}(k)$, for $k = 0, 1, \dots, N-1$,
- `p2_y_cv`: the corresponding output signal of the cross-validation experiment, $y_{\text{cv}}(k)$, for $k = 0, 1, \dots, N-1$.

Note that this requires the file `HS2019_SysID_final_p2_GenerateData.p` to be in your working directory. Use the provided data sets to solve the following problem.

Let $\theta = [a_1, a_2, a_3, b_1, b_2]^T$ be the parameter vector. Consider an open-loop identification problem with the following configuration:

$$y(k) = \frac{B(z,\theta)}{A(z,\theta)} u(k) + e(k),$$

where $\{e(k)\}$ are i.i.d. random variables drawn from $\mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma^2 = 0.25$. All signals are zero for negative time, i.e. $y(k) = u(k) = e(k) = 0$ for $k < 0$.
The polynomials are of the form:

$$A(z,\theta) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}\,,$$
$$B(z,\theta) = b_1 z^{-1} + b_2 z^{-2}\,.$$

Solve the following tasks:

1. Perform an instrumental variables estimation of the parameter vector $\hat{\theta}^{\text{IV1}} = [\hat{a}_1^{\text{IV1}}, \hat{a}_2^{\text{IV1}}, \hat{a}_3^{\text{IV1}}, \hat{b}_1^{\text{IV1}}, \hat{b}_2^{\text{IV1}}]$. In order to do this, use the input and output

data of experiment 1, $\{u_1(k), y_1(k)\}$, to compute a least squares estimate of the parameters. Then, use these estimates to obtain instruments with which you should calculate an estimate of the parameter vector $\hat{\theta}^{\mathrm{IV1}}$ using the data from experiment 2, $\{u_2(k), y_2(k)\}$.

Display in the Matlab command window your brief and concise answer to the following questions: Are the least squares estimates obtained here unbiased? Are the instrumental variables estimates obtained here unbiased? Explain your answer.

2. Perform again an instrumental variables estimation of the parameter vector $\hat{\theta}^{\mathrm{IV2}} = [\hat{a}_1^{\mathrm{IV2}}, \hat{a}_2^{\mathrm{IV2}}, \hat{a}_3^{\mathrm{IV2}}, \hat{b}_1^{\mathrm{IV2}}, \hat{b}_2^{\mathrm{IV2}}]$. This time, however, use delayed inputs $u(k-i)$, $i = 1, .., d$ to construct the instruments, by choosing the right number d of delayed inputs in the instruments. Use both the data $\{u_1(k), y_1(k)\}$ from experiment 1 and $\{u_2(k), y_2(k)\}$ from experiment 2 for this estimation.

   Display in the Matlab command window your brief and concise answer to the following question: Are the instrumental variables estimate obtained here unbiased? Explain.

3. Compare the estimates $\hat{\theta}^{\mathrm{IV1}}$ and $\hat{\theta}^{\mathrm{IV2}}$. Use the cross-validation experiment data $\{u_{\mathrm{cv}}(k), y_{\mathrm{cv}}(k)\}$ to compute the mean square error (MSE) for outputs obtained for each of the parameter estimates. Return as output variables `p2_mse_ex1` corresponding to the MSE obtained for output for $\hat{\theta}^{\mathrm{IV1}}$, and `p2_mse_ex2`, the MSE obtained for output from using $\hat{\theta}^{\mathrm{IV2}}$.

   Display in the Matlab command window your brief and concise answer to the following questions: How do you explain the difference between the parameter values estimated for $\hat{\theta}^{\mathrm{IV1}}$ and $\hat{\theta}^{\mathrm{IV2}}$? What are the requirements for the parameter estimates generated from least squares based instruments to be equal to that generated from delayed inputs?

Your function should return the following variables:

- `p2_a_ex1` (vector of dimension 3): the coefficients of the polynomial $A(z, \hat{\theta}^{\mathrm{IV1}})$, i.e., the vector $[\hat{a}_1^{\mathrm{IV1}}, \hat{a}_2^{\mathrm{IV1}}, \hat{a}_3^{\mathrm{IV1}}]^T$,
- `p2_b_ex1` (vector of dimension 2): the coefficients of the polynomial $B(z, \hat{\theta}^{\mathrm{IV1}})$, i.e., the vector $[\hat{b}_1^{\mathrm{IV1}}, \hat{b}_2^{\mathrm{IV1}}]^T$,
- `p2_a_ex2` (vector of dimension 3): the coefficients of the polynomial $A(z, \hat{\theta}^{\mathrm{IV2}})$, i.e., the vector $[\hat{a}_1^{\mathrm{IV2}}, \hat{a}_2^{\mathrm{IV2}}, \hat{a}_3^{\mathrm{IV2}}]^T$,
- `p2_b_ex2` (vector of dimension 2): the coefficients of the polynomial $B(z, \hat{\theta}^{\mathrm{IV2}})$, i.e., the vector $[\hat{b}_1^{\mathrm{IV2}}, \hat{b}_2^{\mathrm{IV2}}]^T$,
- `p2_mse_ex1` (scalar): the MSE result for your $\hat{\theta}^{\mathrm{IV1}}$ estimate using the cross-validation data set,
- `p2_mse_ex2` (scalar): the MSE result for your $\hat{\theta}^{\mathrm{IV2}}$ estimate using the cross-validation data set.

Your grade for this problem will depend on:

- the quality, conciseness and correctness of your answers to the questions,
- the accuracy of your parameter estimates and MSE values.

Problem 3 (Weight 35%)

Write a MATLAB function of the following form:

```
[p3_b_ML,p3_b_MAP,p3_cv_error,p3_prior_best] = HS2019_SysID_final_p3_LegiNumber()
```

Use the template file, `HS2019_SysID_final_p3_12345678.m` for your solution. Do not change the order or names of the output variables as written in the template.

Consider the problem of identifying a linear time-invariant, discrete-time system with input u, output y, and measurement noise e. The system dynamics are described by the following equation,

$$y(k) = \sum_{i=1}^{n} b_i u(k-i) + e(k)$$

where $n = 8$ and $e(k)$ is independent and identically normally distributed with $\mathcal{N}(\mu, \sigma^2)$, $\mu = 0$ and $\sigma^2 = 0.5^2$. Assume that the system is initially at rest, i.e., for all $k < 0$, $u(k) = y(k) = e(k) = 0$.

Your task is to obtain maximum likelihood and maximum a posteriori estimations of $\mathbf{b} = [b_1\ b_2\ \cdots b_8]^\mathsf{T}$ from given input-output data. We have supplied the following function that generates the input-output data:

```
[p3_u,p3_y,p3_u_cv,p3_y_cv] = HS2019_SysID_final_p3_GenerateData
```

The input of the function is

- `LegiNumber`: Your leginumber, entered as shown in the solution template file. The leginumber is used to generate unique experiment data for you to estimate.

The outputs of the function are

- `p3_u`: The input data, $u(k)$, for identification.
- `p3_y`: The output data, $y(k)$, for identification.
- `p3_u_cv`: The cross-validation input data, $u_{\text{cv}}(k)$.
- `p3_y_cv`: The cross-validation output data, $y_{\text{cv}}(k)$.

Both the identification data and the cross validation data consist of 100 data points, which corresponds to the measurements at $k = 0, 1, \cdots, 99$.

Using the provided experiment data, you should do the following:

a) Using the maximum likelihood method and the identification data $u(k)$ and $y(k)$, find the maximum likelihood estimate $\hat{\mathbf{b}}_{\mathrm{ML}}$.

Please explain the following in the command window:

- The likelihood function and the equation for computing $\hat{\mathbf{b}}_{\mathrm{ML}}$.
- The derivation of your estimate.

b) In order to calculate maximum a posteriori estimates of $\mathbf{b}$, the parameters $\mathbf{b}$ are assumed to be a normally distributed random vector subject to the distribution $\mathcal{N}(0, S)$, where

$$S = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

is the covariance matrix. The probability density function of the distribution is given by

$$p(\mathbf{b}) = \frac{1}{\sqrt{(2\pi)^n \det(S)}} \mathrm{e}^{-\frac{1}{2}\mathbf{b}^\mathsf{T} S^{-1} \mathbf{b}}.$$

From our prior knowledge about the system, the following five choices of S are proposed:

- $S_1 = I$ is the identity matrix, i.e., $a_{ii} = 1, a_{ij} = 0$ for $i \neq j$;
- S_2 is a diagonal matrix with $a_{ii} = 0.8^i, a_{ij} = 0$ for $i \neq j$;
- S_3 is a diagonal matrix with $a_{ii} = 0.5^i, a_{ij} = 0$ for $i \neq j$;
- S_4 is a full matrix with $a_{ij} = 0.8^{\max(i,j)}$.
- S_5 is a full matrix with $a_{ij} = 0.5^{\max(i,j)}$.

Using the maximum a posteriori method and the identification data $u(k)$ and $y(k)$, find the maximum a posteriori estimate $\hat{\mathbf{b}}_{\mathrm{MAP}}(S_m)$ for $m = 1, 2, 3, 4, 5$.

Hint: Let Q be an n-by-n symmetric matrix with non-negative eigenvalues, c be a real vector of length n and d be a real scalar. If x^* is the solution of the following optimization problem

$$\min_x \; \frac{1}{2} x^\top Q x + c^\top x + d,$$

Then, x^* is a solution of $Qx + c = 0$.

To find the optimal choice of S, consider the following estimation error formula on the cross-validation data $u_{cv}(k)$ and $y_{cv}(k)$:

$$\epsilon(S) = \frac{1}{100} \sum_{k=0}^{99} \Big(y_{cv}(k) - \hat{y}(k \mid \hat{\mathbf{b}}_{\mathrm{MAP}}(S), u_{cv}) \Big)^2$$

Calculate $\epsilon(S_m)$ for $m = 1, 2, 3, 4, 5$ and find out the optimal S_m that gives the smallest cross-validation error.

Please explain the following in the command window:

- The optimization problem and the equation for computing $\hat{\mathbf{b}}_{\text{MAP}}(S)$.
- The derivation of your estimate.
- The difference between the $\hat{\mathbf{b}}_{\text{ML}}$ and $\hat{\mathbf{b}}_{\text{MAP}}(S)$.
- The effect of different choices of S.

You may use up to TWO figures to support your argument.

Important notes:

- *Please only use the identification data $u(k)$ and $y(k)$ for obtaining the estimates.*
- *We expect the length of a concise explanation to be around 2-4 sentences. We may deduct points for explanations that are too long at our discretion.*
- *Make sure to have exactly 4 outputs, and to keep the right order as displayed above and written in the template. If you don't have results for one or more of the outputs, keep the null variables as in the template.*

Your MATLAB function should return the following outputs of correct dimensions and sequences in the order listed:

- `p3_b_ML` (vector of dimension 8×1): Maximum likelihood estimate of $\mathbf{b}$, i.e., $\hat{\mathbf{b}}_{\text{ML}}$.
- `p3_b_MAP` (matrix of dimension 8×5): Maximum a posteriori estimate of $\mathbf{b}$. The m-th column of the matrix should be the estimate $\hat{\mathbf{b}}_{\text{MAP}}(S_m)$.
- `p3_cv_error` (vector of dimension 5×1): The cross-validation errors of maximum a posteriori estimates. The m-th element of the vector should be the cross-validation error $\epsilon(S_m)$ of $\hat{\mathbf{b}}_{\text{MAP}}(S_m)$.
- `p3_prior_best` (scalar integer in the set $\{1, 2, 3, 4, 5\}$): The index m of the optimal prior covariance matrix S_m.

Your grade for this problem will depend on:

- The accuracy of the outputs `p3_b_ML,p3_b_MAP,p3_cv_error`. We will compare your outputs with the correct outputs obtained for your individual experiment data using the methods specified in the task description.
- The correctness of the output `p3_prior_best`.
- The correctness of the derivation of your maximum likelihood and maximum a posteriori estimators.
- The quality, clarity and accuracy of your explanations, answers, and/or figures in all parts.