System Identification
227-0689-00L
Final Examination

Due: 19:00 on Friday, December 22^{nd}, 2023

Overview

The final exam is composed of THREE problems. To solve these problems you need to 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.

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

Your functions will be submitted through Moodle. In this submission 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 `HS2023_SysID_final_DO_LegiNumber.pdf`, where “LegiNumber” must be your legi-number, without any dash, slash, backslash, etc., e.g.,

`HS2023_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.

The Moodle forums are closed during the final examination. If you encounter any problems with the examination, please send an email to `sysid@ee.ethz.ch` or Prof. Roy Smith.

Your submission must be submitted to Moodle through the link

`https://moodle-app2.let.ethz.ch/mod/assign/view.php?id=943808`

by the due date and time given above. The submission should contain exactly 4 separate files (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, 35% for Problem 2, and 35% for Problem 3.

Downloadable data

The required files for the individual parts will be provided through the Moodle platform under the “final Matlab project” section. Here is the list of uploaded files:

File	Problem	Summary description
`HS2023_SysID_final_p1_GenerateData.p`	Problem 1	Generate the problem data
`HS2023_SysID_final_p1_12345678.m`	Problem 1	Solution template
`HS2023_SysID_final_p1_check.p`	Problem 1	Code compliance check function
`HS2023_SysID_final_p2_GenerateData.p`	Problem 2	Generate the problem data
`HS2023_SysID_final_p2_12345678.m`	Problem 2	Solution template
`HS2023_SysID_final_p2_check.p`	Problem 2	Code compliance check function
`HS2023_SysID_final_p3_GenerateData.p`	Problem 3	Generate the problem data
`HS2023_SysID_final_p3_12345678.m`	Problem 3	Solution template
`HS2023_SysID_final_p3_check.p`	Problem 3	Code compliance check function
`HS2023_SysID_final_DO_12345678.pdf`	-	Declaration of originality

MATLAB function format

Each problem has a solution template (named `HS2023_SysID_final_p?_12345678.m`) which can be downloaded from Moodle. 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 calls a function called

`HS2023_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 from Moodle. When running your solutions, please avoid using ‘Run section’ (or Ctrl + Enter). This is because your legi-number is extracted from the filename but ‘Run section’ doesn’t preserve the filename.

Each problem also has a compliance check function (named `HS2023_SysID_final_p?_check.p`) which can be used to check if your functions meet the minimal requirements for submission. Running this function, with your legi-number as the input argument, will perform this check on the script 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 and the choices you are making. These explanations must be given in the command window of MATLAB, using the `disp` function. Explanations written in the comment of your functions will NOT be graded. Please keep your explanation concise (around 2-4 sentences). You will be graded on the quality of your explanation.

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 for Problems 1 and 2. In addition to this, you may also use the System Identification Toolbox for Problem 3. Functions from other toolboxes are not permitted. This is checked by the compliance check function.
- 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.
- The script should run unattended. Do not use the commands `pause`, `clear`, `clc`, `clf`, or `close` within your solution function. Do not define any global variables in your functions or subfunctions. If you skip one part of a problem, please keep the dummy variable as in the solution template.
- For any figures requested in the problem, use the command `figure(xxx)` (where `xxx` is specified in the problem) 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 the magnitude and the 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 are not covered.

Before submitting your functions, restart MATLAB and test them by running the compliance check function for each problem in a folder/directory with no other files.

Problem 1 **(Weight: 30%)**

In this problem, you are asked to identify two discrete-time systems $G_1(z)$, and $G_2(z)$.

1) In this part, we consider the system $G_1(z)$. System outputs are measured according to

$$y_1(k) = G_1(z)u_1(k) + H_1(z)e(k), \tag{1}$$

where e is a zero-mean i.i.d. noise with a variance of 0.1 and $H_1(z)$ is a discrete-time transfer function. You are given experimental data in vectors `p1_u1`, `p1_y1`, containing the inputs u_1 and corresponding outputs y_1, respectively. Using this, and assuming the following parametrization for the transfer functions

$$G_1(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \qquad H_1(z) = \frac{1 + h_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \tag{2}$$

the goal is to estimate the parameter vector $\theta_1 = [a_1, a_2, b_1, b_2]^\top$. Complete the following tasks by assuming all the signals have a value of 0 for $k<0$.

a) Taking $h_1 = 0$, provide an *asymptotically unbiased* and a *minimum variance* estimate for θ_1 and return it in `p1_theta1_a`.

b) Repeat part a) by now taking $h_1 = 1$. Return the new estimate for θ_1 in `p1_theta1_b`.

c) Explain your implemented methods for parts a) and b), justifying with clearly defined symbols the properties of the estimates.

d) Assume now that

$$H_1(z) = 1 + h_1 z^{-1},$$

with $h_1 = 1$. Can you provide an *asymptotically unbiased* and a *minimum variance* estimate in this case? Explain why or why not.

[**Weight: 10%**]

2) In this part, we consider the system $G_2(z)$ and its output measurements given by

$$y_2(k) = G_2(z)u_2(k) + \frac{H_1(z)}{D(z)}e(k) + \frac{H_2(z)}{D(z)}w(k), \tag{3}$$

where e is zero-mean i.i.d. noise with variance 0.1 as in part 1), and w and u_2 are deterministic inputs provided in variables `p1_w` and `p1_u2`, respectively. The output measurements, y_2 are provided in the variable `p1_y2`. Furthermore, $H_1(z)$ is the same as in part 1), equation (2) with $h_1 = 1$, and the other transfer functions are parameterized as

$$G_2(z) = \frac{f_1 z^{-1} + f_2 z^{-2}}{D(z)}, \qquad H_2(z) = 1 + h_2 z^{-1}, \qquad D(z) = 1 + d_1 z^{-1} + d_2 z^{-2}.$$

The goal is to estimate the parameter vector $\theta_2 = [d_1, d_2, f_1, f_2, h_2]^\top$ using the provided data and the estimate `p1_theta1_b` from part 1). Complete the following tasks by assuming all the signals have a value of 0 for $k<0$.

a) Provide an estimate for θ_2, such that the following prediction error objective is minimised

$$\sum_{k=0}^{N-1} \|\hat{y}_2(k|k-1) - y_2(k)\|_2^2,$$

where $\hat{y}_2$ is the output predicted by your estimate of θ_2. Return this estimate in `p1_theta2`.

b) Return the predicted vector, $[\hat{y}_2(0), \ \dots, \hat{y}_2(N-1)]^\top$ in `p1_y2_hat`.

c) Explain your implemented method for part a), justifying with clearly defined symbols the properties of the estimate.

d) Is your estimate `p1_theta2` asymptotically unbiased? Does it have minimum variance?

e) Generate one MATLAB figure where you plot the measured output `p1_y2` and the output `p1_y2_hat` predicted by your estimates. Provide a legend, a title and label the axes.

[**Weight: 20%**]

Problem 2 (Weight: 35%)

This problem consists of two parts.

1) Imagine a situation where it is required to identify three separate FIR systems of order 9. For this purpose, a unit impulse input ($u(0) = 1$, and $u(t) = 0$ for all $t \neq 0$), was applied to each of the three systems, with zero initial conditions, and the output was measured:

$$y_i(t) = g_i(0)u(t) + g_i(1)u(t-1) + \cdots + g_i(9)u(t-9) + v_i(t), \qquad t = 0, \ldots, 9,$$

where $i = 1, 2,$ and 3 is an index referring to the system. The measurement noise v_i for each estimation experiment has the same statistical properties: i.i.d. Gaussian, with zero-mean, and variance $\sigma^2 = 0.5^2$.

The recorded outputs are given in the three vectors `Y1`, `Y2`, and `Y3`, where

$$\texttt{Y1} = \begin{bmatrix} y_1(0) \\ \vdots \\ y_1(9) \end{bmatrix} \quad \text{and similarly for } \texttt{Y2} \text{ and } \texttt{Y3}.$$

Denote the parameter vector of each system as $\theta_i = \begin{bmatrix} g_i(0) \\ \vdots \\ g_i(9) \end{bmatrix} \in R^{10}, \qquad i = 1, 2, \text{ and} 3.$

Complete the following tasks:

a) Find the least-squares estimate of each system. Return the estimates as three separate vectors `p21_thetaHat1`, `p21_thetaHat2`, and `p21_thetaHat3` (each with dimension 10×1).

b) Compute the bias vector, the covariance matrix, and the MSE of the least-square estimator of the first system `p21_thetaHat1`. Return the values in `p21_bias1` (dimension 10×1), `p21_covMat1` (dimension 10×10), and `p21_MSE1` (dimension 1×1).

c) For each system, construct a regularized estimator that is *guaranteed* to have an MSE smaller than the least-squares estimator. Return the estimates as three separate vectors `p21_thetaHat1_reg`, `p21_thetaHat2_reg`, and `p21_thetaHat3_reg` (each with dimension 10×1).

d) Describe the estimator you used in part c) and justify your choice.

[**Weight: 15%**]

2) In this part, you are asked to identify the pulse-response of an unknown linear dynamical system under the condition of missing data.

The following prior knowledge is available:

- The system is stable, linear and time-invariant,
- The pulse response of the system decays exponentially to a negligible value after 100 samples,
- There is one sample delay at the input.

The experimental configuration is as follows:

- Three independent data sets of input-output pairs are collected using three different input signals and three different sensors
  - `dataSet1` of length 250 samples
  - `dataSet2` of length 250 samples
  - `dataSet3` of length 350 samples

  The first two data sets (`dataSet1` and `dataSet2`) contain missing data values (represented as `NaN`) due to faulty sensors,
- Initial values of inputs and outputs in all data sets are unknown,
- The output is corrupted by additive zero-mean white noise. The noise variances of the sensors used to collect the three data sets (denoted λ_1, λ_2 and λ_3 respectively) are unknown. But it is known that $\lambda_1 = \lambda_3$ and the ratio (λ_2/λ_1) is given in `var_ratio`.

Your task is to estimate the first significant part of the system's pulse response using an FIR model

$$y(t) = \quad g(0)u(t) + \sum_{k=1}^{n} g(k)u(t-k) \quad = \quad \sum_{k=1}^{n} g(k)u(t-k), \quad n \geq 1,$$

where $g(0)$ is known to be zero (due to the one sample delay at the input).

Complete the following tasks and for all plots provide a **legend**, a **title** and **label** the axes.

a) Under the assumption that the true pulse response $g_\circ(k) = 0$ for all $k > 100$, find the best unbiased linear estimate using the above FIR model with $n = 100$, and all available data. Return the estimate as a vector `p22_ghat_ls_100` (dimension 100×1).

In the next two tasks (b) and (c), you will use the cross-validation method where the available data is split into two independent parts: a part for estimation (estimation data), and a part for validation (validation data). This is known as hold-out cross validation. Split the data as follows:

- Combine the first two data sets (`dataSet1` & `dataSet2`) and use their combination as your estimation data.
- Use `dataSet3` as your validation data.

b) Use weighted least-squares and the hold-out cross-validation method (with a data split as indicated above) to find the best FIR model order n that minimizes the mean-square error on the validation data.

   i. plot the validation mean-square error for each $n = 1, \ldots, 100$ in figure `figure(221)`.

   ii. return the best order as `p22_n_cv` (dimension 1×1, positive integer in $\{1, \ldots, 100\}$).

   iii. return the weighted least-squares estimate when $n =$ `p22_n_cv` as a vector `p22_ghat_ls_n_cv` (dimension `p22_n_cv` $\times 1$).

c) Find the best regularized FIR model using a TC-kernel when $n = 100$. Use the hold-out cross-validation method (with a data split as indicated above) to tune the regularization parameter and kernel's hyper-parameters. Return your estimate as a vector `p22_ghat_tc_cv` (dimension 100×1).

[**Weight: 20%**]

Problem 3 (Weight: 35%)

Consider the following configuration where a plant $G(z)$ operates in closed-loop with a controller $C(z)$.

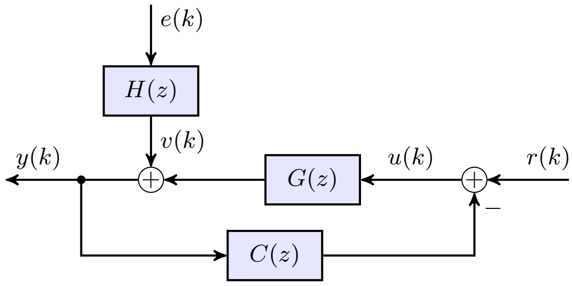

Figure 1: Structure of the control loop

This problem is divided into 2 parts.

1) Choose two reference signals $r(k)$, namely, `p31_r1` and `p31_r2`, and use the provided data generation function to generate `p31_y1`, which is the output $y(k)$ of the closed-loop system resulting from the reference `p31_r1`, and `p31_u2`, which is the input $u(k)$ to the plant G resulting from the reference `p31_r2`.

   - The reference signals $r(k)$ must satisfy $r(k) \in [-1, 1]$, for all k, with a maximum size of 1024×1 each.
   - The system controller $C(z)$ and discrete-time plant $G(z)$ run at a frequency of 255 Hz, thus defining the sample rate for `p31_r1`, `p31_r2` and `p31_y1` as well.
   - $H(z) = 1$, that is, $e(k) = v(k)$ and $e(k) \sim \mathcal{N}(0, \sigma^2)$.
   - Before being excited with the references that you provide, the system was given an unknown non-zero reference $r(k)$.

   You may call the data generation function *only once* in the solution you submit. However, you may call it as many times as you like when developing your solution.

   Assuming an unknown controller $C(z)$, **your task is** to estimate the transfer function T_{ur} from r to u, the transfer function T_{yr} from r to y and the plant transfer function G in the frequency domain using an input-output method. We want to estimate the plant at $f = \{1, \ldots, 127\}$ Hz.

   a) Explain your choice of reference signals.
   b) Explain how you used the generated data to compute the estimates.
   c) Are your estimates of T_{ur}, T_{yr}, and G asymptotically unbiased?
   d) Return the vector of estimates `p31_Tur_hat`, `p31_Tyr_hat`, and `p31_G_hat`, of T_{ur}, T_{yr}, and G, respectively, the vector of frequencies `p31_omega` in rad s^{-1} in increasing order (each vectors of 127×1) as well as the references used `p31_r1`, `p31_r2`.
   e) If you could perform a second experiment, how would you improve these estimates?

[**Weight: 15%**]

2) Another experiment was performed for a *different* system of the same form as in Figure 1, but here the signal $u(k)$ is *not* measured. However, the structure of the system is partially known. In particular, for this part you know that

$$\begin{aligned} G(z) &= \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} = \frac{B(z)}{A(z)}, \\ H(z) &= \frac{1}{A(z)}, \\ C(z) &= \frac{0.5z}{z - 0.9}. \end{aligned}$$

Use the provided data generation function to generate the reference sequence $r(k)$ `p32_r` and output sequence $y(k)$ `p32_y` of this experiment as vectors. You can assume that the system was at rest before the experiment, that is, $y(k) = u(k) = r(k) = 0$ for $k < 0$.

Given `p32_y`, and `p32_r`, **your task is** to estimate the unknown system parameters $\theta = [a_1, a_2, a_3, b_1, b_2]^\top$. Describe how you formulated the least squares estimation problem and provide details about the components that appear in the regression equation. Calculate the best solution to your regression problem using linear least squares. If your solution is nonlinear, calculate a linear least squares approximation.

a) Is it possible to obtain an unbiased estimate of the unknown system parameters θ by formulating a linear regression problem? If not, explain why.

b) Is your regression consistent? Justify your answer.

c) Return the estimates $\hat{\theta} = [\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{b}_1, \hat{b}_2]^\top$ as `p32_theta_hat`.

[**Weight: 20%**]