HOMEWORK EXERCISE III

FILTERING AND IDENTIFICATION (SC42025)

Hand in pictures / scans of your hand-written solutions as a PDF for theoretical exercises. For the MATLAB exercise, please export your live script as a PDF (instructions in template). Then, **merge all files as a single PDF and upload them through Brightspace** on **16-12-2024** before **18:00**. You are allowed and encouraged to discuss the exercises together but you need to hand in individual solutions.

Please highlight your final answer!

Exercise 1

You are given the following "ARARX" model

$$A(\mathrm{q}^{-1})y(k) = B(\mathrm{q}^{-1})\,u(k) + \frac{1}{D(\mathrm{q}^{-1})}\,e(k)\,, \tag{1}$$

with

$$\begin{aligned} A(\mathrm{q}^{-1}) &:= 1 + a_1\mathrm{q}^{-1} + a_2\mathrm{q}^{-2}\,, \\ B(\mathrm{q}^{-1}) &:= b_1\mathrm{q}^{-1} + b_2\mathrm{q}^{-2}\,, \\ D(\mathrm{q}^{-1}) &:= 1 + d_1\mathrm{q}^{-1} + d_2\mathrm{q}^{-2}\,, \end{aligned}$$

with a canonical state-space parametrization as:

$$x(k+1) = \begin{pmatrix} -a_1 & 1 & 0 & 0 & 0 & 0 \\ -a_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -(a_1+d_1) & 1 & 0 & 0 \\ 0 & 0 & -(a_2+d_2+a_1d_1) & 0 & 1 & 0 \\ 0 & 0 & -(a_1d_2+a_2d_1) & 0 & 0 & 1 \\ 0 & 0 & -a_2d_2 & 0 & 0 & 0 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ b_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ 0 \\ -(a_1+d_1) \\ -(a_2+d_2+a_1d_1) \\ -(a_1d_2+a_2d_1) \\ -a_2d_2 \end{pmatrix} e(k),$$

$$y(k) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} x(k) + e(k). \tag{2}$$

Please note that for this exercise you are encouraged to use the notation (k) to indicate a variable at discrete time k. The goal of this exercise is to prove that there is a one-to-one correspondence between equations (1) and (2).

(a)
1. Start with writing out the update equation for each state variable $x_1(k+1), \dots, x_6(k+1)$. Also write out the output equation for $y(k)$ as a function of the state variables $x_1(k), \dots, x_6(k)$ and noise $e(k)$.
2. Which state variables are directly related to the output $y(k)$?
3. Use the equations you wrote down for question (a)-1 to write $x_1(k)$ as the output of a second-order transfer function model with input u_k, i.e., $x_1(k) = F^u(\mathrm{q}^{-1})\,u_k$.
4. Likewise, derive an expression for $x_3(k)$ as the output of a 4th-order transfer function model with input e_k, i.e., $x_3(k) = F^e(\mathrm{q}^{-1})\,e_k$.
5. Now prove that there is a one-to-one correspondence between equations (1) and (2).

(b) Derive the predictor for both the transfer-function and state-space models. What are the transfer functions $\hat{G}$ and $\hat{H}$ for the predictor in transfer-function form and how many poles and zeros do $\hat{G}$ and $\hat{H}$ have?

Exercise 2

In this exercise we assume that the data we measured was generated by the following SISO ARMAX input-output model:

$$y_k = \frac{b_1 \mathrm{q}^{-1}}{1 + a_1 \mathrm{q}^{-1}}\, u_k + \frac{1 - c_1 \mathrm{q}^{-1}}{1 + a_1 \mathrm{q}^{-1}}\, e_k, \tag{3}$$

with u_k and e_k ergodic, and statistically independent white-noise sequences with $\mathbb{E}[u_k] = \bar{u}$, $\mathbb{E}[e_k] = 0$ as well as variances σ_u^2 and σ_e^2, respectively. From this it follows that y_k is also ergodic. Given a dataset $\{u_k,\, y_k\}_{k=1}^{N}$, the goal of this exercise is to design a predictor $\hat{y}_k$ for the provided system that can be computed by solving a regular least-squares problem.

Hint: *You may find the following information helpful.*

- Note that if the random process x_k is ergodic, the following limit holds:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} x_k = \mathbb{E}[x_k].$$

- In the limit of $N \to \infty$, $\mathbb{E}[y_k] = \mathbb{E}[y_{k-j}]\ j \geq 0$.
- The finite sum of a geometric series can be computed as

$$\sum_{n=0}^{p} r^n = \frac{1 - r^{p+1}}{1 - r}.$$

(a) A possible predictor for the system in equation (3) is written as

$$\hat{y}_k = \phi_k\, \theta$$

where

$$\theta = \begin{pmatrix} b_1 & -(c_1 + a_1) & c_1 \end{pmatrix}^{\mathrm{T}}.$$

Derive the entries of the row vector ϕ_k. Can θ be learned from solving a linear least-squares problem? Motivate your answer.

In the following, we aim to approximate the given first-order ARMAX system with a high-order ARX model. We will show that this approximation is unbiased if we select the order sufficiently large. To this end, the required steps are provided as follows:

(b) Let the Maclaurin series expansion of $\dfrac{1}{1 - c_1 \mathrm{q}^{-1}}$ be written as

$$\frac{1}{1 - c_1 \mathrm{q}^{-1}} = \sum_{n=0}^{\infty} m_n\, q^{-n}.$$

Derive explicit expressions for m_0, m_1, m_2, m_3, m_p. In addition, provide the condition on c_1 under which the summation is convergent.

(c) Assume the Maclaurin series is convergent. Use a p-th order Maclaurin series that you derived in subquestion (b) to rewrite (3) as an ARX model, i.e.,

$$y_k^{\mathrm{ARX}} = \frac{\mathcal{B}(\mathrm{q}^{-1})}{\mathcal{A}(\mathrm{q}^{-1})} u_k + \frac{1}{\mathcal{A}(\mathrm{q}^{-1})} e_k.$$

Provide the polynomials $\mathcal{B}(\mathrm{q}^{-1})$, $\mathcal{A}(\mathrm{q}^{-1})$, and the order of the resulting ARX model. How many pole-zero cancellations are there in the transfer function $\mathcal{B}(\mathrm{q}^{-1})/\mathcal{A}(\mathrm{q}^{-1})$?

(d) Write the corresponding predictor for the driven ARX model in (c) in the form of $\hat{y}_k^{\mathrm{ARX}} = \phi_k\, \theta^{\mathrm{ARX}}$.

(e) Now show that

$$\lim_{p\to\infty} y_k - y_k^{\mathrm{ARX}} = 0\,,$$
$$\lim_{N\to\infty} \mathbb{E}[y_k] = \frac{b_1\bar{u}}{1+a_1}\,,$$
$$\lim_{N,p\to\infty} \mathbb{E}\Big[y_k - \hat{y}_{k|k-1}^{\mathrm{ARX}}(\theta^{\mathrm{ARX}})\Big] = 0\,.$$

Exercise 3

Instruction Note: Use the live-script template Homework3_template.mlx for this exercise. In this exercise, there are **four Test-Function parts** within the Homework3_template.mlx file that you need to execute. To this end, we have provided you with *all the functions' templates* (m-files) in a folder named "**function_folder**". You will see it in the assignment package. It is essential not to change the name of the folder at all in order to perform the Test-Function parts correctly. After finishing each function, please copy the code into the same function template in the Function section of the Homework3_template.mlx file. Once you have completed everything, execute the Homework3_template.mlx file and export it as **a single PDF-file**.

Consider the following Output Error system:

$$y(k) = \frac{b_2 q^{-2} + b_3 q^{-3} + b_4 q^{-4}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3} + a_4 q^{-4}} u(k) + e(k). \tag{4}$$

You will use MATLAB to identify the parameters and initial conditions of the given system using the input-output data from iodata.mat.

(a) Parameterize the system (4) using the observable canonical form (according to the lecture slides!).

(b) How many parameters do we have in part (a) ($\theta \in \mathbb{R}^p$)? Represent as $\theta = \begin{pmatrix} \theta_{\bar{A}} \\ \theta_{\bar{B}} \\ \theta_{x_0} \end{pmatrix}$.

(c) We are now going to implement a Prediction Error method (pem.m) for the Output Error system (4). We will do this in the following four steps:

   I. Derive expressions for $\dfrac{\partial \bar{A}(\theta)}{\partial \theta_p^{(i)}}, \dfrac{\partial \bar{B}(\theta)}{\partial \theta_p^{(i)}}, \dfrac{\partial K(\theta)}{\partial \theta_p^{(i)}}, \dfrac{\partial C(\theta)}{\partial \theta_p^{(i)}}, \dfrac{\partial D(\theta)}{\partial \theta_p^{(i)}}, \dfrac{\partial x_0(\theta)}{\partial \theta_p^{(i)}}$ (for all p).

   II. Based on the result in part (a), implement a MATLAB function that computes the state-space matrices from the parameter vector θ. (*Look at* theta2matrices.m *in the provided template*).

   III. Write a function simsystem.m that simulates a dynamic system given any input vector u and matrices A, B, C, D, as well as an initial condition $x(0)$.

   IV.
   - Implement a function jacobian.m that calculates the Jacobian vector.
   - Implement a function hessian.m that calculates the approximated Hessian matrix.

   Note: The convergence loop for the regularized Gauss-Newton algorithm is already implemented in the pem.m function. There is no reason to edit this.

(d) Choose two different initial guesses for θ. For each guess, train two models such that the first model is obtained from the first 500 samples (from the given iodata.mat as (ut1,yt1)) and the second one is obtained from the first 1000 samples (from the given iodata.mat as (ut2,yt2)). Then, apply all four identified models on **the validation set** (from the given iodata.mat as (uv,yv)) and plot the predicted output of all models (only for the validation set) in one figure. Add the numerical values of the VAF and the RMSE to the legend of this figures. (Do not forget to label the models properly!)

(e) According to the result in (d), report the system parameters a_1, a_2, a_3, a_4, b_2, b_3, b_4 for the best identified model. Explain how you chose the best model.