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 **19-12-2022** 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 [30 points]

You are given the Box-Jenkins model

$$y_k = \frac{b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a}} u_k + \frac{1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_{n_c} q^{-n_c}}{1 + d_1 q^{-1} + d_2 q^{-2} + \cdots + d_{n_d} q^{-n_d}} e_k. \tag{1}$$

In this exercise, you are also given the following parameterization of a SISO state-space model:

$$x_{k+1} = \begin{pmatrix} -a_1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n_a-1} & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ -a_{n_a} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & -d_1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & -d_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -d_{n_d-1} & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 & -d_{n_d} & 0 & 0 & \cdots & 0 \end{pmatrix} x_k + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_b-1} \\ b_{n_b} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} u_k + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ c_1 - d_1 \\ c_2 - d_2 \\ \vdots \\ c_{n_d-1} - d_{n_d-1} \\ c_{n_d} - d_{n_d} \end{pmatrix} e_k,$$

$$y_k = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \end{pmatrix} x_k + e_k. \tag{2}$$

(a) **[10 points]** Prove that there is a one-to-one correspondence between (1) and (2) for the case of $n_a = 3$, $n_b = 2$, $n_c = 1$ and $n_d = 2$.

(b) **[5 points]** Derive the predictor for both the transfer-function model and state-space model for the given case. Will the predictor in transfer-function and state-space form have the same number of poles?

(c) **[15 points]** Assume an additional term b_0 is added to the Box-Jenkins model such that:

$$y_k = \frac{b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a}} u_k + \frac{1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_{n_c} q^{-n_c}}{1 + d_1 q^{-1} + d_2 q^{-2} + \cdots + d_{n_d} q^{-n_d}} e_k. \tag{3}$$

Derive the parameterization for the corresponding state-space model in the same form as in (2) for the case of $n_a = 3$, $n_b = 2$, $n_c = 1$ and $n_d = 2$. What are the differences with respect to the state space model shown in (2)? Is there a time delay from input to output when $b_0 \neq 0$?

Exercise 2 [25 points]

Consider the following generic ARX model:

$$y_k = \frac{b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}} u_k + \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}} e_k \tag{4}$$

We have access to N measurements of $y_k \in \mathbb{R}$ and $u_k \in \mathbb{R}$. Suppose additionally that $e_k \sim \mathcal{N}\left(0, \sigma_e^2\right)$.
In this exercise, we aim to apply Bayes' theory to learn the parameters $\theta = (b_1,\ b_2,\ \cdots\ b_{n_b}\ a_1\ a_2\ \cdots\ a_{n_a})$ from the data.

(a) **[5 points]** Suppose that there are N measurements of y_k and u_k available. We define the vector

$$\mathbf{y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_N \end{pmatrix}^T \in \mathbb{R}^N \tag{5}$$

and likewise we define the vectors $\hat{\mathbf{y}} \in \mathbb{R}^N$ and $\mathbf{e} \in \mathbb{R}^N$. Before we can apply Bayes' theorem to learn the model parameters from data, we first need to rewrite the model as

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{e} = \mathbf{\Phi}\theta + \mathbf{e}.$$

Derive the expression for the k-th row of the $\mathbf{\Phi} \in \mathbb{R}^{N \times (n_a + n_b)}$ matrix.

(b) **[5 points]** We now assume that e is white Gaussian noise for the ARX model in equation (5). Give explicit expressions for the mean vector $\mathbb{E}[\mathbf{e}] = \mu_e$ and covariance matrix $\mathbb{E}\left[(\mathbf{e} - \mu_e)(\mathbf{e} - \mu_e)^T\right] = P_e$.

(c) **[10 points]** Considering the noise model $p(e) = \mathcal{N}(\mu_e, P_e)$ from question (b) and prior $p(\theta) = \mathcal{N}(\mu_\theta, P_\theta)$, use Bayes' theorem to derive expressions for the mean and covariance matrix of the posterior $p(\theta \mid y)$. Use the expressions for μ_e, P_e from question (b) in your answer.

(d) **[5 points]** Denote the mean and covariance matrix of the posterior $p(\theta \mid y)$ by θ_{post} and P_{post}, respectively. We will now use this posterior as a "prior" when making a prediction for y_{N+1}. What are then the mean and variance of the predictive distribution $p(y_{N+1} \mid y)$?

Exercise 3 [45 points]

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:

$$y(k) = \frac{b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3}}{1 + a_1 q^{-1} + 2_2 q^{-2} + a_3 q^{-3} + a_4 q^{-4} + a_5 q^{-5}} u(k) + e(k). \tag{6}$$

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

(a) **[5 points]** Parameterize the system (6) using the observable canonical form.

(b) **[2 points]** 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 (6). We will do this in the following four steps:

I. **[5 points]** Derive expressions for $\frac{\partial \bar{A}(\theta)}{\partial \theta_p^{(i)}}, \frac{\partial \bar{B}(\theta)}{\partial \theta_p^{(i)}}, \frac{\partial K(\theta)}{\partial \theta_p^{(i)}}, \frac{\partial C(\theta)}{\partial \theta_p^{(i)}}, \frac{\partial D(\theta)}{\partial \theta_p^{(i)}}, \frac{\partial x_0(\theta)}{\partial \theta_p^{(i)}}$ (for all p).

II. **[5 points]** 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. **[5 points]** 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. **[5 points]**

- 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) **[15 points]** Choose two different initial guesses for θ. For each guess, train two models such that the first model is obtained from the first 5000 samples (from the given `iodata.mat` as `(ut1,yt1)`) and the second one is obtained from the first 10000 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) **[3 points]** According to the result in (d), report the system parameters a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3 for the best identified model. Explain how you chose the best model.