Course: 5SMB0 System Identification 2018/2019
Final Assignment

Final Assignment

General setup

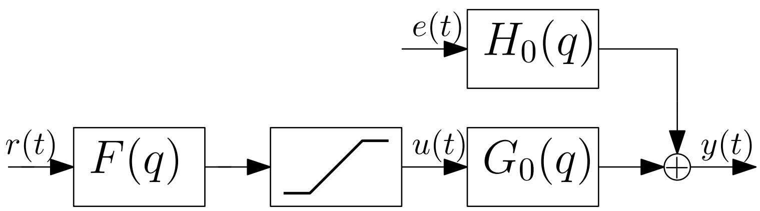

Figure 1: Block scheme of the system to identify

We are given a linear time-invariant system $G_0(q)$ which we are asked to identify as accurately as possible. The output $y(t)$ obeys the equations:

$$y(t) = G_0(q)u(t) + H_0(q)e(t)\,,$$

where $H_0(q)$ is a stable and monic filter and $e(t)$ is white noise.

The system is represented through an encrypted Matlab function (extension `.p`) named `assignment_sys_xx.p`, where xx is the Group number. The function implements the block scheme of Fig. 1: it accepts as input a vector r representing the samples of the signal $r(t)$ and sends out two vectors u and y representing the samples of the input and output signals of $G_0(q)$.

For example, the function can be called as follows:

```
[u,y] = assignment_sys_xx(r)
```

Therefore, the experimenter has the freedom to feed the system with a prescribed signal $r(t)$, which will be converted inside the function into the actual input $u(t)$. The conversion is made through the two blocks of Fig. 1. The first block is a linear time-invariant filter; more precisely, it is a Butterworth filter given by

$$F(q) = \frac{0.505 + 1.01q^{-1} + 0.505q^{-2}}{1 + 0.7478q^{-1} + 0.2722q^{-2}}.$$

The filter acts as safety device for the system actuator, which cannot be excited at high frequencies. The filter damps all the frequency contents that are above a certain cut-off frequency. The second block is a saturation $S(\cdot)$, namely a static function that transforms every sample of x as follows:

$$S(x) = \begin{cases} M & \text{if } x \geq M \\ x & \text{if } -M < x < M \\ -M & \text{if } x \leq -M \end{cases}$$

In the above equation, M is a real number that is unknown.

Part 1: Understanding saturation and Butterworth filter

1.1 It is important to understand the characteristics of $F(q)$. Plot its frequency response (Bode diagram). What is the cut-off frequency at 3dB (expressed as $x\,\pi$, where $x \in (0,1)$)? **Hint:** A linear scale in the horizontal axis may be preferable in this case.

1.2 It is required to determine M. To this end, design a signal $r(t)$ that allows to accomplish this task. Explain why it is important to determine M.

Part 2: Nonparametric identification

2.1 We first want to get a rough idea of the frequency behavior of $G_0(q)$. To this end, we want to perform a preliminary experiment and apply nonparametric system identification using the collected data. The task is the to design an input length $N = 1024$ that excites the system at 128 frequencies. In designing the input, we have to take into account the presence of $F(q)$. What type of input shall be used? Motivate the choice.

2.2 Display the Bode plot of the identified system. Can we conclude anything about:

- High-frequency behavior of $G_0(q)$;
- Resonance peaks.

2.3 Derive an expression of the noise spectrum and plot the magnitude of the noise filter in a Bode diagram.

Part 3: Experiment design

3.1 Based on the findings of the nonparametric identification, we want to get a parametric model of $G_0(q)$. To get the most from this experiment, we want to design a PRBS input that takes into account the filter $F(q)$ and the saturation. Design the signal such that the total energy is lower than $E = 12 \times 10^3$. Motivate the design choices. What is the maximum length of the experiment that you can obtain when the power of the signal is equal to M^2?

3.2 Explain briefly why in your opinion a filtered Gaussian process is not a preferable choice for $r(t)$ in this setup.

Part 4: Parametric identification

4.1 Using the designed input, perform a parametric identification of the system using an OE model of the type

$$\mathcal{M}_{OE} = \left\{ G(q,\theta) : G(q,\theta) = \frac{B(q)}{F(q)} = \frac{\sum_{i=1}^{n_b} b_i q^{-i}}{1 + \sum_{i=1}^{n_f} f_i q^{-i}} \right\},$$

where $n_b = 1$ and $n_f = 2 \times \#$ resonance peaks. Plot the obtained residuals. Is the result satisfactory? How could we improve it?

4.2 Adjust n_f and n_b until we get better results from the residuals test. Comment on these results: why are they not completely satisfactory yet?

4.3 Further improve the outcome of the identification procedure by considering a different model class.

- What additional characteristics should this model class have?
- Which model class would choose between ARX, ARMAX, Box-Jenkins?

4.4 Identify a model of the system using the newly chosen model class. Tune the model order such that the residual test give a good result. Motivate the choice of model order. Report the chosen model orders and a plot of the residuals.

Part 5: Validation

5.1 We have the chance to perform a new experiment on the system, which we can use to further assess our choice of model class and order. Using the input designed in the previous part, identify several models of the same model class chosen previously, but with different orders for $G(q,\theta)$. The orders to test are specified in Table 1.

5.2 Simulate the identified model using a new RBS input of length 3000 with suitable amplitude and band. Plot the results in a plane that has $\mathcal{M}_i$ in the horizontal axis and $V_N(\hat{\theta}_i)$ in the vertical axis, with $\hat{\theta}_i$ corresponding to the identified parameter for model $\mathcal{M}_i$. Give a brief explanation on the behavior of the curve. **Hint:** You may repeat the procedure several times to get more insights (the curve may change using different data sets).

5.3 Does the minimum of $V_N(\theta_i^*)$ correspond to the model order established previously? If not, adjust your selected model accordingly.

Model	n_b	n_f
$\mathcal{M}_1$	2	3
$\mathcal{M}_2$	2	4
$\mathcal{M}_3$	3	3
$\mathcal{M}_4$	3	4
$\mathcal{M}_5$	5	5
$\mathcal{M}_6$	8	8
$\mathcal{M}_7$	9	10
$\mathcal{M}_8$	11	12

Table 1: Model orders to test.

Part 6: Theoretical verification

6.1 We would like to get an understanding of the theoretical performance of the identification procedure. To this end, we are going to perform Monte Carlo simulations on the system as follows:

1. Generate a RBS signal $r(t)$ of length $N = 1000$ and suitable band and range;
2. Generate an input/output data set using $r(t)$ and, using the collected data, identify an OE (not a Box-Jenkins) model having the orders of $B(q)$ and $F(q)$ found in the last part;
3. Store the identified parameter vector;
4. Repeat the previous two steps 100 times.

6.2 Inspect the parameters of obtained from the Monte Carlo simulations. Do we always get similar results from the simulations? If not, could you explain why?

6.3 Compute the mean and the variance of the identified coefficients of $B(q)$ and $F(q)$ (over the 100 Monte Carlo runs). The corresponding Matlab commands are `mean` and `var`. Compare the obtained variances with the theoretical variances. Note that the theoretical variances are directly obtained from Matlab using the command `getcov(Model_Name)`, where `Model_Name` is the name of the variable containing the identified model (e.g., `Model_Name = oe(data,[nb nf nk])`. Note that the parameters variances are the diagonal elements of the matrix that you get from the command. Do the Monte Carlo variances match the theoretical ones? Why?

6.3 Repeat the Monte Carlo simulations, this time initializing the parameters search to a prescribed point. To do so, replace the vector of orders in the Matlab commands for model identification with an object of the type `idpoly`. For example, we can initialize the parameter search for an OE model to the polynomials $B_{init}(q)$, $F_{init}(q)$ as follows:

```
M_init = idpoly([],B_init,[],[],F_init);
M_arx = armax(data, M_init);
```

The polynomials to use in this case are obtained by computing the *median* (not the mean!) of the polynomials obtained from the previous Monte Carlo simulations (command `median`). Compute the mean and the variance of the identified coefficients of $B(q)$ and $F(q)$ (over the 100 Monte Carlo runs). What differences do we observe compared to the previous simulations? Do the Monte Carlo variances match the theoretical ones? Why?

6.3 Repeat the Monte Carlo simulations, this time using a model order where n_b and n_f are both decreased by 2. Comment on the results and explain the differences with respect to the previous results.

Instructions and general recommendations

- Submit your report through Canvas. The report consists of a typewritten pdf file. Handwritten reports shall not be considered.
- Give concise and focused answers that get straight to the point, do not get lost into lengthy explanations.
- Try to fit your report to 6 pages.
- In using Matlab, you are free to either use the System Identification Toolbox GUI, or the inline commands (e.g. `arx()`), or both. If you use the GUI, try to describe the procedure (which commands you click on, etc.); if you use inline commands, you can write them in the report (at least the most important ones).
- Most of the algorithms to generate inputs, identify systems, etc., are already implemented in Matlab! The following is a list of useful commands:
  - `bode,freqz`: frequency response (in logarithmic or linear scale)
  - `iddata`: format data for the system identification toolbox commands
  - `idinput`: generate various input signals
  - `etfe`: nonparametric identification
  - `arx,armax,bj,oe`: identify parametric models
  - `Model_Name.a,Model_Name.b,...`: get the polynomials $A(q)$, $B(q)$, ..., from an identified model called `Model_Name`
  - `resid`: residuals test
  - `sim`: simulate the response of a model given an input

  Type `help ident` for a complete list.