Course: 5SMB0 System Identification 2021/2022

Final Assignment

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 into 10 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
  - `pe`: allows to calculate the prediction error (residual) signal

  Type `help ident` for a complete list.

General setup

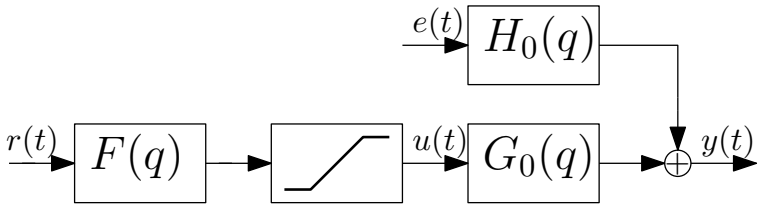

Figure 1: Block scheme of the system to identify

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 then to design an input with length $N = 1024$ that excites the system at 128 frequencies within the passband of the Butterworth filter. What type of input shall be used? Give an expression of $r(t)$ and motivate your 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 estimating the noise spectrum Φ_v and plot the magnitude of the estimated noise spectrum in a Bode diagram. **Remark:** In case you choose to use the `cpsd` matlab function, be aware that it implements Welch's averaging method.

Part 3: Experiment design

3.1 Based on the findings of the nonparametric identification, we want to get a parametric model of $G_0(q)$. However, the experiment is limited to only $N = 3000$ data points. Given the choice between a filtered white noise process with a Gaussian pdf, and a PRBS, explain briefly which choice for $r(t)$ is preferable in this setup.

3.2 Design the external input signal r in such a way that within the experimental constraints, maximum power is supplied to the plant input u, while there is no substantial power loss in the step from r to u.

Part 4: Parametric identification and validation

4.1 Using the designed input from 3.2, we want to perform a parametric identification of the system that yields *a consistent* estimate of G_0. For this purpose use a data set

of length $N = 3000$. Motivate the choice(s) that you make for the model structure and model orders and provide at least two validation tests (for example, residual test, cross validation, model simulation, AIC...).

Part 5: Experimental verification of variance estimates

5.1 We would like to get an understanding of the mechanism behind the estimation of the variance of identified parameters. To this end, we are going to introduce experiment repetition as a way to estimate the variance of an estimated model. We do this by performing Monte Carlo simulations on the system as follows:

1. Generate a RBS signal $r(t)$ of length $N = 1000$ and suitable frequency band and amplitude;
2. Generate an input/output data set using $r(t)$ and, using the collected data, identify *an OE* model having the orders of $B(q)$ and $F(q)$ that correspond to the validated $G(q, \theta)$ found in the previous section;
3. Store the identified parameter vector;
4. Repeat the previous two steps 100 times.

5.2 Inspect the parameters obtained from the Monte Carlo simulations. Do we always get similar results from the different simulation runs? If not, could you explain why?

5.3 Using the collected data, we would like to compare the simulated statistics (via Monte Carlo) with what we will call the *theoretical variance* (estimated covariance matrix based on one single experiment). This estimated covariance matrix can be obtained directly from Matlab using the command `getcov(M_i)`, which implements the covariance matrix estimate (4.121)-(4.122)[1] for all model structures with a parametrized noise model, and (4.138)-(4.140) for model structures with a non-parametrized noise model. The estimated parameter variances occur as the diagonal elements of the estimated covariance matrix. The simulated statistics are obtained by computing the mean and variance of the identified coefficients of $B(q)$ and $F(q)$ (over the 100 Monte Carlo runs)— commands are `mean` and `var`.
Do the Monte Carlo variances match the theoretical ones? Why?

5.4 In the previous part, every OE run starts from a different point "randomly" chosen by the toolbox. This time, repeat the Monte Carlo simulations 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:

[1] Equation numbers are taken from the Lecture Notes.

```
M_init = idpoly([],B_init,[],[],F_init);
M_oe = oe(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?

Part 6: Estimation of a Box Jenkins model for minimum variance

6.1 We are now extending our aim from estimating a consistent model $G(q, \hat{\theta}_N)$ towards estimating a model $G(q, \hat{\theta}_N)$ with minimum variance. We do this by estimating a Box Jenkins model in the same setting (input signal, data length) as applied in Part 5. Estimate an appropriate Box Jenkins model, motivate the choice of the model order of the noise model, e.g., by adding a plot of the residual tests.

6.2 Evaluate the estimated theoretical covariance matrix related to the estimated $G(q, \hat{\theta}_N)$ of the Box Jenkins model in 6.1, and relate it to the result obtained from the variance analysis of the output error model in Part 5. What can you conclude with respect to the parameter variances?
Hint: you can remove the columns and rows from the covariance matrix that are related to the noise model parameters.