5SMB0 System Identification

Final Assignment

Date:
19-4-2019

Names:	**Student number:**
Bob Clephas	1271431
Job Meijer	1268155

Introduction

In this exercise the following system is identified:

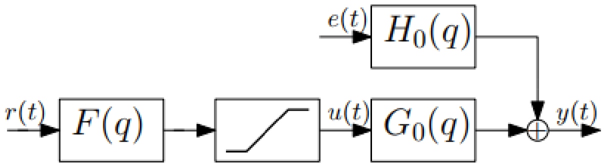

Figure 1 – Block scheme of system to identify

Where $F(q) = \frac{0.505+1.01q^{-1}+0.505q^{-2}}{1+0.7478q^{-1}+0.2722q^{-2}}$ and the output of $F(q)$ contains a saturation $S(x)$ resulting in the signal $u(t)$. $S(x)$ is described as follows:

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

The signal $u(t)$ is inserted in the linear time-invariant system $G_0(q)$, which is identified in this exercise as accurately as possible. The only controllable input of the system is $r(t)$ which is inserted into $F(q)$. Furthermore $H_0(q)$ is a stable and monic filter and $e(t)$ is white noise. This results in the output $y(t) = G_0(q)u(t) + H_0(q)e(t)$.

Part 1: Understanding saturation and Butterworth filter

1.1 The cutoff frequency of $F(q)$ is 0.7. This is determined by inserting the transfer function of $F(q)$ in Matlab, creating a Bode plot shown in Figure 2 and determining the frequency at which the amplitude is -3dB. This resulted in a frequency of 2.2 rad/s on a scale of 0 to π, resulting in $\frac{2.2}{\pi} = 0.7$.

1.2 A ramp signal for $r(t) = t$ is used to determine the saturation M. This resulted in $M = 2$ because $u(t)$ did not increase beyond this value as shown in Figure 3. A negative ramp resulted in the same behavior.

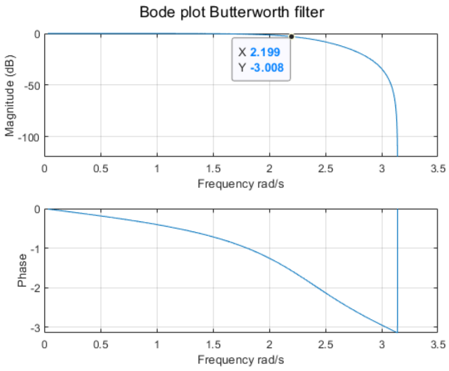

Figure 2 – Bode plot of F(q)

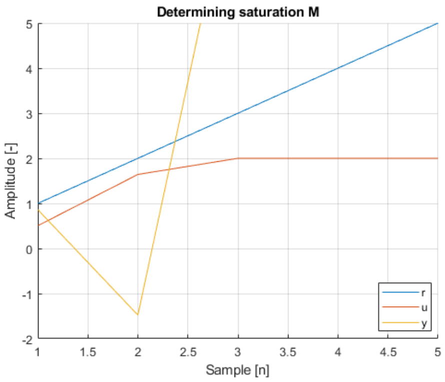

Figure 3 – Determining saturation M

Part 2: Nonparametric identification

2.1 To get a rough idea of the frequency behavior of $G_0(q)$ a preliminary experiment is designed to do nonparametric system identification. For this experiment an input is designed with length $N = 1024$ that excites the system at 128 different frequencies. Because the system can only be excited via input $r(t)$, the presence of $F(q)$ and $S(x)$ has to be taken into account. The designed $r(t)$ is described below:

$$r(t) = \sum_{i=0}^{N_f} A \cdot \cos\left(2\pi\left(\frac{F}{F_s}\right)n + \phi\right)$$

Where

$$A = 0.06 \qquad \phi = 2\pi \cdot rand(1,1) \qquad F = Fsteps \cdot i$$
$$N = 1024 \qquad n = [0, N-1] \qquad N_f = 128$$
$$F_s = 1024 \qquad F_{max} = 512 \cdot 0.7 \qquad F_{steps} = round\left(\frac{F_{max}}{N_f}\right)$$

Applying this signal to the system and estimating the frequency behavior of $G_0(q)$ resulted in Figure 4. A random phase shift and a relatively low amplitude is applied to the cosine in order to keep the amplitude below the maximum.

2.2 From the Bode plot in Figure 4 can be concluded that there are two resonant peaks in $G_0(q)$ one around 0.22π and one around 0.5π. At higher frequencies, the magnitude of $G_0(q)$ drops and $H_0(q)$ becomes active.

2.3 An expression of the noise spectrum and a Bode plot of the noise filter $H_0(q)$ are determined. To determine the noise spectrum, it is recalled that $e(t)$ is white noise and that $H_0(q)$ is stable and monic.

To determine the noise of the system, first $v(t) = H_0(q)e(t)$ is determined. When the input $r(t) = 0$ the output of the system is only determined by $v(t)$. Now a Power Spectral Density (PSD) estimate via Welch's method is used to determine the frequency behavior of $H_0(q)$. This is done using the Matlab function $pwelch\left(y, hann\left(\frac{N}{256}\right)\right)$ with and experiment length of $N = 2^{16}$ of which the results are shown in Figure 5.

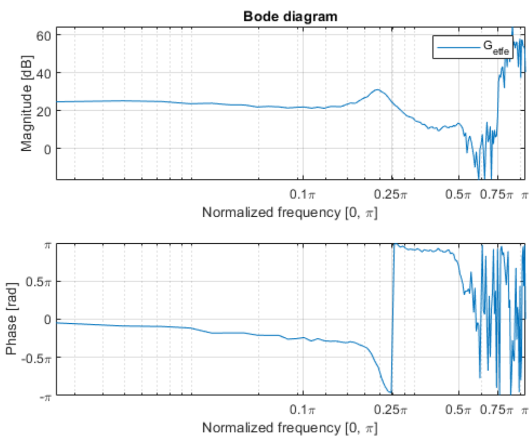

Figure 4 – Estimated frequency response of G(q)

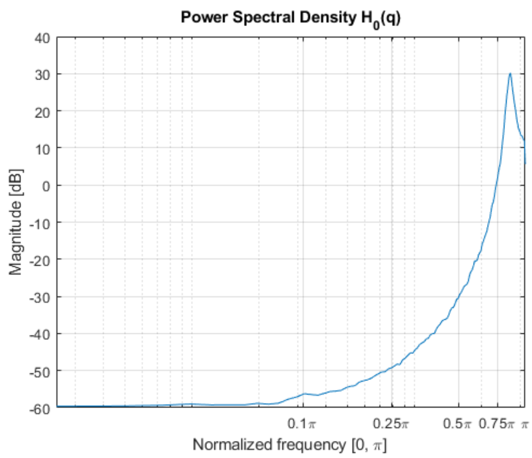

Figure 5 – Welch Power Spectral Density Estimate of $H_0(q)$

Part 3: Experiment design

3.1 To generate a suitable PRBS input signal, a clock based PRBS is used for which the power spectrum can be changed by changing the clock period. With a clock period of 1, the PRBS has a flat spectrum. To account for the filter which otherwise suppresses the high frequencies of the signal, a clock period of 2 is chosen which shifts the power from the higher frequency range to the lower frequency range. The smoothened power spectrum of the signal r can be seen in Figure 6.

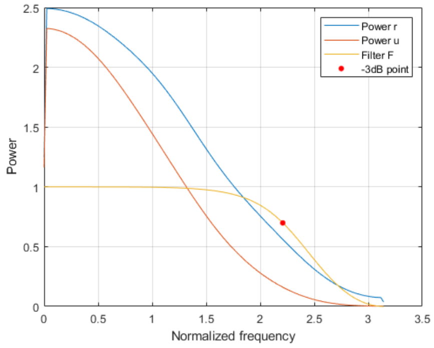

Figure 6 – Power spectrum PRBS signal

The length of the experiment is set to $N = 3000$ which results in an input energy of $E = M^2 \cdot N = 4 \cdot 3000 = 12 \cdot 10^3$. This corresponds with the measured energy in MATLAB. Using the same input 'r', generates a signal 'u' with an energy of $E = 8654$ which is lower than the energy of 'r' due to the filter.

3.2 A Gaussian process can generate samples between $[-\infty, \infty]$. If a filtered Gaussian process was used as input in this setup, the filter could be chosen such that most of the samples are in between the saturation bounds but then a lot of signal power would be lost as most of the signal would have an amplitude close to zero.

Part 4: Parametric identification

4.1 The earlier designed input signal $r(t)$ is now used to perform a parametric identification of the system using an OE model of the following 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

$n_b = 1$

$n_f = 4\ (2 \cdot number\ of\ resonance\ peaks)$

This identification is executed using the System Identification toolbox GUI in MATLAB, resulting in the model residuals shown in Figure 7. The PRBS input designed in part 3 is used during this assignment.

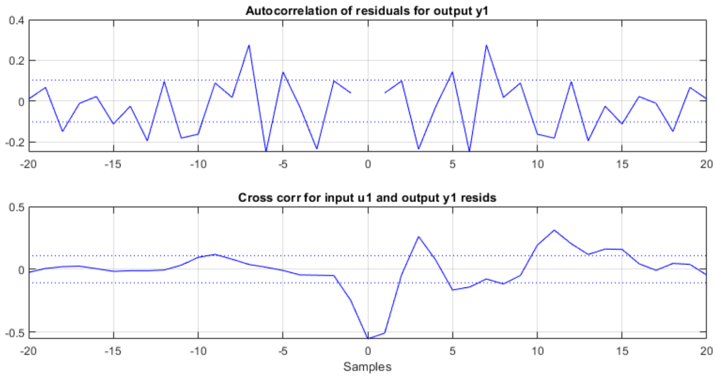

Figure 7 – Model residuals of $\mathcal{M}_{OE}$ fit

The cross correlation does not have the characteristics of white noise, indicating that the model $\mathcal{M}_{OE}$ does not completely fit the system $G_0(q)$. This could be improved by increasing the order of the $\mathcal{M}_{OE}$ model. Also does the autocorrelation not have a dominant peak at zero, indicating that it is not only white noise.

4.2 Now n_f and n_b are adjusted until the cross correlation is between the bounds, with the smallest possible model order. This resulted in $n_f = 5$ and $n_b = 7$ with the residuals visible in Figure 8. Using this model structure, it is not possible to improve both the autocorrelation and the cross correlation, because the noise model is threated as being one, which it is not.

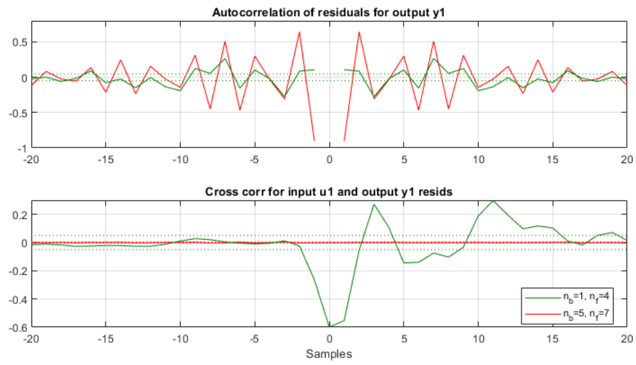

Figure 8 – Model residuals of $\mathcal{M}_{OE}$ fit with increased model order

4.3 Further improving the outcome of the identification procedure is done by considering a different model class that models the noise dynamics in $H_0(q)$. This results in the options ARX, ARMAX and Box-Jenkins. The ARX and ARMAX model structure share the denominator between $H_0(q)$ and $G_0(q)$ and ARMAX also has a numerator therm. With a Box-Jenkins model structure $G_0(q)$ and $H_0(q)$ are independently parameterized, resulting in the denominator of $H_0(q)$ being unrelated to $G_0(q)$.

The choice for best model structure depends on the expected denominator of $H_0(q)$. In this case it is expected that the denominator is different, thus a Box-Jenkins model structure is chosen. This expectation is based on the earlier obtained results in Figure 4 and Figure 5 in which is visible that $G_0(q)$ has low frequent behavior and $H_0(q)$ has high frequent behavior.

4.4 Now the model of the system is identified with the Box-Jenkins model structure. After some tuning, this resulted in the following settings: $n_b = 5, n_f = 7, n_c = 1, n_d = 2, n_k = 0$. The order of the noise model for $H_0(q)$ is based on the single resonance peak that it has, resulting in two poles ($n_d = 2$). The parameter n_c is set to 1, resulting in a single gain. Lowering the earlier determined parameters n_b and n_f resulted in worse residuals. The resulting residuals of the determined model are shown in Figure 9. The autocorrelation and cross correlation are close to 0, indicating a good model fit.

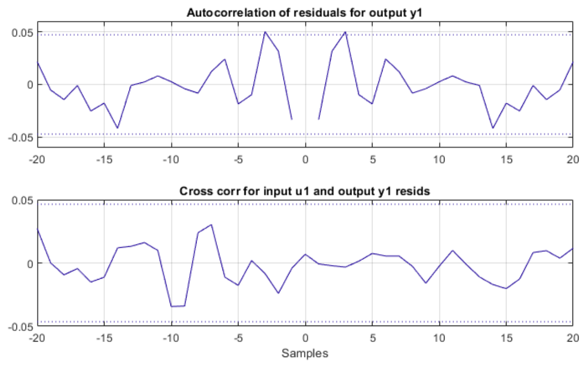

Figure 9 – Auto- and cross correlation of determined Box-Jenkins model

Part 5: Validation

5.1 To further assess the choice of model class and order, the earlier designed PRBS input is applied again to the system. Now the Box-Jenkins model is determined again but with the orders for $G(q, \theta)$ shown in Table 1.

Table 1 – Model orders for validation

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

5.2 Now the identified models are simulated using both the old PRBS input signal used for determining the model and a newly generated PRBS input signal. The results are plotted in Figure 10, where the model number is plotted on the horizontal axis and the cost $V_N(\hat{\theta}_i)$ in the vertical axis.

The overall $V_N(\hat{\theta}_i)$ decreases over an increasing model number. There is a minimal difference in the cost when calculating it with a new dataset. Most of the time model 5 has the lowest cost when repeating this experiment, indicating that this is the best fitting model. The curve can vary wildly with different data sets, sometimes one model can have a relatively high cost for a certain dataset as can be seen in Figure 11.

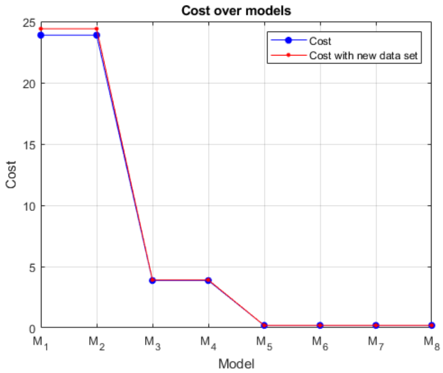

Figure 10 – Cost over different models

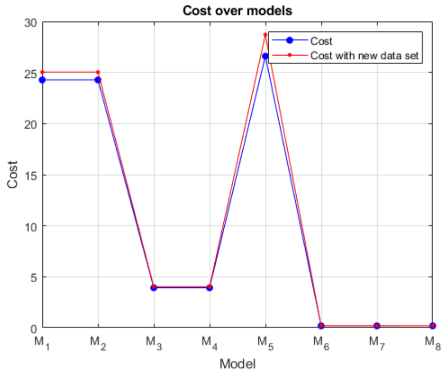

Figure 11 – Cost over different models 2

5.3 The model does almost correspond to the earlier determined model, where model number 5 has an order of 5 instead of the determined 7.

Part 6: Theoretical verification

6.1 For the Monte Carlo simulations an RBS input with a length of 1000, a clock of 2 and an amplitude of 2 is generated. Using this signal, an input/output data set u and y is generated. Afterwards this data set is used to identify an OE model with $n_b = 5$ and $n_f = 5$. The process of generating the data set and identifying a model is repeated 100 times.

6.2 As can be seen in Figure 12, the parameters vary a lot over the different iterations. This is probably due to different local minima for the optimization problem.

6.3 In Figure 13 the computed variance of the identified coefficients is compared with the theoretical variance. This theoretical variance is computed with median of the variance calculated with the 'getcov' command for each iteration. The calculated variances do not match the theoretical ones which is probably since the theoretical variance is calculated according to one local minima of the optimization problem and the calculated variance is based on identifications which can have different local minima.

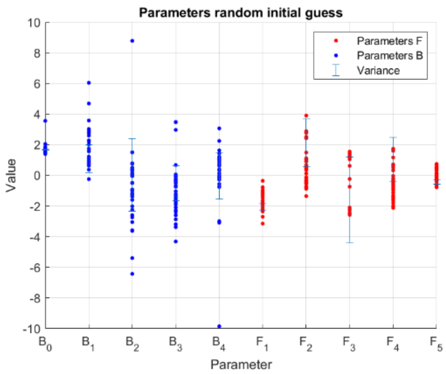

Figure 12 – Parameters random initial guess

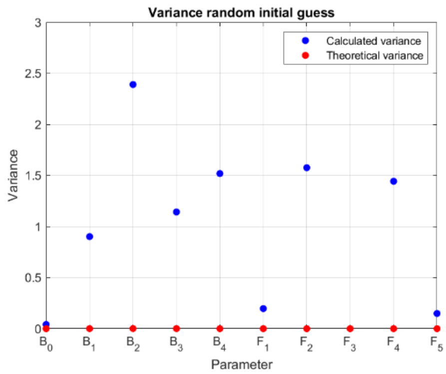

Figure 13 – Variance random initial guess

6.4 When using the median of the identified coefficients as initialization for the new identification, the variance is a lot smaller and matches the theoretical variance a lot better. The reason probably is that the optimization now converges to the same local minimum each time which is closest to the initialization point.

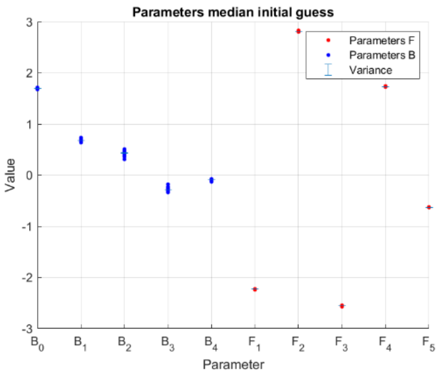

Figure 14 – Parameters median initial guess

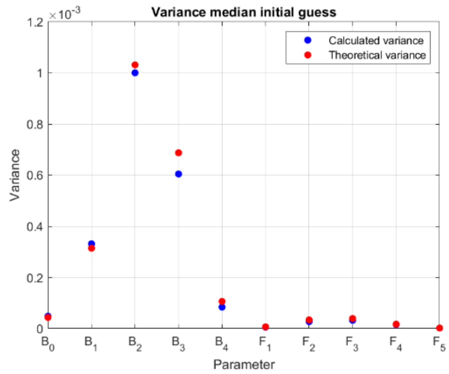

Figure 15 – Variance median initial guess

6.5 When repeating the experiment with a lower model order, the results are different. In this case the variance of the parameters with a random initial point are already low because they all converge to the same local minimum. The calculated variance is in this case with both the random initial guess as well as the median initial guess lower than the theoretical variances. This is probably because the model order is wrong and therefor the theoretical variance is calculated wrong. The parameters and variances with a random initial guess can be seen in Figure 16. For the median initial guess, the results are almost identical, because these parameters already converge to same local minimum.

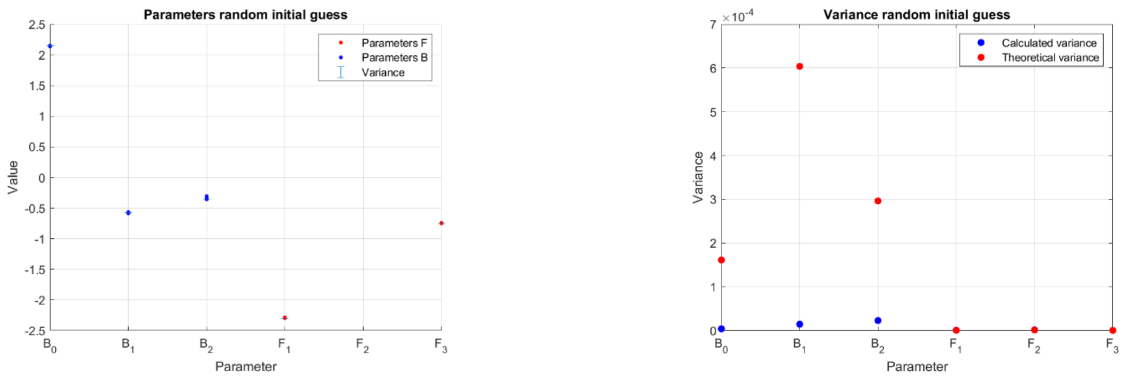

Figure 16 – Monte Carlo simulation lower order nb and nf