5SMB0 - System Identification

Graded Assignment - Group 12

Julian Vos (ID: 0890397)
Gregorius Rafael Widojoko (ID: 1773828)

Department of Electrical Engineering

April 22, 2022

1 Understanding saturation and Butterworth filter

In this part, for the identification process an input is needed, but the system is limited by a Butterworth filter and a saturation block.

Question 1.1

The input data is entered into the system and might contain noise. Therefore, the input should be filtered to minimize its effect.

The transfer function of the filter is given, in Fig. 1 the bode plot of the Butterworth filter is shown. a red star is used to indicate the bandwidth points, which is defined as the frequency at -3dB. The cut-off frequency at -3 dB is 0.69966 π rad/sample, found using the `bandwidth()` command.

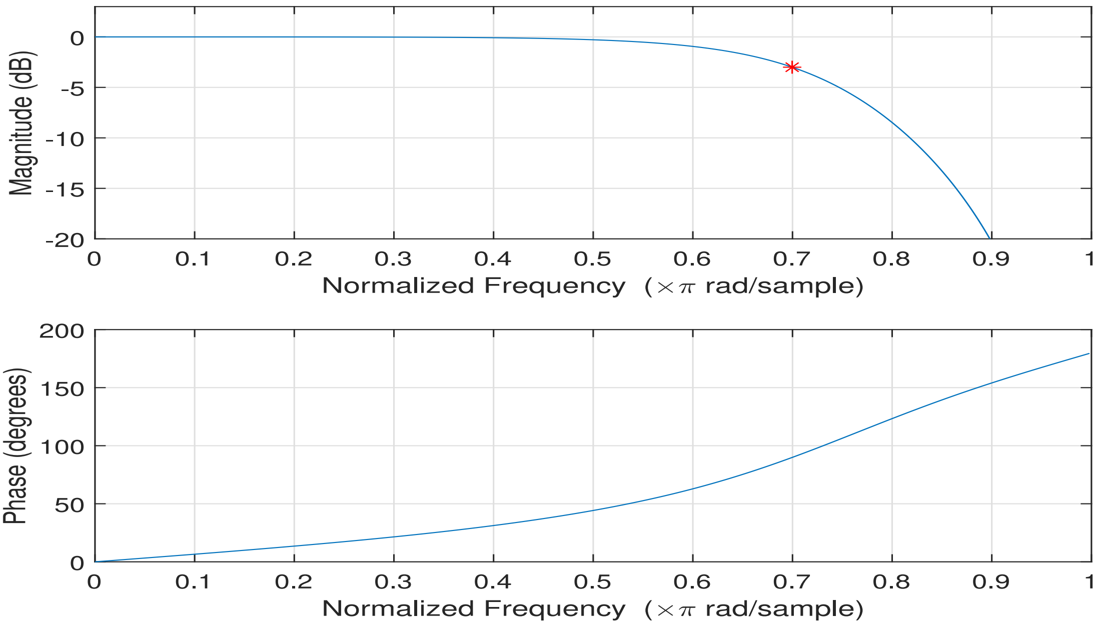


Figure 1: Bode plot of $F(z)$

Question 1.2

Since the input signal $r(t)$ might excite amplitudes beyond the system's capability a saturation block is present. Another reason for saturation is not to lose the input power due to the transmission process of $u(t)$ while determining the future input profiles.

We measured M by feeding the input $r(t)$= -1000:0.001:1000 to the given `assignment_sys_xx.p` file and toke the maximum value of the output. In this case, for `[y, u] = assignment_sys_12(r)` the M=`max(u)`=2. The same procedure applies for the minus counterpart.

$$u(t) = \begin{cases} 2 & r(t) > 2 \\ r(t) & |r(t)| \leq 2 \\ -2 & r(t) < -2 \end{cases} \tag{1}$$

2 Non-parametric identification

To have a rough idea about the system's behaviour, one can estimate the system characteristics, especially frequency behaviour, specifically its resonance peaks. In this section, the non-parametric identification approach is applied.

Question 2.1

With the non-parametric identification process, it is preferable to use a periodic input. As periodic inputs with period N_0, input $R_N(\omega) = 0$ at $\omega = k\frac{2\pi}{N_0}$, and so $\breve{G}_N(e^{i\omega})$ are unbiased for those frequencies.

With this motivation and the given $N_0 = 1024$ and $N = 128$ (excited frequency points), the input is defined as

$$r(t) = \frac{1}{\sqrt{128}} \sum_{k=1}^{128} \sin(k\omega_0 t) \text{ ; with } \omega_0 = \frac{2\pi}{1024} \approx 0.0061359 \text{ rad/s} \tag{2}$$

The defined $r(t)$ ensures there is no bias introduced to the non-parametric identification setup, based on the notion of general signals $\breve{G}_N(e^{i\omega})$, which are asymptotically unbiased for those frequencies where $\frac{1}{\sqrt{N}}U_N(\omega) \neq 0$. in Matlab, this can be achieved with the `idinput` command as follows: `[r] = idinput(1024, 'sine', [-x x],[-M M],[128, 128, 1])`, where M and x are the saturation value and cut-off frequency values respectively.

Question 2.2

Provided with $\breve{G}_N(e^{i\omega}) = G_0(e^{i\omega}) + \frac{V_N(\omega)}{U_N(\omega)} + \frac{R_N(\omega)}{U_N(\omega)}$, it is observed that the only errors possible in ETFE are due to the usage of a non-periodic input signal (leads to $R_N(\omega) \neq 0$), lack of excitation ($U_N(\omega) = 0$ at certain frequencies) or relatively large $V_N(\omega)$.

The high-frequency ($\omega > 0.7\pi$) behaviour can be disregarded as invalid estimation due to no excitation at these frequencies ($U_N(\omega) = 0 \ \forall \omega > 0.7\pi$). However, for the frequencies around the resonance-peak, there is lower excitation energy which implies a valid estimate. The excitation is periodic, which implies $R_N(\omega) = 0$. therefore, this leads to increased noise as a last resort to introduce an error.

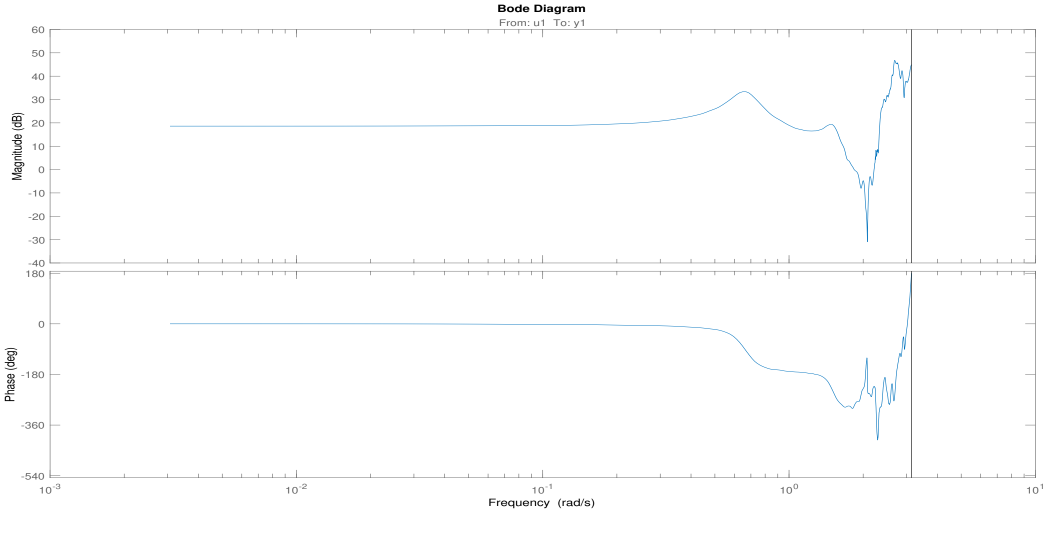


Figure 2: Bode plot of $\breve{G}_N(e^{i\omega})$

Question 2.3

Assuming the model is $y(t) = G_0(q)\,u(t) + v(t)$ with $u(t)$ and $v(t)$ are uncorrelated, we can say $\Phi_y(\omega) = |G_0(e^{i\omega})|^2\,\Phi_u(\omega) + \Phi_v(\omega) \implies \Phi_v(\omega) = \Phi_y(\omega) - |G_0(e^{i\omega})|^2\,\Phi_u(\omega)$. Since the $G_0(e^{i\omega}) \approx \breve{G}_N(e^{i\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_u(\omega)}$, thus

$$\boxed{\Phi_v(\omega) = \Phi_y(\omega) - |\breve{G}_N(e^{i\omega})|^2\,\Phi_u(\omega)}$$

where $\Phi_{yu}(\omega) = G_0(e^{i\omega})\,\Phi_u(\omega)$ and $\Phi_u(\omega) = \sum_{\tau=-\infty}^{\infty} R_u(\tau)\,e^{i\omega\tau}$. The latter expression is computed using Matlab `cpsd()` function.

In Fig. 3, the pressence of noise is noticed around $\omega \approxeq 0.7\pi$. This error grows along with the increasing variance $Var(\breve{G}_N(e^{i\omega}) = \frac{\mathbb{E}(\frac{1}{N}|V_N(\omega)|^2)}{\frac{1}{N}|U_N(\omega)|^2}$. This result implies that so-called noise is actually an error estimate.

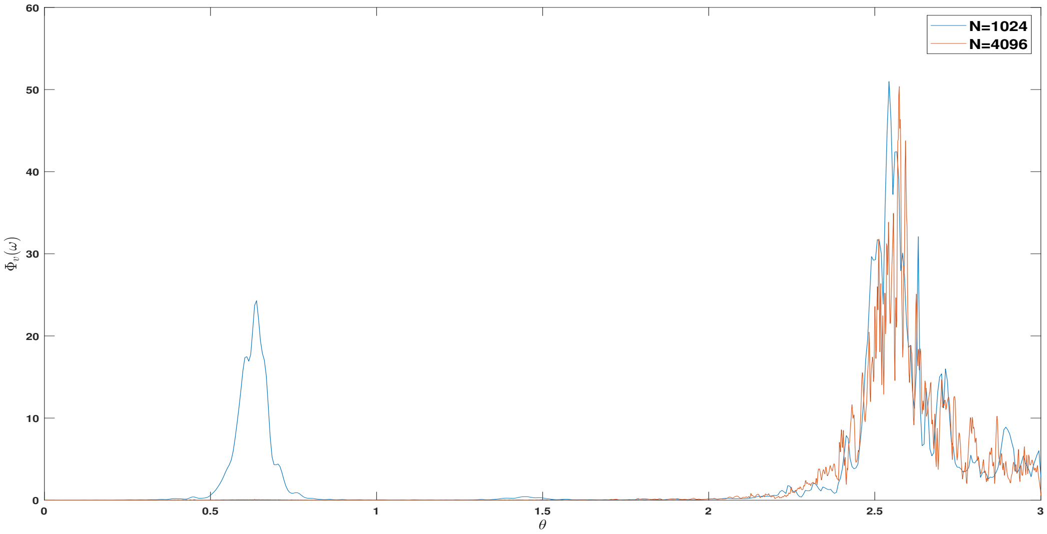


Figure 3: Power Spectral Density of $v(t)$

3 Experiment Design

Provided with insight into the system's frequency behaviour, one can start the parametric system identification process. Firstly, the experimental design is considered, where one designs the proper input for the system.

Question 3.1

To have proper input for the parametric identification process, three criteria have to be fulfilled, which are:

1. The input signal must have a significant power contribution towards a certain frequency spectrum.
2. The input signal provides high power to achieve a high signal-to-noise ratio (SNR).
3. The input signal's power is small enough to ensure the linear process properties.

The first item would be satisfied by both the PRBS and Gaussian PDF because both have the white noise process characteristics and deliver equitable power contributions. However, since the white noise of the Gaussian process tends to have a zero-mean outcome, it is very likely for the Gaussian PDF's signal power to be less than the actual needed signal's power to identify the system. On the other hand, the PRBS signal always utilises the maximum signal power due to the saturation value of its formulation.

Therefore, the PRBS input signal is preferable, as the Gaussian PDF has a higher probability of the magnitude exceeding the saturation level (regardless of amplitude scaling). However, no guarantee to ensure the linear process property. However, the PRBS is always constrained by the saturation level.

Question 3.2

The PRBS signal was designed using the Matlab code: `r = idinput(N,'prbs',[0 1/Nc],[-M M])` with given values of N and M, and N_c is the maximum clock period. The objective of the input signal design $r(t)$ is to make sure the selected clock period will make $r(t)$ excite persistently with the lowest power loss from its transition to $u(t)$, i.e. $\mathcal{P}_u \simeq \mathcal{P}_r$. Theoretically speaking, $\forall T \to \infty : \mathcal{P}_u = \mathcal{P}_r$ holds.

However, this is useable as the excitation power is focused on the lower frequencies. In assignment 2, the SNR is used as a way to measure the variance of the ETFE. It is inconsistent due to the area of frequencies that are excited by it. Stepping up the clock period also does not help because the filter $F(z)$ will not attenuate a constant signal of 1 clock signal for 3000 samples. Thus it drops down the order of persistence of excitation, i.e. no consistency for the parametric identification method. In this situation we have to make a trade-off between the order of persistence of excitation versus the losses of $\mathcal{P}_u \to \mathcal{P}_r$.

A way to solve this is to take a N_c value such that $\frac{2\pi}{N_c} \approxeq \omega = 0.7\pi$. This ensures the chosen clock period will be spread wide enough along the frequency band ω. Thus, we end up opting for $N_c = 3$.

4 Parametric identification and validation

To estimate a black-box system consistently under the assumption of $\mathbf{G_0} \in \mathcal{G}$, it must yield that:

i The input data persist enough excitation (i.e. $n_a + n_b \leq n$)

ii The G_0 and H_0 are parametrised independently.

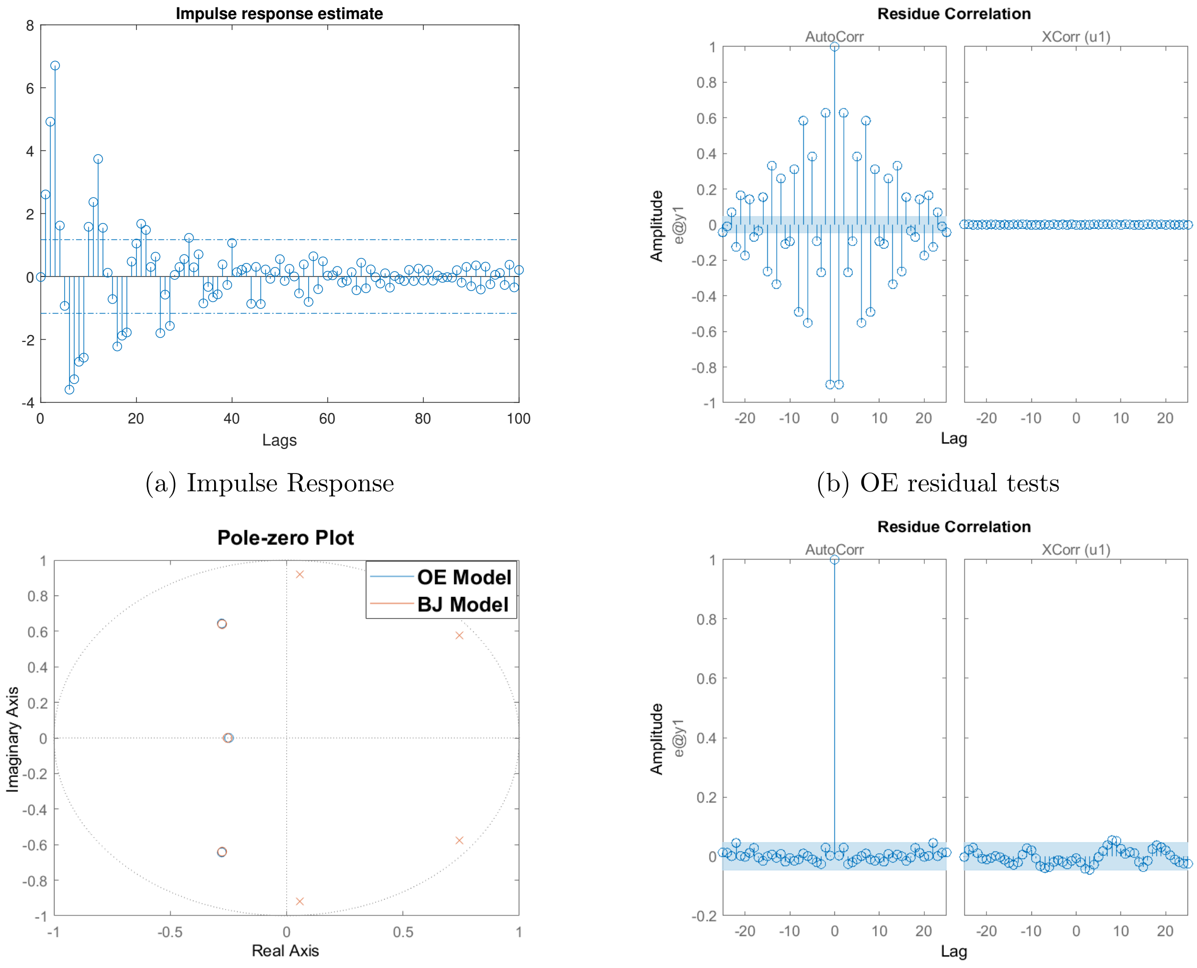
Regarding this, the designed input (in `idinput` format) was checked with `pexcit()` command and the order of persistence of excitation is 50. Evaluating from Fig. 2, we find that it is sufficient to identify the system. For the second prerequisite, we are opting to fit the system into OE (Output Error) and BJ (Box Jenkins). Meanwhile, the other choices, like ARX and ARMAX are not considered due to the coupled denominator. For the FIR (Finite Impulse Response) model structure, it is not considered as the phase characteristics from Fig. 2 is non-linear, which would result in IIR (Infinite Impulse Response). Nevertheless, the FIR model structure can be used as an intermediate step to find the n_k.

The first step we need to undertake is to estimate the n_k by using the FIR model structure. Next, we estimate the n_a and n_b using OE (OE $\subseteq$ BJ). If no satisfactory fit can be achieved, we will try to estimate the system using the BJ model structure.

Using the property of cross-correlation $R_{yu}(\tau) = g_0(\tau) \cdot \sigma_u^2$ with the assumption of uncorrelated $u(t)$ and $e(t)$, we can obtain the $R_{yu}(0)$. In Matlab, we are using the `[ir,~,~] = cra(data,100, [],1)` command. From Fig. 4a, we find the $R_{yu}(0) \approxeq -0.0166$ for our case, which implies that the $n_k = 1$, even though the $R_{yu}(0)$ value is really small. In addition it is noticeable that after $\tau = 40$, the dynamics converge into the certainty bounds.

With the delay parameter n_k known, the identification process with the OE model structure is tried first. A guess is used as an estimate such that n_b and n_f can be traced back from the ETFE Bode plot (Fig. 2). The 2 resonance peaks and the final -2 slope signal that the system should have at least 2 zeros and 4 poles. This observation will shrink our estimation process into $2 \leq n_b \leq 8$ and $2 \leq n_f \leq 10$. We are trying to sweep the numbers. Note that the upper bound is chosen arbitrarily. The setup used has an input delay of $\tau = 40$ based on the FIR impulse response. In Fig. 4b, the residual test shows that $G(\hat{\theta}_N)$ is validated for $n_b = 4$ and $n_f = 4$, but the $H(\hat{\theta}_N)$ is still invalid (i.e. the autocorrelation test still shows some dynamics). Thus, the next step is to use the BJ model structure to attempt to validate the $H(\hat{\theta}_N)$ by sweeping the n_c and n_d from 2 up to 4, as maximum zeroes and poles from $G(\hat{\theta}_N)$. The n_k, n_b and n_f will remain unchanged. The sweeping process ended up with $n_c = 3$ and $n_d = 4$.

To strengthen the motivation for choosing the model estimate, we need to take a look at the



(a) Impulse Response

(b) OE residual tests

(c) Pole-zero plot

(d) BJ residual tests

Figure 4: Parametric identification plots, from FIR to BJ

minimum variance estimator, which is formulated by a set of equations:

$$\frac{P_\theta}{N} = \frac{\hat{\sigma}_e^2}{N}\left[\frac{1}{N}\sum_{t=1}^{N}\psi(t,\hat{\theta}_N)\cdot\psi(t,\hat{\theta}_N)^T\right]^{-1} \tag{3}$$

$$\psi(t,\hat{\theta}_N) := \left.\frac{\partial\hat{y}(t|t-1;\theta)}{\partial\theta}\right|_{\theta=\hat{\theta}_N} = -\left.\frac{\partial\epsilon(t,\theta)}{\partial\theta}\right|_{\theta=\hat{\theta}_N} \tag{4}$$

$$\epsilon(q,\theta) = H^{-1}(q,\theta)\cdot(y(t)-G(q,\theta)\cdot u(t)) = \cdot\left(\frac{G_0(q)-G(q,\theta)}{H(q,\theta)}\right)\cdot u(t) + \left(\frac{H_0(q)}{H(q,\theta)}\right)\cdot e(t) \tag{5}$$

where P_θ represents the covariance matrix, $y(t|t-1;\theta)$ = one-step ahead predictor, $G_0(q)$ and $H_0(q)$ the true plant and noise model respectively, $G(q,\theta)$ and $H(q,\theta)$ are the estimates of $G_0(q)$ and $H_0(q)$ respectively.

The formulated equations above can be summarised as follows:

$$\forall\, H_0(q), H(q,\theta) : \left[H_0(q) = H(q,\theta) \implies \arg\min_\theta \epsilon^2(t,\theta) = \min\right] \tag{6}$$

$$\forall\, N \to \infty : \left[\left[\frac{P_\theta}{N} \to \infty \implies \arg\min_\theta P_\theta = \min\right] \implies \text{convergence}\right] \tag{7}$$

$$\therefore H(q,\theta) \text{ should be modelled properly.} \tag{8}$$

To verify the estimated model, we ran several tests. The first is the pole-zero plot (Fig. 4c) to ensure there are no pole-zero cancellations. None were found. the second test is the residual test (Fig.

4d). The residual test result shows that the residual $R_{yu}(\tau)$ is completely random and the $R_y(\tau)$ is effectively a δ-pulse. This implies there are no residual dynamics present in the residual.

Besides those two tests, we also used the AIC test, with nAIC=-1.64483 and nAIC=-1.54427 (validation using 1000 data points). The optimal BJ-structure according to the AIC test was $n_b = 4$,$n_c = 3$,$n_d = 4$,$n_f = 4$, and $n_k = 1$, which is the same as the manual-tuning of the model. For now, $\boxed{\mathcal{M} = \text{BJ}(4,\ 3,\ 4,\ 4,\ 1)}$

5 Experimental verification of variance estimates

To maximise the modelled dynamics' performance, it is necessary to optimize the smallest value of variance. In this section, an experimental verification is used, but it is one of many ways to minimise the variance introduced into the model.

Question 5.1

Matlab code is designed for this question.

```
load('est_model_and_dataPRBS.mat') % Loading data_prbs
nb = 4;nf = 4;nk = 1; % OE structure derived from question 4
oe441 = oe(data_prbs, [nb nf nk]);
B = oe441.B;F=oe441.F;      %Theoretical values
% Monte Carlo without init
N5 = 1000; N_c = 3; M = 2;
Range5 = [-M, M]; Band5 = [0 1/N_c];
nb = 4; nf = 4; nk = 1; % From Q4

for i = 1:100
    r5(:,i) = idinput(N5,'rbs',Band5,Range5);
    [u5(:,i), y5(:,i)] = assignment_sys_12(r5(:,i));
    OE_modelout = oe(iddata(y5(:,i),u5(:,i)), [nb nf nk]);
    OE.B(i,:) = OE_modelout.B; OE.F(i,:) = OE_modelout.F;   % Monte Carlo
end
T_cov = diag(getcov(OE_modelout))'; % Getting the theoritical variance
OE.meanB = mean(OE.B); OE.meanF = mean(OE.F);
OE.varB = var(OE.B); OE.varF = var(OE.F);



% monte carlo with initialisation
OE.medianB = median(OE.B); OE.medianF = median(OE.F);
M_init = idpoly([],OE.medianB,[],[],OE.medianF); % Initialisation for MC with initialisation

for i = 1:100
    r5med(:,i) = idinput(N5,'rbs',Band5,Range5);
    [u5med(:,i), y5med(:,i)] = assignment_sys_12(r5med(:,i));
    M_oe = oe([y5med(:,i) u5med(:,i)],M_init);
    OEmed.B(i,:) = M_oe.B; OEmed.F(i,:) = M_oe.F;
end
T_med_cov = diag(getcov(M_oe))'; % Getting the theoritical variance
OEmed.meanB = mean(OEmed.B); OEmed.meanF = mean(OEmed.F);
OEmed.varB = var(OEmed.B); OEmed.varF = var(OEmed.F);
```

Question 5.2

The parameters obtained change per simulation run. The reasons for this is three fold. Firstly, the derivation caused by the optimization process of θ, which is given by

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta, Z^N) \tag{9}$$

where

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N} \epsilon^2(t,\theta). \tag{10}$$

In case the optimization problem is convex, i.e. if ϵ is linear in θ, the result will be the same per simulation run. However, the OE structure is non-convex. So, when performing this optimization for this structure it is essential to choose a non-convex optimization algorithm.

Secondly, each simulation run has a dataset generated using a similar but different noise realization.

	b_0	b_1	b_2	b_3	f_0	f_1	f_2	f_3
Theoretical	0.0141	0.2072	0.5423	0.0323	0.0143	0.0069	0.0113	0.0129
Monte Carlo	0.0093	2.0458	2.4121	1.4112	0.2988	1.0136	0.7857	0.1468

Table 1: Comparison of theoretical variance and Monte Carlo variance without initial conditions

Question 5.3

As can be seen in Tab. 1, there is a large mismatch between the theoretical and the Monte Carlo simulation values. As the OE structure is not a convex one, it is possible for the simulation iteration to get stuck in a local minima. These minima have different mean and variance. As the Theoretical method avoids this issue, it results in the mismatch.

Question 5.4

Comparing the data from table 1 and table 2, the Monte Carlo with initial conditions has smaller variance differences with the theoretical values for the variance compared to the Monte Carlo variance without. This is due to the optimisation algorithm which tends to utilise the same initial condition. Consequently, the results shown, from the Monte Carlo simulation with initialisation, are more likely to reach the similar minimum (whether it is local or global minima). Despite the theoretical reasoning the corresponding mean and variance must be equal. However, the values will not be equal due to the different noise realisations (which holds the white process characteristics).

One of the solution to this problem would be modelling the $H_0(q)$ independently, i.e. $G_0(q)$ and $H_0(q)$ must be individually parameterised.

	b_0	b_1	b_2	b_3	f_0	f_1	f_2	f_3
Theoretical	0.0141	0.2072	0.5423	0.0323	0.0143	0.0069	0.0113	0.0129
Monte Carlo ($\times 10^{-5}$)	6.70	34.29	62.64	9.05	0.1870	0.4168	0.3695	0.1280

Table 2: Comparison of theoretical variance and Monte Carlo variance with initial conditions

6 Estimation of Box Jenkins model for minimum variance

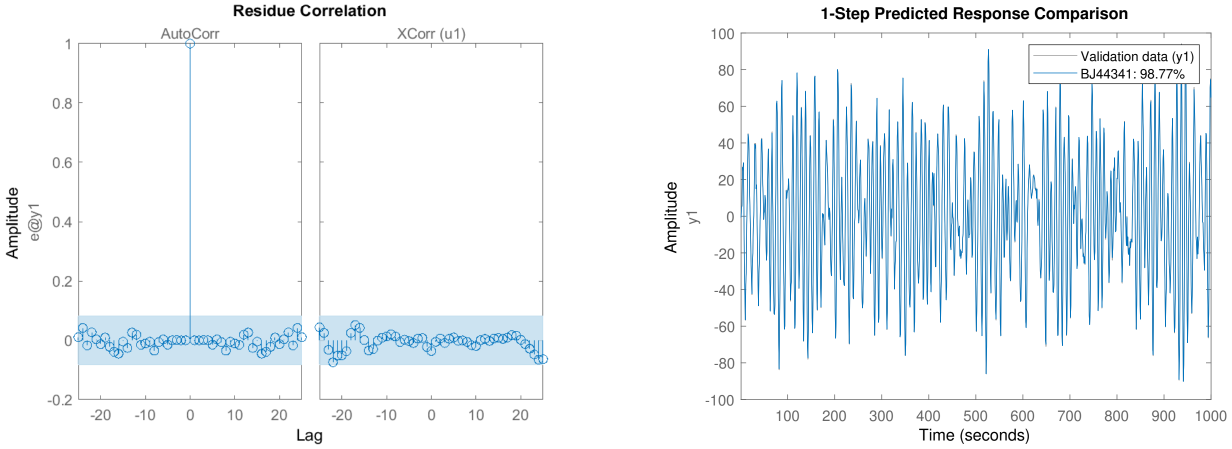
The objective of system identification method is to estimate a system with smallest possible variance and no bias. Earlier in part 5, we have tried to minimise the variance with experimental Monte Carlo simulations and found a missing link between smallest possible variance and consistency, which is the conditions of $\mathcal{S} \in \mathcal{M}$. In this part, we are trying to model a system with smallest variance possible using BJ model structure.

Question 6.1

Previously, we have modelled a system with $\boxed{\mathcal{M}_{BJ}(n_b = 4, n_c = 3, n_d = 4, n_f = 4, n_k = 1)}$. However, the variance of the model is not minimal (due to non-convex optimisation), even though the consistency properties still hold. This means the parameters obtained from part 4 are the solutions for the local minima. To achieve the minimum variance, we have to solve the system identification process for the global minima solution, i.e. taking the experimental verification of variance estimates by simulating Monte Carlo simulation 100 times, utilising the median value as an initial condition for identification setup.

It is difficult to say if a global minimum can always be reached. Partly because of the non-convex optimisation problem (which the optimisation algorithm might not handle) And partly because the noise realisation varies over time (due to not being parameterised independently and the white-noise process). Reaching a global minima solution would also mean we should adjust the optimisation algorithm, which does not guarantee that there is a global minimum to reach. The best possible solution would be as close as possible to the global minimum.

Thus, we set our BJ model structure again for the identification setup and feed the optimisation strategy (as in part 5) and validate the identified model with a residual and one-step-predictor test (in Fig.5a and 5b).



(a) Residual Test

(b) 1-step-ahead predictor test

Figure 5

It is noticeable in Fig. 5a that the result is better than the BJ model structure performed without the initialisation. Moreover, the confidence level for the model is 98.76% with mean-squared error of 0.1974. The end result of parameter is as follows:

$$B(z) = (2.299 \pm 0.003478)z^{-1} + (1.857 \pm 0.008738)z^{-2} + (1.439 \pm 0.01205)z^{-3} + (0.284 \pm 0.0045)z^{-4}$$

$$C(z) = 1 - (0.6486 \pm 0.1966)z^{-1} + (0.7643 \pm 0.2181)z^{-2} - (0.4432 \pm 0.102)z^{-3}$$

$$D(z) = 1 + (1.535 \pm 0.2037)z^{-1} + (1.384 \pm 0.3886)z^{-2} + (1.023 \pm 0.3741)z^{-3} + (0.5494 \pm 0.1743)z^{-4}$$

$$F(z) = 1 - (1.602 \pm 0.0009176)z^{-1} + (1.902 \pm 0.001559)z^{-2} - (1.36 \pm 0.001514)z^{-3} + (0.7476 \pm 0.000815)z^{-4}$$

Question 6.2

Comparing all values of the variances from part 4 up to part 6, the $G(q, \theta)$: It is noticeable that the variance is significantly lower for the OE model due to the $G(q, \theta)$ and $H(q, \theta)$ are not parametrised independently.

	b_0	b_1	b_2	b_3	f_0	f_1	f_2	f_3
Output Error*	0.0141	0.2072	0.5423	0.0323	0.0143	0.0069	0.0113	0.0129
Monte Carlo ($\times 10^{-5}$)	6.70	34.29	62.64	9.05	0.1870	0.4168	0.3695	0.1280
Monte Carlo[1]	0.0093	2.0458	2.4121	1.4112	0.2988	1.0136	0.7857	0.1468
Box Jenkins	0.0000	0.0001	0.0001	0.0000	0.0386	0.0476	0.0104	0.0415

Table 3: Summary of OE, Monte Carlo, and tuned BJ variances