Final Assignment

Jiaxuan Zhang (1551957)
Yiting Li (1567713)

April 22, 2022

Course:	5SMB0 System Identification
Professor:	Prof.dr.ir. P.M.J. Van den Hof
Date:	April 22, 2022

1 Part 1 | Understanding saturation and Butterworth filter

1.1 Butterworth Filter

According to the given expression of the Low-pass filter, we can plot its bode diagram, by command `tf`. We set the frequency scale to linear scale, the bode diagram is shown in the figure 1.

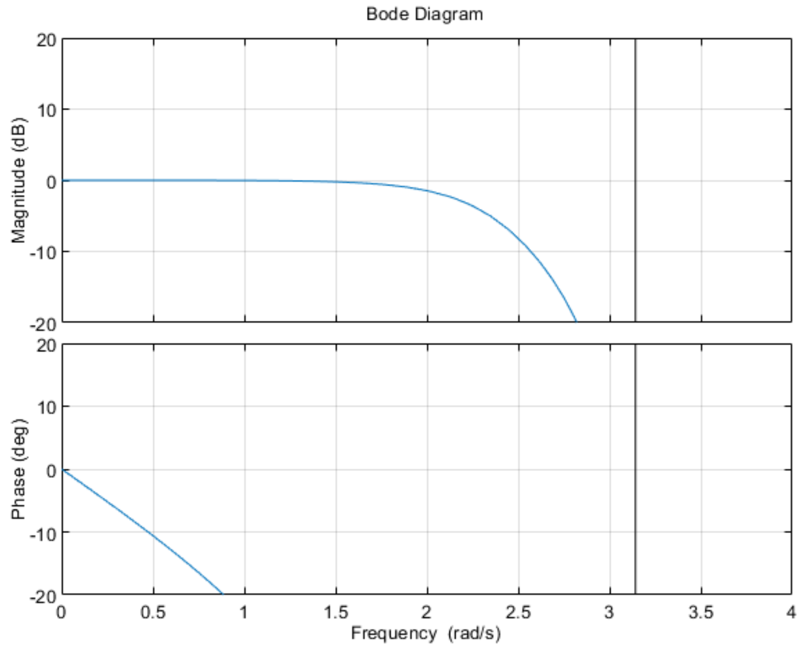


Figure 1: Bode Diagram of Butterworth Filter

From the obtained bode diagram, we conclude that the cut-off frequency(-3 dB point) is around 2.1954 rad/s.

1.2 Saturation Module

We generate a sin-wave signal with different amplitudes to check the gain of the saturation module, and we found that in our case, the M is 3.

From figure 2, we can see a flat line in 3. The upper bound of the saturation module is important, because if our reference signal has a large offset, the generated input will be a flat line and cause an unsatisfied input signal.

2 Part 2 | Nonparametric identification

2.1 Input of ETFE Identification

In order to excite the system at 128 frequencies, the input signal should be excited at 128 frequencies point. we use the reference signal as follows. The $\omega_0 = 2/128$, where 2 is the bandwidth of the filter $F_0(q)$.

$$r(t) = \sum_{k=1}^{128} \sin(k\omega_0 t) \tag{1}$$

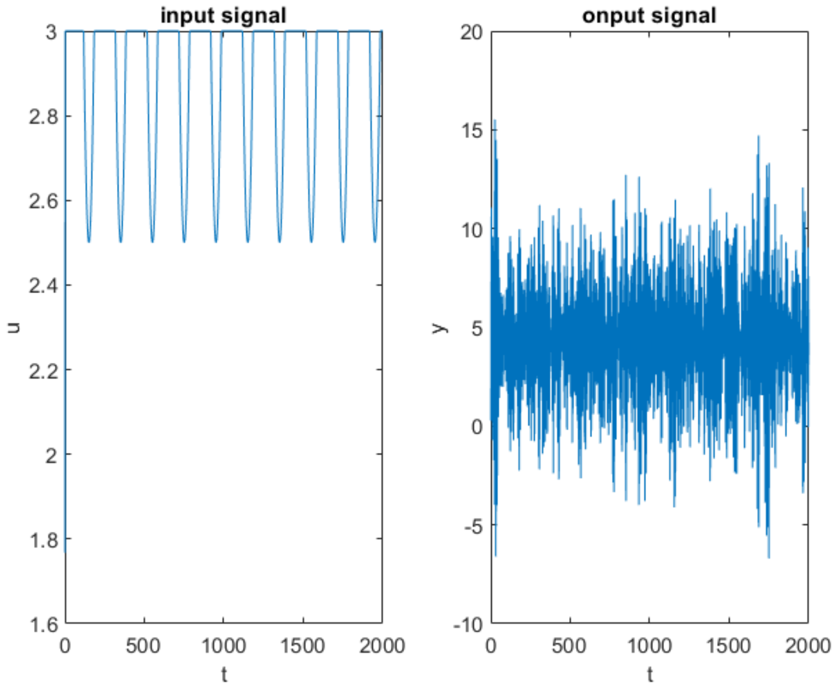


Figure 2: Input Signal after The Saturation Module

2.2 Identified ETFE Model

By using the `etfe` function in MATLAB, we got an estimated ETFE model. The Bode plot of the identified ETFE model is shown in Figure 3. It can be seen that the signal has a resonance peak in the passband of the Butterworth filter, at a frequency around $0.565 rad/s$ In the high-frequency area, the $G_0(q)$ has a related high magnitude and a large phase-shifting.

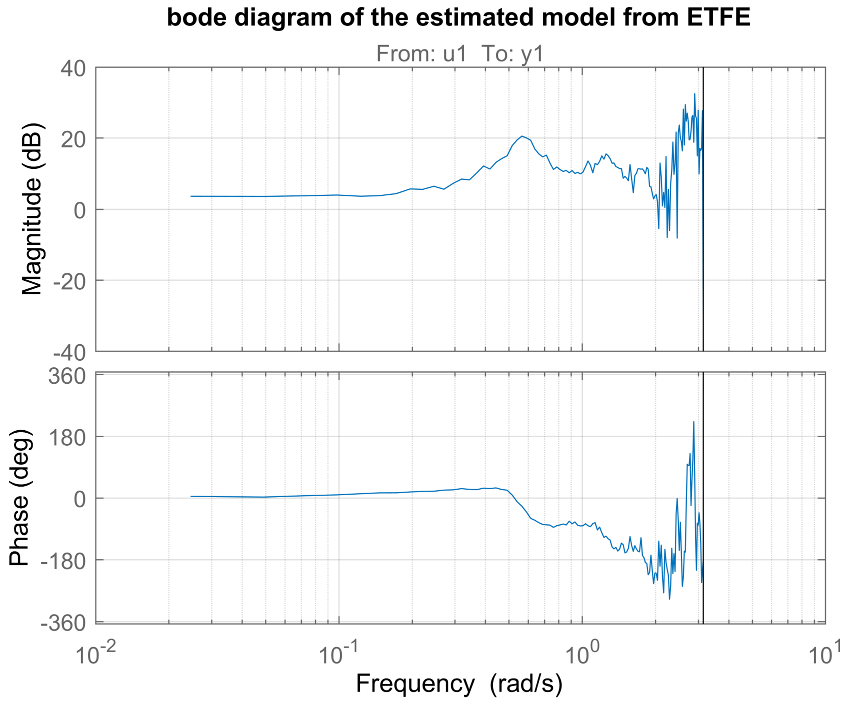


Figure 3: Identified ETFE Model

2.3 Estimated Noise Spectrum

The noise signal can be estimated if we inject reference signal $r = 0$, which will cause $u = 0$. That is, the system output y is totally excited by the noise signal v. After getting the output y, we can directly use `cpsd` function to generate the noise spectrum. In the `cpsd` function, we use 1024 as a segment number to invalidate the Welch's averaging method. The Bode diagram of the magnitude of the estimated noise spectrum is shown in Figure 4.

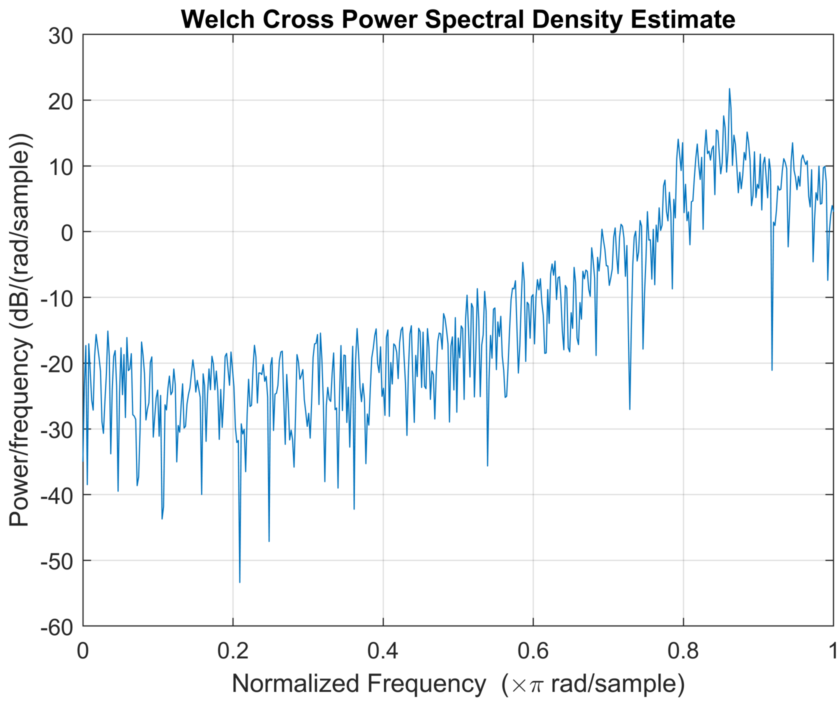


Figure 4: Noise Spectrum

3 Part 3 | Experiment design

3.1 Choice of Reference Signal

We prefer to use PRBS. The main reason is by using PRBS, the reference signal r will be binary, and we can easily choose an amplitude of r that make the input u has maximum power after the saturation. Another possible reason is, by using PRBS, we can easily adjust the signal's bandwidth so that we can actually get what we want after filtering, while by using Gaussian pdf, we cannot easily predict the input signal u after filtering and saturation.

3.2 Design the Reference Signal

The power of a discrete-time signal u can be calculated by $\sum u^2(t)$. In order to guarantee the maximum power requirement, we should use a reference signal r that always makes input signal $u(t) = M = 3$. We prefer to choose the PRBS signal $r(t)$ based on the following algorithm and set $|r(0)| = M = 3$. Because the magnitude of r keeps a constant, and by changing p we can easily adjust the power spectral density of r more easily compared to the $u(t) = c \cdot \text{sign}[w(t)]$ solution.

$$r(t) = \begin{cases} r(t-1) & \text{with probability } p \\ -r(t-1) & \text{with probability } 1-p \end{cases} \tag{2}$$

In order to guarantee no substantial power loss requirement, most of the power of reference signal r should locate within the bandwidth of the low pass filter $F(q)$. After several groups of tests, we choose $p = 0.83$. The power spectral density of r is shown in Figure 5. It can be seen that, after about frequency point 0.7π, most times the power is lower than $0db$, which means after filtering, there will be no substantial power loss.

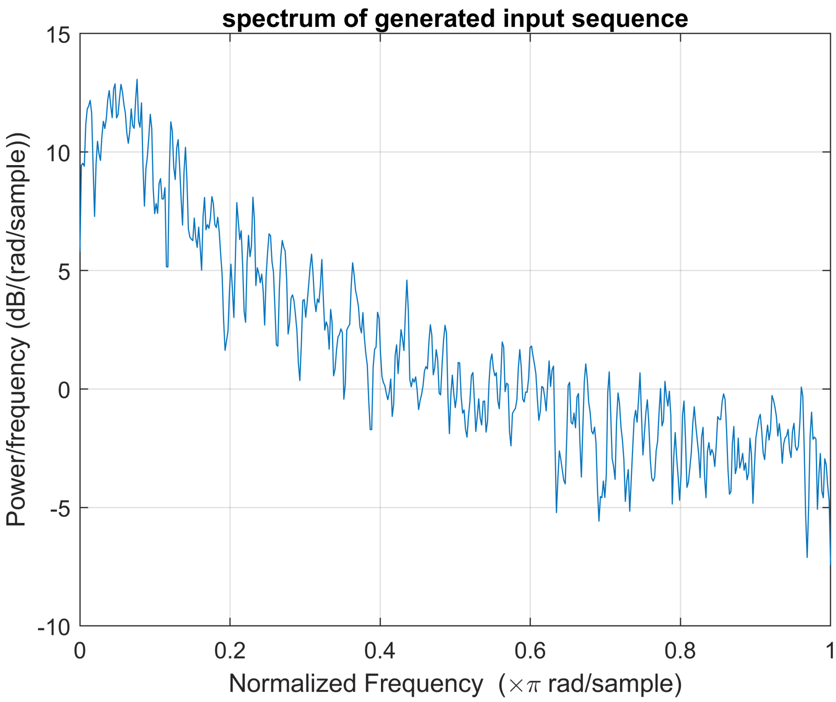


Figure 5: Spectrum of Designed PRBS

4 Part 4 | Parametric identification and validation

4.1 Experiment with OE Model

For the OE model, there is no existing function in MATLAB that can help us estimate and select the order automatically. So we first use cross-validation to choose a possible order configuration. Then we use the residual test to find whether the chosen order configuration meets $G_0 \in \}$.

For cross-validation, we first divide the data into training data set and validation data set with a ratio of 7:3. Then we tried to learn an OE model based on the training data and then test with the simulation method on the validation set (actually, for the OE model, simulation is the same as prediction). We tested order summation up to 20 and the simulation errors are shown in Figure 6 (some very large error configuration is not shown in the Figure). Although a smaller simulation error does not always mean the perfect fitting of the model, we still determined to choose the order configuration with the smallest simulation error for the first step of order selection. The best configuration order is $n_b = 4, n_f = 14, n_k = 1$. We also test some other configurations with a smaller order, and they do not show a good result in the residual test. (Please note that, because u, y are not fixed in each running, so the optimal configuration may not be the same in each run).

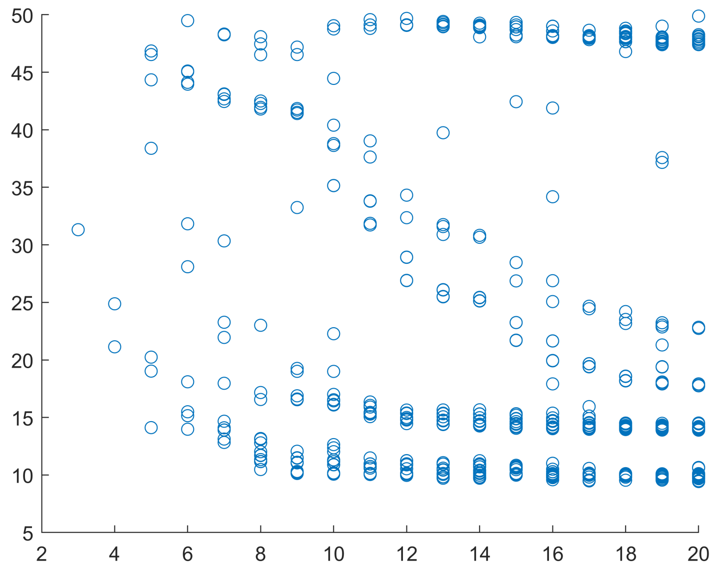


Figure 6: OE Model Order Selection Error Figure

The residual test result of the OE model with selected order configuration is shown in Figure 7. It can be seen that, estimated G is validated (although if we zoom in there are some points very close to the confidence interval boundary), while estimated H is not good. However, we still get a nice order configuration.

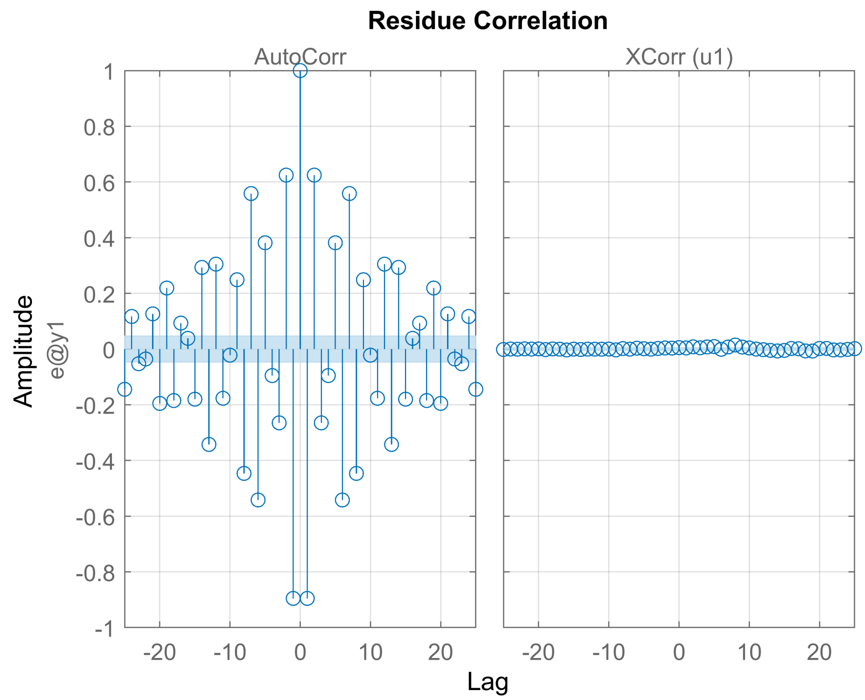


Figure 7: OE Model Order Selection Residual Test

4.2 Experiment with ARX Model

For the ARX Model, we can use the GUI, since the Toolbox offers an order selection tool.

Firstly, we used the r to generate u and y, then import these two time-domain data into the Toolbox, then we used the polynomial model estimation with ARX model(order selection), shown in the figure 8. The searching range is set as $n_a = [1:15]$, $n_b = [1:15]$, $n_k = [1:5]$.

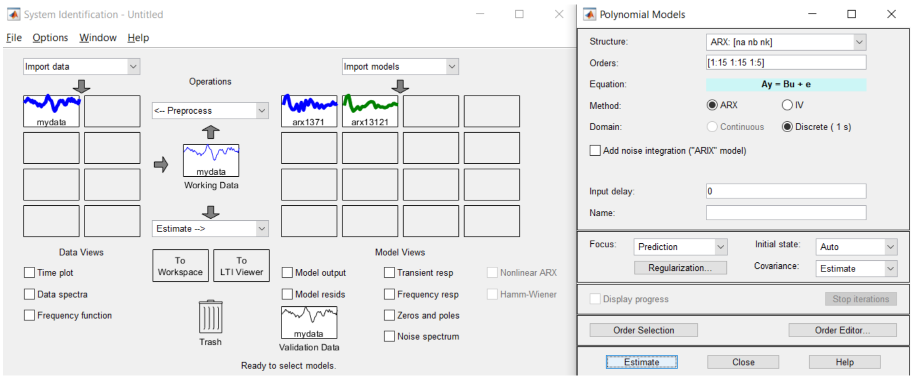


Figure 8: The Interface of The GUI

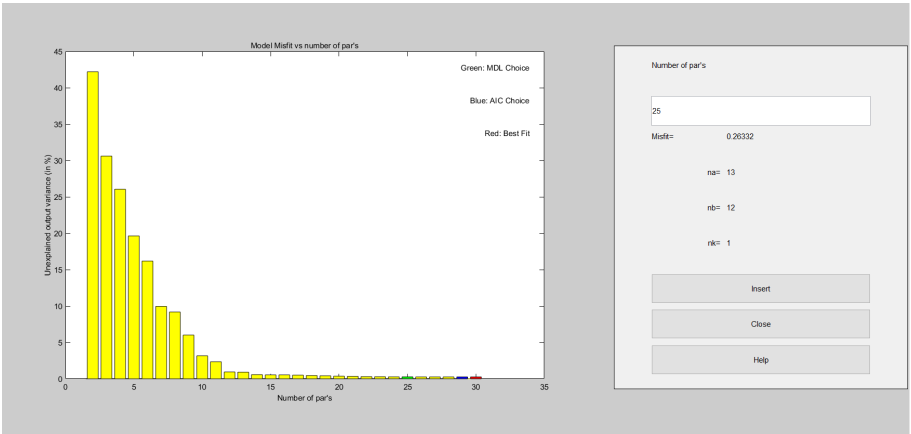
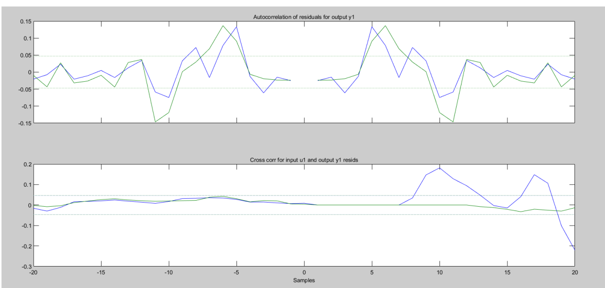
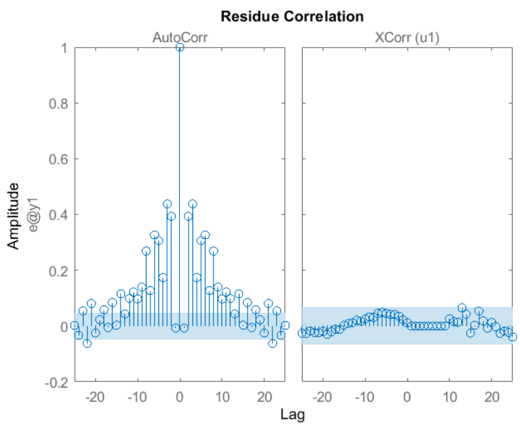


Figure 9: Order Selection Result

From the order selection result, figure 9, we chose some examples that have a lower variance to do the residual test. We found that the order of ARX model should be almost equal to each other.

Then we tried to reduce the order and manually tested other parameters, we reached the conclusion that $n_a = 10, n_b = 1, n_k = 1$ is the parameter set that can pass the cross correlation test.


(a) Residual Test of Model(blue: 13, 7, 1), (green: 13, 12, 1)



(b) Residual Test of Model(10, 10, 1)

Figure 10: Residual Test

4.3 Order & Model Structure Selection

We chose our order as $n_b = 4, n_f = 14, n_k = 1$, same as the OE model, for two reasons.

Firstly, we expect the summation of model order should as small as possible to avoid possible overfit in G.

Secondly, in the ARX model, the H and G are not modeled independently.

5 Part 5 | Experimental verification of variance estimates

5.1 Monte Carlo Simulations

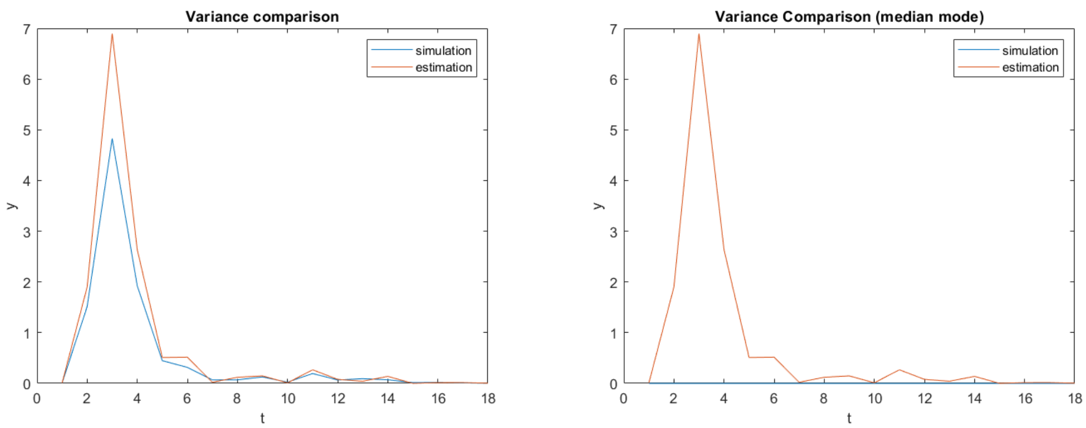
We used the same method to generate PRBS reference signal and used OE Model parameter($n_b = 4, n_f = 14, n_k = 1$) obtained from previous section.

In each Monte Carlo simulation, it can be observed that the parameter identified are not the same. There are several possible reasons:

1. In each simulation, the input noise sequence is not the same. Although theoretically with an infinite sequence, we can guarantee an asymptotical convergence result, we cannot generate or use an infinite length input-output sequence in reality.
2. Another possible reason may be because the PRBS generated is a random binary sequence. Its properties like power density spectrum are all from a statistical perspective while in each run, the property of the random binary sequences is not strictly the same.
3. Besides, in `oe` function in MATLAB, it uses some search method to find the optimal solution instead of an analytical solution. The starting point of the searching method is randomly selected, which causes varying results in each experiment.

5.2 Theoretical Variance Comparison with Different Start Point(Randomly)

We obtained the theoretical variance using the `getcov()`, and used two different search options to get the simulated variance, the comparison is shown in the left part of the figure 11.



(a) Variance Comparison Between simulation(random start point) and estimation

(b) Variance Comparison Between simulation(prescribed start point) and estimation

Figure 11: Variance Comparison

After multiple experiments(loop was set as 100), we observed from the comparison, that when we used randomly start point initialization, the simulation result has the same trend as the estimation curve. Both curves showed that there was a big variance in the B part(the first four elements on the x-axis).

Furthermore, during multiple experiments, we observed that those two curves sometimes were close to each other, and sometimes they didn't. This may be because the loop number is not big enough, if the loop number is larger, the curve dis-match might occur less frequently than the case with a smaller loop number.

5.3 Theoretical Variance Comparison with Different Start Point(Prescribed Point)

the comparison is shown in the right part of the figure 11. Since we used a prescribed point, and in each search round the prescribed point didn't change, we can only see a flat line. It may be because although the identified parameter (actually the prescribed starting point) may not be the global optimal solution, the true global error value is very close to the error based on the prescribed starting point. Then the solver just keeps at the prescribed point because it is already good enough and meets the stop criterion.

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

6.1 Estimate an BJ Model

In BJ model, for G part, we directly use the configuration we get in the OE model $n_b = 4, n_f = 14, n_k = 1$. When selecting the order of the H part, there is little clue to find a small order region directly. One clue is from the noise spectrum in section 2. From the trend of the spectrum graph (the power increase in the high-frequency region, we can conclude that in the BJ model, the $D(q)$ part should have a lower order than the $C(q)$ part. We test several groups of configuration, and when we choose $n_c = 3, n_d = 5$. The residual test result of this configuration is shown in Figure 12. The estimated BJ model is validated based on the result of the residual test.

Two additional tests can be done: comparing the identified ETFE model and the bode graph of the G part in the BJ model, comparing the noise spectrum from section 2 and the bode graph of H part in the BJ model. The result of G comparison is shown in Figure 13. The estimated H of the BJ model is shown in Figure 14.

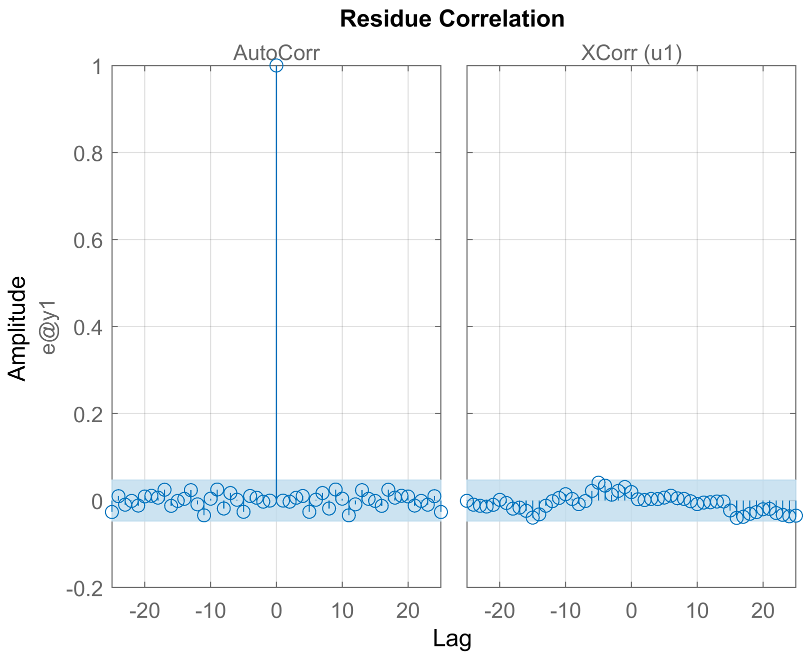


Figure 12: Residual Test of BJ Model

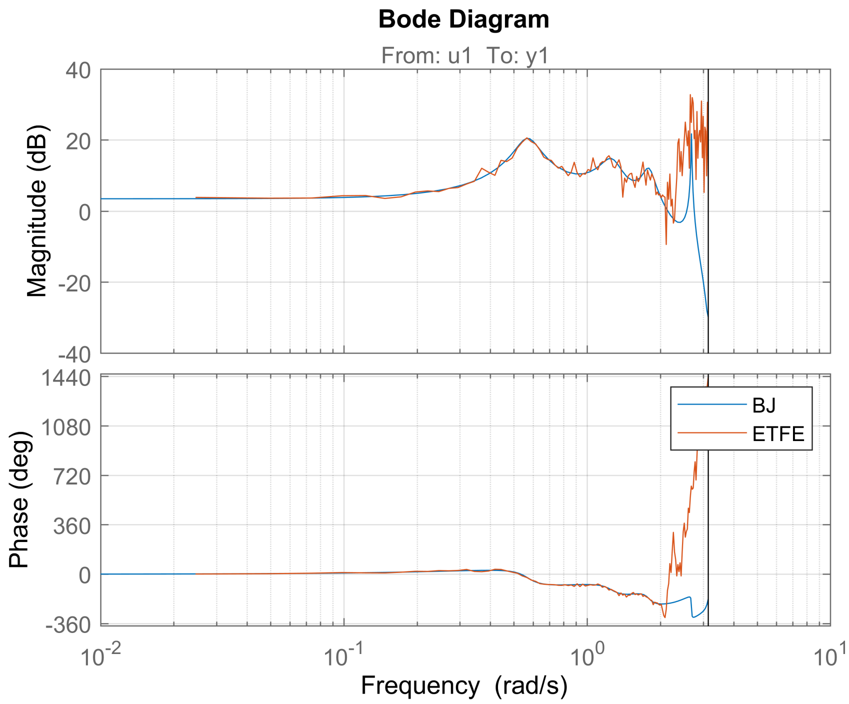


Figure 13: G BJ ETFE Comparison

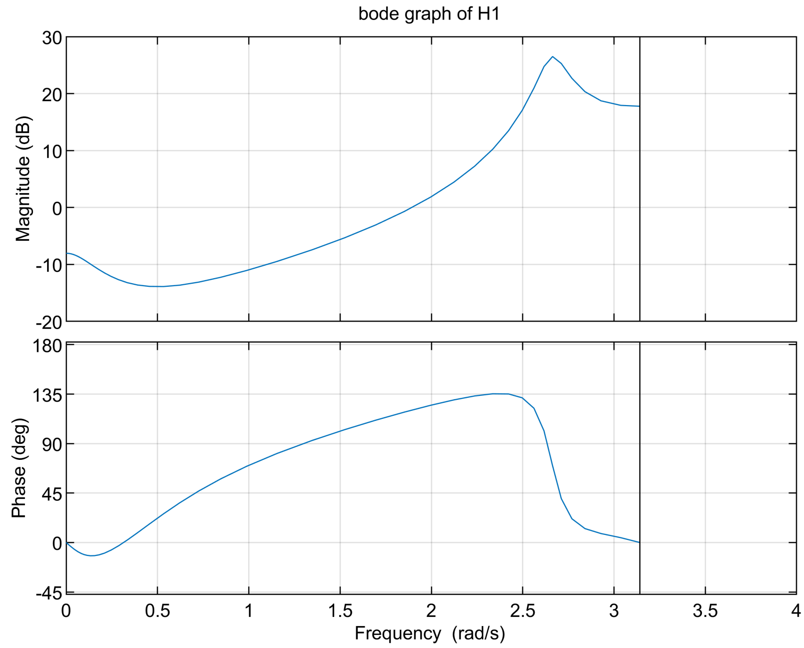


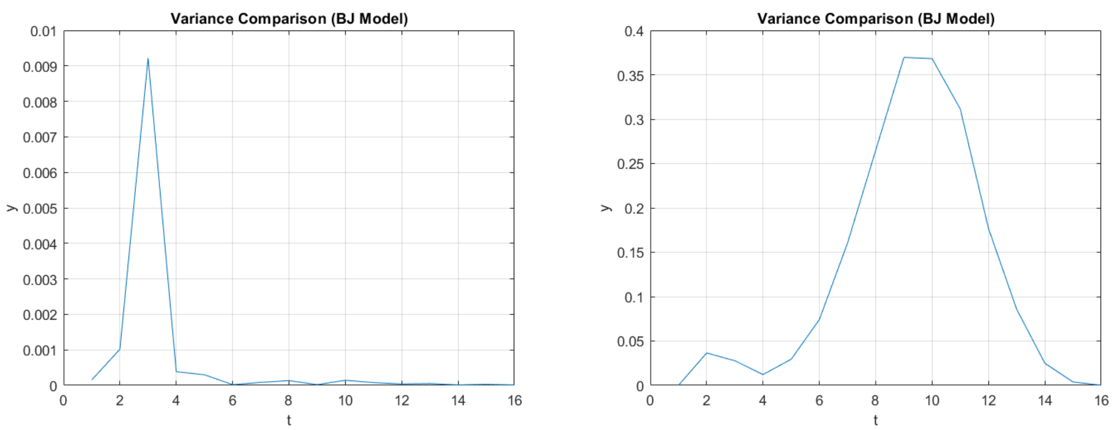
Figure 14: H of BJ model

The BJ model and the ETFE model fit very well for the frequency in the passband of the Butterworth Filter. The H and the estimated noise spectrum has a very similar trend although the magnitude value is not the same. These tests also indicate that our estimated BJ model is suitable.

6.2 Covariance Comparison

We obtained the theoretical variance using the `getcov()`, we ignore those parameters related to noise model H, the comparison is shown in the figure15. We observe there is a peak in the first three elements(belonged to B), which is similar to the comparison result of OE model.

After multiple experiments, we also observed a different pattern, while the peak was mainly located at those elements that belonged to F. The reason for that may relate to the input signal. In practice, we would prefer pattern 1 since the peak value is smaller and the other estimated variance is not large.



(a) Variance Comparison (BJ Model) pattern 1

(b) Variance Comparison (BJ Model) pattern 2

Figure 15: Variance Comparison (BJ Model)