RYERSON UNIVERSITY

Department of Electrical, Computer, and Biomedical Engineering

Course Number	ELE829
Course Title	System Models & Identification
Semester/Year	Fall 2022
Instructor	Professor M.S. Zywno, Ph.D.

Tutorial Report No.	2
Tutorial Title	Diagnostic Tools and Non-Parametric Models in Time Domain

Group No.	**10**
Section No.	**04**
Submission Date	**October 12, 2022**
Due Date	**October 13, 2022**

Name	Student ID	Signature
Hansel Kalathil	500768880	H.K
Tenzin Dhargye	500776621	T.D

(Note: remove the first 4 digits from your student ID: *xxxx12345*)

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*

www.ryerson.ca/senate/current/pol60.pdf

ELE829 Fall 2022

Tutorial # 2 Grading Sheet

Your Tutorial 2 OE Process Number: 10	
Any deductions will be recorded here - please refer to the Tutorial Instructions for specifics.	
ONE PAGE EXECUTIVE SUMMARY Anything that is important about this report should be included on this page - it is your "bottom line". If you don't know what to include, think about a busy CEO of your company who will not want to thumb through the whole report - he/she needs to know "the why, "the what" and the end result. The rest is for "the middle management" to pore over.	**/10**
Part 1 - Diagnostics in Time Domain	**/20**
Part 2 - OE Model	**/10**
Part 3: Conventional Model ("Clean" data only)	**/20**
General: Clarity, writing style, grammar, spelling, layout of the report	**/20**
Total Mark for the Collaborative Part of the Report	**/80**
Partner 1 (name): Hansel Kalathil	**Partner 2 (name): Tenzin Dhargye**
D2L Quiz mark for Tutorial 2 **/20**	**D2L Quiz mark for Tutorial 2** **/20**
TOTAL: /100	**TOTAL: /100**

Executive Summary

This tutorial will be looking at multiple diagnostic tools and comparing them to each other. Similarly to tutorial 1, the tutorial will be broken up into three parts. The first part looks at non-parametric models in the time domain that can be used as diagnostic tools for Modern System Identification of parametric models. The second, looks at the OE model which can be created through characteristics found in part 1. Finally, the third part looks at the classical approach that was used in ELE639 to find a workable transfer function of the system.

Part 1 - Time Domain Data as a Diagnostic Tool

In this tutorial we experiment with new subroutines and tests on the discrete system structure. The experiments were conducted with two different data sets of "noisy" and "clean" systems. The usage of new subroutines ***imweigh, cra, correl*** and ***hanktest*** were used to collect data and plots. The diagnosing tool of amplitude and number of samples help create a better defined plot. When comparing the different data, the "clean" system indicates the system structure better while the "noisy" system creates subpar diagrams for all the subroutines. To find the plot the ***imweigh*** produces more defined plots in the "clean" data while ***correl*** subroutine produces better "noisy" data plots. The impulse response estimate helps gather the sampling delay and the ***hanktest*** assists with the system order. The discrete system structure is then found to be $[n_b \quad n_f \quad n_k] = [2 \quad 2 \quad 13]$

Part 2 - Simple System Identification Using OE Model

Based on the diagnostics from part 1, where [n_b, n_f, n_k] were determined to identify the unknown system. Using the ***oe*** function in MATLAB generates a Discrete Time (DT) model. As expected, when comparing the "clean" data, the oe model shows a very good fit of 87.59% to the original data signal as seen in figure 6. The fit is not 100% even for the clean data because there is a delay in the system, so the first few samples will not be matched correctly . Next the DT model is converted to a continuous time (CT) model using the ***d2c*** command in MATLAB to produce a G(s) transfer function and a pole-zero map showing the system is 2nd order with a zero. The same procedure is done with the "noisy" data but the "fit" is much further off at 75.92%, as seen in figure 8, which is to be expected. When transformed to a CT model, it also shows up as a second order model with a zero.

Part 3 - Conventional Model ("Clean" data only)

In this part, using the noise-free step response graph (figure 5) created in part 1, an estimated conventional G(s) model is created using the information learned in ELE639. After the estimated model "fits" well to the graph in figure 5, as seen in figure 10, it is then compared to the OE model created in Part 2 using the ***compare*** command in MATLAB. As can be seen in figure 12 the OE model (marked as M_OE) has a very similar error to the conventional model (marked as M3_OE). The error to the original waveform is due to the system lagging in the beginning but soon after a few instances both models fit very well.

Part 1. Diagnostics in Time Domain

When deciding on which methods to perform Step/Impulse weight, an easy method to compare is to observe the plots created. When performing the system in a noise-free environment, the best method to observe the data is to use the de-convolution method as both the delays and the trends are clearly visible in figure 1 and 5. The correlation method does perform this task well but the trends and delays are not as easily recognizable. In the noisy system, the correlation and CRA plots show similar plots and the trends can be recognizable (figure 2). The de-convolution however completely breaks in the noisy system. As both show clear advantages and disadvantages in which data is being used. In the noise-free data, the deconvolution method performs the best and correlation methods achieve the best results in noisy data.

Similar to the Step-Impulse weight, in the noise-free data the deconvolution method proves to show the best plot. The Hankel test in the noise-free data using the de-conv indicates a clear drop off with no additional slopes and with easier identification of the system using system order estimate plot (figure 3). Using this method we can identify the order of our model to be 2 ($n_b = n_f = 2$). In the noisy data the correlation method can be seen having minimal changes in slope (figure 4). The system order estimate plot also indicates a second order plot with the third point being ignored due to n=3 being significant.

When comparing the different subroutines, they all perform the task better depending on the given data. In the noise free data the impweigh subroutine is shown to best plot the graph, while the correl showed the best figure in the noisy data. Taking into account all relevant data, I can conclude that correl subroutine proved to be the most reliable in the both datas. Although the impweigh shows a much clear trend in noise-free data, the subroutine completely breaks when noise is introduced. Cra subroutine is similar to the correl in our experiment but the correl plots tend to show a more defined trend. The use of different input signals produces a more defined trend,and making changes to amplitude and number of samples outline the trend better.

Making use of the Hankel test and Step/Impulse weight , we can determine the unknown model structure based on the “clean data”. In the “clean” Hankel test the system order estimate indicate the order of the model to be 2 ($n_b = n_f = 2$). With the impulse response estimate, the number of sample delay are 13($n_k = 13$) in figure 1. These three values are collected to produce the model structure $[n_b \quad n_f \quad n_k] = [2 \quad 2 \quad 13]$

When looking into the effects of the noise in the system, the subroutines show stark differences depending on the data. As the impweigh subroutine practically becomes unusable while the other two subroutines are able to produce subpar results. The representation of the data can be seen the most precisely when there is no noise present in the system as shown in figure 1 and 2.

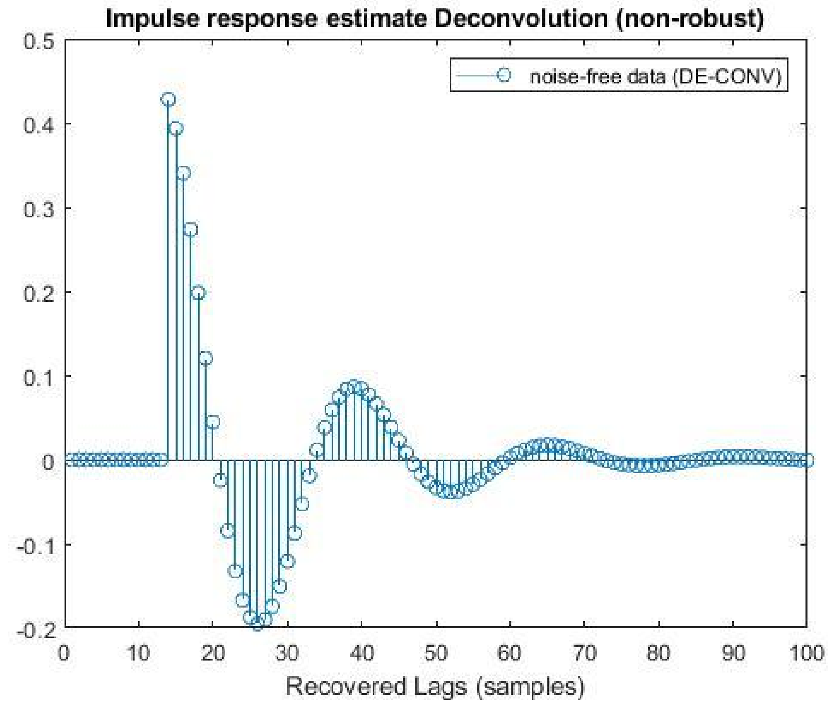

Figure 1: Deconvolution Method Impulse Response estimate (noise-free)

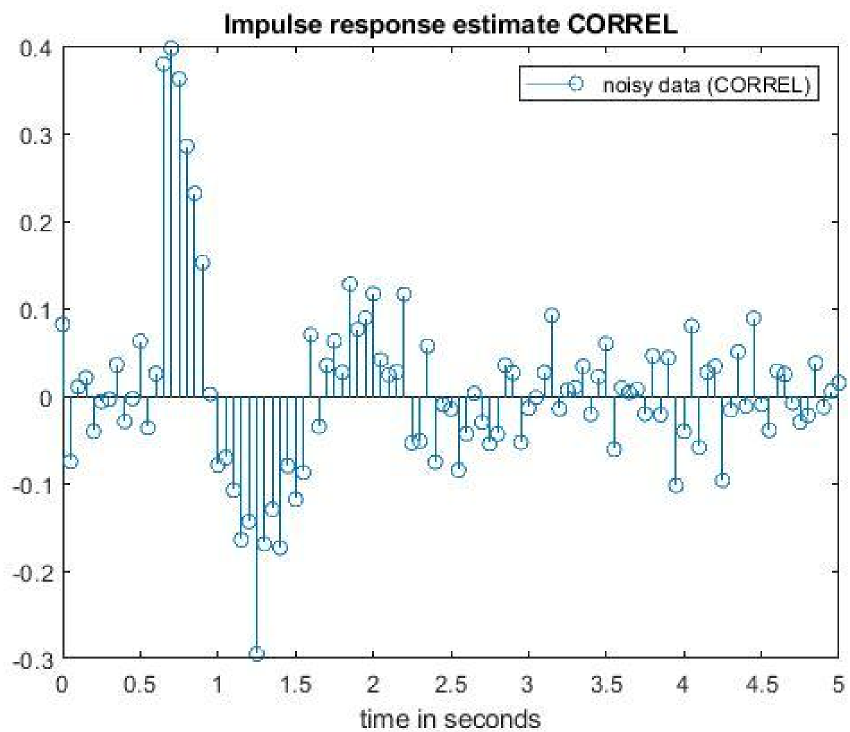

Figure 2: Correl Method Impulse response estimate (Noisy).

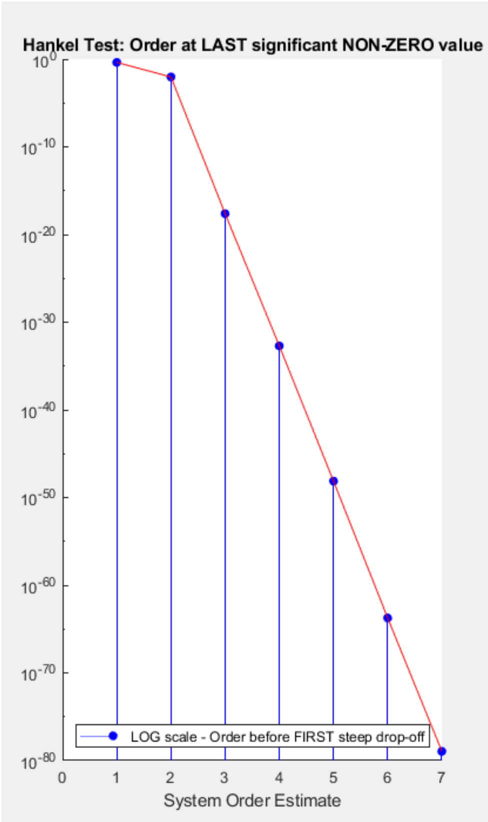

Figure 3: Deconvolution Method Hank Test (noise-free)

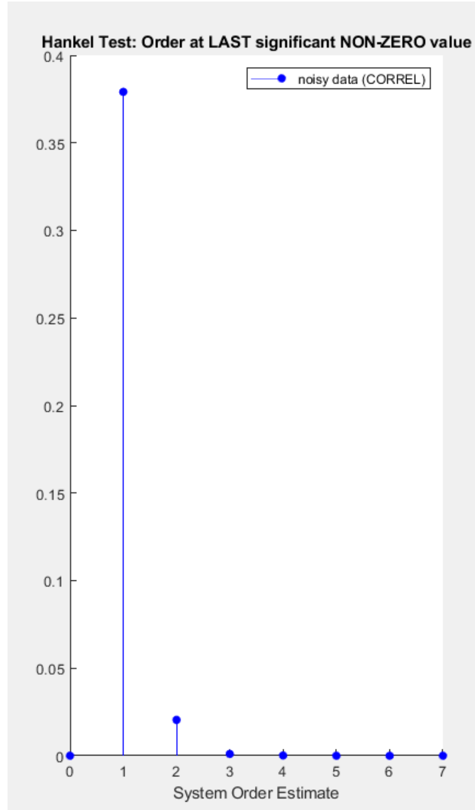

Figure 4: Correl Method Hank Test (noisy)

Recover Step Response (noise-free)

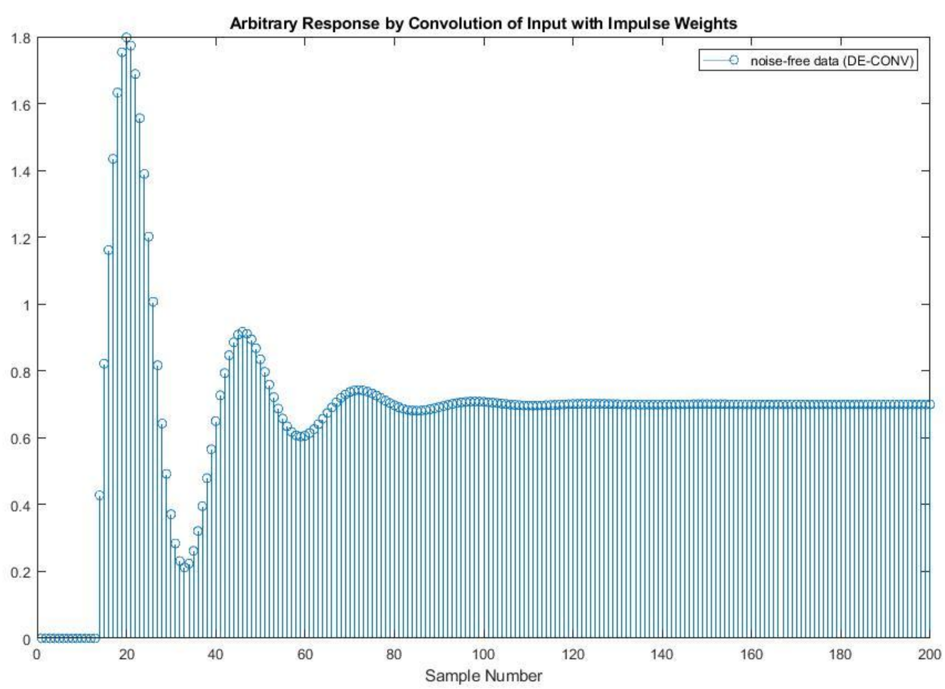

Figure 5: Deconvolution Method step response (noise-free)

Part 2 - OE Model

Compared to tutorial 1, where the noisy data file heavily affected the OE models enough to produce an extra zero. In this case the noisy file did not significantly affect the transient response as can be seen in figure 7 and figure 9 where the noise-free data and the noisy data show similar pole and zero placements.

Noise-Free Data:

G(z) = [B(z) / F(z)] u(t) + e(t)
B(z) = 0.4279 (+/- 1.411e-16) z^-13 - 0.387 (+/- 1.399e-16) z^-14
F(z) = 1 - 1.824(+/- 6.975e-17) z^-1 + 0.8825(+/- 6.895e^-17) z^-2

G(s) = [B(s) / F(s)] u(t) + e(t)
B(s) = 8.75s + 17.5
F(s) = s^2 + 2.5s + 25

Although the noise-free data should show a 100% fit, the compare function gives a 87.59% fit. This is because the delay of the system makes it impossible for the system to fit perfectly; as can be seen below in figure 6, where the curve fits perfectly after T = 2.5s but before that the delays cause the curve to misalign.

Noisy Data:

G(z) = [B(z) / F(z)] u(t) + e(t)
B(z) = 0.4187 (+/- 0.004668) z^-13 - 0.3773 (+/- 0.004716) z^-14
F(z) = 1 - 1.825 (+/- 0.001721) z^-1 + 0.884 (+/- 0.001752) z^-2

G(s) = [B(s) / F(s)] u(t) + e(t)
B(s) = 8.541s + 17.67
F(s) = s^2 + 2.467s + 25.03

The noisy data fits the curve quite well. Although there are obvious artifacts in the original waveform due to noise, the overall trend of the OE model follows the original curve well. By observing figure 8 below, it can be seen that the compare function gives a 75.92% fit.

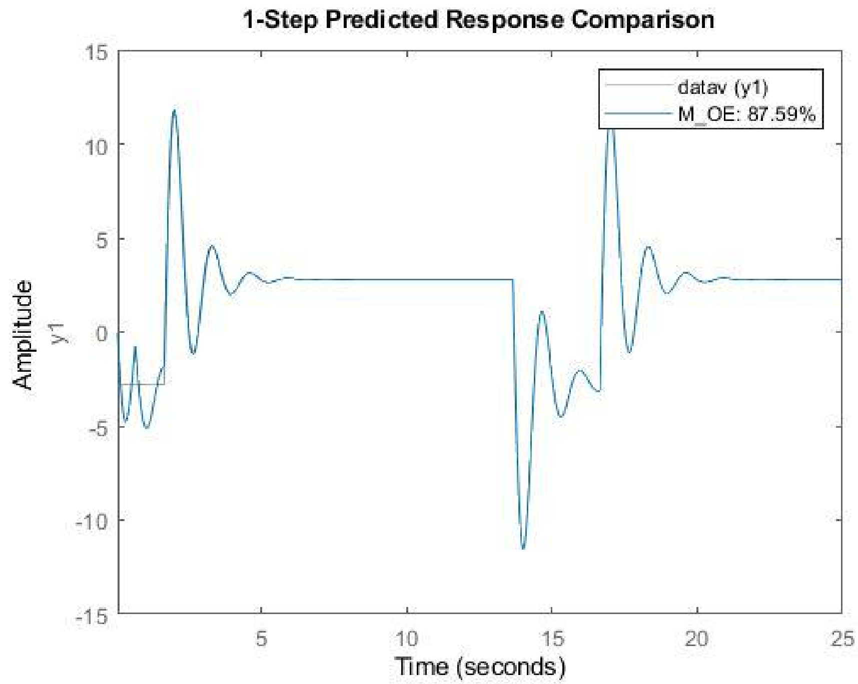

Figure 6: OE compared to original data (noise free)

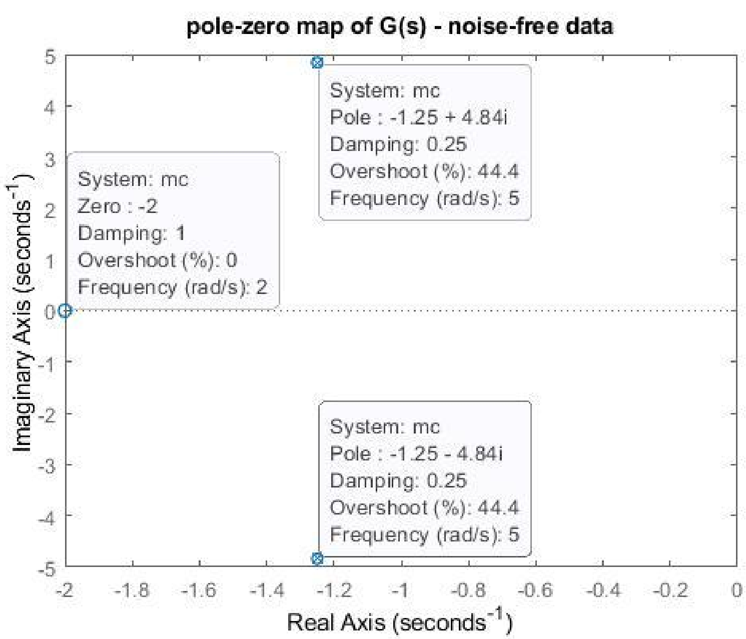

Figure 7: Pole-zero map of OE (noise-free)

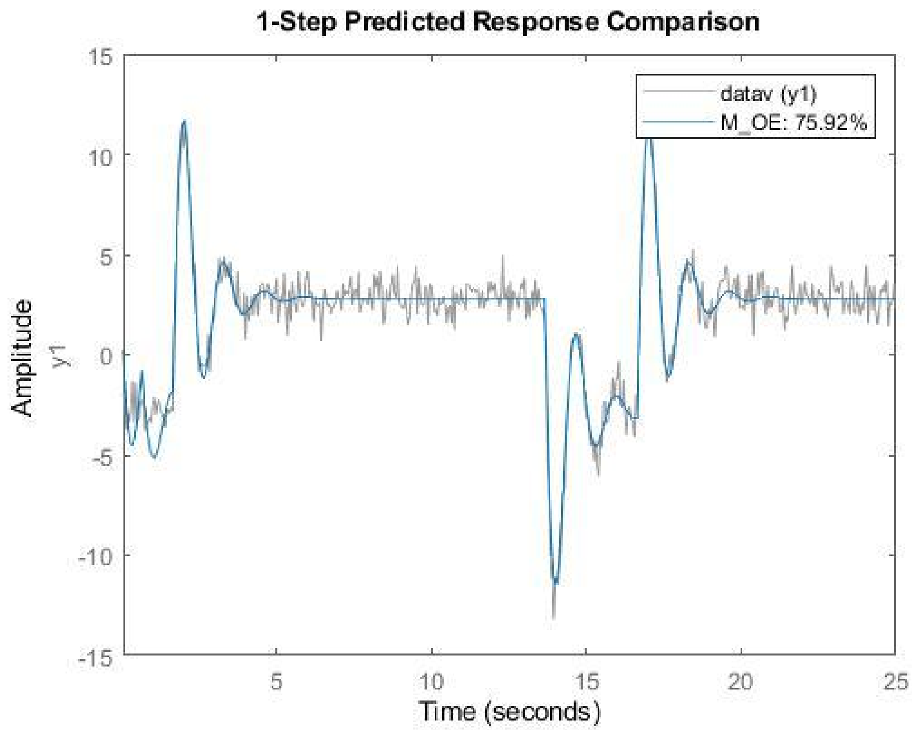

Figure 8: OE compared to original data (Noisy)

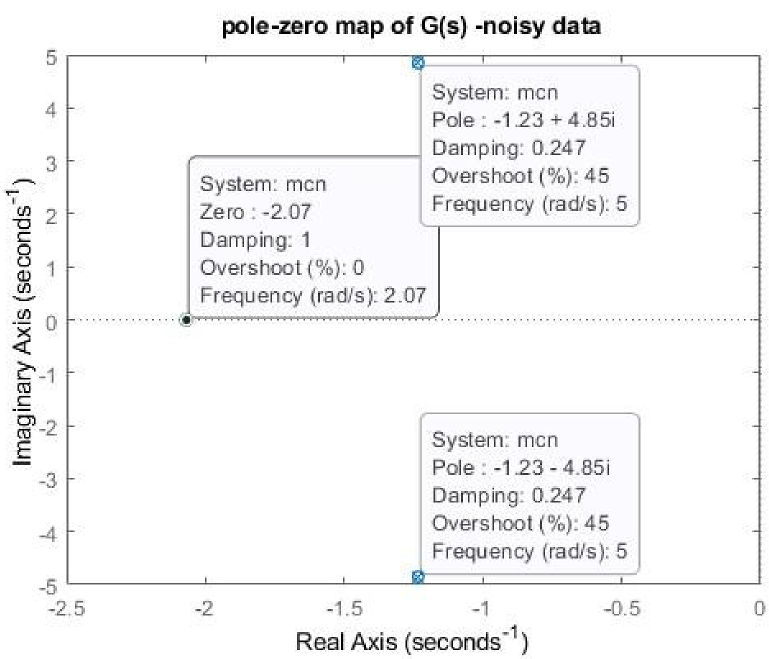

Figure 9: Pole-zero map of OE (noisy)

Part 3 - Conventional Model

The conventional model fits very well with the OE model created in part 2 as can be seen in figure 12 where there is only 0.04% difference between the two, and in figure 11 where the pole-zero map shows the poles and zeros to be very close to the values seen in figure 7 and figure 9. In this tutorial the OE model in part 2 was much easier to create as MATLAB does most of the work for you compared to the conventional method where characteristics of the graph in figure 5 had to be manually extracted to calculate the transfer function.

Calculation

This course uses a standard Ts = 0.05
The period can be found using the model found in figure 5 using the equation
Period = (peak2 - peak1)*Ts → (46-20)*0.05 = 1.3
K_dc = 0.7 (found from figure 5)
w_d = 2*pi*period
Damping ratio zeta can be found with slight trial and error but since we know that the system has a high P.O with slight oscillations, but the oscillations do not go past 0, we know the zeta value must be somewhere in the range of 0.15 ↔ 0.3
Zeta = 0.25
w_n = wd / sqrt(1-zeta^2)
a = 1.6*zeta*w_n
These values were used in MATLAB to generate the conventional method transfer function.

G(s) = exp(-0.6*s) * $\frac{17.44s + 34.83}{1.977s^2 + 4.983s + 49.75}$

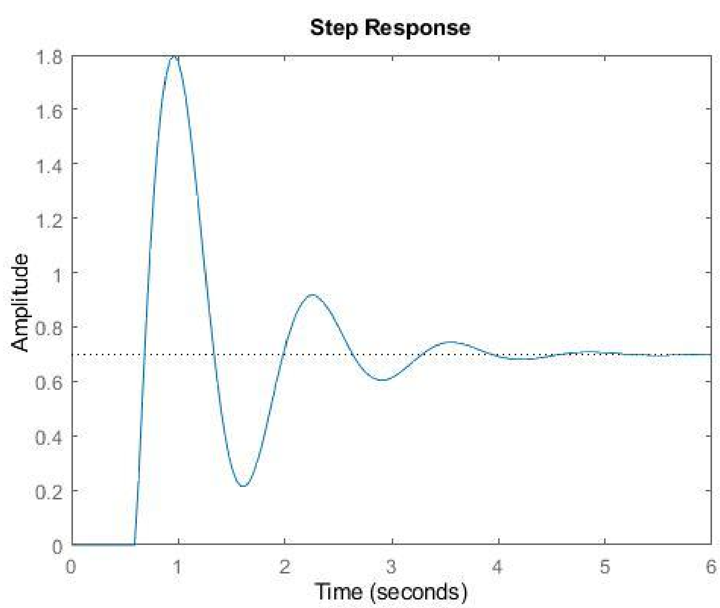

Figure 10: Step Response of conventional method

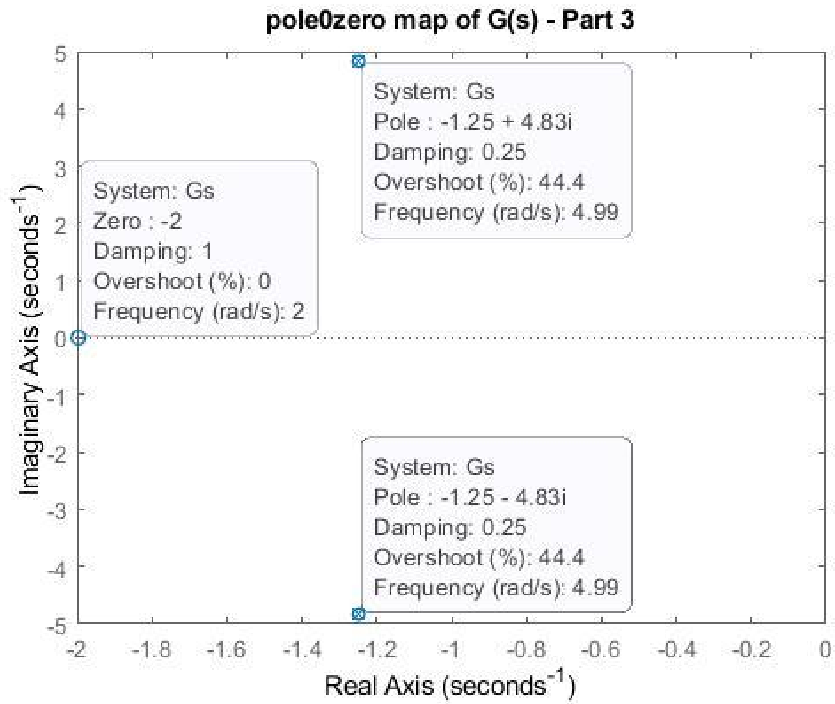

Figure 11: Pole-zero map of conventional method

Validate

G(z) = [B(z)/F(z)]u(t) + e(t)

B(z) = 0.4272 z^-1 - 0.3864 z^-2

F(z) = 1 - 1.824 z^-1 + 0.8827 z^-2

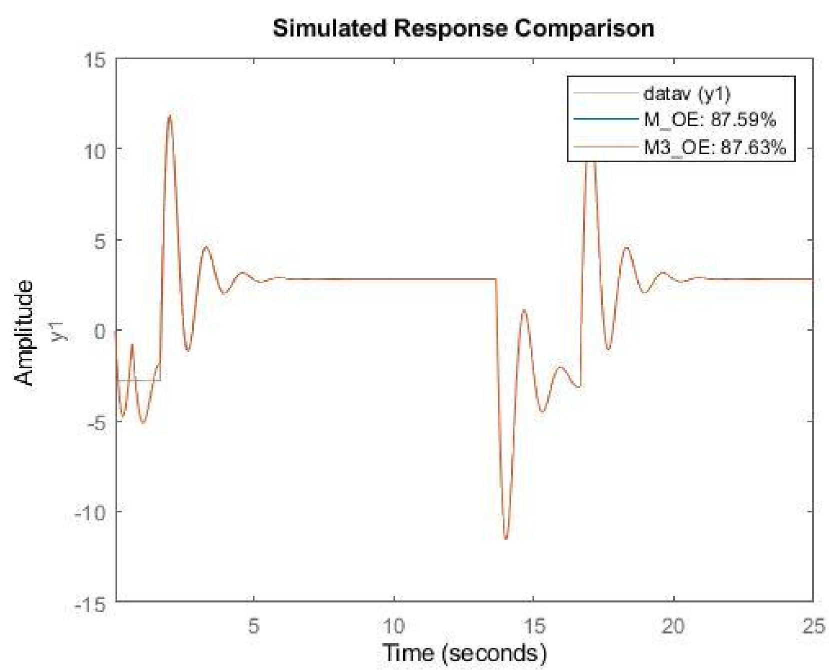

Figure 12: Comparison of conventional and OE to the original file