Computer Control

Laboratory

Identification and Computer Control of a Flexible Robot Arm Joint

Part 1

Authors:
Daniel Batista 86965
Maria Adelaide Ambrósio 87064
Ricardo Antão 87107

Faculty:
João Miranda Lemos
João Pedro Gomes

October 2020

Contents

1 Introduction

This project aims to identify and computer control design the position of a flexible arm joint. This arm joint is simulated by a flexible bar connected to a motor.

The design will be done using a simulation of the real model, using Simulink.

The work consists of two parts:

- 1^{st} Using system identification methods and data given by the professor, attainment of a plant model relating the motor with the angular position of the tip of the bar.
- 2^{nd} Design of a controller using the previous model in order to drive the tip to a desired position.

2 Plant Description

2.1 Q1 - Description of reasons that cause this control objective to be nontrivial

The bar used to simulate the arm is a flexible one. This means that when there is an oscillation the tip of the bar has a slight whip. This means that the angle dislocated at the tip of the bar attached to the motor is different from the angle dislocated at the end of the bar. This makes it a non-trivial problem.

Applying different tensions to the motor will imply different angular positions in the bar. This will eventually lead the bar to the final position in a series of velocities. However, the bar will still oscillate (whip effect at its tip) and the problem remains the same.

Applying a constant tension to the motor will cause a delay between the ends of the bar. This would imply some time until reaching a constant angular velocity.

Analyzing the system's open loop transfer two types of oscillations can be spotted:

- Low frequency, high amplitude and low dampening
- High frequency, low amplitude and high dampening

These oscillations are presented as a pair of complex-conjugate poles in the transfer function. Both have dampening which means the system is stable and therefore both pairs are placed in the left semi plane.

The "whip effect" is represented as a non-minimum phase zero that puts a zero in the right semi plane.

The DC motor also introduces a pole at the origin.

After observing that the zero in the right plane attracts a pole, the system becomes unstable very quickly. It is possible to conclude that that feedback with a controller made of single gain amplifying the tracking error would not solve the problem.

3 Software for data acquisition and control

3.1 Q2 - Open-loop experiments and data collection

When the command signal is constant the tip of the bar connected to the motor moves at constant angular velocities. Therefore angular position will change accordingly in a linear way. This can be verified by looking at the step response system.

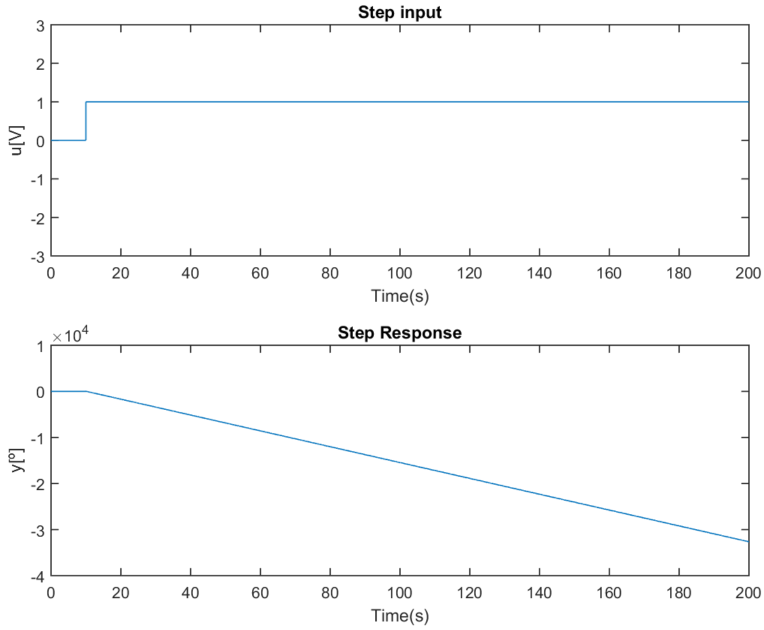
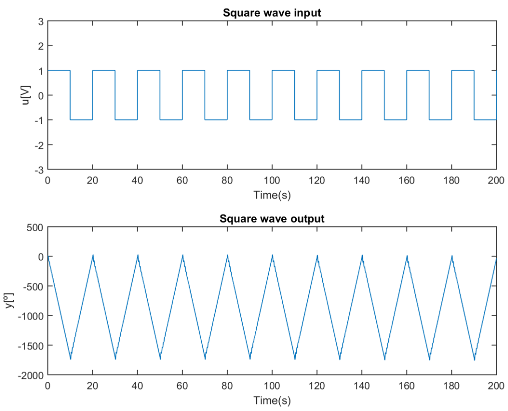


Figure 1: Step Response

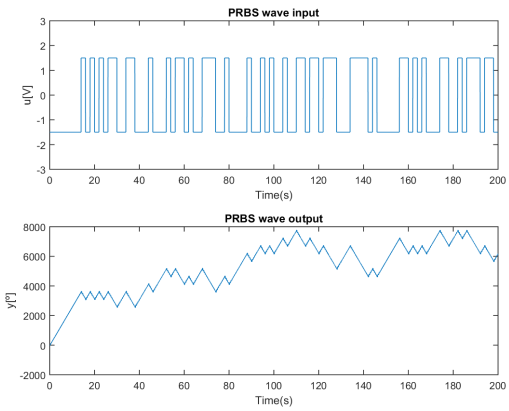
4 Plant Model Identification

4.1 Input Signals

In order to obtain the model, two input signals were used. A square wave with frequency of 0.05Hz and amplitude of 1 V and a PRBS wave with amplitude of 1.5 V and with B = 0.01, in other words, in a period of 100s the signal will spend as much time as possible at 1.5V and at -1.5V. The square wave was used to train the model and the PRBS was used as the test set of that model:



(a) Square Wave.



(b) PRBS Wave.

Figure 2: Input Signals.

If a too high frequency is used, the whip effect cannot be seen, as the system never leaves the transient regime, thus never stabilizing. This creates an output with a unexpected format, which is not good for the study of the system.

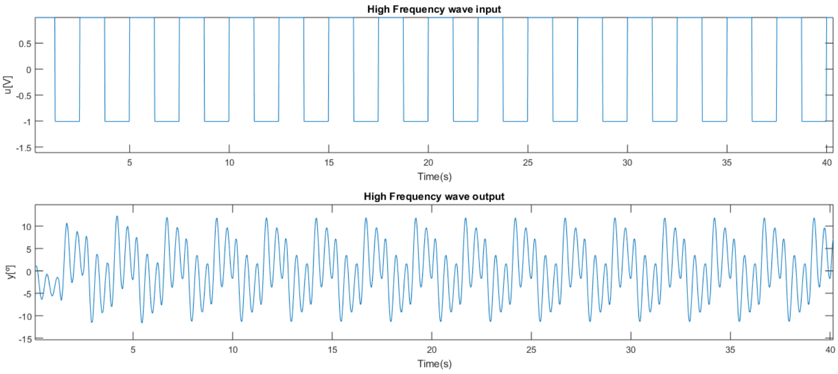


Figure 3: System when given a signl with too high frequency.

If a too low frequency is used, the system cannot excite the transient regime.

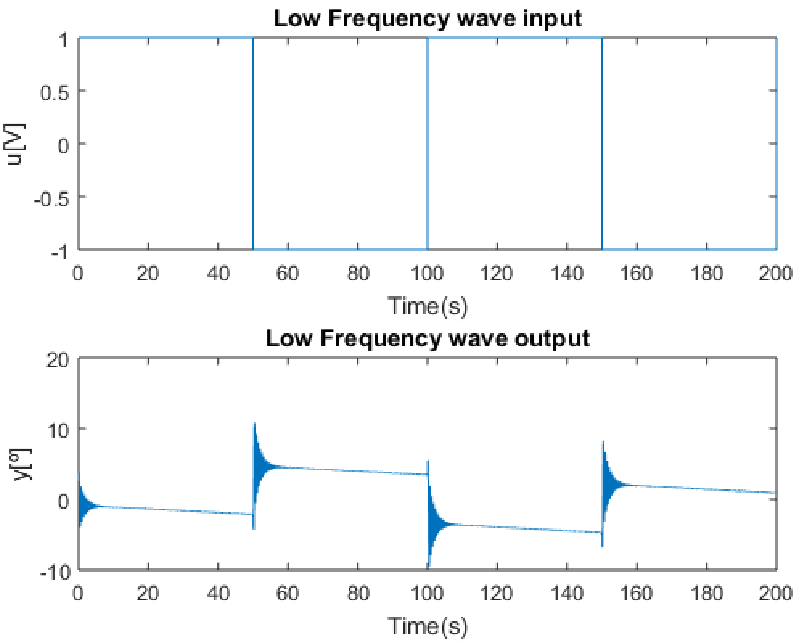


Figure 4: System when given a signl with too low frequency.

With the simulink model provided, the transient regime and the whip effect could only be simulated at lower frequencies than the ones recommended.

Regarding the amplitude of the input, changing it would only alter the amplitude of the output, not affecting any of the other properties. This is due to the outputs being generated by a simulink model and not a physical setup subject to the limitations of the hardware (like the gearbox backlash) and other imperfections.

4.2 Sampling frequency

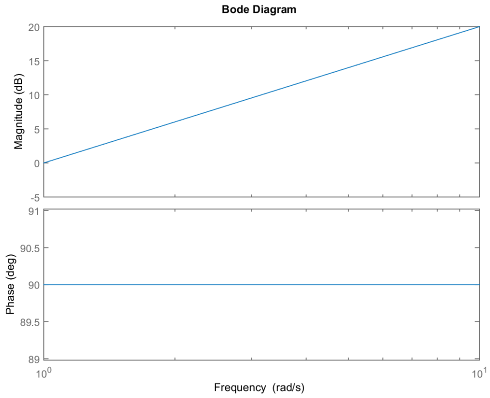
The sampling frequency chosen was 50Hz. This choice was due to the fact that higher sampling frequencies would add noise to the output. Therefore making it much worse for the identification task than a lower sampling frequency. A sampling frequency too low would violate the Nyquist theorem.

4.3 Filtering

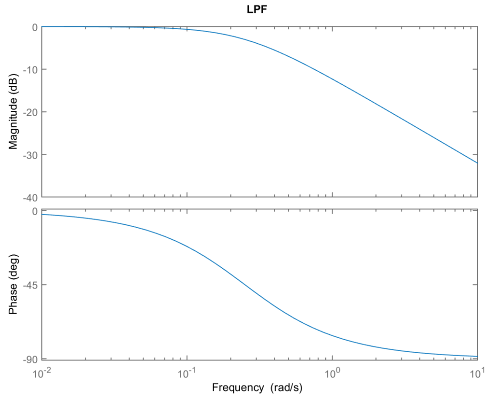
In order to obtain good identification results, the integral effect must be removed. This effect appears due to the relation between the angular velocity and the angular position of the motor shaft, which leads to an integrator in the motor, and because there is a non-zero average in the input result.

The integral effect (pole at the origin) was removed by using a differentiatior. The differentiator leads too noise at high frequencies, this was removed with a LPF filter.

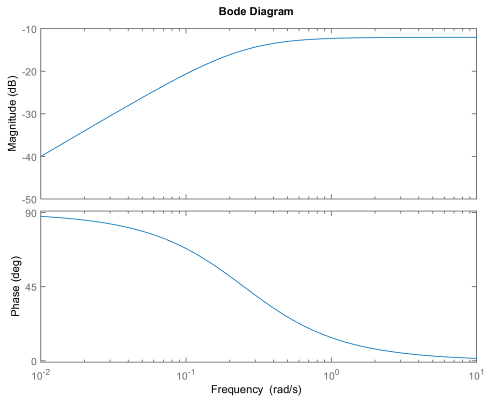
The total filter and its components Bode Diagram are the following:



(a) Differentiator



(b) Low Pass Filter



(c) The total filter applied

Figure 5: Filter

The transfer functions are given by:

- Differentiator:

$$D(z) = (1 - z^{-1}) \tag{1}$$

- Low pass filter:

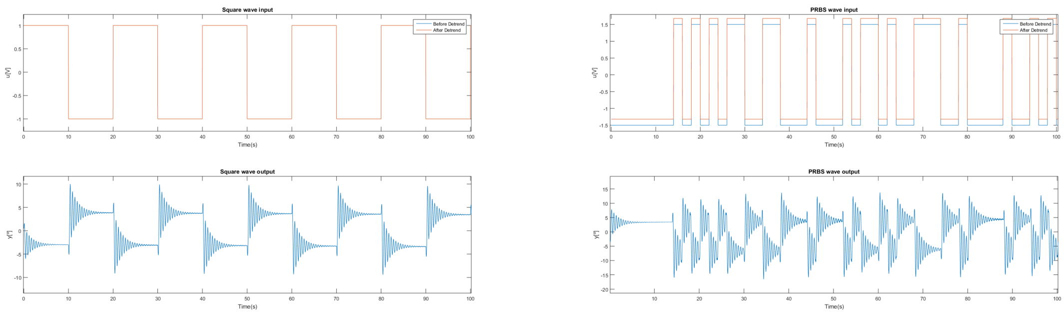
$$LPF(z) = \frac{0.2}{-0.8z^{-1} + 0.2} \tag{2}$$

- Total filter:

$$TF(z) = \frac{0.2(1 - z^{-1})}{-0.8z^{-1} + 0.2} \tag{3}$$

The non-zero averages of the input and outputs were removed with the *detrend()* functionof matlab.

The signals obtained after the filtering are as follow:



(a) Square signal filtered

(b) PRBS signal filtered

Figure 6: The output signals after filtering

4.4 Armax Model

A linear system with noise is described by:

$$y = Gu + He \tag{4}$$

The input signal is u and the noise is e. Therefore G is the transfer function of the system and H models the noise disturbance. To model G and H we are going to use the ARMAX model, where G and H will become $\frac{B(q)}{A(q)}$ and $\frac{C(q)}{A(q)}$ respectively. Thus, the ARMAX model is described by:

$$A(q)y(t) = B(q)u(t - n_k) + C(q)e(t) \tag{5}$$

Where $y(t)$ and $u(t - n_k)$ are the output and input signals respectively. Furthermore, n_k is the the dead time of system or system delay (input sample $u(t)$ will affect output sample $y(t + n_k)$). $A(q)$, $B(q)$ and $C(q)$ are the polynomials in respect to the delay operator $q = z^{-1}$. So the polynomials are given by:

$$\begin{cases} A(z) = 1 + a_1 z^{-1} ... + a_{n_a} z^{-n_a} \\ B(z) = 1 + b_1 z^{-1} ... + b_{n_b} z^{-n_b} \\ C(z) = 1 + c_1 z^{-1} ... + c_{n_c} z^{-n_c} \end{cases} \tag{6}$$

Where n_a, n_b and n_c are the order of the polynomials. From this we can infer that n_a is the number of poles and n_b is the number of zeros +1 of the system.

ARMAX refers to Autoregressive Moving Average eXtra. The 'AR' (autoregressive) part is described by the $A(z)$ polynomial. The 'MA' (moving average) is described by the $C(z)$ polynomial. The 'X' (eXtra) referes to the extra input described by $B(z)u(t)$.

4.4.1 Model Orders

The final model orders were reached by training the model with various possibilities of the set of parameters $n_n = [n_a\ n_b\ n_c\ n_k]$. The training was done recurring to the *armax()* function from the Control System Toolbox of MATLAB. The sets of n_n had to satisfy certain conditions: first $na >= nb + n_k - 1$ and second $n_a = n_c$. This is to ensure that in the final State Space Model the scalar D = 0. As mentioned before, the square wave in Figure2 was used for training and the PRBS wave was used for testing. After this process the 10 combinations of n_n which had the best testing accuracy were taken represented in Table 1.

na	nb	nc	nk	Train Accuracy(%)	Test Accuracy(%)
6	6	6	1	99.94005259	88.00802951
8	5	8	1	99.94180108	88.00180654
8	6	8	1	99.94181999	88.00020557
8	7	8	1	99.9418183	87.99928019
7	6	7	1	99.94181086	87.9992636
7	5	7	1	99.94146154	87.99894883
7	7	7	1	99.94182031	87.99892885
8	6	8	2	99.88091989	87.8092166
8	7	8	2	99.88091973	87.80822772
7	6	7	2	99.88091671	87.8080626

Table 1: Top 10 Training results

After this process the Pole-Zero plot of each of the best models was analysed and the final n_n chosen was [6 6 6 1]. This choice was due to the fact that, along with being the best performing one, it was also the one with the smallest order.

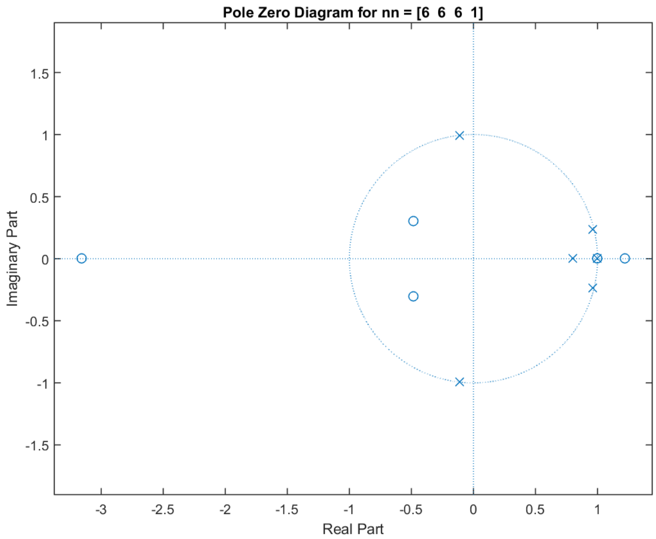


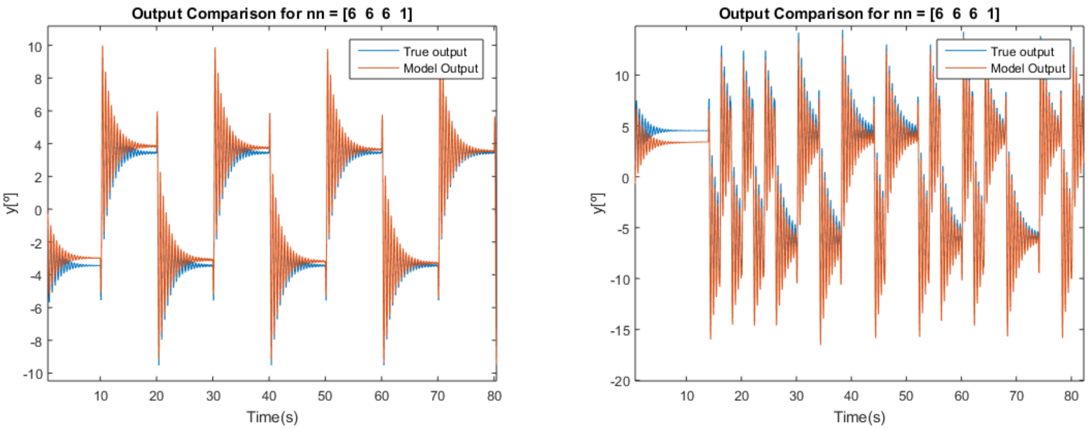
Figure 7: The output signals after filtering

4.4.2 Final ARMAX Model

With $n_n = [6\ 6\ 6\ 1]$ the final transfer function is described by:

$$H(z) = \frac{0.0405z^{-1} + 0.0773z^{-2} - 0.1847z^{-3} - 0.0593z^{-4} + 0.0750z^{-5} + 0.0511z^{-6}}{1 - 3.5031z^{-1} + 5.4218z^{-2} - 5.8653z^{-3} + 5.2821z^{-4} - 3.1150z^{-5} + 0.7795^{-6}} \tag{7}$$

From this, it can be concluded that the model has 6 poles and 5 zeros, which are represented in Figure 7. To obtain the final state-space model the integrator must be added back into the system. However, before that, the model must be validated by testing its response to the two input signals. This response can be visualized in Figure 13



(a) Square Wave. (b) PRBS Wave.

Figure 8: Validation of the model.

4.4.3 Pole-Zero Analysis

The integrator transfer function is $\frac{1}{1-z^{-1}}$, through the MATLAB function *eqftlength()* to also change the delay operator to forward operator. Adding the integrator will add a zero at the origin and a real pole at s = 1. The pole-zero plot of the final model is shown in Figure 9 and the transfer function of the final model becomes:

$$G(z) = \frac{0.0405z^6 + 0.0773z^5 - 0.1847z^4 - 0.0593z^3 + 0.0750z^2 + 0.0511z^1}{z^7 - 4.503z^6 + 8.925z^5 - 11.29z^4 + 11.15z^3 - 8.397z^2 + 3.894z - 0.7795} \tag{8}$$

Zeros	Poles
0	-0.1088 + 0.99234i
-3.1591	-0.1088 - 0.9924i
1.2216	0.9598 + 0.23456i
1	0.9598 - 0.23456i
-0.4855 + 0.3022i	1
-0.4855 - 0.3022i	1
	0.8011

Table 2: Location of the poles and zeros of the system.

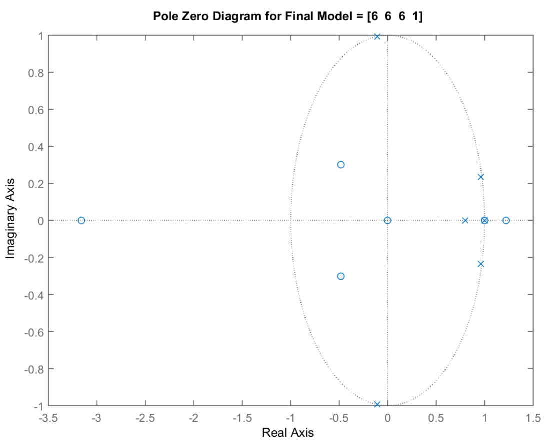


Figure 9: Pole-Zero Plot of plant transfer function.

To be accurate the system must model the whiplash effect and the oscillations of the output signal. The whiplash effect is modeled by the two zeros, one in the right side of the plane outside the unit circle at $s = 1.2216$ and another on the left side of the plane $s = -3.1591$. The oscillations are modeled by both pairs of complex poles at $s = -0.1088 \pm 0.99234i$ and $s = 0.9598 \pm 0.23456i$. Therefore the oscillations in the output are the sum of two sinusoidals one for each pair. The pole at $s = 0.80$ is due to the differentiator. The rest of the poles and zeros contribute to the gain of the system, the poles contribute with terms that decay in time and the zeros contribute with terms that grow in time. It can also be seen that, as expected, the integrator added the zero at the origin and a pole at the unit circle.

4.4.4 Final State-Space Model

It is possible to convert the ARMAX model to an equivalent state-space model:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k+1) = Cx(k) + Du(k) \end{cases} \tag{9}$$

Where:

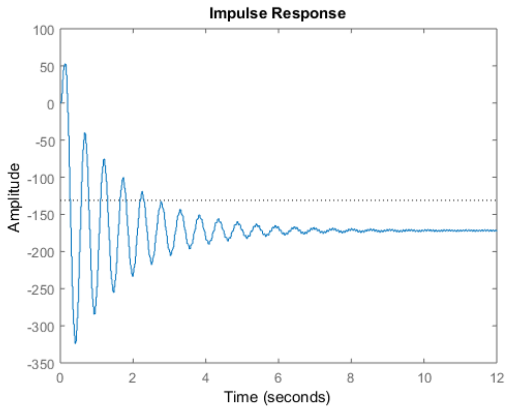
$$A = \begin{bmatrix} 4.503 & -8.925 & 11.29 & -11.15 & 8.397 & -3.894 & 0.7795 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

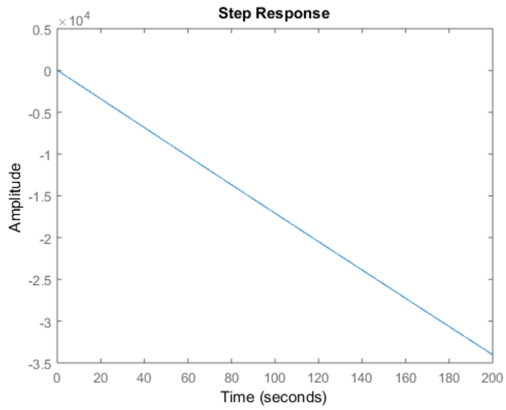
$$C = \begin{bmatrix} 0.041 & 0.077 & -0.185 & -0.059 & -0.075 & -0.051 & 0 \end{bmatrix}$$

$$D = 0$$

4.4.5 Model Time and Frequency Response



(a) Impulse Response.



(b) Step Response.

Figure 10: Validation of the model.

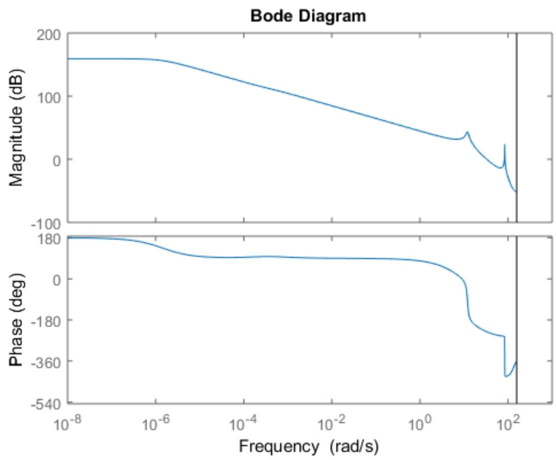


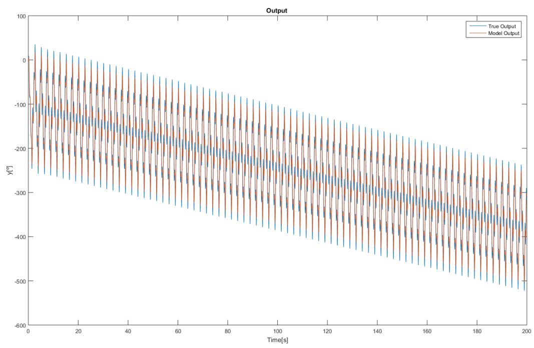
Figure 11: Bode Diagram.

Figure 12: Validation of the model.

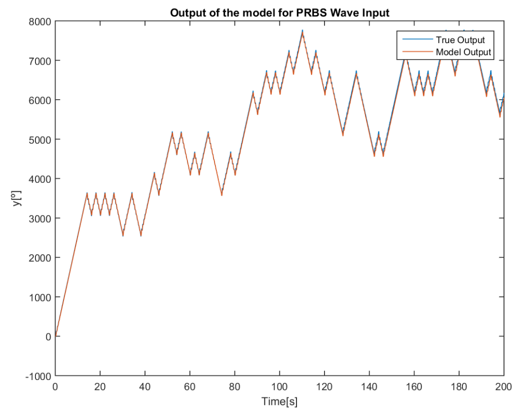
The two peaks visible in the bode diagram are caused by the conjugated poles. Observing the magnitude of the graph, the magnitude starts with a value of 159 dB, which corresponds to the pole added by the integrator.

For low frequencies, the system has high amplitude and for high frequencies it has low magnitude. This mimics the behaviour of a low pass filter.

4.4.6 Model Validation



(a) Square Wave Input (f = 0.4Hz).



(b) Step Response.

Figure 13: Validation of the model.

References

Miranda Lemos, J. 2018. Lecture Slides