Ryerson University

**Department of Electrical, Computer, & Biomedical Engineering**
Faculty of Engineering & Architectural Science

| | |
|---|---|
| Course Title: | Signal and Systems II |
| Course Number: | ELE 632 |
| Semester/Year (e.g.F2016) | Winter 2021 |
| Instructor: | Dr. Dimitri Androutsos (TA: Mahdi Shamsi) |

| | |
|---|---|
| Assignment/Lab Number: | 04 |
| Assignment/Lab Title: | Discrete Time Fourier Transform |

| | |
|---|---|
| Submission Date: | March 21, 2021 |
| Due Date: | March 21, 2021 |

| Student LAST Name | Student FIRST Name | Student Number | Section | Signature* |
|---|---|---|---|---|
| ALAM | ABRAR | 500725366 | 11 | Abrar Alam |
| BHATT | SAGAR | 500837713 | 10 | Sagar |
| | | | | |

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report 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: http://www.ryerson.ca/senate/current/pol60.pdf

## Table of Contents

## Abstract

In Part A1, A2 the efficiency of the Matlab's **Fast Fourier Transform (FFT)** algorithm subroutine *fft* was demonstrated to compute DTFT of an N-point signal. Besides, DTFT plots of the signals were presented within $(\pi, \pi)$ interval, due to the fact that DTFT of any signal repeats in every $\Omega = 2\pi$. Besides the necessity of *fftshift* command for accurate DTFT plotting was also shown in these parts. In part A3 the use of *ifft* command to reconstruct a time domain signal from its DTFT was shown.

In Part B1, B2 an alternative approach to calculation of DTFT (using matrix-based approach) was examined. The results from these parts were then used to prove the fact that **convolution in time domain is equivalent to taking product in the frequency domain**, in part B3, B4, B5, and B6.

For parts C1 and C2 we followed the instructions given in section 9.7-3 of *Linear Systems and Signals (3e)*. $H_d$ was customized to apply to a high-pass filter. Using $H_d$, $H(\Omega)$ and $h[n]$ were retrieved using the instructions. We did it for both linear phase and zero phase. Linear phase was found to be more accurate. The commands for C1 and C2 were put in a function to be reused for C4. Questions C3 and C5 were answered by referring to the diagrams.

# Part A

## Part 1

Matlab's built in *fft* command was used to calcu;ate the DTFT of the signal $x[n]$ (refer to the lab 4 manual for the graph of $x[n]$). **Figure 1** shows the magnitude and phase plots of $X(\Omega)$ in $(-\pi, \pi)$. *fftshift* command was used to center the zero-frequency component before generating these plots. Please refer to the attached codefile file named "partA_part1.m" if necessary.

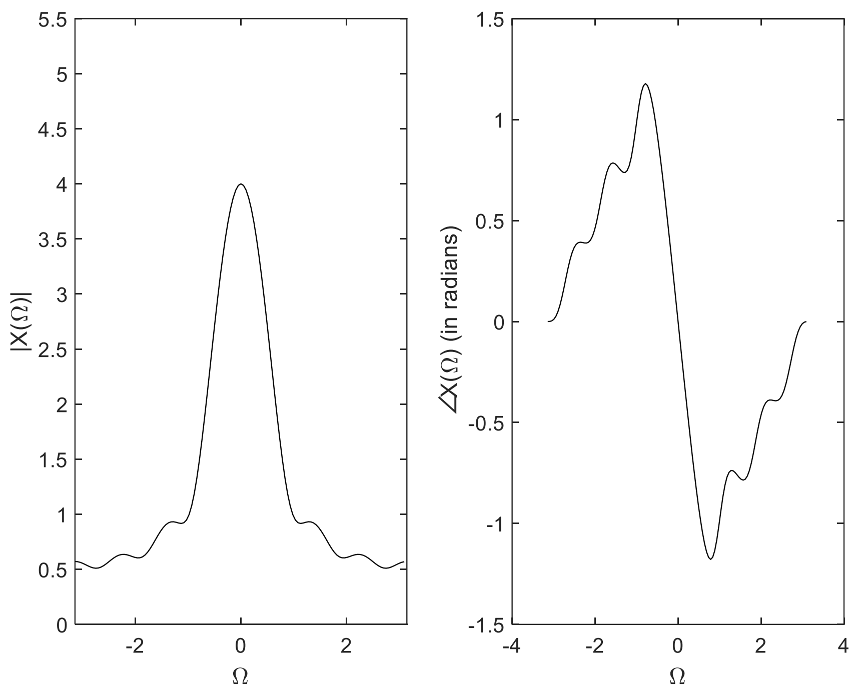


*Figure 1:* Magnitude and phase plot of $X(\Omega)$ from part A part 1.

## Part 2

The expression for the DTFT of $x[n]$ was calculated as follows:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \qquad (1)$$

Equation **1** can be rewritten as,

$$X(\Omega) = \sum_{n=0}^{6} (-\frac{1}{7}n + 1) e^{-j\Omega n}$$

$$= \left(-\frac{1}{7}(0) + 1\right)e^{0} + \left(-\frac{1}{7}(1) + 1\right)e^{-j\Omega} + \left(-\frac{1}{7}(2) + 1\right)e^{-j2\Omega} + \left(-\frac{1}{7}(3) + 1\right)e^{-j3\Omega} + \left(-\frac{1}{7}(4) + 1\right)e^{-j4\Omega} + \left(-\frac{1}{7}(5) + 1\right)e^{-j5\Omega} + \left(-\frac{1}{7}(6) + 1\right)e^{-j6\Omega}$$

$$= 1 + \frac{6}{7}e^{-j\Omega} + \frac{5}{7}e^{-j2\Omega} + \frac{4}{7}e^{-j3\Omega} + \frac{3}{7}e^{-j4\Omega} + \frac{2}{7}e^{-j5\Omega} + \frac{1}{7}e^{-j6\Omega} \qquad (2)$$

Since we are using Matlab to plot the equation **2**, we do not need to simply this any further. Thus equation **2** is the DTFT expression of $x[n]$.

**Figure 2** shows the magnitude and phase plots generated using equation **2**. Please refer to the attached code file named "partA_part2.m" if necessary.

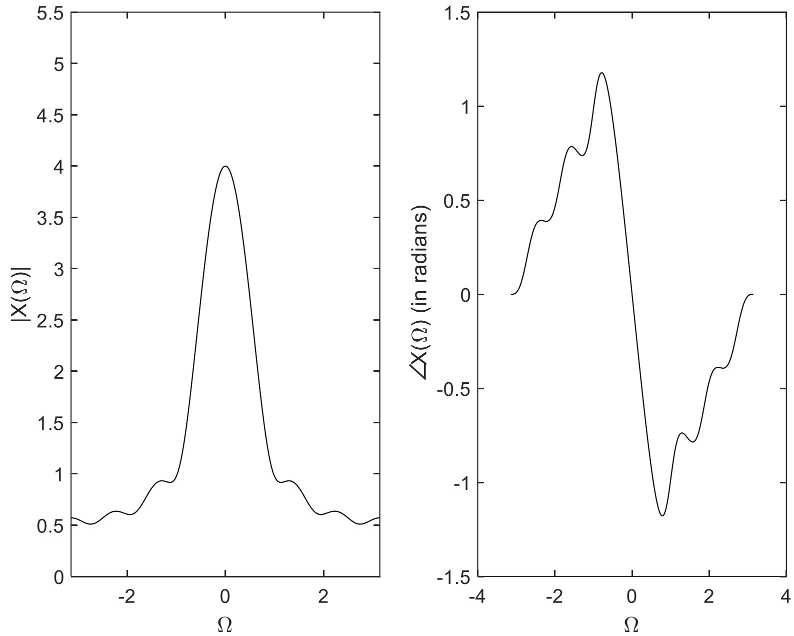


*Figure 2:* Magnitude and phase plot generated using hand calculation of DTFT of $x[n]$ (part A, part 2).

Clearly, **figure 2** is exactly the same as the plot presented in **figure 1** from part 1, which is expected.

## Part 3

Here *ifft* command from Matlab was used to reconstruct the time domain signal from its DTFT spectrum $X(\Omega)$. **Figure 3** shows the synthesized $x[n]$ signal. Please refer to the attached code file named "partA_part3.m" if necessary.

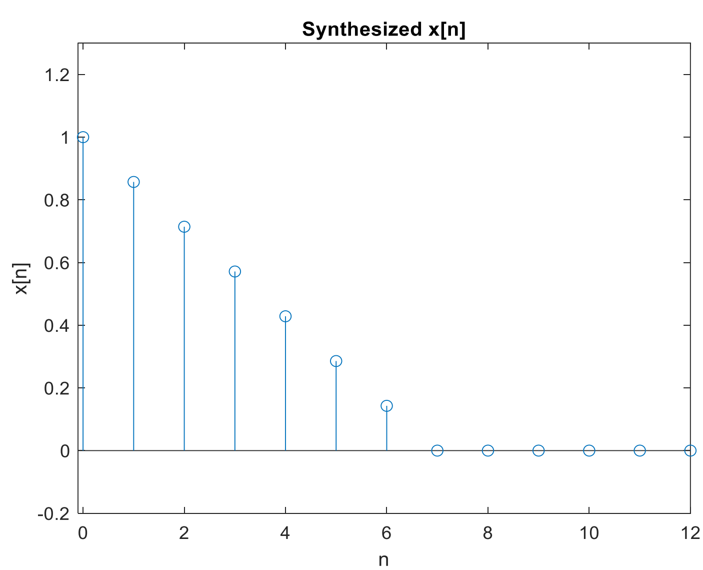


*Figure 3:* Reconstructed x[n] from its DTFT spectrum.

In **figure 3** the time domain signal is exactly the same as the time domain signal $x[n]$ presented in the lab manual, which is what we expect.

# Part B

## Part 1

Here the DTFT of the signal $x[n] = \sin\left(\frac{2\pi n}{10}\right)(u[n] - u[n-10])$ was computed using a matrix-based approach instead of using *fft* command. **Figure 4** shows the magnitude and phase plot of the DTFT of this signal. Please refer to the code file named “partB_part1.m” if needed.

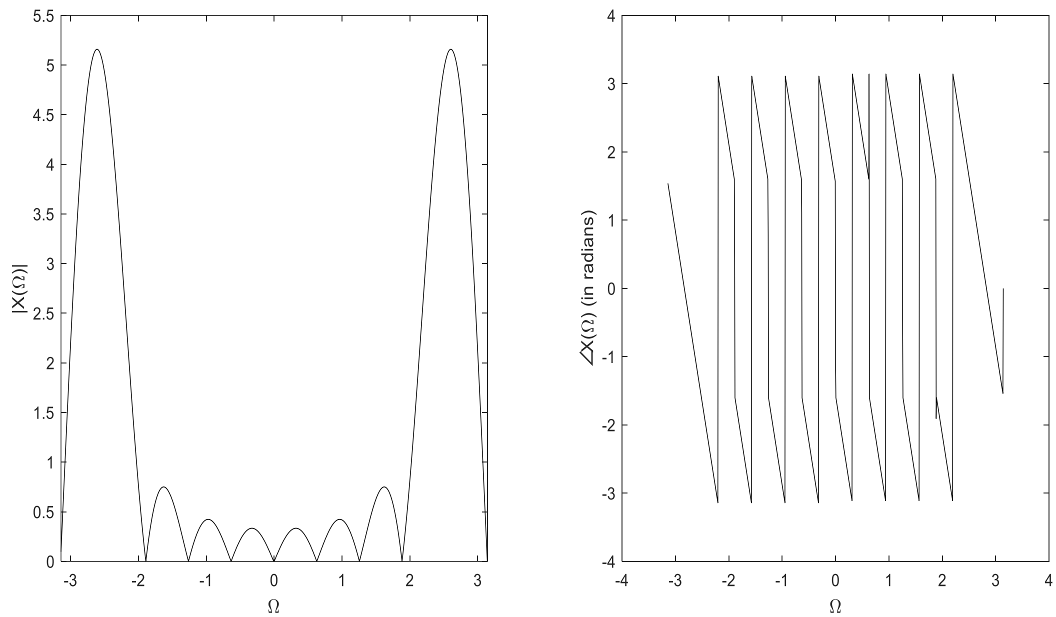


*Figure 4*: DTFT of the signal $x[n] = sin\left(\frac{2\pi n}{10}\right)(u[n] - u[n-10])$ generated using a matrix-based approach

## Part 2

Here the same matrix-based approach was used to generate the DTFT of the signal $h[n]$ (please refer to the lab 4 manual for the time-domain graph of this signal). **Figure 5** shows the DTFT of the signal. Please refer to the attached code file named “partB_part2.m” if necessary.

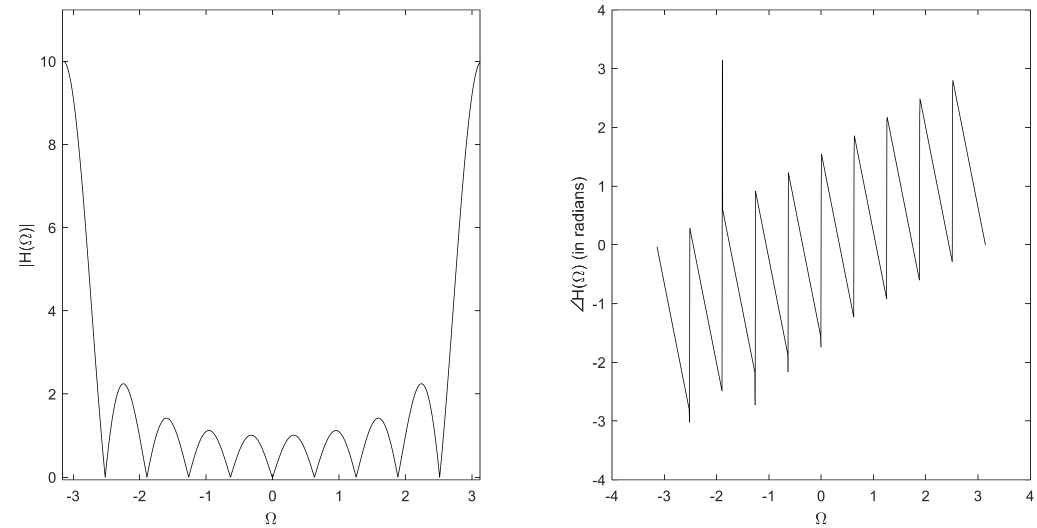


*Figure 5:* DTFT magnitude an phase plot of the signal $h[n]$.

## Part 3

Here the DTFT of the time domain signal $y[n] = x[n] * h[n]$ is calculated by taking the product of $X(\Omega)$ and $H(\Omega)$. **Figure 6** shows the plot of $X(\Omega)H(\Omega)$. Please refer to the code file named "partB_part3.m" if needed.

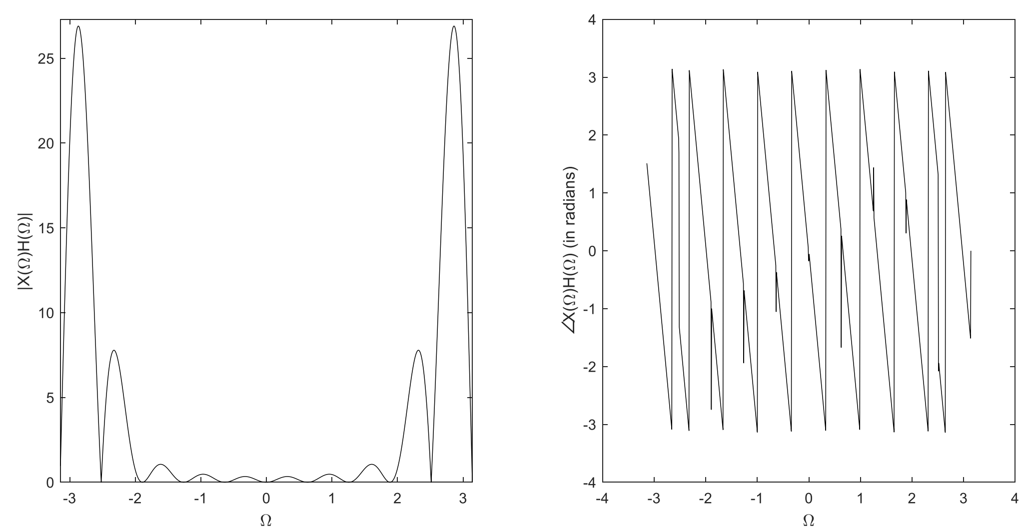


*Figure 6:* Magnitude and phase plot of $X(\Omega)H(\Omega)$.

## Part 4

Here Matlab *conv* command was used to compute $y[n] = x[n] * h[n]$. **Please note that since the lab manual does not ask for a plot of $y[n]$, this plot has not been presented in the report for the sake of brevity. However, for the verification purposes, please refer to the attached code file named "partB_part4_5_6.m".**

## Part 5

Here we plot the DTFT of the time-domain signal $y[n]$ calculated in part 4. **Figure 7** shows the plot of the DTFT. Please refer to the attached code file named "partB_part4_5_6.m" if required.

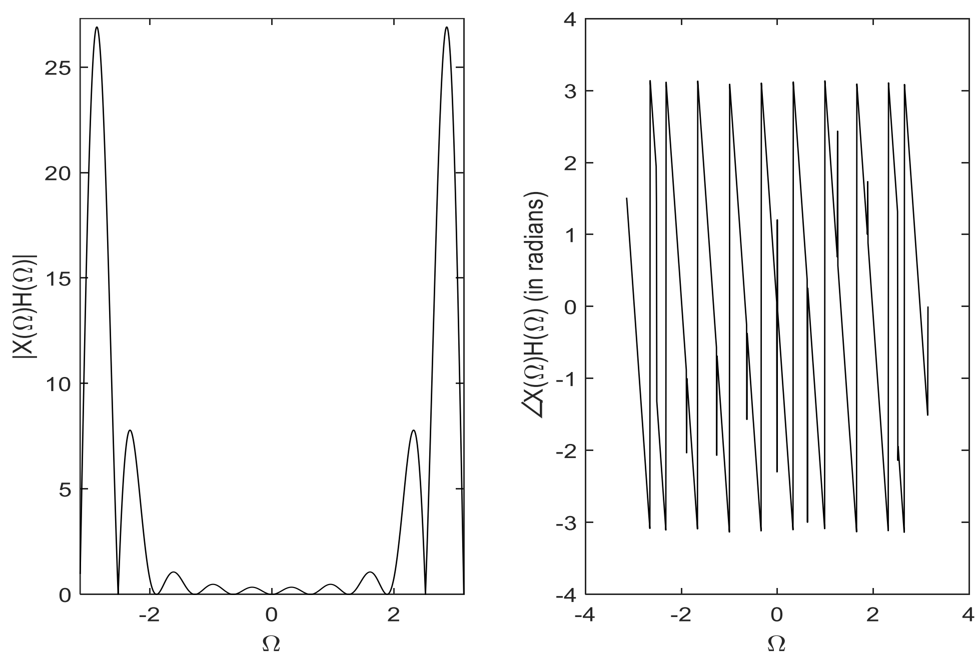


*Figure 7:* DTFT of the signal $y[n] = x[n] * h[n]$.

## Part 6

Examination of **figure 7** and **figure 6** reveals that these plots are exactly the same. This is to be expected because **convolution in time domain is equivalent to product in the frequency domain.** For example, if we have $y[n] = x[n] * h[n]$ then $Y(\Omega) = X(\Omega)H(\Omega)$, that is $Y(\Omega)$ can be calculated either by taking DTFT of $y[n]$, or by just taking the product of $X(\Omega)$, and $H(\Omega)$.

## Part C

Elaborated on in the abstract.
The following are the diagrams for C1 and C2. As you can see the passband allows frequencies higher than 2pi/3 and is approximately unity, whereas the stopband is approximately 0.

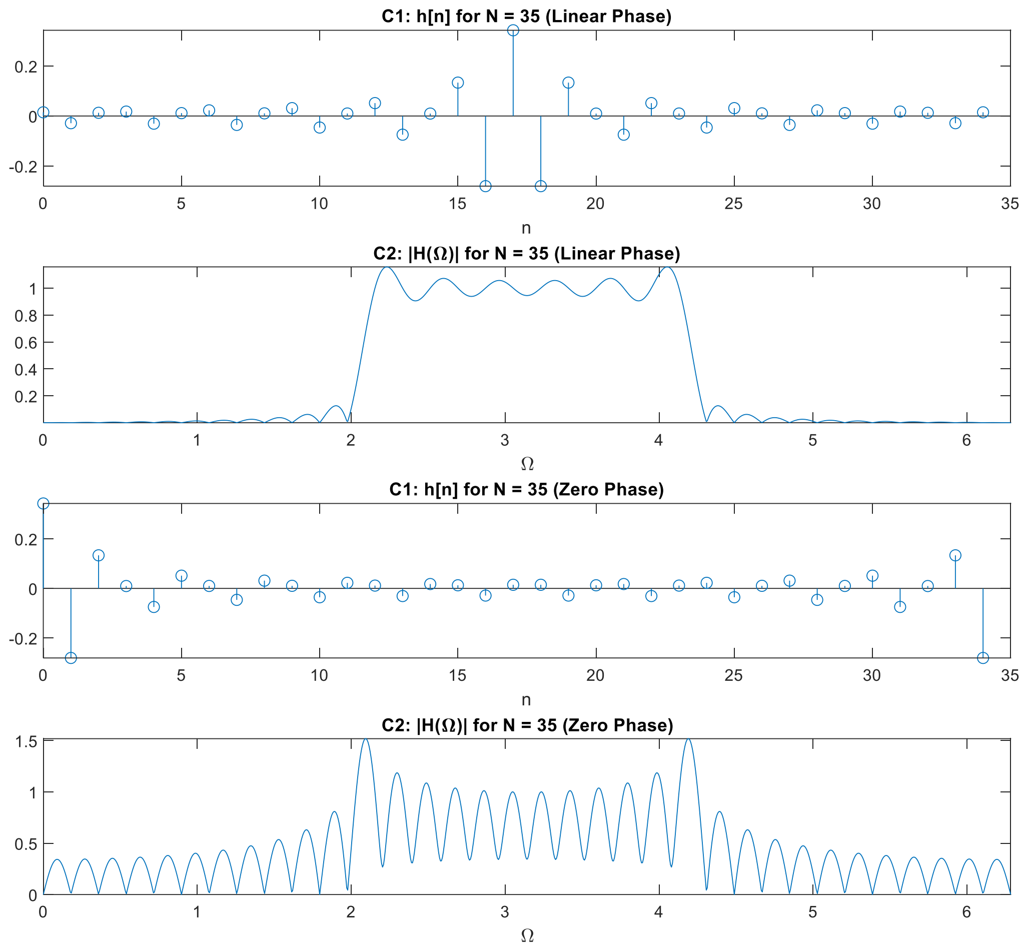


*Figure 8 C1 and C2*

C3: The ideal highpass filter allows distortionless transmission of signals above $\frac{2\pi}{3}$ in the passband while suppressing other frequencies perfectly. The gains are either zero or unity over certain bands – there is a sudden transition with infinite slope from the passband (unity gain) to the stopband (zero gain). On the other hand, the filter in C2 has a transition band where the transitions between the passband and stopband are gradual – large slope but not infinite. The stopband in C2 is where the gain is small, but not zero. Likewise the passband in C2 does not have a perfect unity gain.

On the next page are the diagrams for C4. As you can see the passband allows frequencies higher than 2pi/3 and is approximately unity, whereas the stopband is approximately 0.

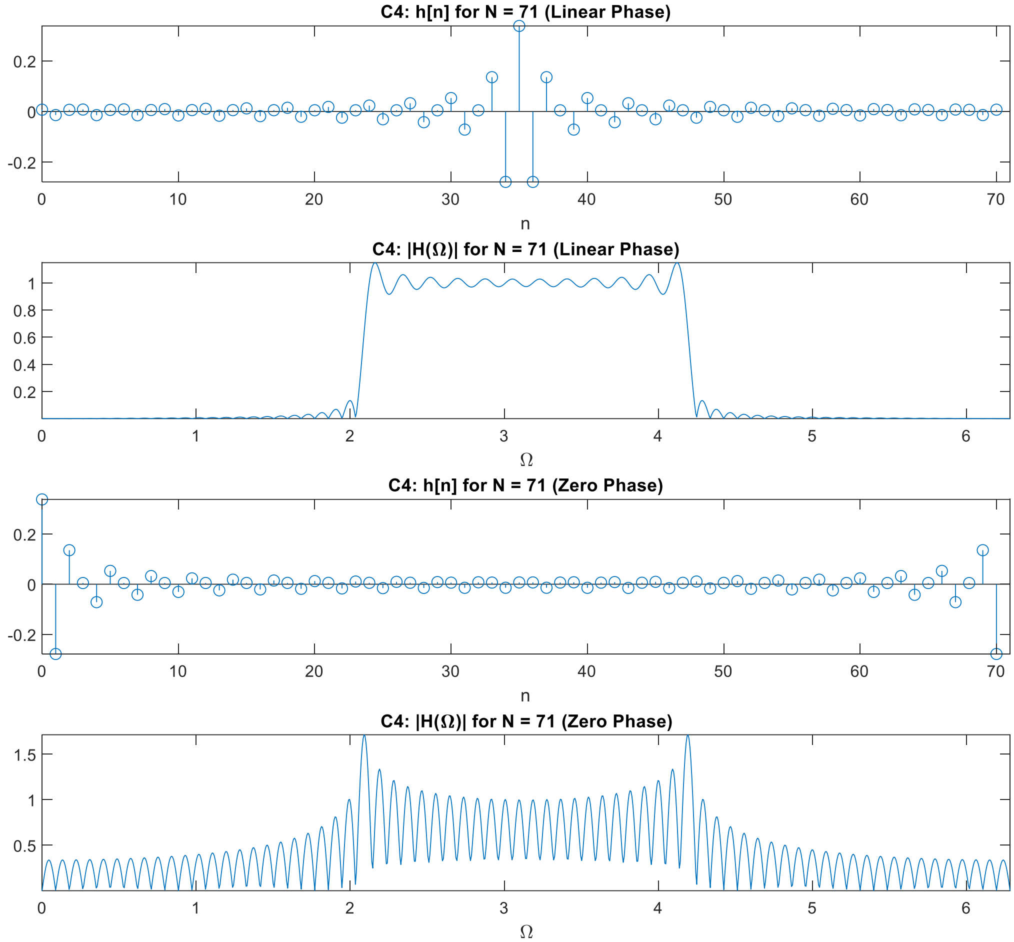


*Figure 9 C4*

C5: Increasing the length makes the result more accurate, closer to the ideal filter.