

# Fast Fourier Transform

## APPM 4600 Fall 2024 Section 001

Elizabeth Cutting, Edward Wawrzynek, Erick White

DATE

## 1 Overview

In this project, we investigate the Fast Fourier Transform (FFT), a method for trigonometric interpolation on equispaced points along an interval that uses symmetry to reduce computational complexity from  $\mathcal{O}(n^2)$  to  $\mathcal{O}(n \log n)$ . We begin with a derivation of the Discrete Fourier Transform (DFT), then show its drawbacks, focusing on computational complexity, and propose the FFT as an alternative. We explore the properties of the FFT, including its restrictions on the sample points, and compare its performance to that of the DFT. We continue with an exploration of the Short-Time Fourier Transform (STFT), a common tool in signal analysis and processing.

## 2 The Discrete Fourier Transform

### 2.1 Derivation

#### 2.1.1 Finite Fourier Series

Consider some piecewise smooth function  $f(t) \in \mathbb{C}$  defined over the interval  $t \in [-\pi, \pi]$ . Over that interval, we are interested in approximating  $f$  with a trigonometric basis, that is, a basis composed of sines and cosines. This approximation is called the Fourier series for  $f$ , and is given as

$$\bar{f}(t) = A_0 + \sum_{n=1}^N A_n \cos(nt) + \sum_{n=1}^N B_n \sin(nt). \quad (1)$$

Euler's identity allows us to write an individual term in the series as

$$A_n \cos(nt) + B_n \sin(nt) = \frac{A_n - iB_n}{2} e^{int} + \frac{A_n + iB_n}{2} e^{-int},$$

so we can write the full series (1) as

$$\begin{aligned} f(t) &\approx A_0 + \sum_{n=1}^N (c_n e^{int} + c_{-n} e^{-int}) \\ &= \sum_{n=-N}^N c_n e^{int}, \end{aligned} \quad (2)$$

where the coefficients are given by

$$c_n = \begin{cases} \frac{1}{2} (A_{-n} + iB_{-n}) & n < 0 \\ A_0 & n = 0 \\ \frac{1}{2} (A_n - iB_n) & n > 0 \end{cases}.$$

We can invert this relationship to give the coefficients

$$A_0 = c_0, \quad A_n = c_n + c_{-n}, \quad \text{and} \quad B_n = ic_n - ic_{-n}. \quad (3)$$

**Theorem 1.** *Let  $f(t)$  be a piecewise smooth function defined over  $t \in [-\pi, \pi]$ , and let  $\bar{f}$  be the fourier series*

$$\bar{f}(t) = \sum_{n=-N}^N c_n e^{int}.$$

*Then the mean square error between  $f$  and  $\bar{f}$ ,*

$$E = \int_{-\pi}^{\pi} (f(t) - \bar{f}(t))^2 dt,$$

*is minimized when  $c_n$  is chosen as*

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} dt. \quad (\text{for } n \neq 0) \quad (4)$$

*Proof.* The minimum of  $E$  occurs at a stationary point, that is,

$$0 = \frac{\partial E}{\partial c_m} = \frac{\partial}{\partial c_m} \int_{-\pi}^{\pi} \left( f(t) - \sum_{n=-N}^N c_n e^{int} \right)^2 dt.$$

Since both  $f$  and  $\bar{f}$  are piecewise smooth, by the Leibniz rule we can take the partial derivative inside the integral and we have

$$\begin{aligned} 0 &= \int_{-\pi}^{\pi} \frac{\partial}{\partial c_m} \left( f(t) - \sum_{n=-N}^N c_n e^{int} \right)^2 dt \\ &= \int_{-\pi}^{\pi} 2 \left( f(t) - \sum_{n=-N}^N c_n e^{int} \right) \frac{\partial}{\partial c_m} \left( f(t) - \sum_{n=-N}^N c_n e^{int} \right) dt \\ &= -2 \int_{-\pi}^{\pi} \left( f(t) - \sum_{n=-N}^N c_n e^{int} \right) e^{imt} dt \\ &= -2 \int_{-\pi}^{\pi} \left( f(t) e^{imt} - \sum_{n=-N}^N c_n e^{int} e^{imt} \right) dt. \end{aligned}$$

We have that our basis functions  $\{e^{int}\}$  are orthogonal under the complex inner product, so we can reduce the sum on the right to a single term where  $n = -m$ , that is,

$$0 = \int_{-\pi}^{\pi} (f(t) e^{imt} - c_{-m}) dt,$$

which implies that we should choose  $c_m$  as

$$c_m = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-imt} dt$$

to guarantee a stationary point. We wish to show that this point is also a minimum. Observe that we can

expand  $E$  as

$$\begin{aligned}
E &= \int_{-\pi}^{\pi} \left( f(t) - \sum_{n=-N}^N c_n e^{int} \right)^2 dt \\
&= \int_{-\pi}^{\pi} \left( f(t)^2 - 2f(t) \sum_{n=-N}^N c_n e^{int} + \sum_{n=-N}^N \sum_{m=-N}^N c_n c_m e^{int} e^{imt} \right) dt \\
&= \int_{-\pi}^{\pi} \left( f(t)^2 - 2f(t) \sum_{n=-N}^N c_n e^{int} + \sum_{n=-N}^N c_n c_{-n} \right) dt \\
&= \sum_{n=-N}^N \left( 2\pi c_n^2 - 2c_n \int_{-\pi}^{\pi} f(t) e^{int} dt \right) + \int_{-\pi}^{\pi} f(t)^2 dt.
\end{aligned}$$

We complete the square in the first term to get

$$E = \sum_{n=-N}^N (2\pi (c_n -))^2 + \int_{-\pi}^{\pi} f(t)^2 dt$$

□

### 2.1.2 Discrete Signals and Coefficients

A discrete signal  $f$  with uniformly spaced samples  $f_0, f_1, \dots, f_{N-1}$  has a discrete Fourier transform (DFT) given by

$$\hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-i2\pi jk/N}. \quad (5)$$

By analogy to the relationship in (3), we have

$$A_0 = \hat{f}_0, \quad A_n = \hat{f}_n + \hat{f}_{-n}, \quad \text{and} \quad B_n = i\hat{f}_n - i\hat{f}_{-n}.$$

We wish to find the coefficients  $\hat{f}_0, \hat{f}_1, \dots, \hat{f}_{N-1}$  in terms of the samples  $f_0, f_1, \dots, f_{N-1}$ . Equation (5) gives us a system of  $N$  equations for our  $N$  unknown coefficients. We observe that the fundamental frequency in the DFT is  $\omega_N = e^{-i2\pi/N}$ . We can write the system as  $\mathbf{\hat{f}} = F_N \mathbf{f}$ , where the system matrix is

$$F_N = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & & \omega_N^{2(N-1)} \\ \vdots & & & \ddots & \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & & \omega_N^{(N-1)^2} \end{bmatrix}.$$

Evaluating this system directly is expensive.  $F_N$  is of size  $N \times N$ , which means that the system requires  $\mathcal{O}(N^2)$  time to evaluate, which can be severely limiting when try to process a large number of sample points.

### 2.1.3 Improving the DFT

We consider a signal with a number of samples which is a power of two, that is, we restrict ourselves to  $N = 2^k, k \in \mathbb{Z}^+$ . As a first step towards improving the performance of our DFT calculation, we will split

the problem in two. Let  $n = N/2$ . By breaking (5) into sums over even and odd indices, we have

$$\begin{aligned}
 \hat{f}_k &= \sum_{j=0}^{n-1} f_{2j} e^{\frac{-i2\pi k}{N}(2j)} + \sum_{j=0}^{n-1} f_{2j+1} e^{\frac{-i2\pi k}{N}(2j+1)} \\
 &= \sum_{j=0}^{n-1} f_{2j} e^{\frac{-i2\pi k}{N/2}j} + e^{\frac{-i2\pi k}{N}} \sum_{j=0}^{n-1} f_{2j+1} e^{\frac{-i2\pi k}{N/2}j} \\
 &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{kj} + \omega_n^{k/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{kj}.
 \end{aligned} \tag{6}$$

For  $k = 0, 1, \dots, n-1$ , these two sums bear a resemblance to our original DFT: the sum over the even indices looks like a DFT of the even samples, while the sum over the odd indices looks like a DFT of the odd samples. However, we want to compute the coefficients for the entire range  $k = 0, 1, \dots, 2n-2$ . Observe that from periodicity of the exponential, we have

$$\omega_n^{\frac{1}{2}n} = e^{\frac{-i2\pi(n/2)}{n}} = e^{-i\pi} = -1 \quad \text{and} \quad \omega_n^n = e^{\frac{-i2\pi n}{n}} = e^{-i2\pi} = 1.$$

Thus,

$$\begin{aligned}
 \hat{f}_{k+n} &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{(k+n)j} + \omega_n^{(k+n)/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{(k+n)j} \\
 &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{kj} (\omega_n^n)^j + \omega_n^{k/2} \omega_n^{n/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{kj} (\omega_n^n)^j \\
 &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{kj} - \omega_n^{k/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{kj}.
 \end{aligned} \tag{7}$$

Together, (6) and (7) allow us to compute the coefficients  $\hat{f}_k$  in terms of the DFT of the even sample terms and the DFT of the odd sample terms.

We rearrange our sample vector  $\mathbf{f}$  to place even indexed samples before odd indexed samples,

$$\mathbf{f}^* = \begin{bmatrix} f_0 \\ f_2 \\ \vdots \\ f_{2n-2} \\ f_1 \\ f_3 \\ \vdots \\ f_{2n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\text{even}} \\ \mathbf{f}_{\text{odd}} \end{bmatrix}.$$

Our system now becomes  $\hat{\mathbf{f}} = F^* \mathbf{f}^*$ , where

$$F^* = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \dots & \omega_n^{n-1} & \omega_n^{\frac{1}{2}} & \omega_n^{\frac{1}{2}+1} & \dots & \omega_n^{\frac{1}{2}+(n-1)} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} & \omega_n^{\frac{n-1}{2}} & \omega_n^{\frac{n-1}{2}+n-1} & \dots & \omega_n^{\frac{n-1}{2}+(n-1)^2} \\ 1 & 1 & \dots & 1 & -1 & -1 & \dots & -1 \\ 1 & \omega_n & \dots & \omega_n^{n-1} & -\omega_n^{\frac{1}{2}} & -\omega_n^{\frac{1}{2}+1} & \dots & -\omega_n^{\frac{1}{2}+(n-1)} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} & -\omega_n^{\frac{n-1}{2}} & -\omega_n^{\frac{n-1}{2}+n-1} & \dots & -\omega_n^{\frac{n-1}{2}+(n-1)^2} \end{bmatrix}.$$

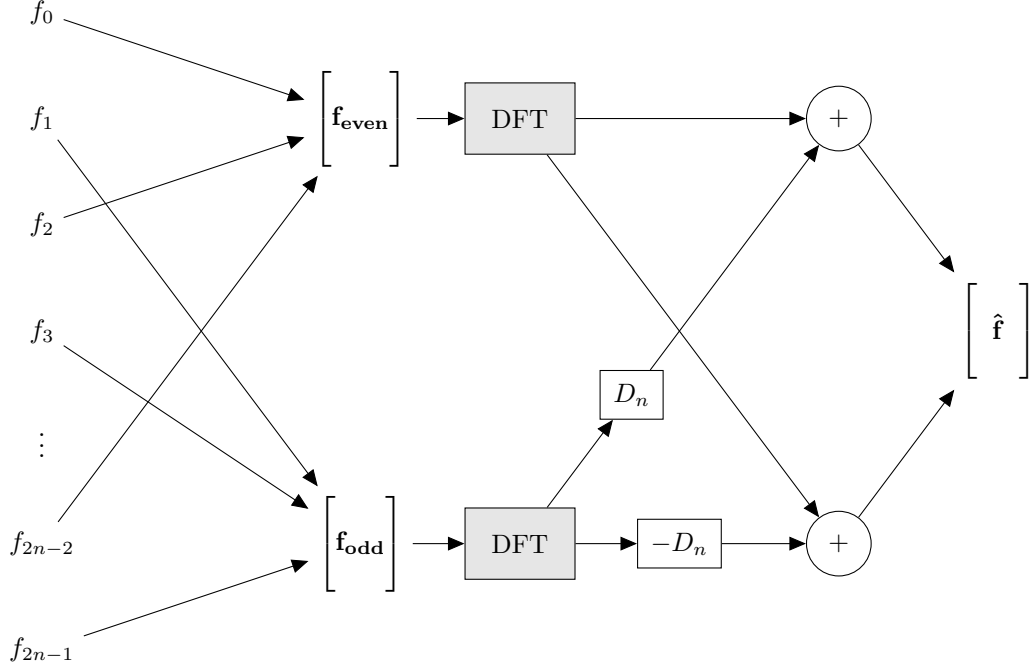


Figure 1: A single step of computation for the FFT on samples  $f$ . The input data is split into two components  $f_{\text{even}}$  and  $f_{\text{odd}}$ , over which the DFT is computed. The DFT may be computed by recursive application of the FFT to  $f_{\text{even}}$  and  $f_{\text{odd}}$ . The resulting coefficients  $\hat{f}$  are formed from a linear combination of the two DFT results.

The top half of the  $F^*$  matrix is from (6) while the bottom half is from (7). We can factor  $F^*$  as

$$\begin{aligned}
 F^* = RG &= \begin{bmatrix} 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \omega_n^{\frac{1}{2}} & \dots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & \omega_n^{\frac{n-1}{2}} \\ 1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & -\omega_n^{\frac{1}{2}} & \dots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & -\omega_n^{\frac{n-1}{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 \\ 1 & \omega_n & \dots & \omega_n^n & 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 & 1 & \omega_n & \dots & \omega_n^n \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} \end{bmatrix} \\
 &= \begin{bmatrix} I_n & D_n \\ I_n & -D_n \end{bmatrix} \begin{bmatrix} F_n & 0 \\ 0 & F_n \end{bmatrix},
 \end{aligned}$$

where  $D_n$  is the diagonal matrix multiplied before the odd part of the DFT,

$$D_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & \omega_n^{\frac{1}{2}} & 0 & \dots & 0 \\ 0 & 0 & \omega_n^{\frac{2}{2}} & \dots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \dots & \omega_n^{\frac{n-1}{2}} \end{bmatrix}.$$

The importance of this method simply cannot be understated. By exploiting this symmetry repeatedly, the time complexity is reduced to  $\mathcal{O}(n \log n)$ .

To emphasize precisely how impactful this is on real world analysis, we tested an example signal of three superimposed sine waves on both the DFT and FFT for a number of points ranging from 1000 to 1000. Each interval was tested ten times, and the average is plotted. The results, shown in ?? demonstrate how much time can be saved.

Further, in order for the FFT to take just as much time as the DFT processing a signal with 10,000 samples, the FFT must process a signal of roughly 1,000,000 samples.

Arguably just as paramount, the spatial complexity of the methods also contributes to the favorability of the fft. In order to create the matrix shown in ??? REF, we require the allocation of  $n^2$  floating points in memory. As such, our machines struggle to process anything larger than 20,000 samples when computing the DFT. In contrast, the FFT can easily compute signals with millions of sample points without issue.

### 3 Short-Time Fourier Transform

Since many signals in the real world are time-dependent, the STFT is an invaluable tool for analyzing and understanding real-world data. The FFT exists only in the frequency domain, so it can provide no information as to a signal's behavior over time. Often, we desire to know information about the time domain of a signal, namely how that signal changes over a period. To do this, we explore the Short-Time Fourier Transform, which we derive and apply in the following section.

#### 3.1 Derivation

##### 3.1.1 Continuous

With continuous data  $f$ , we analyze a given section to determine behavior in the time domain. For some window function  $g(x)$ , we can perform the analysis with an integral, where

$$\int_{\mathcal{D}} f(t) \overline{g(t-x)} e^{-2\pi i \omega t} dt \quad (8)$$

.

Note the similarity to the standard Fourier transform, which takes the form

$$\int_{\mathcal{D}} f(t) e^{-2\pi i \omega t} dt$$

. With this insight, we see more intuitively how the introduction of a window function introduces

##### 3.1.2 Discrete

Similar to section ??? we discretize this process by considering a finite number of points. As such, we have the resulting sum:

$$\sum_{n=0}^{N-1} g(n) f(n+mH) e^{-i \frac{2\pi}{N} \omega n} \quad (9)$$

### 3.2 Application

## 4 Project Overview

In this project, we investigate the Fast Fourier Transform (FFT), a method for trigonometric interpolation on equispaced points along an interval that uses symmetry to reduce computational complexity from  $\mathcal{O}(n^2)$  to  $\mathcal{O}(n \log n)$ . We begin with a derivation of the Discrete Fourier Transform (DFT), then show its drawbacks, focusing on computational complexity, and propose the FFT as an alternative. We explore the properties of the FFT, including its restrictions on the sample points, and compare its performance to that of the DFT. We continue with an exploration of the Short-Time Fourier Transform (STFT), a common tool in signal analysis and processing.

## 5 Introductory Material

### 5.1 Introduction

We introduce the Fourier transform as a tool for understanding the spectral content of a signal and show how the transform is useful in a wide variety of applications. We motivate the desire to apply the transform over sampled signals with a finite number of points.

### 5.2 Discrete Fourier Transform (DFT)

We introduce the Fourier series representations of periodic signals and its extension to the Fourier transform on aperiodic signals. We show how discrete signals over a finite time interval can be represented by a truncated Fourier series, and introduce the discrete Fourier transform (DFT) as a means to compute such a representation [5]. We show how the DFT can be computed through the solution to a linear system in  $\mathcal{O}(n^2)$  time complexity. We demonstrate the relationships between Fourier series representations in complex exponential and real forms and show how to convert between the two.

### 5.3 Fast Fourier Transform (FFT)

We then demonstrate a method to overcome  $\mathcal{O}(n^2)$  complexity and reduce to  $\mathcal{O}(n \log(n))$  time. We restrict ourselves to discrete signals with a number of points that are an integer power of 2 and rearrange the signal into odd and even data points. From here, we derive a matrix factorization that exploits sparsity and show how the structure of the matrix allows us to continually divide the signal and save computations via symmetries.

## 6 Extension: Short-Time Fourier Transform

### 6.1 Introduction

For our independent extension, we explore the Short-Time Fourier Transform (STFT), a method for analyzing signals with time-varying spectra. [1] Since many signals in the real world are time-dependent, the STFT is an invaluable tool for analyzing and understanding real-world data; furthermore, the STFT leads to useful data visualization methods, such as spectrograms and waterfall plots, which we will also investigate.

### 6.2 Derivation

We introduce the mathematical methods used to derive the STFT. We explore both the continuous[3] and discrete[2] variants, focusing on the discrete variant, its parallels with the DFT, and its applications. After introducing several different windowing functions and analyzing their advantages and shortcomings, we motivate their use for specific scenarios; we also discuss the other input parameters to the STFT algorithm, such as window size, window overlap, and number of FFT points.

### 6.3 Application

We demonstrate the applicability of the STFT by analyzing several example signals using the different methods explored in the previous section. We conclude with examples of visualizations and discussion of how these tools have contributed to various fields of science. We use magnetometer data from the EMFISIS instrument suite aboard the Van Allen Probes to demonstrate the utility of the STFT in visualizing charged particle events in the Earth’s magnetosphere. [4]

## 7 Timeline and Work Distribution

Our proposed timeline for the project is given below in Table 1.

| Date range      | Proposed work   | Lead      |
|-----------------|---|-----------|
| Oct 24 - Nov 1  | Work on project proposal  | Everyone  |
| Oct 24 - Nov 8  | Research, find sources  | Everyone  |
| <b>Nov 1</b>    | <b>Project proposal due</b>   | Everyone  |
| Nov 1 - Nov 8   | Work on theoretical foundations (Sec 2.1 - 2.3, Start 3.1)          | Elizabeth |
| Nov 8 - Nov 15  | Work more on 3.1 - 3.3  | Edward    |
| Nov 8 - Nov 15  | Write scripts for computing FFT/STFT and visuals                    | Erick     |
| Nov 15 - Nov 22 | Finish rough draft  | Everyone  |
| <b>Nov 22</b>   | <b>Project rough draft due</b>                                      | Everyone  |
| Nov 22 - Nov 29 | Minor edits over break, clean up figures while waiting for feedback | Everyone  |
| Nov 29 - Dec 6  | When we get feedback, meet and discuss revisions & next steps       | Everyone  |
| Nov 29 - Dec 6  | Finish sections not fully in rough draft, extend where possible     | Everyone  |
| Dec 7 - Dec 15  | Improve Sec 2   | Erick     |
| Dec 7 - Dec 15  | Improve Sec 3   | Elizabeth |
| Dec 7 - Dec 15  | Improve Formatting & Visuals  | Edward    |
| <b>Dec 15</b>   | <b>Project final draft due</b>                                      | Everyone  |

Table 1: Proposed project timeline.

## References

- [1] Ronald L. Alle and Duncan W. Mills. *Signal Analysis: Time, Frequency, Scale, and Structure*. John Wiley & Sons, 2004.
- [2] Andersen Ang. *Discrete STFT*. Mathematique et recherche operationnelle UMONS, Belgium. 2020. URL: [https://angms.science/doc/SP/SP\\_STFT.pdf](https://angms.science/doc/SP/SP_STFT.pdf).
- [3] Karlheinz Gröchenig. “The Short-Time Fourier Transform”. In: *Foundations of Time-Frequency Analysis*. Boston, MA: Birkhäuser Boston, 2001, pp. 37–58. ISBN: 978-1-4612-0003-1. DOI: 10.1007/978-1-4612-0003-1\_4. URL: [https://doi.org/10.1007/978-1-4612-0003-1\\_4](https://doi.org/10.1007/978-1-4612-0003-1_4).
- [4] C. A. Kletzing et al. “The Electric and Magnetic Field Instrument Suite and Integrated Science (EMFISIS) on RBSP”. In: *Space Science Review* (2012).
- [5] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Education, 1997.