Signal Parameter Estimation of Damped Sinusoidal Waveforms Using Deep Learning

by

Dawn Idoko

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, MB, Canada

Examining Committee

This thesis was examined and approved by the following examining committee on December 16, 2022:

- **Dr. Behzad Kordi** (advisor)
  Department of Electrical & Computer Engineering
  University of Manitoba
- **Dr. Greg E. Bridges** (co-advisor)
  Department of Electrical & Computer Engineering
  University of Manitoba
- **Dr. Ahmed Ashraf**
  Department of Electrical & Computer Engineering
  University of Manitoba
- **Dr. David Swatek**
  Department of Electrical & Computer Engineering
  University of Manitoba

Abstract

Sinusoids and damped signals are a fundamental part of different engineering fields. Analysis of these signals to give an accurate estimation of certain parameters such as frequency, damping factor, and phase angle is important in many engineering fields as an accurate estimation of these parameters is needed to ensure the smooth running of various processes. The need for higher levels of precision and accuracy in the signal-processing domain has resulted in the development of several algorithms based on different methods of operation. These algorithms can be divided into two classes, namely, parametric and non-parametric algorithms. The former assumes that the signal follows a particular model and estimates the signal parameters based on that assumption, while the latter makes no assumptions regarding the signal. Intuitively, the non-parametric class of algorithms seem to be a better choice for real-life applications as the model of the signal is usually unknown. However, algorithms under this class suffer from the issue of spectral leakage. Both classes of algorithms for signal analysis have their strengths as well as shortcomings.

In this thesis, the concept of using machine learning methods in signal analysis is explored. To achieve this, the DeepFreq model is extended by modifying its architecture and applying it to damped sinusoidal signals to provide an estimate of signal parameters such as frequency and damping factor. The developed algorithm can estimate the number of frequencies as well as the value of the frequencies contained in a signal waveform with an R^2

score of 0.88 even in noise levels of up to 0 dB. The algorithm's performance was evaluated using data samples of sinusoidal signals within the ISM band range of 2.4GHz to 2.65GHz. The algorithm was tested on synthetic data and data from lab experiments, and the results show that the deep learning model can perform frequency and damping factor estimation for damped multi-frequency sinusoidal signals.

Acknowledgements

I would like to express my appreciation to my advisors, Dr. Behzad Kordi and Dr. Greg Bridges, for their patience, encouragement, support, and supervision during this research. I deeply appreciate all their valuable comments and guidance.

I would also like to extend my appreciation to my thesis committee, Dr. Ahmed Ashraf, and Dr. David Swatek, for their comments and suggestions that helped improve this thesis.

I am also grateful to Amirmasoud Amirkabiri, whose invaluable guidance and direction helped me during this research work. I would like to express many thanks to the University of Manitoba High Voltage Lab (HVL), Daryl Hamelin, and Ehsan Zeyneli, for all their practical consultations, cooperation, and guidance for test, setup, design, and preparation.

Financial support from the University of Manitoba's FGS, Manitoba Hydro, and MITACS is gratefully acknowledged.

My heartfelt gratitude goes to my parents, Gabriel Idoko and Mary Idoko, and my siblings, Sonnia Idoko and Karen Idoko, for their support, encouragement, and prayers.

To my dear friends Donald Ufuah, Chinedu Nnorom, and Uche Elekwa, your endless words of encouragement and support are deeply appreciated.

Dedications

To my parents for their love and encouragement, without whom I would never have enjoyed so many opportunities.

Table of Contents

List of Figures

List of Tables

Nomenclature

DL	Deep Learning
MPM	Matrix Pencil Method
POF	Pencil of Functions
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
DFT	Discrete Fourier Transform
LTI	Linear Time Invariant
SVD	Singular Value Decomposition
FFT	Fast Fourier Transform
DTFT	Discrete-Time Fourier Transform
EMF	Electro-Motive Force
BY1	Bertocco-Yoshida Order 1
BY1-LC	Bertocco Yoshida Order 1 with Leakage Correction
IpDFT	Interpolated Discrete Fourier Transform
IDTFoE	Interpolated Discrete Fourier Transform of an Exponent
MSD	Maximum Side-lobe Decay
IQML	Iterative Quadratic Maximum Likelihood

CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
FNR	False Negative Rate
VNA	Vector Network Analyzer
ANNs	Artificial Neural Networks
IFFT	Inverse Fast Fourier Transform
ReLu	Rectified Linear Unit
LO	Local Oscillator

Chapter 1

Introduction

1.1 Background

In signal processing and communications, one of the main problems faced is frequency estimation of noisy sinusoidal waves [1]. In the field of wireless electromagnetic sensors (that is of interest to this research), the signals under consideration are transient responses of electromagnetic scatterers and are often noisy. These noisy transient responses, upon closer inspection, appear to be made up of small numbers of damped sinusoids. In the frequency domain, these damped sinusoids are poles of the Laplace transformed transient response [2]. To characterize the frequency domain responses of electromagnetic scatterers in terms of their singularities, the Singularity Expansion Method (SEM) is used. This method was introduced in 1971 by Baum and it is based on the premise that the natural modes and resonances of an electromagnetic scatterer which are characterized by poles in the complex frequency plane, or the Laplace transform s-domain may be used to describe the scatterers electromagnetic response to an incident wave [3]. It is particularly useful in the calculation or characterization of transient responses from antennas and other passive scatterers of electromagnetic radiation [4]. SEM was developed based on the observation of objects subjected to electromagnetic pulse experiments. In these experiments, the objects were exposed to a continuum of frequencies contained in a broadband transient excitation. The late time transient response of the objects was observed to be dominated by damped sinusoids which in the complex frequency plane corresponds to a pair of pole singularities [4]. The challenge lies in accurately estimating the frequency of these transient responses.

Over the years, different theoretical techniques have been proposed to tackle the challenge of frequency estimation. Some of these techniques include discrete Fourier transformation [5, 6], Phase Locked Loops [7, 8], Least Squares Method [9, 10] and Matrix Pencil Method [11, 12]. The one thing these techniques have in common is that they aim to give an accurate estimate of the frequency contained in the signal without trading their performance in terms of speed, accuracy, and efficiency.

In recent times, artificial intelligence techniques have been applied in different areas of science to automate processes, lower human error rates, increase efficiency, and improve workflows [13–18]. In speech recognition and natural language processing, these techniques have helped develop state-of-the-art systems for speech transcription, language identification, and speaker verification [19–21]. Through advances in knowledge representation, multiple knowledge sources can now be incorporated into a single framework. This development has led to improvements in speech transcription as pronunciations across different accents are better understood by these systems. Also, the introduction of statistical methods into existing system models and algorithms, has led to these systems being able to characterize certain sets of speech features such as tone, and idiolect with increased accuracy. This has led to significant improvements in language identification and speaker verification. In image detection and classification, these techniques have helped in robotic fruit harvesting, and detection of abnormalities in plant leaves [22–27]. The advancements in image processing techniques have resulted in systems that are able to perform a wide range of precision tasks that are applicable to agriculture such as, fruit detection, fruit variety classification, fruit disease detection, fruit sorting, fruit counting and fruit grading. These techniques have been proven to be non-invasive, non-destructive, and effective with high accuracy when analyzing fruit characteristics. In medical diagnosis, these techniques have been used to identify and classify different types of skin lesions, cancers, and brain tumors [28–30]. The techniques have improved discriminative representation abilities. They make use of feature maps and attention maps that help the algorithm focus on discriminative parts of the ailment. This focusing ability helps improve the accuracy of the algorithm when classifying conditions that have similar symptoms. In the field of signal processing, deep learning has been applied in different areas such as symbol detection, anti-interference, and modeling channel information [31]. Deep learning techniques have the advantage of being independent of the signal model while being able to extract useful information from the observed data. They have been used to identify time domain signals based on their modulation type e.g., Binary Phase Shift Keying (BPSK) as well as perform end-to-end wireless communication involving encoding, decoding, modulation, and demodulation of signals. They have also been used in anti-interference schemes to mitigate radio frequency (RF) interference and the results show that deep learning methods have comparable accuracy with existing mitigation algorithms. In [32–36], deep learning was used in estimating the direction of arrival (DOA) of electromagnetic waves. The use of deep learning in DOA estimation was explored because the existing signal processing methods could only work well on data samples that were uncontaminated with noise. Deep learning was used to extract certain features from noisy signals and map those extracted features to the DOA. The results showed an improved root-mean-square error (RMSE) when compared to existing methods. In [1, 37–39], the use of deep learning networks for signal parameter estimation of sinusoidal signals was introduced.

The use of artificial intelligence techniques for frequency estimation introduces the concept of learning-based techniques to signal processing which is a significant improvement over the model-based methods that previously existed. The application of learning-based techniques would imply that algorithms can be trained to identify slight variations across different signal waveforms, and depending on the robustness of their training dataset, these trained algorithms could be used on a wide range of signal waveforms.

Applying machine learning methods to signal processing gives rise to numerous possibilities including unsupervised real-time data analysis and evaluation. In this thesis, a modified algorithm for frequency estimation based on [1] is proposed. The algorithm uses a Deep learning architecture to estimate the frequency of damped sinusoidal waveforms over a frequency range of 0 Hz to 2.65 GHz. The modification produces significantly better frequency estimations for SEM waveforms than the model proposed in [1].

1.2 Research Objectives

The objective of this thesis is to explore the use of deep learning (DL), an artificial intelligent technique in the area of signal analysis. Specifically, this thesis aims to explore using DL to estimate signal parameters of damped sinusoidal signals. In this work, the signals being considered are that of a wireless passive sensor proposed in [40]. Figure 1.1 shows a schematic diagram of the operation of the wireless passive sensor. An interrogator transmits RF pulses to the resonator and receives ringback signals from the resonator. The sensor is designed

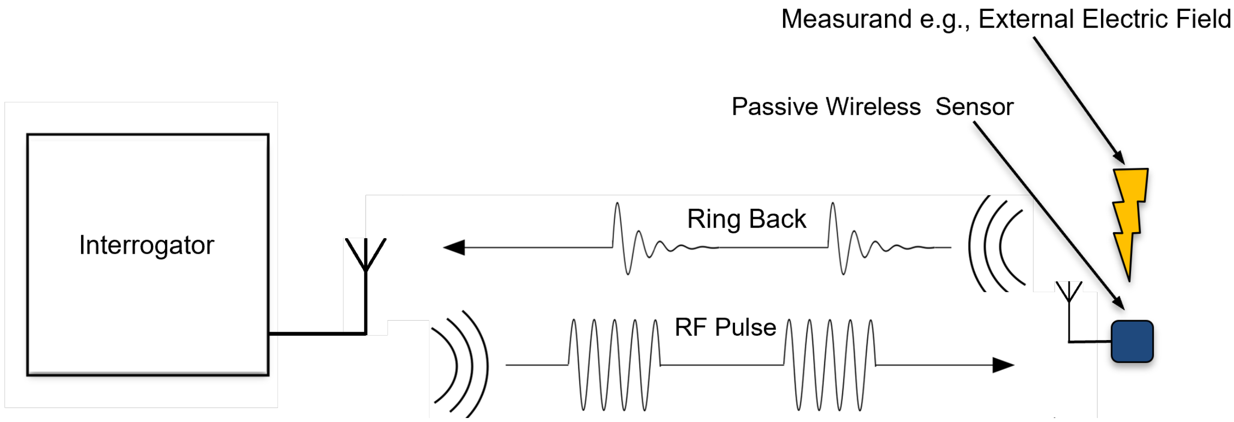


Fig. 1.1: Schematic diagram of the interrogation system for the wireless passive sensor. Adapted from [40] with permission.

to have a resonance frequency of 2.57 GHz and the interrogator sends RF pulses with a frequency of 2.60 GHz to the sensor. The 30 MHz difference in frequency helps to differentiate the ringback frequency of the sensor from that of the interrogator for accurate signal analysis. The interrogator system consists of two 10 dB horn antennas, one of which is used to transmit the RF pulses to the resonator while the other is used to receive the ringback signals from the resonator. The algorithm developed in this work, takes in data samples comprising of damped sinusoidal waveforms as inputs and gives the number of frequencies or damping factors contained in each of the waveforms and the estimated value of these frequencies or damping factors as an output.

The first part of this study involved training and testing the algorithm using artificial real-valued damped and undamped sinusoidal signals. The damped sinusoidal signals were generated using random values of damping factors and for each generated sample, a random noise level was added.

The next part of this study involved training and testing the algorithm using data samples collected from lab simulations and measurements. The lab simulations were performed using the Multisim 14.1 environment where a wireless passive sensor including a transmitter and receiver was designed. The ringback signals from this simulation were passed to the deep learning algorithm. In the lab measurements, the data samples were collected when a resonator was used as the wireless sensor. These experiments were not conducted in an anechoic chamber and as such the data samples contain noise as well as interference signals. These experimental conditions were employed to model real-world conditions. The results of these tests are reported in Chapter 4.

The third part of this study involved comparing the performance of the algorithm with that of existing signal processing algorithms. For this part, the different algorithms were tested on the same data samples and their performances were evaluated based on accuracy and computational complexity. The tests results and observations are reported in Chapter 3.

1.3 Research Contributions

The contributions of this thesis are as follows:

- The use of deep learning for frequency estimation of damped sinusoidal signals is investigated. Different architectures were considered and tested. The results from these tests were used to redesign the architecture and optimize the training parameters to improve the accuracy of the algorithms' frequency estimates.

- A robust signal model that consists of a wide range of frequencies is designed. This allows for the proposed algorithm trained using this model to estimate the frequency of sinusoidal signals over a wide range of frequencies.

- The application of the deep learning algorithm for signal parameter estimation allows for transfer learning which reduces the time required to estimate parameters of 'new' signal waveforms.

- The proposed algorithm is computationally less expensive than the traditional signal processing methods. The proposed algorithm once trained can be used on a wide range of datasets whereas the existing traditional signal processing methods which are computationally expensive need to be run on the dataset every time to estimate the signal's frequency.

- The proposed method in comparison to traditional signal processing methods can be used for real-time signal analysis as it is time efficient. The algorithm is able to determine the number of frequencies and damping factors contained within a waveform sample as well as estimate the value of these frequencies and damping factors without prior knowledge of the exact model of the signal.

- The proposed algorithm is able to estimate the signal parameters of damped noisy sinusoidal waveforms in comparison to the algorithm proposed in [1] that estimates the frequencies of undamped sinusoidal waveforms.

- The proposed algorithm was trained using double exponential signal waveforms that have an envelope similar to the signals obtained from a wireless interrogation system and the results are presented.

- The proposed algorithm has a significantly reduced number of layers in comparison to [1], and gives accurate results within a shorter training time.

- The proposed algorithm provides signal parameter estimates for multiple components of damped multi-sinusoidal signals whereas existing traditional signal processing methods are able to provide single signal parameter estimates for damped sinusoidal signals.

- The application of deep learning to estimate the damping factors of damped multi-sinusoidal signals was also explored and the results are discussed in Chapter 4 of this work.

1.4 Thesis Outline

This thesis is divided into six chapters as defined below:

Chapter 1: Introduction, motivation, research objectives, and research contributions are presented.

Chapter 2: Background on different signal processing algorithms for frequency estimation including a brief history on the Matrix Pencil method algorithm and other traditional signal processing methods is presented. The advantages and shortcomings of the algorithms are discussed in this chapter.

Chapter 3: The architecture of the deep learning algorithm for frequency estimation is presented in this chapter. The modifications made to the algorithm are discussed and the different modules of the algorithm's architecture are explained.

Chapter 4: The performance of the deep learning algorithm is evaluated in this chapter. The data-sample preparation process for training, validation and testing is described. The structure of the wireless interrogation system simulated with Multisim 14.1 is explained and the proposed algorithm is evaluated using waveform samples from the simulation. Results from tests run on data-samples collected from the wireless passive sensor are presented. Also, the results from the comparison between the proposed algorithm and traditional signal processing algorithms for signal parameter estimation are discussed.

Chapter 5: Conclusions of this thesis and potential future research work are discussed in this chapter.

Chapter 2

Literature Review

Signal processing methods can be classified into two types: parametric methods and non-parametric methods [41]. For spectral analysis, parametric methods are based on models while non-parametric methods are based on the Fourier transform. The parametric methods are able to provide high resolution estimates and are suitable for data samples that are relatively short. However, these methods are computationally more expensive than the non-parametric methods and typically require an estimation of the model orders i.e., they require that the exact model of the signal and the disturbances be known. The Prony algorithm is an example of a parametric method [41]. The algorithm extracts information from a signal under analysis and uses this information to define the signal as a sum of damped complex exponentials. This definition is used to estimate the signal's parameters such as its frequency, its phase, its amplitude, and its damping factor. On the other hand, non-parametric methods which are based off the FFT algorithm, have low computational costs when compared to parametric methods and can easily be implemented because they do not require that the exact model of the signal be known. These methods are considered practical for real-life applications as the signal of interest may have components that cannot be precisely modeled such as an unknown number of sinusoids, an unknown number of frequencies, noise with an unknown distribution, amongst others. These components invariably affect the accuracy of the parametric methods which are also computationally expensive as much of their estimation depends on the computation of the pseudo-inverse of a matrix [42]. However, non-parametric methods are not without their shortcomings as these methods have two main challenges; spectral leakage and picket fence effects [43]. Spectral leakage is caused by the truncation of a signal to a finite length. FFT and DFT algorithms work on the assumption that the signal being analyzed is continuous and repeats itself after the measured time interval. This leads to errors in the assumed signal when the length of the DFT is not an integer product of the period of the signal. In multi-frequency signals, when the length of the DFT is not an integer multiple of all the components of the signal, it results in a smeared spectrum caused by interference amongst the sinusoidal components of the multi-frequency signal. Spectral leakage is not the same as aliasing, which is caused by sampling a signal at a rate lower than the Nyquist rate. On the other hand, the picket fence effect is the fall off in frequency between two frequency bins. Frequency bins are intervals between samples in the frequency domain i.e., they are storage bins for spectral energy. The frequency is only accurate at these specific, regular intervals (bins). The parametric and non-parametric methods of signal processing are discussed in detail below.

2.1 Non-parametric Signal Processing Algorithms

2.1.1 Windowed Interpolation Algorithms

Previously, the well-established methods of determining parameters of continuous signals involved counting the zero-crossings of the signal to get the frequency, peak to peak measurement to get the amplitude and window averaging to get the dc value of the signal. These methods had their shortcomings as they were most effective when moderate accuracy of 1% or lower was desired [44].

Over the years, the need for fast and computationally efficient algorithms to process distorted signals has given rise to the development of a variety of signal-processing algorithms. A number of these algorithms are based on the fast or discrete Fourier transform (FFT or DFT), which is a good way to perform power spectrum analysis and filter simulation [45]. FFT is a computationally-efficient method of determining the DFT of a time series [46]. The DFT is defined by [46]

$$A_n = \sum_{m=0}^{N-1} (S_m)\, e^{(-2\pi j/N)\times nm}, \qquad 0 \leq n \leq N-1 \tag{2.1}$$

where A_n is the n^{th} coefficient of the DFT, S_m is the k^{th} sample of the time series and N denotes the number of samples in the time series. The DFT of a time series is a reversible mapping operation that defines a time series and it is closely related to the Fourier transform of the continuous waveform that samples were taken from to form the time series. However, a shortcoming of using the DFT or FFT is spectral leakage which allows a single tone signal to be spread across multiple frequencies making it hard to distinguish the actual frequency of the signal. Although spectral leakage is affected by the sampling period of the signal, it is important to note that it is not caused by it [47]. Another point worthy of note is that the phase of the signal impacts the accuracy of the frequency and damping factor estimates for signals with a small number of cycles. Also, the phase that minimizes systematic error for frequency estimation does not necessarily minimize the error for damping factor estimation [48].

To solve the challenge of signal parameter estimation due to spectral leakage, a method based on the interpolation of DFT points on a spectrum was used to estimate signal parameters in the presence of spectral leakage [44]. Interpolation DFT algorithms compute the sample of the continuous spectrum of the DFT between frequency bins [48]. The purpose of DFT interpolation is to determine the value of frequency correction and from that value the angular frequency of the signal can be obtained [49]. However, it was found that only short-range spectral leakage (i.e., spectral leakage caused by positive frequencies [43]) could be efficiently countered by the interpolation scheme [45]. The method proved to be impractical in countering the effects of long-range spectral leakage (i.e., spectral leakage caused by negative frequencies [43]) when estimating signal parameters [45].

In [45], a new method to characterize the parameters of multifrequency signals was introduced. The multifrequency signals considered are of the form represented by [45]

$$s(m\Delta t) = \sum_{k} A_k e^{2\pi j f_k m \Delta t}, \qquad 0 \leq m \leq N-1. \tag{2.2}$$

In this method, the signals are weighted using a Hanning window before the discrete Fourier transform is calculated. An example of a Hanning window is shown in Fig. 2.1. The Hanning window is described as [45]

$$w(m\Delta t) = 0.5\left(1 - \cos(\frac{2\pi m\Delta t}{T})\right), \qquad 0 \leq n \leq N-1, \tag{2.3}$$

and the DFT of the signal in (2.2) is defined as [45]

$$S(n\Delta f) = \sum_k A_k D_N((n\Delta f - f_k)T) \tag{2.4}$$

where

$$D_N(\theta) = \frac{\sin(\pi\theta)}{N sin(\pi\theta/N)} e^{(\frac{-j\pi\theta(N-1)}{N})} \tag{2.5}$$

and $\Delta f = T^{-1}$ where $T = N\Delta t$. The tapered time function which is calculated by applying the Hanning window to the multifrequency signal i.e. $s(m\Delta t) \cdot w(m\Delta t)$ is given by [45]

$$S_H = \sum_k A_k B_N((n\Delta f - f_k)T) \tag{2.6}$$

where

$$B_N(\theta) = 0.5(D_N(\theta) - 0.5(D_N(\theta+1) + D_N(\theta-1))). \tag{2.7}$$

In (2.4), D_N represents a Dirichlet kernel which is an expression of the partial sum of the Fourier series of a function. In the frequency domain, (2.6) can be written as

$$S_H(n\Delta f) = 0.5[S(n\Delta f) - 0.5(S((n+1)\Delta f) + S((n-1)\Delta f))]. \tag{2.8}$$

An algorithm based on (2.8) is of a lower computational complexity than an algorithm based on (2.6). For a function, the summation of three consecutive Dirichlet kernels that have different phases cancels out the side-lobe structure of the function [47]. The Hanning

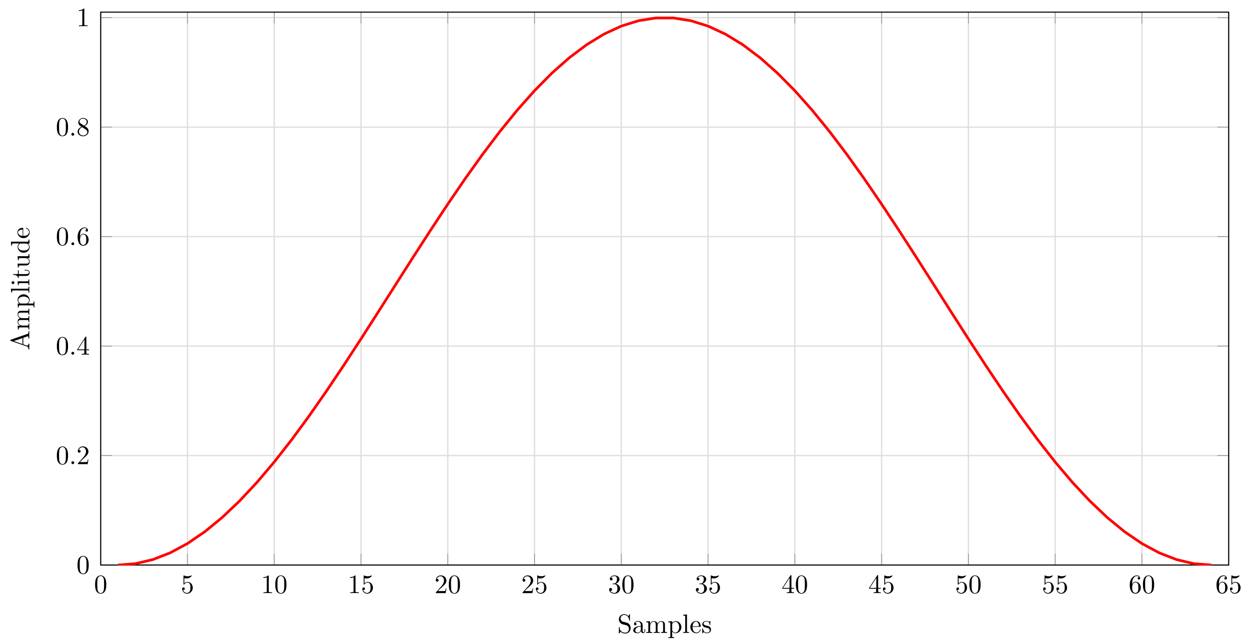


Fig. 2.1: The Hanning window in the time domain.

window acts as a tapered window which is used to remove discontinuity in the signal. It is applied in the time domain to improve the properties of the signal in the frequency domain. Tapered windows help remove the effect of long-range spectral leakage. After windowing, the DFT of the signal is calculated and the interpolation method is used to counter the effect of short-range spectral analysis. The method effectively combines the advantages of windowing with that of interpolation resulting in an improved efficiency in estimating signal parameters. The Hanning window used in the method was adopted for its simplicity and because leakage across the boundaries of the window is negligible except for frequencies close to 0 and $f_{\max}$. The proposed method, when compared with other interpolation algorithms that existed at the time has a higher accuracy and gives a better estimate of the fundamental frequency of signals in the case of additive white noise [45]. The method has its short comings in that it is only viable for signals whose noise variance does not exceed 3% and, the frequencies of

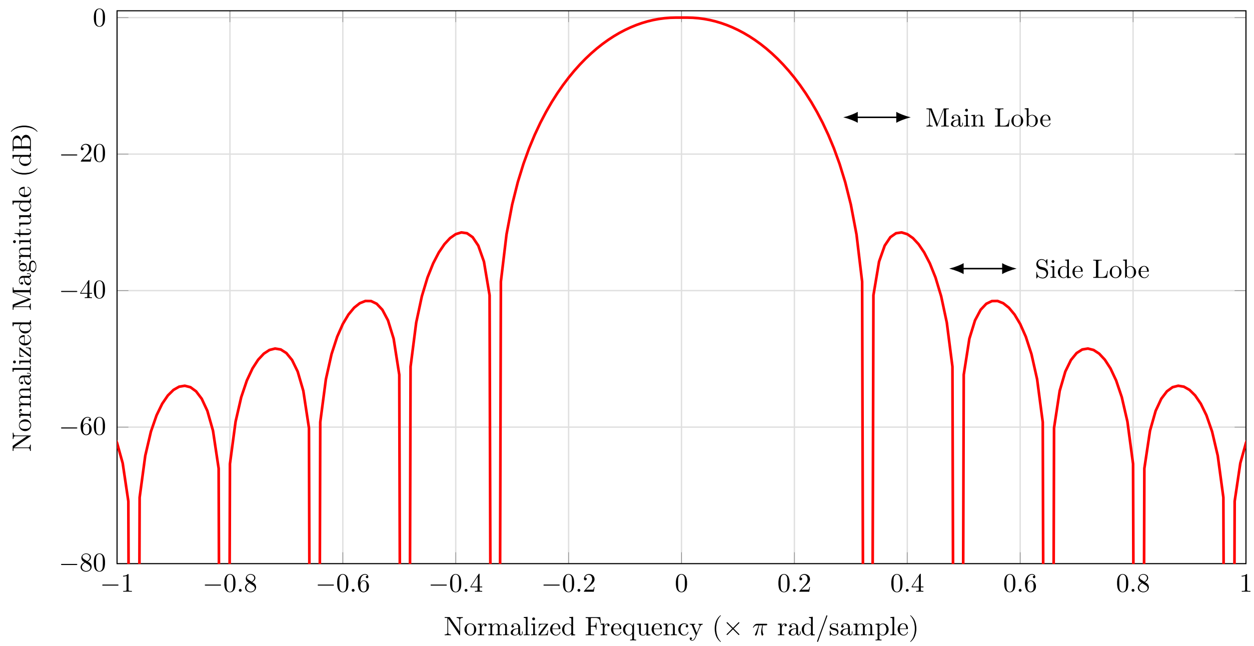


Fig. 2.2: The Hanning window in the frequency domain.

the signal have to be spaced sufficiently.

In [50], the Rife and Vincent family of weighting functions were used to remove the effect of short-range spectral leakage and the results were further examined under noisy conditions. The Rife and Vincent weighting functions were specifically chosen in this approach because they improve the accuracy of measurements over a wide range of applications. Rife and Vincent in their work [51] discussed three classes of weighting functions to be used for windowing. The Class I weighting functions provide minimum high-order side-lobe amplitude for large frequencies. The Hanning window falls under this category. Figure 2.2 shows the Hanning window in the frequency domain. The central peak is known as the main-lobe while the other peaks at regular intervals are called side-lobes. Each sidelobe also represents a frequency bin. Class II weighting functions provide minimum main-lobe width at the expense of higher side-lobe amplitudes. These weighting functions are the result of applying

Taylor approximations to the Dolph-Tchebycheff functions. They give good results when the spectral components of the signal under evaluation are very close. However, the wider side-lobe width increases the distortion of the signal in the frequency domain. Class III weighting functions combine the desired properties of Class I and Class II weighting functions. They have a narrower main-lobe width when compared with Class I algorithms, but they have slightly higher side-lobes. The slightly narrower main-lobe widths of Class III weighting functions helps them provide a better resolution for small frequency separated tones than Class I weighting functions. They also provide better resolution for weighting large frequency separated tones than Class II weighting functions.

An advantage of the Rife and Vincent family of weighting functions is that they allow simple interpolation algorithms that counter the effect of short-range spectral leakage to be obtained. These algorithms help make accurate estimations of sinusoidal signal parameters such as frequency, magnitude, and phase with little to no increase in computation time. This method of using the Rife and Vincent family of weighting functions works on the assumption that the minimum distance between two continuous frequencies is large enough and as such, the harmonic interference caused by long-range spectral leakage is completely removed by the windowing operation. When additive Gaussian noise was introduced to the signal, it was found that for noise levels less than 40 dB, the difference in the error curves for the performance of the three classes was more evident than for practically noiseless signals with noise levels greater than 100 dB [50].

The algorithms discussed in [50] outperforms those proposed by [44], specifically the Class I and Class II algorithms which provide optimal filtering with the least computational effort. The method when used with interpolation algorithms gives root mean square (rms) voltage estimates accurate to within 0.1% and frequency estimates accurate to within 0.01%

even at relatively high noise levels. This proves that the weighting functions used to combat the effect of long-range spectral leakage play a role in the accuracy of the signal parameters estimated.

2.1.2 Yoshida's Algorithm

The Yoshida algorithm, originally designed for an inverted torsion pendulum, was proposed in 1981 [49]. The algorithm was applied to damped sinusoidal signals containing multiple frequencies which were the result of an electro-magnetic force (EMF) generated in a coil [52]. The generated damped sinusoidal signals were assumed to have three distinct group of frequencies. The first group represented the main signal obtained from the damped oscillation. The second group represented a sum of the parasitic motions of the signal which consist of several harmonic motions whose angular frequencies are far from the angular frequency of the signal in the first group. The third group represented every other frequency not considered in group one and group two, and all forms of noise fall in this category. A DFT of the damped sinusoidal signal was calculated in a finite time interval and the four largest DFT bins were used to estimate the signal parameters according to [52]

$$R = \frac{X(k-2) - 2X(k-1) + X(k)}{X(k-1) - 2X(k) + X(k+1)} \tag{2.9}$$

where the damping factor α and the frequency ω_o are calculated using the following equations [49]:

$$\alpha = \frac{2\pi}{N}\, \mathrm{Im}\,(\frac{-3}{R-1}) \qquad \textit{and} \qquad \omega_o = \frac{2\pi}{N}\, \mathrm{Re}\,(k - \frac{3}{R-1}). \tag{2.10}$$

In the above equations, k is the index of the DFT bin with the largest magnitude, X is used to denote the DFT bins, and R represents the ratio calculated using the four largest DFT bins. This ratio is used to estimate the signals damping factor and frequency according to (2.10). The results showed that the frequency components of the waveform analysis were well separated. Also, there was a significant improvement in the signal to noise ratio and parasitic motion disturbances were suppressed.

2.1.3 Bertocco's Algorithm

In [53], a frequency domain interpolation algorithm was proposed. The proposed algorithm can estimate parameters for signals modelled as linear combinations of damped sinusoids. In multi-frequency sinusoidal signals, spectral leakage is a function of not only the effect of the spectrum for negative frequencies but also each sinusoid is a leakage source for the other sinusoids. The interpolation algorithms previously discussed that involved the use of windowing and interpolation considered only the case of undamped sinusoids [54]. For damped sinusoids, those methods are no longer applicable. The Bertocco's algorithm performs a Discrete-Time Fourier Transform (DTFT) of the signal to get its frequency domain estimation after which a DFT interpolation scheme is applied. The DFT interpolation scheme uses the two largest DFT bins to estimate the signal parameters according to [53]

$$R = \frac{X(k \pm 1)}{X(k)} \tag{2.11}$$

where k is the index of the frequency bin with the highest magnitude. The damping factor α and the frequency q are calculated by [49]

$$q = \frac{N}{2\pi} \arg\left(\frac{1-R}{R \times e^{\pm(\frac{-j2\pi}{N})}}\right) \qquad \text{and} \qquad \alpha = \ln \frac{1-R}{R \times e^{\pm(\frac{-j2\pi}{N})}} \tag{2.12}$$

The main difference between the DTFT and the DFT is that the DTFT is the Fourier transform of a discrete time signal and its output is continuous and periodic whereas the DFT is a frequency domain sampled version of the DTFT output [55]. The accuracy degree of the proposed algorithm is very close to those reported for undamped signals in the case where the damping factor $(\alpha) \leq 0.5$. Under noisy conditions, Bertoccos's algorithm was compared to other time-domain approaches and it was found that it provides poor signal parameter estimates [49]. The computational efficiency of this algorithm is significantly better and can be used for real-time measurements in conditions where the spectral resolution of the signal is not a major concern. It is important to note that although the Yoshida algorithm precedes the Bertocco algorithm by thirteen years, the Yoshida algorithm outperforms the Bertocco algorithm [49]. One major shortcoming of the Bertocco algorithm is that it becomes rather impracticable for signals consisting of multiple frequencies because of the harmonic interference from spectral leakage [56].

2.1.4 Bertocco-Yoshida Algorithm

In the Bertocco-Yoshida algorithm (BY1), concepts from the Yoshida algorithm and the Bertocco algorithm are combined to form a more robust algorithm [49]. In the Yoshida method, the second-order differences of the DFT bins are used in the signal parameter estimation while in the Bertocco method, the zero-order differences of the DFT bins are used.

Intuitively, it is noticed that the first-order differences are missing across both implementations. The BY1 algorithm explores the use of first-order differences in estimating the signal parameters according to [49]

$$R = \frac{X(k-1) - X(k)}{X(k) - X(k+1)}. \tag{2.13}$$

The damping factor α and frequency ω_o are calculated according to [49]

$$\alpha = -\mathbb{R}(\ln y) \qquad \text{and} \qquad \omega_o = \mathbb{I}(\ln y) \tag{2.14}$$

where

$$y = e^{j\omega_k} \frac{p - R}{p(e^{\frac{-j2\pi}{N}}) - R(e^{\frac{j2\pi}{N}})} \qquad \text{and} \qquad p = \frac{-e^{-j\omega_k} + e^{-j\omega_{k-1}}}{-e^{-j\omega_{k+1}} + e^{-j\omega_k}}. \tag{2.15}$$

The amplitude A and phase ϕ are given by [57]

$$A = \left| \frac{2X_k}{c} \right| \qquad \text{and} \qquad \phi = \arg \frac{2X_k}{c} \tag{2.16}$$

where X_k is the k^{th} frequency bin and $c = \frac{1-y^N}{1-ye^{-j\omega_k}}$. In this algorithm, the three frequency bins with the largest magnitudes are used. The proposed algorithm can estimate the frequency of damped sinusoids with a higher accuracy and a lower standard deviation (STD) level than the Yoshida algorithm [49] and it has better noise immunity in estimating the frequency and damping factor of sinusoids as well as lower systematic errors than the Bertocco algorithm [48].

2.1.5 Bertocco Yoshida Algorithm with Leakage Correction

In [58], the interpolated discrete Fourier transform of an element (IDTFoE) algorithm was proposed. This algorithm is also referred to as the Bertocco-Yoshida order 1 with leakage correction (BY1_LC) algorithm [42]. This algorithm extends the BY1 algorithm in that it can reduce the effect of spectral leakage through iterative estimation of the signal parameters and subtraction of the negative part of the spectrum from these estimates. The algorithm makes use of the rectangular time window which offers the highest level of noise robustness. The algorithm can analyze aperiodic signals that are assumed to be of exponential form and estimate the frequency, phase, amplitude, and damping factor of these signals in several steps. The signal is first sampled and then transformed into the frequency domain using DFT. The frequency bins are then selected and a DFT interpolation is applied on the bins. The result of this interpolation is used to estimate the signal parameters. The effect of leakage from the negative frequencies of other spectral lines is then subtracted from each of these frequency bins, and a new estimate for the signal parameters is found. For the frequency bin X_k, subtracting the effect of the negative frequencies of other spectral lines is given by [42]

$$X_k^{\text{new}} = X_k - \frac{A}{2} e^{-j\phi} \frac{1 - y^{*N}}{1 - y^* e^{-j\omega_k}} \tag{2.17}$$

where $A \geq 0$ is the signal's amplitude and y^* represents the complex conjugate of the variable y which is defined in (2.15). This process of subtracting leakage from the frequency bins and estimating new signal parameters is repeated until the estimated signal parameters are within an acceptable error range. The method boasts of improved accuracy in estimating the signal parameters and excellent convergence rates since approximate values are first obtained using the interpolated DFT before leakage correction is applied.

2.1.6 Two or Three-Point Interpolated DFT Algorithm

Over the years, research has been carried out on how to improve the accuracy of signal parameter estimates when performing signal analysis using interpolated discrete Fourier time (IpDFT) algorithms. Specifically in the area of using two or three-point IpDFT algorithms. In the classical three-point IpDFT algorithms, a maximum side-lobe decay (MSD) window is used for the signal analysis. The three frequencies with the largest magnitude are selected and used to estimate the signal's frequency. In [59], the classic three-point algorithm method is further modified to reject the effect of spectral leakage on the estimation accuracy of the algorithm. This modification is achieved by introducing weighted coefficients (α and β) in the estimation process. The modified three-point algorithm is represented by [59]

$$R = \frac{|X(k)| + \alpha|X(k-1)|}{|X(k)| + \beta|X(k+1)|}. \tag{2.18}$$

The coefficients (α and β) depend on the number of acquired sinusoid cycles. The resulting algorithm has an improved accuracy when there is a significant amount of spectral leakage due to the signal image component. The method outperformed other IpDFT-based methods for signal to noise ratio (SNR) levels of 30 dB or higher and when the acquired sinusoidal cycles for the signal under analysis is small.

In [43,60], a three-point IpDFT algorithm was proposed to estimate the signal frequency and damping factor by completely removing the effect of spectrum leakage due to negative frequencies. It achieves this by using the three frequency bins with the highest magnitude specifically, the part of the bins corresponding to its negative frequency are used. The proposed algorithm was coined I3PNDFT. The method was compared against other existing IpDFT algorithms such as the BY1, Bertocco and IDFToE algorithms and it provided a higher accuracy than all the listed algorithms even when a small amount of signal samples were used [60]. The algorithm was further improved on to reduce its computational complexity in [61]. Here, the negative part of the two frequency bins with the highest magnitude was used. The resulting algorithm was compared against other existing IpDFT algorithms such as the BY1, Bertocco, and IDFToE algorithms and the I3PNDFT algorithm and it was found to have results very similar to the I3PNDFT algorithm while outperforming all the other IpDFT algorithms.

2.2 Parametric Signal Processing Algorithms

2.2.1 Matrix Pencil Method

The Matrix Pencil Method (MPM) which is based on the Prony method dates as far back as 1985. The Prony Method itself was developed in 1795 by Gaspard Riche de Prony [62]. It is similar to the Fourier Transform method in that it allows for the estimation of frequency, magnitude, phase, and damping factors by building a series of damped complex exponentials through the information it extracts from uniformly-sampled signals.

The Prony method was applied in [63] to the transient response of an electromagnetic scatterer [64]. Prior to this, a conventional and iterative method was used to find the poles of the system by finding the determinant of the zeros of the system in the complex plane. The shortcoming of this conventional and iterative method is that it is unable to extract the poles of the signal from the transient response. The Prony method unlike the conventional method can systematically extract the complex poles and residues from a transient response.

It does this by solving two matrix equations and solving for the zeroes of an N^{th} degree polynomial. To find N desired poles and residues using the Prony algorithm, at least $2N$ equally spaced transient data samples are needed [63]. The Prony method works on the premise that the impulse response of electromagnetic scatterers can be represented by the sum of residues multiplied by exponentially damped sinusoids. The observed time response of an electromagnetic scatter ($d(t)$) is given by [65]

$$d(t) = x(t) + n(t) = \sum_{i=1}^{M} R_i e^{s_i t} + n(t), \qquad s_i = -\sigma_i + j\omega_i \tag{2.19}$$

where $x(t)$ is the signal, $n(t)$ is the noise associated with the signal, R_i represents the residues or complex amplitudes of the signal, σ_i represents the damping factors of the signal, ω_i represents the angular frequencies of the signal and M represents the number of poles used to estimate the sequence. The downside of the Prony's method is that it does not work for noise contaminated data and non-equispaced data samples even though it is a straightforward process to determine useful information from the signal [66].

To address the shortcomings of the Prony's method, the Pencil of Functions (POF) method was developed. The Pencil of Functions method was first suggested in 1974 by V.K. Jain [67]. It is based on the premise that a mathematical entity known as a "pencil of function" is produced when a pair of linear functions are combined by a parameter. Just like the Prony method, it can find the poles of a signal in two steps. The first step involves solving a matrix equation and the second step involves finding the root of a polynomial [11]. The method is generally insensitive to noise and can find the desired poles and residues from the output of a system when a known input is given [66]. The downside of this method is that it is computationally expensive.

The Pencil of Functions method was further improved by using the estimation of signal parameters via rotational invariance techniques (ESPRIT) approach. The ESPRIT algorithm assumes that an array of antennas comprises of two identical sub-arrays that may overlap [68]. The ESPRIT algorithm exploits the "rotational invariance of the underlying signal subspace induced by the translational invariance of the sensor array" [69]. This means that the algorithm works based on the premise that the signal stays the same no matter how it is oriented in space since the array of antennas produces the same output regardless of how its input is shifted. This premise is based on the assumption that the requirement for rotational invariance in space is satisfied because the sub-arrays are identical and the spacing between them is known. The ESPRIT algorithm, which can produce signal parameter estimates as generalized eigenvalues, was applied to the sinusoidal sequence problem, and it resulted in the Matrix Pencil Method [12].

The Matrix Pencil Method, also known as the Generalized Pencil of Function method, was suggested in [11] - [12]. The MPM models a given signal dataset as a sum of complex exponentials [65] according to

$$d(kT_s) = x(kT_s) + n(kT_s) \approx \sum_{i=1}^{M} R_i q_i^k + n(kT_s), \qquad 0 \leq k \leq N-1 \tag{2.20}$$

where $q_i = e^{s_i T_s} = e^{\sigma_i + j\omega_i}$, T_s is the sampling period and k is the index number of the data samples. The time variable t in (2.19) is replaced by kT_s in (2.20) after the data is sampled. The poles of the sampled signal are the generalized eigenvalues of the matrix pencil equation [11] and can be estimated from the singular values of one of the linear functions of the matrix pencil. The model is considered valid because the system generating the dataset is treated as a linear time invariant (LTI) system which implies that the eigenfunctions of the operator are expressed as decaying exponentials from which the poles of the system can be derived.

For a noiseless dataset, a pencil parameter L is chosen such that $M \leq L \leq N - M$. The data matrix D is defined as

$$[D] = \begin{bmatrix} d_0 & d_1 & \cdots & d_L \\ d_1 & d_2 & \cdots & d_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N-L-1} & d_{N-L} & \cdots & d_{N-1} \end{bmatrix}_{(N-L)\times(L+1)} \tag{2.21}$$

which can be written as [65]

$$\begin{aligned} [D] &= [e_1, D_1] \\ &= [D_2, e_{L+1}] \end{aligned} \tag{2.22}$$

where e_i represents the i^{th} column of the data matrix. The matrices D_1 and D_2 can be written as [65]:

$$\begin{aligned} [D_1] &= [Q_1][R][Q_0][Q_2] \\ [D_2] &= [Q_1][R][Q_2] \end{aligned} \tag{2.23}$$

where

$$[R] = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_M \end{bmatrix}_{M\times M} \tag{2.24}$$

$$[Q_0] = \begin{bmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_M \end{bmatrix}_{M \times M} \tag{2.25}$$

$$[Q_1] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ q_1 & q_2 & \cdots & q_M \\ \vdots & \vdots & \ddots & \vdots \\ q_1^{N-L-1} & q_2^{N-L-1} & \cdots & q_M^{N-L-1} \end{bmatrix}_{(N-L) \times M} \tag{2.26}$$

$$[Q_2] = \begin{bmatrix} 1 & q_1 & \cdots & q_1^{L-1} \\ 1 & q_2 & \cdots & q_2^{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & q^M & \cdots & q_M^{L-1} \end{bmatrix}_{M \times L} . \tag{2.27}$$

The matrix pencil equation is defined by [65]

$$[D_1] - \lambda[D_2] = [Q_1][R]([Q_0] - \lambda[I])[Q_2], \qquad \text{provided} \qquad M \leq L \leq N - M \tag{2.28}$$

Assuming, $\lambda = q_i$, where $i = 1, 2, \ldots M$ in (2.28), the matrix pencil would be of rank $M - 1$, which implies that the generalized eigenvalues of the matrix pair D_1 and D_2 are q_i. Therefore (2.28) can be written as [65]

$$[D_1][\theta] = q_i D_2[\theta] \tag{2.29}$$

where θ is the generalized eigenvector. The eigenvalues of the system are computed by multiplying one of the pencil matrices by the Moore-Penrose pseudo inverse of the other, and the poles are calculated directly from the eigenvalues as a one-step process [65] using the equivalent form of (2.29) given by

$$([D_2]^{\dagger}[D_1] - q_i[I])[\theta] = 0 \tag{2.30}$$

where $[D_2]^{\dagger}$ is the Moore-Penrose pseudo-inverse of $[D_2]$ which is given by: $D_2]^{\dagger} = ([D]^H[D]^{-1}[D]^H)$ where the superscript H denotes the complex conjugate transpose of a matrix. The Moore-Penrose inverse of a matrix is also known as the generalized inverse of a matrix. It is the shortest length least squares solution to a system of linear equations that lacks a solution. Once the exponents q_i have been calculated using the Moore-Penrose inverse of the matrix, the residues at the poles can now be computed using [65]

$$[Y] = [Q][A] \tag{2.31}$$

where

$$[Y] = \begin{bmatrix} d_{(0)} \\ d_{(1)} \\ \vdots \\ d_{(N)} \end{bmatrix}_{(N+1)\times 1} \qquad [Q] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ q_1 & q_2 & \cdots & q_M \\ q_1^2 & q_2^2 & \cdots & q_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ q_1^N & q_2^N & \cdots & q_M^N \end{bmatrix}_{(N+1)\times M} \qquad [A] = \begin{bmatrix} R_{(1)} \\ R_{(2)} \\ \vdots \\ R_{(M)} \end{bmatrix}_{(M)\times 1} . \tag{2.32}$$

The residues at the poles can be calculated by [65]

$$[A] = [Q]^{\dagger}[Y] = [Q]^H[Q]^{-1}[Q]^H[Y] \tag{2.33}$$

which is the least square solution of (2.32) using the Moore-Penrose inverse of the matrix Q.

For a noisy dataset, the procedure is slightly different and involves using Singular Value Decomposition (SVD) for efficient signal noise filtering. The pencil parameter L, for noisy data samples is chosen to be between $\frac{N}{3}$ to $\frac{N}{2}$ where N is the number of samples. An SVD of the data matrix is computed and errors due to noise are reduced by selecting the M largest singular values of the decomposition and defining submatrices corresponding to those singular values which are used to reconstruct the matrix pencil algorithms. The SVD of the data matrix D is defined as [65]

$$[D] = [U][\Gamma][V]^H \tag{2.34}$$

The matrices $[U]$ and $[V]$ are unitary matrices of size $(N-L)\times(N-L)$ and $(L+1)\times(L+1)$ respectively, while the matrix $[\Gamma]$ is a diagonal matrix of size $(N-L)\times(L+1)$ with the singular values of the data matrix $[D]$ in descending order. When dealing with noiseless data, the data matrix $[D]$ would have M non-zero singular values. However, for noisy data, the data matrix $[D]$ would have several small non-zero singular values and the M largest singular values are chosen by a ratio comparing each singular value to the largest one given by [65]

$$\frac{\sigma_i}{\sigma_{max}} \approx 10^{-f} \tag{2.35}$$

where f is the number of accurate significant digits of the data. The noise in the given data is suppressed by defining submatrices denoted by $(*)$ according to the following [65]

$$[U^*] = [U(:, 1:M)] \tag{2.36}$$

$$[V^*] = [V(:, 1 : M)] \tag{2.37}$$

$$[\sum^*] = [\sum(1 : M, 1 : M)] \tag{2.38}$$

$$[D^*] = [U^*][\sum^*][V^*]^H. \tag{2.39}$$

The matrices $[D_1]$ and $[D_2]$ are defined from the submatrix D^* which is noise filtered as [65]

$$[D_1] = [U^*][\sum^*][V_1^*]^H \qquad and \qquad [D_1] = [U^*][\sum^*][V_2^*]^H \tag{2.40}$$

where V_1^* is equal to V^* without its first row and V_2^* is equal to V^* without its last row. The eigenvalues and poles are computed in the same way as in the noiseless case. It is important to note that if the smallest singular value of the given data is smaller than the round-off error for the data, more data samples must be acquired.

The Matrix Pencil Method is generally more insensitive to noise than the Prony method. It approximately reaches the Cramér-Rao bound, which implies that it achieves the lowest possible mean squared error in comparison to other techniques and no other technique can perform better in estimating the poles of a signal in a noisy environment. The Matrix Pencil Method is computationally more efficient than the previous signal processing methods discussed in [11].

2.2.2 Steiglitz McBride Algorithm

The Steiglitz McBride (STMB) algorithm works by identifying a linear system from its input and outputs by minimizing the mean-squared error (MSE) between the system, and its outputs [70]. The STMB algorithm has two modes: mode-1, and mode-2. It is an iterative algorithm used to compute the pole-zero model. This algorithm was proposed in 1965 [70] as a system identification technique specifically for control systems where the examination of the observed system signals provide a good understanding of the system's operation in a changing environment. The algorithm uses an iterative procedure to minimize the MSE, its mode of operation is like the Kalman linear regression solution [71] except that it prefilters the inputs and outputs. Equations (2.41 - 2.45) below explain the process. For N signal samples, the following matrix equations are formed [42]

$$s = Sc + e, \quad s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{N-1} \end{bmatrix} \quad S = \begin{bmatrix} s_1 & s_0 \\ s_2 & s_1 \\ \vdots & \vdots \\ s_{N-2} & s_{N-3} \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad e = \begin{bmatrix} e_2 \\ e_3 \\ \vdots \\ e_{N-1} \end{bmatrix} \tag{2.41}$$

where

$$c_1 = 2\cos(\omega)e^{-\alpha} \qquad \text{and} \qquad c_2 = -e^{-2\alpha}. \tag{2.42}$$

The vector s and the matrix S are known variables of the equation in (2.41), they are made up of data samples, while the vectors c and e are unknown variables. Assuming that error (e) is negligible, from (2.41) we can derive the Kalman estimate (c) as [42]

$$c = S^{\text{pinv}} s \qquad\qquad S^{\text{pinv}} = (S^T S)^{-1} S^T \tag{2.43}$$

where S^{pinv} is the Moore-Penrose Psuedo inverse matrix of S and S^T represents the transpose of matrix S. The Kalman estimate (c) is derived from the original input-output values. The estimate is then used as a denoising filter which has the following transfer function [42]

$$H_c(z) = \frac{1}{(1 - c_1 z^{-1} - c_2 z^{-2})}. \tag{2.44}$$

The input-output values are multiplied by the filter to give new input-output values. These new values are then used as inputs and outputs into the system and new Kalman estimates are determined. The entire process is repeated with these new values until successful convergence is reached i.e., the MSE falls within a desirable range. Once this occurs, the complex root λ of the polynomial can be found by solving [42]

$$1 - c_1 z^{-1} - c_2 z^{-2} = (1 - \lambda z^{-1})(1 - \lambda^* z^{-1}) = 0. \tag{2.45}$$

The frequency ω_o and damping factor α can then be calculated according to (2.14). The method was noted to converge within 10-20 iterations and the results showed a significant improvement when compared to the Kalman linear regression solution [70].

In [72], the convergence and accuracy properties of the STMB algorithm were further investigated. It was discovered that in cases where the additive output noise was white, the algorithm's estimates gave a true representation of the system. It was found that the model is not globally convergent [72] as was implied in [70], however, if the SNR is sufficiently large or the signal has only one pole, the estimates would converge to the true value. It was concluded that the mode-2 version of the STMB algorithm gives the most accurate estimates. The main difference between both modes is that the method of computing the signal estimates is slightly different, while the procedure stays the same. In the mode-1 version of the algorithm, the initial estimates are computed from the output of the system while in the mode-2 version of the algorithm, the initial estimates are a random set of values. It was found that the

STMB method is like the iterative quadratic maximum likelihood (IQML) algorithm and other algorithms that predate the IQML [73] and is great for estimating the parameter of single damped sinusoids in white noise [74].

2.3 Machine Learning Methods

With the introduction of machine learning and AI techniques into different fields, the signal processing field has not been left out. In recent times, artificial intelligence (AI) techniques have been applied to improve the estimation of signal parameters. They have the advantage of being re-used over a wide range of data sources provided the dataset is robust enough to account for different scenarios. Also, they can be used for real-time signal analysis and depending on the learning type involved (supervised or unsupervised), the algorithm is able to train itself and constantly evolve. In supervised learning, the input data is provided to the algorithm model alongside the output data whereas in unsupervised learning only input data is provided and the algorithm learns patterns from the unlabelled data. The goal is to train the model or algorithm to predict an accurate output when given new input data.

In [38], a deep learning algorithm for frequency estimation was proposed. A three-layer model was used and the resulting model was able to accurately estimate the frequency of new inputs with low error rates and in very little time. The authors stated that the proposed model could be scaled up to accommodate high frequencies in the GHz range. In [37], a learning-based approach to estimate the frequencies in a multi-sinusoidal signal from a finite number of samples was proposed. The algorithm is coined PSnet and its aim is to perform line-spectra super resolution using deep learning. The estimation accuracy was determined by comparing the PSnet algorithm against two non-parametric models for traditional line spectra estimation, and it was found that the PSnet algorithm significantly outperformed the other algorithms.

The algorithm proposed in [1], builds on the previously discussed algorithm in [37]. The proposed algorithm is coined DeepFreq and is able to produce more accurate signal parameter estimates than the PSnet algorithm and outperforms other algorithms when the SNR levels range from medium to high. In [39], the DeepFreq algorithm was extended to apply to two-dimensional signals. The resulting algorithm is coined ResFreq and is able to estimate the frequencies of two-dimensional signals. It does this by transforming the signals from the time domain to the frequency domain. This transformation helps reduce the number of network parameters needed by the model and also improves the amplitude estimation of the ResFreq algorithm. The results of the performance evaluation tests show that the algorithm is able to operate at SNR levels ranging from -15 dB to 25 dB and that it provides better performance in terms of accuracy and resolution that the DeepFreq algorithm for two-dimensional signals.

2.4 Summary

In this chapter, a brief overview of signal processing algorithms to estimate signal prarameters such as frequency, phase, amplitude, and damping factor was discussed. The two classes of signal processing algorithms, parametric and non-parametric algorithms were discussed in detail. The parametric algorithms are based on models and can provide high-resolution estimates for relatively short data samples. However, one major drawback of algorithms within this class is that they require the exact model of the signal and its disturbances to be known which is not practical in real-life applications as the signal of interest may have components that cannot be modelled, such as an unknown number of frequencies amongst others. The non-parametric algorithms are more suited for real-life applications as no prior knowledge of the signal of interest is required. Algorithms within this class are based on the FFT algorithm and have lower computational costs in comparison to algorithms under the parametric class. However non-parametric algorithms have the issue of spectral leakage and picket fencing when used to estimate the frequency component of sinusoidal signals. Several algorithms under each class were discussed. The similarities between the algorithms were explored and the improvements made on previous algorithms that gave rise to new algorithms were also discussed. The advantage of each algorithm was presented as well as its shortcomings. This chapter also provided information concerning the advantages and main drawbacks of using the FFT/DFT algorithm for signal processing. The different suggested methods in literature to counter the effects of these drawbacks as well as the shortcomings of these suggested methods were discussed. This chapter gave an overview of different signal processing algorithms, their advantages, and shortcomings. It is important to note that none of these algorithms is inherently better than another. The choice of which algorithm to use depends on the signal of interest and the level of accuracy needed while taking into consideration the time complexity of the entire process

The next chapter introduces the deep learning algorithm proposed in this thesis for signal parameter estimation. The structure of the algorithm and the different modules that make up the algorithm structure are discussed. The algorithm is compared with existing signal processing methods for frequency estimation, and the results are presented.

Chapter 3

Deep Learning Models for Signal Parameter Estimation

Deep learning (DL) is considered a subset of machine learning that is based on artificial neural networks (ANNs) with three or more layers [75]. It is a type of machine learning that is modeled based on the way the human brain functions. In deep learning algorithms, a computer learns how to predict and classify data by filtering inputs through layers that are made up of artificial neurons, also known as nodes. It is used to perform complex computations on large datasets. Deep learning algorithms extract features and discover useful patterns from input data during the training process, which involves filtering inputs through the different layers of the algorithm. These features and patterns can then be used to predict or classify data. This thesis aims to predict signal parameters from datasets containing damped sinusoidal signals. Deep learning was chosen because of its ability to extract features and discover useful patterns in data which is an important attribute needed to solve the challenge proposed in this thesis.

Different deep learning algorithms such as generative adversarial networks (GANs), long short term memory networks (LSTMs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs) were considered to solve the signal parameter estimation challenge. Among these algorithms, convolutional neural networks (CNNs) were chosen. CNNs specialize in processing data that is represented as a vector, matrix, or tensor. The layers of a CNN which include a convolutional layer, a rectified linear unit (ReLu) layer and a fully connected layer are structured to detect patterns that increase in complexity as the network gets deeper [75]. The convolutional layer is the main block of the neural network as it performs most of the network's computations. This layer performs a convolution operation which extracts features from the network's input. The first convolutional layer captures the low-level features of this input. As more layers are added to the network, the succeeding convolutional layers adapt and begin to extract high-level features enabling the network to have a wholesome understanding of its input. This means of understanding its input data is similar to the way humans process information.

The ReLu layer introduces non-linearity into the feature maps by applying the Relu activation function to the output of the convolutional layer. An activation function is used to decide on the importance of a node's input to the network. There are different activation functions, some of which are the sigmoid function, the tanh function, the softmax function, and the ReLu function. CNNs make use of the ReLu function which is defined as [76] f(x) $= \arg max(0, x)$ where x is a positive value received by a node. It helps prevent exponential growth in the computation required by the network and it also prevents the occurrence of vanishing gradients which is when the partial derivative of the loss function vanishes as it approaches a value close to zero. This is often seen in deep networks, preventing the network from being trained. The fully connected layer is used to learn non-linear combinations of high-level features in the data.

In this work, a CNN model which is an extension of the DeepFreq model [1] is proposed. The proposed model is named SPED, which stands for signal parameter estimation using deep learning. The neural network architecture of the SPED model is an improvement on the existing DeepFreq model architecture. The SPED model has an intermediate layer that maps the input signal to a high-dimensional feature space and it has fewer hidden layers than the DeepFreq model. The main contribution of this model is its ability to estimate the frequencies and damping factors contained in damped multi-sinusoidal time-domain signals. The DeepFreq model is only capable of estimating the frequencies of undamped sinusoidal waveforms. Algorithm 1 shows the pseudo-code for the training and testing process for frequency estimation using the SPED model. The network architecture of the SPED model is explained in detail in the next section of this chapter.

3.1 SPED Model Architecture

The architecture for the SPED model is discussed in this section. For the purpose of explanation, this section focuses on frequency estimation using the SPED model. The damping factor estimation process is similar to that of the frequency estimation process. The difference lies in the parameter the model is trained on. A schematic diagram of the model is shown in Fig. 3.1 where a damped sinusoidal signal is fed as an input into the model which gives two values as its output. The first is an estimate of the frequency values contained in the signal and the second is an estimate of the number of frequencies contained in the signal.

The model proposed in this work for frequency and damping factor estimation was written using Python programming language, version 3.6.13. The model is divided into two

Algorithm 1: SPED Model

Training Input : Dataset containing input waveforms

Training Output: Frequency values contained in the waveforms

1. Divide the input into training and validation datasets.
2. Generate bs batches of the training dataset and the corresponding labels to be used to train the model.
3. **for** $i \leftarrow 1$ **to** *epochNumber* **do**

    for $j \leftarrow 1$ **to** bs **do**

        - Generate noisy samples of the training dataset with varying SNR levels.
        - Train the model using these noisy samples.
        - Calculate the MSE loss between the predicted output and the true output for this batch of training samples.
        - Update the attributes of the network (weights and learning rate) using ADAM optimizer.
        - Evaluate the performance of the model using the validation dataset.

    end for

    return *model*

end for

Test Input : Input waveforms

Test Output : Frequency values contained in the waveforms

1. Load the trained model.
2. Normalize and process the input signal to get the frequency representation of the signal and the number of frequencies contained in the signal.
3. Locate the spectral peaks of the frequency representation estimate.
4. Return the estimated values of frequency(s) contained in the signal.

modules: a discretization module and an estimation module, both of which are convolutional neural networks (CNNs). The discretization module makes up the first half of the deep learning network model. The output of this module when plotted shows a representation similar to the DFT of a signal. In this plot, the peaks represent the frequencies present in the input waveform. The estimation module makes up the second half of the deep learning model. The output of the estimation module is a single number that represents the number of frequencies present in the input waveform. The algorithm is able to achieve the desired output or a value significantly close to the desired output by updating the weights of the neural network to minimize the error between the predicted output and the expected output.

The algorithm was trained using the supervised learning process. It involved the use of a labeled dataset containing inputs (signal waveforms) and correct outputs (corresponding frequencies) in the training process to teach the algorithm to yield the desired output. There are two branches of supervised learning, namely regression and classification. Classification algorithms separate the data into specific categories, while regression algorithms are used to understand the relationship between dependent and independent variables. They are specifically useful in predicting numerical values based on different data points. The algorithm proposed in this work is a regression-based supervised learning model where the dataset used is labeled, and the model predicts numerical values based on the input data points.

3.1.1 Discretization Module

This module makes up the first half of the network model. Figure 3.2 shows a representation of the structure of the feed-forward neural network implemented by this module. A batch size of 256 is used in this implementation. The batch size is a hyper-parameter that defines the number of samples to work through before updating the internal parameters of the network.

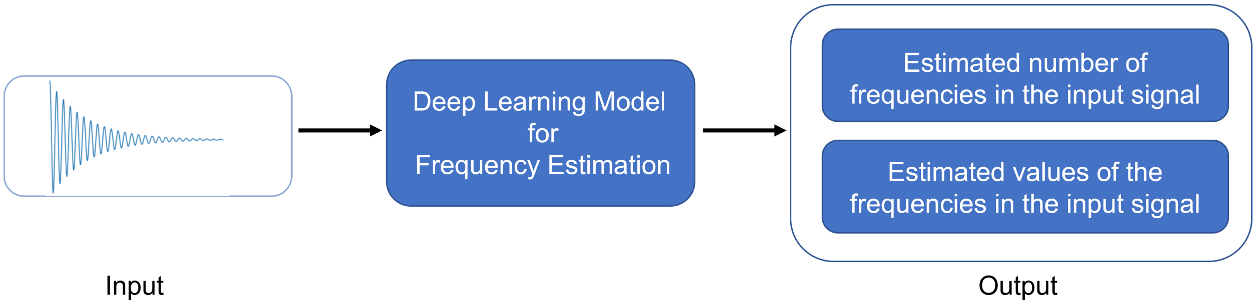


Fig. 3.1: Schematic representation of the frequency estimation process for the SPED model.

A batch size of 256 was chosen to ensure that the model goes through a substantial amount of training samples before updating its weights. In this work, data samples consisting of 500 data points per sample were used. This number of points per data sample was chosen because with 500 data points, a well-defined plot representing the waveform of damped multi-sinusoidal signals containing frequencies ranging from 0 to 2.65 GHz with a frequency resolution of 20 MHz can be obtained. The training dataset consists of $51,200$ data samples while the validation dataset consists of $10,240$ data samples. These numbers were arbitrarily chosen to ensure that the number of samples in the training and validation set is a multiple of 256. From Fig. 3.2, the linear layer maps the 256 by 500 data samples to a feature space consisting of $8,000$ points. These features are then passed through several one-dimensional convolutional layers with Batch Normalization before the rectified linear (ReLu) activation function is applied. Each convolutional layer consists of 32 filters with a kernel size of 3. A kernel size of 3 was chosen because small kernels are better suited to capture and process local patterns.

When dealing with two-dimensional planes, the words kernel and filter are used interchangeably even though a difference exists between them [77]. In two-dimensional planes, a

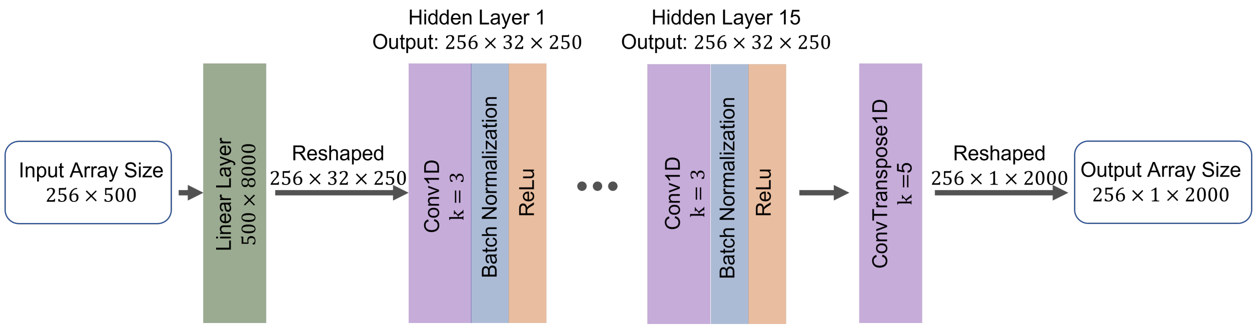
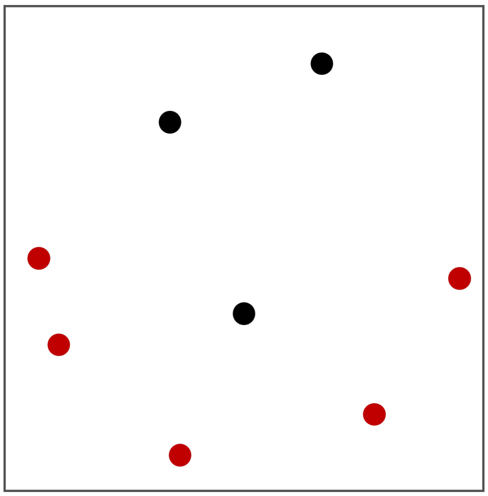
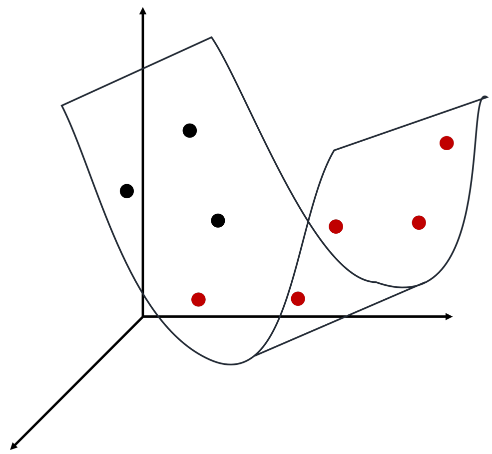


Fig. 3.2: Structure of the neural network that makes up the discretization module of the SPED model.

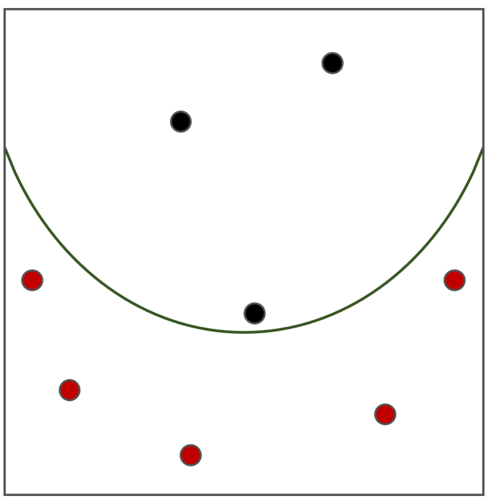
kernel and a filter are the same i.e., they are a two-dimensional array of weights. However in a higher-dimensional plane such as a three-dimensional plane, a filter comprises of several kernels that directly relate to the number of channels the data to be processed has. A kernel is used to generalize linear algorithms to use curved shapes [78]. Kernels can achieve this by transforming a two-dimensional data plane to a higher-dimensional plane. A Gaussian kernel is used in this work as the datasets are of a non-linear nature. Figure 3.3 shows a visual representation of a two-dimensional plane placed in a three-dimensional space such that the data points that were separated by a curved line in the two-dimensional plane are now separated by a plane in the three-dimensional space. The idea behind this is that adding more dimensions increases the flexibility of the lines (two-dimensional) or the planes (three-dimensional), allowing them to move around easily. The data represented in Fig. 3.3a clearly shows that there is no linear relationship between the two sets of data points and as a result, there is no line that can separate the three black points from the five red points in the two-dimensional plane. However, we see that the curve shown in Fig. 3.3b is able to separate the two sets of data points in the two-dimensional plane. When the two-dimensional plane is placed in a three-dimensional space using a kernel, another plane depicted in green as



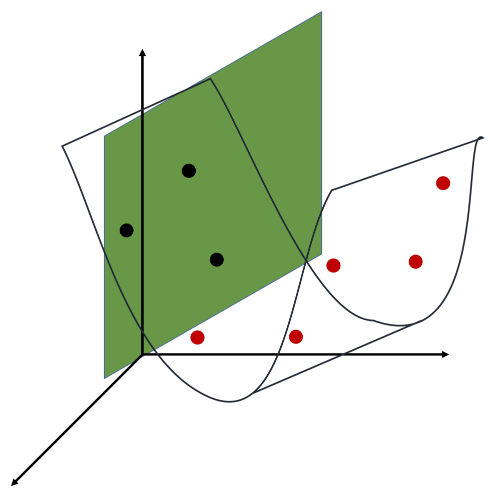
(a) Non-linear Data-points on a 2D plane.



(c) Non-linear Data-points in a 3D space.



(b) Non-linear Data-points on a 2D plane with curve separating the data-points.



(d) Non-linear Data-points in a 3D space with hyperplane separating the data-points.

Fig. 3.3: Visual representation of the transformation of data-points in a 2D plane to a 3D space in order to get a hyperplane to separate the non-linear dataset.

shown in Fig. 3.3c can easily separate the black points from the red points by intersecting the two-dimensional plane as a curve and the three-dimensional space as a plane as shown in Fig. 3.3d. The data samples are transformed to a higher dimensional plane by kernels and then the classification algorithm is run on the data in this higher dimensional plane with the aim of getting a hyperplane that separates the different classes of the dataset. The resulting hyperplane which is a curve in the two-dimensional plane can then be used to make predictions on a new dataset as the hyperplane gives a somewhat linear relationship between the data points and the expected output. This hyperplane when used with a new set of data points, provided the data is similar, would also give accurate results. In this module, the data is transformed from 256×500 to $500 \times 8,000$, and shaped as 256 by 32 by x, where x represents number of columns in the reshaped data sample. This three dimensional array is then passed through each of the 15 layers in the network. The number of hidden layers in the network was set as 15 to ensure that the training time wasn't overly long given the number of sample points contained in each signal waveform. Each layer is made up of a one-dimensional convolutional network, the output of this network is batch normalized after which the ReLu activation function is applied. After all the convolutional layers have acted on this three-dimensional array, the 32 channels are then combined to form one channel by reshaping the data to have a size of $256 \times 1 \times x$ where x represents number of columns in the reshaped data sample. In this case, the value of x is $2,000$. Figure 3.4 shows a graphical representation of applying a one-dimensional kernel to data.

The output of this module as shown in Fig. 3.5 is quite similar to the plot of the DFT transform of a signal. From the plot, dominant peaks are noticed between 0 and 200, while peaks that are similar to noise representation are noticed between 800 and 1200. As with the DFT of a signal, in this representation, the peaks represent the frequencies present in the

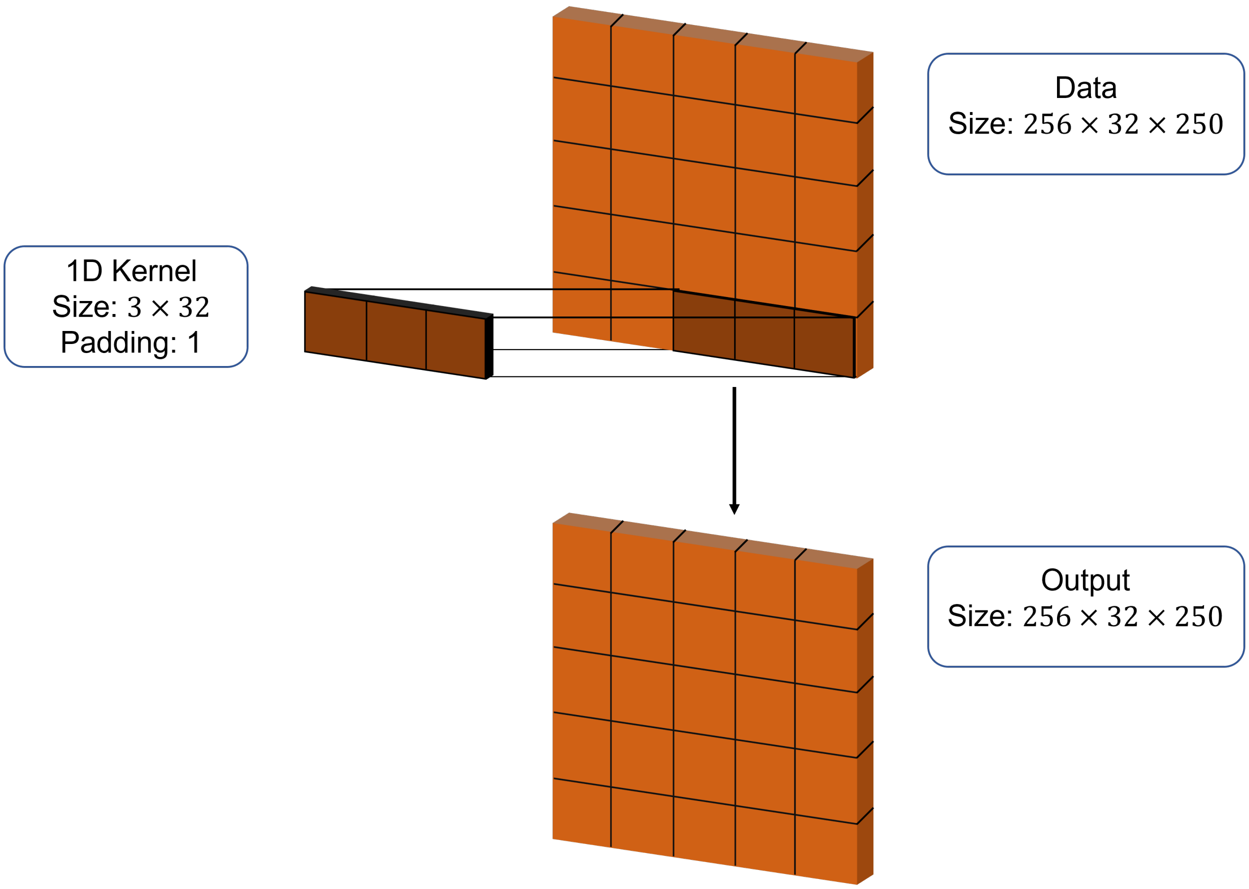


Fig. 3.4: One-dimensional data convolution using a one-dimensional kernel.

input waveform. To get the frequency approximation of the peaks, the output of this module is discretized using a grid of size 2000. The frequency resolution capable by the network using this grid size is 25 MHz. This method of frequency estimation was discovered to provide better frequency estimates than when the network architecture was used to estimate the frequencies [37]. The training process of the discretization module can be summarized as forcing the neural network to learn to produce an output that is similar to the magnitude of the DFT of the input signal. An accurate prediction is achieved by minimizing the error

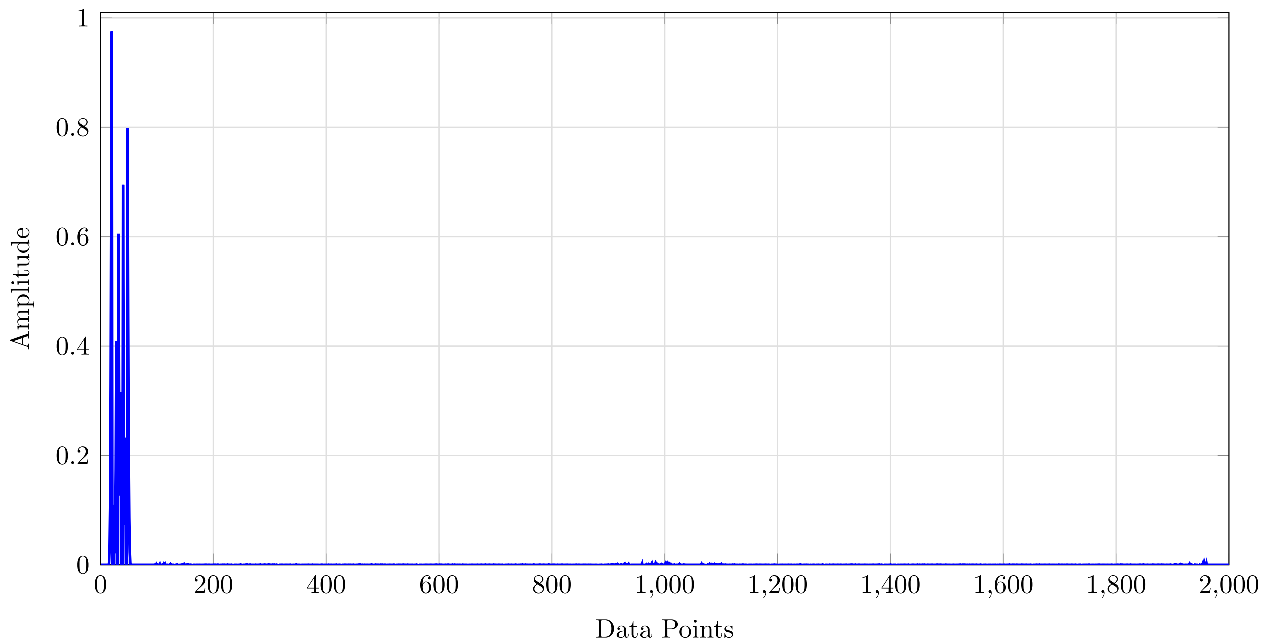


Fig. 3.5: Plot of the Discretization Module Output for an Input Waveform Containing Four Frequencies.

between the network's predicted value and the signal's true value for a number of training samples using the grid as a reference measure.

3.1.2 Estimation Module

The estimation module can approximate the number of frequencies contained in the signal without prior information about the signal model. The discretization module locally concentrates the frequency information of the signal before passing it to the estimation module making it easy for the estimation module to count the number of frequencies.

This module exploits the knowledge that the presence of true frequencies in the signal is unperturbed by translations as far as the noise in the signal is not in the frequency domain. Figure 3.6 shows a representation of the feed-forward neural network represented by this

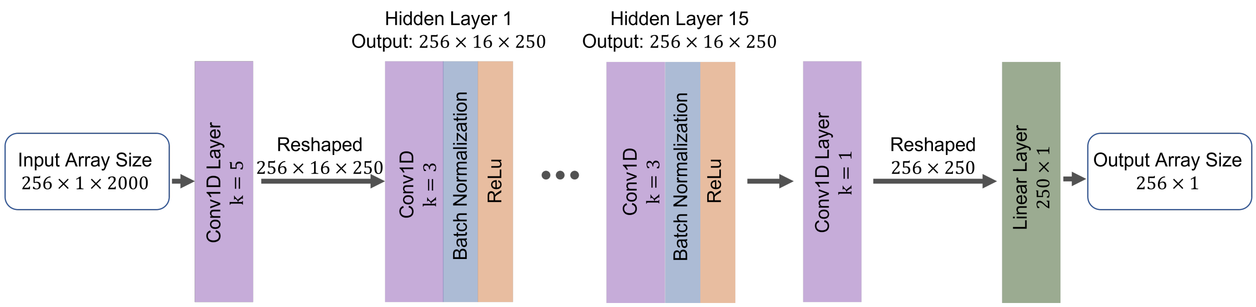


Fig. 3.6: Structure of the neural network that makes up the estimation module of the SPED model.

module. The first convolutional layer has a wider filter than the remaining layers. The output layer is a fully connected layer that outputs a single number which is the frequency estimate of the number of frequencies in the waveform. This module is trained using the results from the discretization module. Both modules are trained separately with the discretization module trained first after which the estimation module is trained based on the results from the discretization module. The performance of the algorithm is largely dependent on the accuracy of the discretization module. In both modules, the training loss is minimized by adjusting the weights of the network to reduce the error between the estimated value and the true value. The network parameters such as the weights and the learning rate are updated using the adaptive movement estimation (ADAM) optimizer. This optimizer was chosen because it combines the benefits of the adaptive gradient algorithm (AdaGrad) and root means square propagation (RMSProp) optimization methods [79].

3.2 Summary

In this chapter, deep learning and convolutional neural networks were discussed. Also, the architecture of the SPED algorithm was presented. The algorithm is a regression-based supervised learning model that is trained using a labeled dataset. It is considered a regression model because its output is continuous in nature i.e., it estimates the frequency and damping factor values of damped sinusoidal signals. The algorithm consists of two modules: the discretization module and the estimation module. The discretization module forces the output of the network to produce a representation similar to the magnitude of the DFT of the input signal. This output is then discretized using a grid to determine the frequency or damping factor estimates depending on what parameter the network is being trained to estimate. This is done by plotting the output of the discretization module across a database grid and taking note of where the peaks fall. The estimation module returns a single number which is an estimate of the number of frequencies contained in the signal. The estimation module plays an important role in the performance of the entire algorithm. An accurate estimate of the number of frequency values in the signal improves the algorithms' ability to estimate the values of these frequencies and their corresponding damping factors.

In the next chapter, the SPED algorithm was tested on waveform samples from different simulated and experimental datasets. The performance of the algorithm was evaluated using specified metrics and the results are presented.

Chapter 4

Performance Evaluation of the SPED Algorithm

In this chapter, the performance of the SPED algorithm is evaluated. The performance of the algorithm is evaluated using regression metrics such as the mean square error (MSE), the mean absolute error (MAE), and the R-squared (R^2) score. Also, the algorithm is tested on three different types of waveform samples that were generated using three different methods. The first set of waveform samples are generated using synthetic data, the second set are generated from a Multisim simulation that models a wireless passive sensor and the third set are generated from laboratory test with the wireless passive sensor. The generation of the waveforms using synthetic data, the Multisim simulation and the laboratory tests with the wireless passive sensor are discussed in detail in this chapter. The evaluation tests performed using these waveforms and the regression metrics used to evaluate the SPED algorithm are also covered.

4.1 Waveform Samples from Simulations

The synthetic data waveform samples were generated using

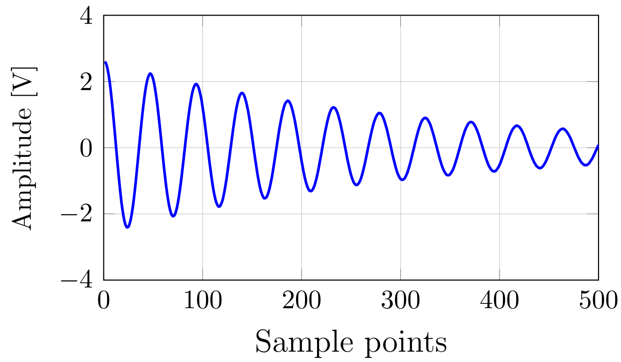
$$y(t) = \sum_{n=1}^{N} R_n e^{(-\alpha + j(2\pi f_n + \phi))t} + q(t) \tag{4.1}$$

for real-valued damped sinusoidal signals, where $R_n \geq 0$ is the signal amplitude, $\alpha \geq 0$ is the damping factor, q is the noise added to the signal, n is the sample index, and N denotes the number of samples. Figure 4.1a shows the plot of a waveform generated using (4.1). From the plot we see that there is a steep rise in the amplitude of the signal from 0 to 2.3. A second signal definition based on the concept of double exponential waveforms was generated using

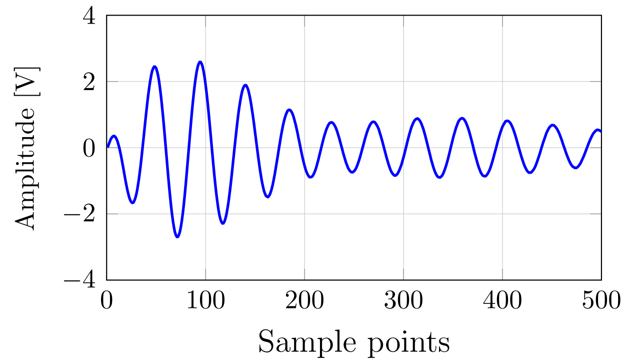
$$y(t) = \sum_{n=1}^{N} R_n(e^{(-\beta)t} - e^{(-\gamma)t}) \times e^{j(2\pi f_n + \phi))t} + q(t) \tag{4.2}$$

where $\beta, \gamma \geq 0$ are used to define the signal envelope which shows a gradual rise in its amplitude from 0 to 0.5 as shown in Fig. 4.1b. The modification of the signal definition for a damped sinusoidal signal (4.2) was proposed to model the signal envelope of the synthetic data waveforms to match of the envelope of ringback signals obtained from the multisim simulations and the laboratory tests. The algorithm was trained on both waveforms and the performance of the trained models was evaluated.

In [1], 100 data points were used in training their model. To effectively capture a damped sinusoidal signal containing frequencies in the GHz, 100 data points would be inadequate. In [38], 2,000 data points were used in training their model. Their proposed network architecture is three layers deep. In this work, three layers would be insufficient to properly train a model that is able to capture the relevant features needed to accurately estimate the fre-



(a) Single exponential waveform sample.



(b) Double exponential waveform sample.

Fig. 4.1: Plot of a single and double exponential waveform sample for a damped sinusoidal signal.

quencies contained in the input waveforms. Neural networks that have a lot of layers have a longer training time when compared with neural networks that have a smaller number of hidden layers. Also, the number of data points of the input waveform contributes significantly to the training time. Datasets consisting of waveforms with large amounts of data points have a longer training time than datasets consisting of waveforms with smaller amounts of data points. To ensure that the training time is not overly long, there is a trade-off between the depth of the neural network and the number of data points in the input signal.

The training dataset used in this work consists of $51,200$ data samples each consisting of 500 data points. This number of data points was chosen after taking into consideration certain design factors such as the number of layers in the network, the training time, and the total number of points needed to accurately represent the damped sinusoidal signal. For each of the waveforms contained in the training, testing, and validation dataset, a random number of frequencies are contained in the waveform. The number of frequencies in each of the waveform samples are chosen at random and range from a minimum of 1 frequency to

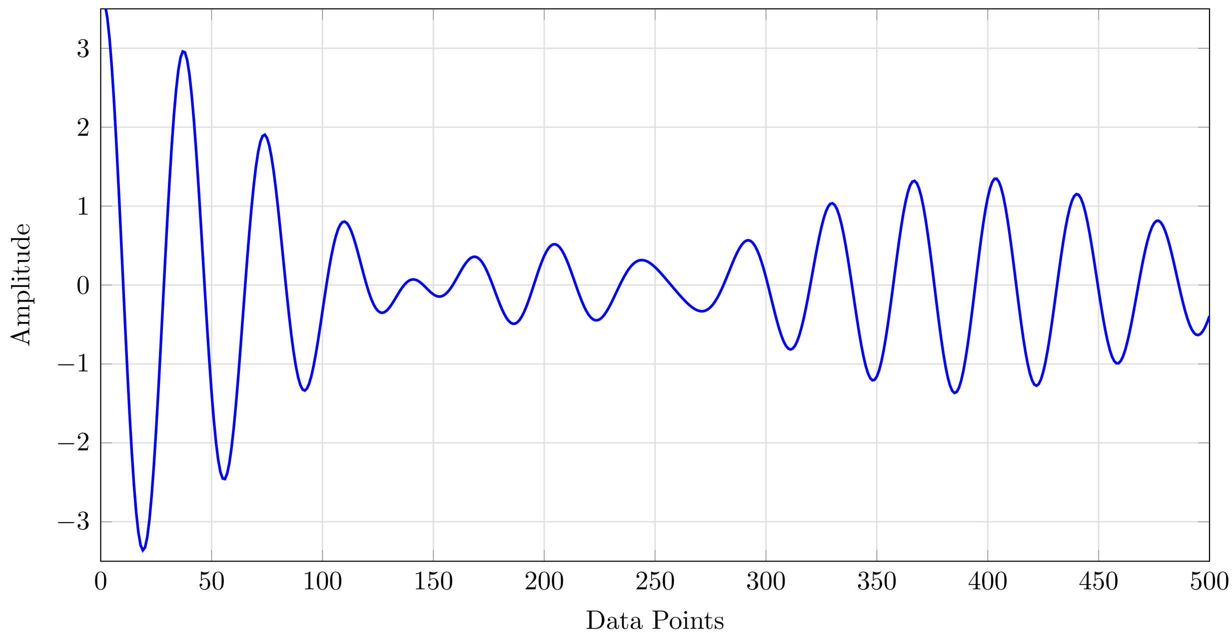


Fig. 4.2: Plot of an input waveform containing three frequencies, 2.07 GHz, 2.29 GHz and 2.48 GHz.

a maximum of 10 frequencies. Figure 4.2 shows a sample of a damped sinusoidal waveform containing 3 frequencies. The frequencies contained in the waveform sample are chosen at random and range from 0 Hz to 2.65 GHz. The model was trained using this range of frequencies to ensure that it is robust and able to handle frequency estimation for a wide range of input signals.

Figure 4.3 shows an example of the ringback signal from a laboratory experiment using the wireless passive sensor. The ringback signal has three main components, the feedthrough, the structural mode, and the antenna mode. The feedthrough and structural mode are typically reflections from the interrogation and sensor antennas and can be removed using time-gating. The antenna mode is the main part of the signal that contains information on the resonant frequency of the sensor which is the frequency we want to estimate as

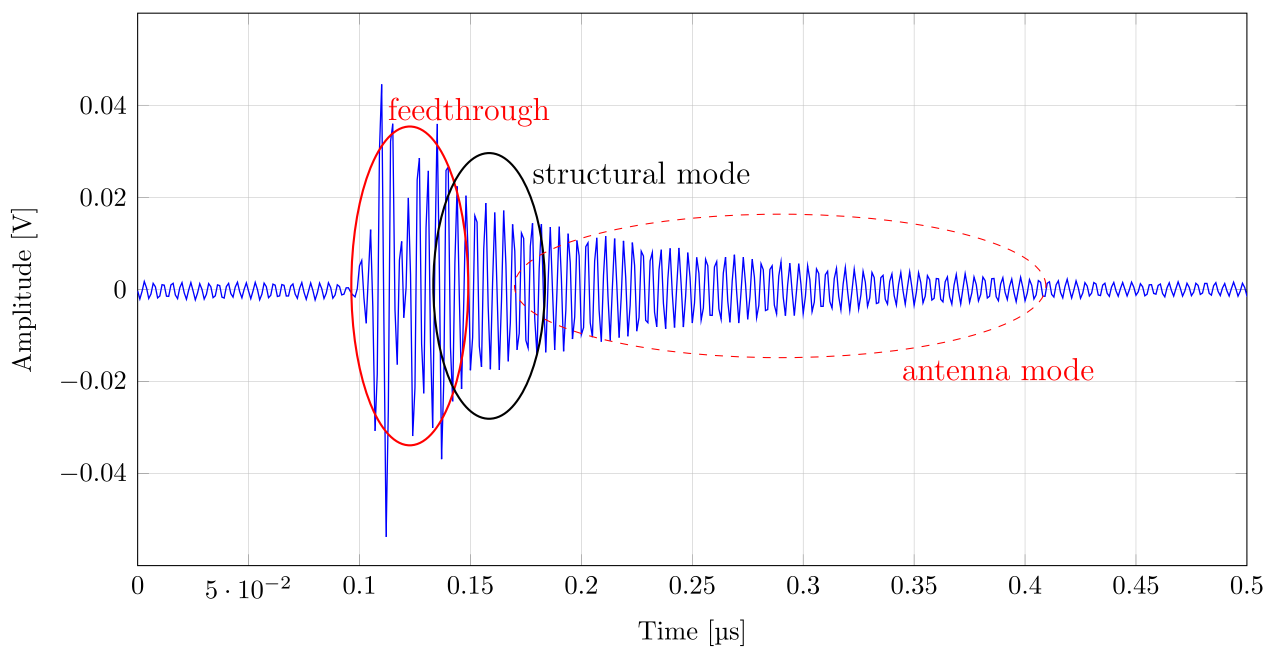


Fig. 4.3: Plot showing a sample of a ringback signal gotten from laboratory experiments with the wireless passive sensor and the different components of the ringback signal. Adapted from [40] with permission.

accurately as possible. Although the ringback signal is expected to have a maximum of three frequencies and noise in the signal, the dataset is generated such that each waveform in the $51,200$ samples has between one and ten different frequencies. The preliminary tests showed that a model trained using a maximum of ten frequencies had a higher frequency estimation accuracy than a model trained using a maximum of four frequencies.

4.1.1 Results and Observations

The performance of the trained models was evaluated using non-linear regression model metrics. In this work, the model trained using synthetic data based on (4.1) would be referred to as the *conventional model* while the model trained using synthetic data based on

(4.2) would be referred to as the *modified model*. After training, the conventional model was evaluated on damped sinusoidal signals with varying SNRs. The results of these tests are shown in Fig. 4.4. The results show that the SPED algorithm can accurately estimate the frequencies contained in damped multi-sinusoidal waveforms even at low SNR values. The results shown in Fig. 4.4 are generated using the validation dataset which consists of $1,000$ test samples that are not part of the training or testing data set used in the learning process for the model. The frequency values are represented in radians by

$$f = \frac{\omega n}{2\pi t} \tag{4.3}$$

where $\omega \leq \pi$, n is the sample index and t is the period. Equation (4.3) is derived by solving (4.4),

$$A\cos(\omega n + \phi)e^{-dn} = A\cos(2\pi f t + \phi)e^{-\phi t} \tag{4.4}$$

where the RHS is a real-valued damped sinusoidal signal in the time domain. A false negative rate (FNR) of 7.69% was achieved during the training process for the conventional model while an FNR of 13.75% was achieved during the training process for the modified model. The false negative rate (FNR) is calculated according to

$$\text{FNR} = \frac{\text{false negative}}{\text{false negative} + \text{true positive}} = \frac{\text{FN}}{\text{FN + TP}}. \tag{4.5}$$

The FNR is the probability that a true positive would be missed by the algorithm. For the conventional model, an FNR of 7.69% means that there is an 92.31% probability that the network correctly identifies the frequency in the signal. Table 4.1 shows the result of the regression metrics for the conventional and modified models evaluated using the test dataset

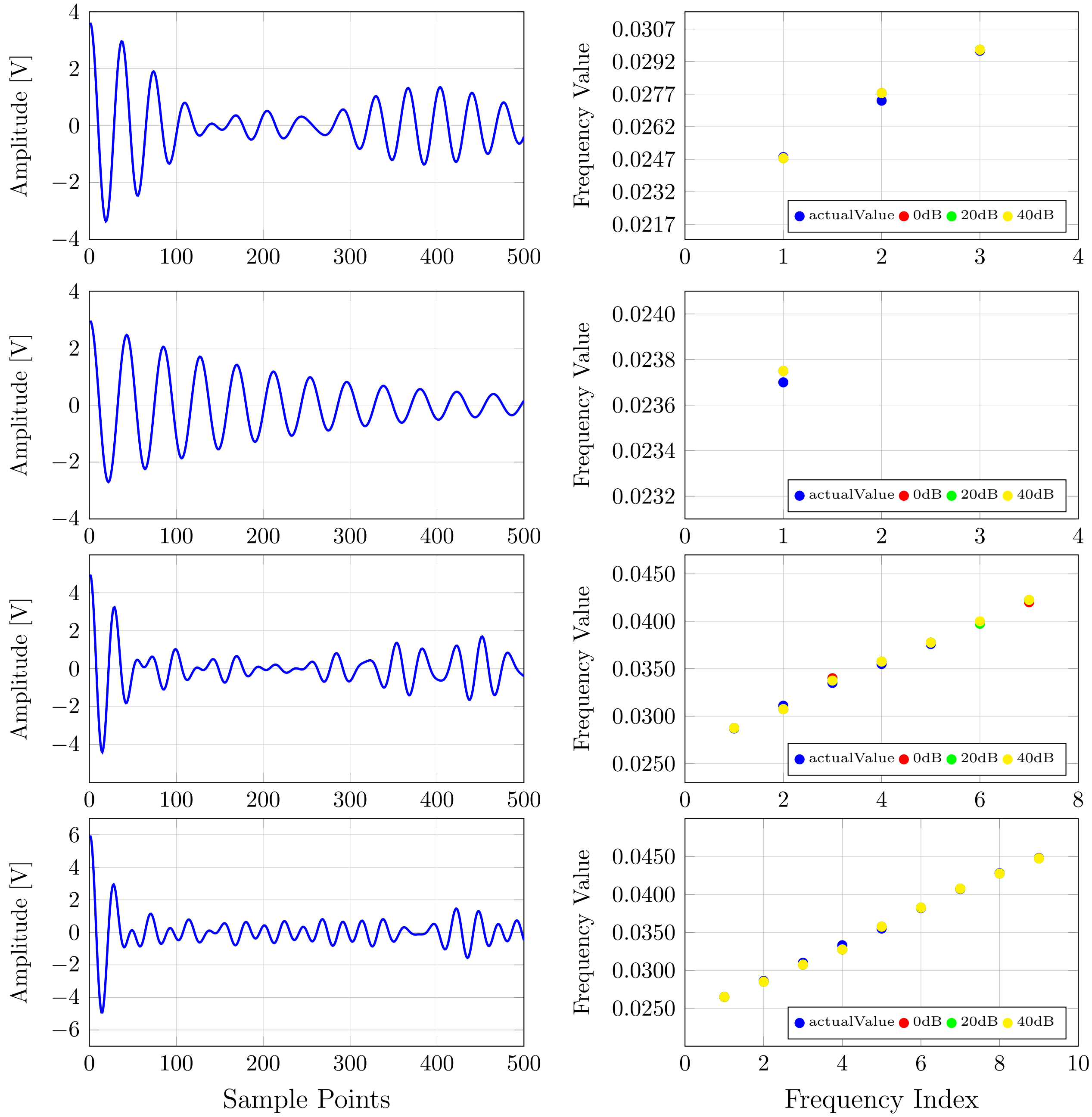


Fig. 4.4: Plot showing four samples of input waveforms containing different number of frequencies (a; 3 frequencies, b; 1 frequency, c; 7 frequencies, d; 9 frequencies). The output of the deep learning algorithm for each of the input waveforms at different SNR levels is shown.

Table 4.1: Evaluation metrics for the SPED algorithm based on the conventional, modified, and damping factor models using the test dataset

Evaluation Metrics	Conventional Model	Modified Model	Damping Factor Model
Mean Absolute Error (MAE)	0.023	0.067	0.036
Mean Squared Error (MSE)	0.023	0.067	0.029
Median Absolute Error (MEAE)	0.001	0.001	0.007
Variance Score	0.89	0.69	0.84
R^2 Score	0.88	0.64	0.83

that consists of $1,000$ waveform samples. The variance score is calculated according to

$$\text{Variance Score} = 1 - \frac{\text{Variance (y - }\hat{y}\text{)}}{\text{Variance (y)}} \tag{4.6}$$

where y represents the target value and $\hat{y}$ reresents the predicted value. It measures how well the observed value differs from the mean of the predicted values. It is a useful metric in determining how well the model handles variations in the dataset. The best possible score is 1.0. The R^2 score is often referred to as the coefficient of determination. It is calculated according to

$$R^2 \text{ score} = 1 - \frac{\text{residual sum of squares}}{\text{total sum of squares}} = 1 - \frac{\text{RSS}}{\text{TSS}}. \tag{4.7}$$

It is a useful metric in determining how well unknown samples would be predicted by the model. As with the variance score, the best possible score is 1.0. From the results in the table, it is evident that the conventional model performs better than the modified model as it has a higher R^2 score. This difference could be because the signal definition of the modified

Table 4.2: Evaluation metrics for the SPED algorithm based on the conventional, modified, and damping factor models using the training dataset

Evaluation Metrics	Conventional Model	Modified Model	Damping Factor Model
Mean Absolute Error (MAE)	0.023	0.062	0.026
Mean Squared Error (MSE)	0.023	0.063	0.026
Median Absolute Error (MEAE)	0.001	0.001	0.007
Variance Score	0.90	0.71	0.86
R^2 Score	0.88	0.67	0.85

model is more complex than that of the conventional model and as such the number of features that need to be learned by the deep learning model increases which could reduce the performance of the model. The use of deep learning to estimate the damping factors of multi-sinusoids was also investigated and the results of the regression metrics are shown in Table 4.1 under the column heading *Damping Factor Model*. The signal definition used for this model is the same as that used for the conventional model. The difference is the algorithm is trained to estimate damping factors instead of frequencies.

The evaluation metrics were also calculated using the training data to determine the performance difference between the training and the test datasets. The results are shown in Table 4.2. Comparing the results in Table 4.2 and Table 4.1, we notice a slight performance drop between the training and the test datasets. The R^2 scores are the same for the conventional model across the two datsets but the variance score for the training dataset is higher than that of the test dataset with a difference of 0.01. In the case of the damping factor model, the R^2 score and the variance score both vary across the two datsets with a difference of 0.02. Based on these results we can conclude that the model does not underfit or overfit as it has good performance values on both the training and test dataset. An overfit model is one that has good performance on the training dataset but performs poorly on the test dataset while an underfit model is one that performs poorly on both the training and test dataset. The results show that deep learning is a viable candidate for signal analysis particularly in the case of frequency and damping factor estimation.

4.2 Multisim Waveform Sample

A resonator-based wireless passive sensor was modeled and simulated using Multisim 14.1. This sensor was proposed in [40]. The passivity of the sensor makes it easy to fabricate and mass produce while its wireless property allows for contactless measurements in environments where the prevailing conditions are not suitable for wired connectivity. The sensor can detect changes in environmental temperature, pressure, and electric field intensity due to changes in its resonant frequency. In this simulation, the parameter being measured by the sensor is electric field intensity. In the circuit model shown in Fig. 4.5, the switch $S2$ is used to simulate switching between the transmitter and receiver. The switch is controlled by a 100 kHz square wave generated by the function generator $XFG2$. The square wave has a high amplitude of $5V$ and a low amplitude of $0V$ which controls the switch. When the amplitude of the square wave is $5V$, the switch $S2$ is in the top position, and the AC voltage source powers the circuit. When the square wave is $0V$, the switch $S2$ is in the bottom position and the ringback signal is measured across the $50\,\Omega$ resistor $R2$ which represents the receiving antenna. Oscilloscopes $Xsc1$ to $Xsc4$ are used to measure the voltage across various points in the schematic diagram. The $50\,\Omega$ resistor $R3$ is used to mimic a $50\,\Omega$

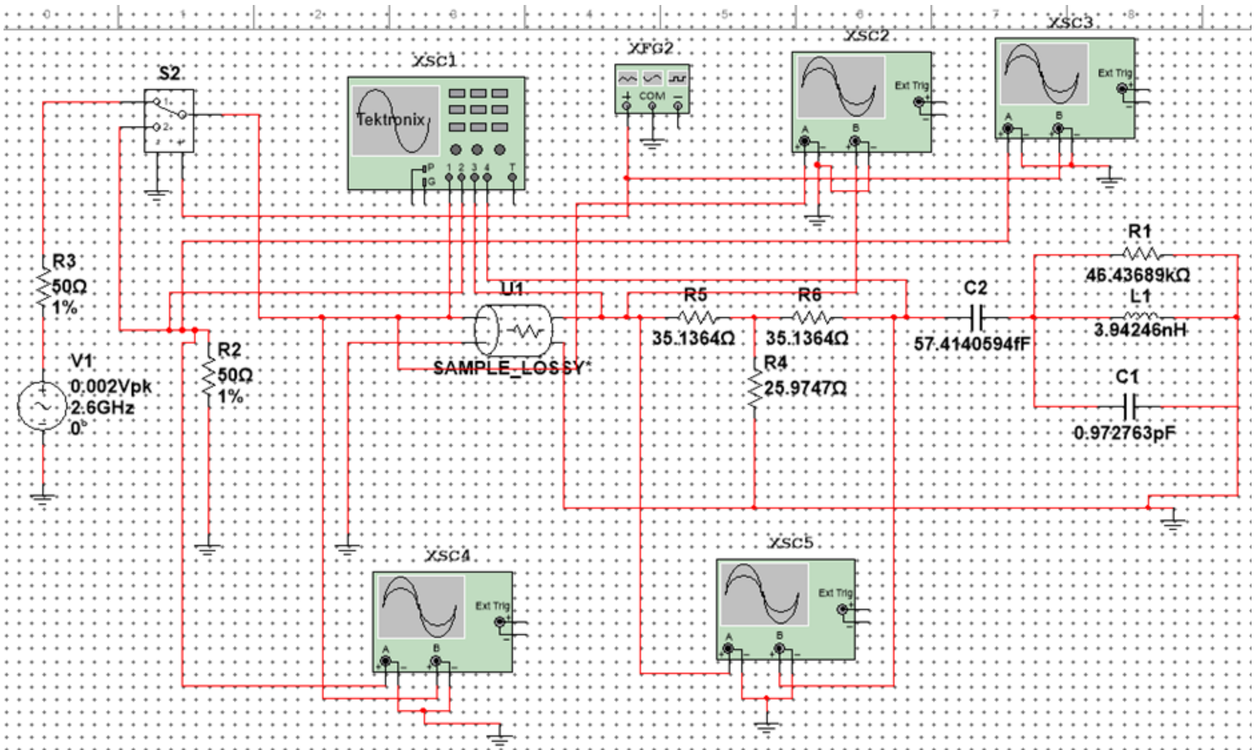


Fig. 4.5: Circuit model of the wireless passive resonator simulated in Multisim 14.1. The block diagram of the interrogation system is shown in Section A.2 of the appendix.

voltage source by connecting it in series to the AC voltage source. A lossless transmission line is used in the simulation software to represent the wireless channel through which the RF signals are transmitted and received. To model a 2-way path loss of -20 dB, a $50\,\Omega$ T-network 10 dB attenuator was added to the output of the transmission line. Figure 4.6 shows an example of the ringback signal measured across the $R2$ resistor. The observed signal waveform closely resembles the waveforms generated using synthetic data at high frequencies that were discussed in the previous section. This simulation model closely matches the fabricated resonator used for field tests. This was done to train and evaluate the SPED algorithm on synthetic and experimental data that are similar in form.

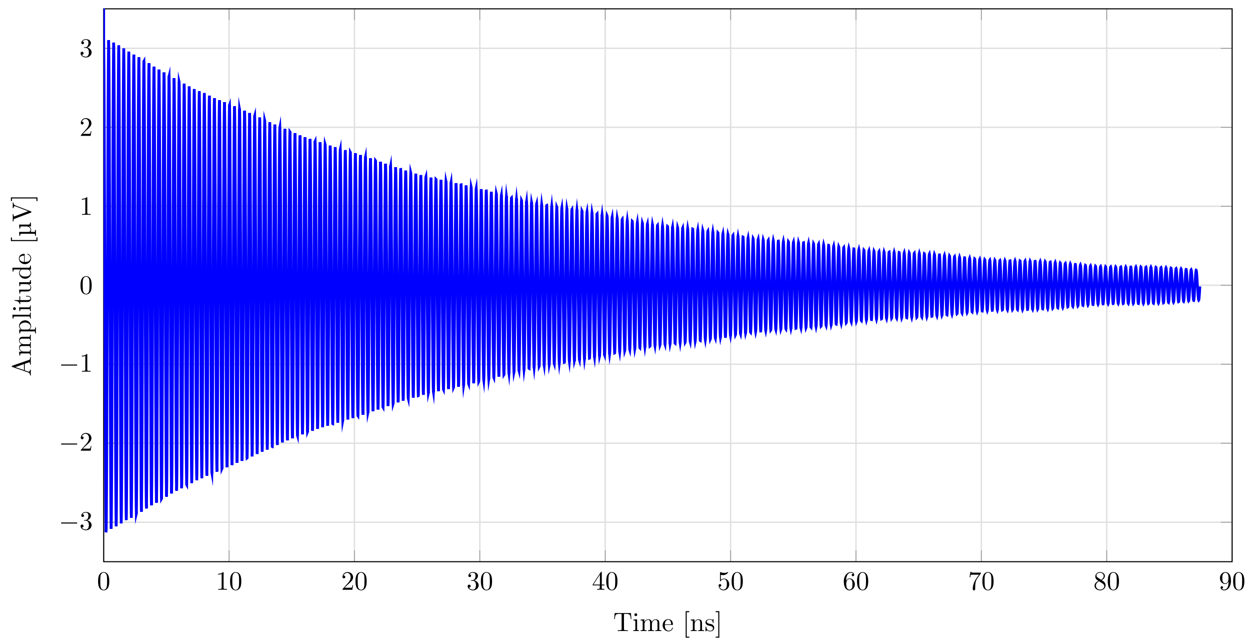


Fig. 4.6: Plot of the ringback signal across resistor R2 for the circuit model shown in Fig. 4.5.

4.2.1 Results and Observations

The conventional model was evaluated on the ringback signals from the Multisim simulation and the results are presented in Table 4.3. The conventional model is used for this test because the results from the regression metrics in Table 4.1 show that the conventional model outperforms the modified model based on their R^2 scores. The conventional model is able to handle variance in data better than the modified model. From the results in Table 4.3, it is evident that the algorithm is able to estimate the frequency contained in the waveforms from the Multisim simulation with an accuracy of $\pm$ 40 MHz. For some frequencies, the algorithm returns more than one frequency estimate. However, both estimates are within an error value of $\pm$ 40 MHz . For a waveform of 3 GHz, the algorithm is able to estimate the frequency but with a larger error value than previously noted for other frequencies. This

Table 4.3: Results of the SPED algorithm model when tested on signals from the Multisim simulation

Signal Frequency	Estimated Frequencies	Error
200 MHz	165 MHz 185 MHz	35 MHz 15 MHz
310 MHz	282 MHz 305 MHz	28 MHz 5 MHz
1.56 GHz	1.54 GHz	20 MHz
2.58 GHz	2.62 GHz	4 MHz
3 GHz	3.06 GHz 3.2 GHz	60 MHz 200 MHz

increase in the error value could be attributed to the fact that the model was trained using frequencies within the range of 0 Hz and 2.65 GHz of which 3 GHz is not included. This shows that the model can predict frequency estimates of signals that are outside its training range albeit with reduced accuracy. To reduce the error value and improve the frequency estimation of the algorithm over a wider range of frequencies, the algorithm can be trained using a more robust dataset that includes more data samples spread over a wider frequency range.

4.3 Passive Wireless Sensor Waveform Sample

The interrogation system used for the laboratory experiments was modeled and simulated in Multisim 14.1 explained in Section 4.2. The data samples, an example of which is shown in Fig. 4.7, are received from the sensor via a wired connection or a wireless connection. The

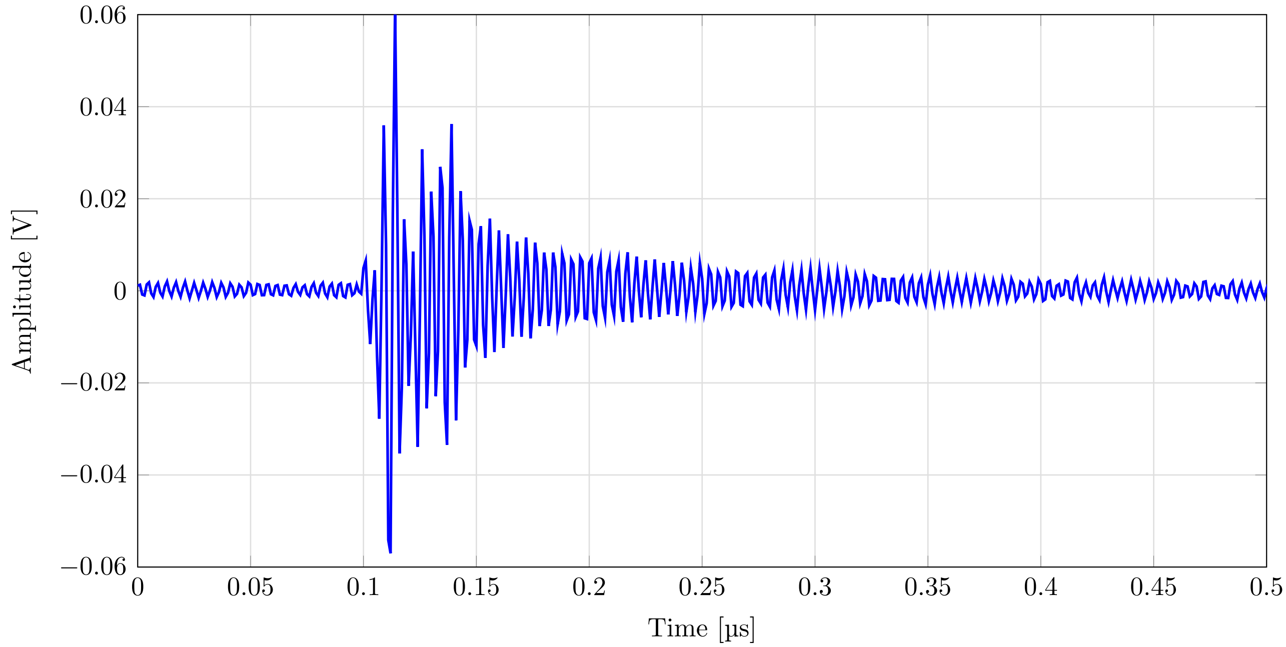


Fig. 4.7: Plot of the measured downconverted ringback signal gotten from the wireless passive sensor after interrogation.

first method involves connecting a coaxial cable to the resonator while the second method involves using the horn antennas to transmit RF signals and receive ringback signals from the resonator. The ringback signal is passed through a low-noise amplifier to amplify it without significantly degrading its signal-to-noise ratio. This amplified signal is denoised using a bandpass filter and down-converted using a mixer. The down-converted signals are captured and stored using an oscilloscope. For this work, the waveforms were measured when the resonator was used for wireless sensing with the horn antennas 80 cm apart. This distance is within the far-field region for the ISM band of 2.4 GHz to 2.5 GHz. The experiments were carried out in a lab environment with many reflective surfaces. This was done to model real-life scenarios where clutter interference exists.

Table 4.4: Results of the SPED model when tested on ringback signals received from the wireless passive sensor

Interrogator Frequency	LO Frequency	Estimated Frequencies	Error
2571.4 MHz	2830 MHz	258 MHz 285 MHz	600 kHz 27 MHz
2574.4 MHz	2830 MHz	228 MHz 243 MHz	28 MHz 13 MHz
2574.5 MHz	2752 MHz	163 MHz 178 MHz	15 MHz 50 kHz
2579.4 GHz	2830 MHz	248 MHz	3 MHz

4.3.1 Results and Observations

The algorithm was tested on the waveforms from lab tests with the wireless passive sensor and the results are shown in Table 4.4. The interrogator system transmits RF pulses at different frequencies to the wireless passive sensor which has a resonant frequency of 2.57 GHz. The ringback signals from the sensor for these transmitted frequencies are down converted using a local oscillator (LO). The down-converted signals are within the range of 177 to 258 MHz and these signals are then evaluated to provide the frequency estimate of the signal using the SPED model. The results show that the SPED algorithm is able to estimate the frequencies contained in the down-converted damped sinusoidal signal with an accuracy of $\pm$ 30 MHz. In most of the test cases, the algorithm provided more than one frequency estimate. This is likely because of reflections and noise contained in the signal. The estimates provided by the algorithm are all close in value to the actual frequency. The observed error value between the actual frequency and the estimated frequency could be because of the precision of the discretization grid within the model's training process. The error between the actual frequency value and the estimated frequency value which is approximately $\leq 35\,\text{MHz}$ could be reduced by increasing the precision of the discretization grid. Overall, the results show that the SPED algorithm trained using a deep learning model can estimate the frequencies of signals in a damped sinusoidal waveform. It has the advantage of not being susceptible to spectral leakage and picket fencing like the non-parametric class algorithms because it does not use an FFT-based approach. The deep learning model is able to estimate the frequency values contained in input waveforms based of the features it learned from signals in its training dataset. Also, no prior knowledge of the signals model is required as in the case of the parametric class. The SPED algorithm when trained using a robust dataset can be used to provide the frequency estimates of a wide range of signals.

4.4 Comparison with Other Methods

In this section, the SPED algorithm is evaluated against existing signal processing algorithms. The three algorithms chosen for this evaluation are: STMB, BY1-LC, and MPM. These algorithms are discussed in Section 2.2.2, 2.1.5, and 2.2.1 respectively. The BY1-LC algorithm was chosen because amongst the non-parametric class of signal processing algorithms, it boasts of improved accuracy in estimating signal parameters and excellent convergence rates since it combines interpolation with an iterative process for signal parameter estimation. The MPM algorithm was chosen because amongst the parametric class of signal processing algorithms, it approximately reached the Cramér-Rao bound which implies that it achieved the lowest possible mean squared error in comparison to other techniques and no other technique can perform better in estimating the poles of a signal in a noisy environment. Details of the Cramér-Rao bound are given in Section A.1 of the appendix.

The STMB algorithm was chosen because like the BY1-LC algorithm, it is a parametric class iterative algorithm used to compute the pole-zero model of damped sinusoidal signals and it boasts of good accuracy values when estimating the parameters of single-damped sinusoidal signals in white noise. These three algorithms were chosen because they are considered the best methods amongst the existing signal processing methods [42]. Also, they make a good evaluation basis for determining the performance of the SPED algorithm when compared with parametric and non-parametric signal processing algorithms.

4.4.1 MPM

The MPM method is a precise but computationally expensive total least square (TLS) solution to find the frequency of a given signal sample. A discussed in Section 2.2.1, the MPM method dates as far back as 1985 and is the result of improvements on other algorithms over time. The MPM method is based on the Prony method which is dated as far back as 1795. The Prony method had the downside of not being able to estimate the frequencies of noise-contaminated data and non-equispaced data samples. The Pencil of Functions method was then developed to address the shortcomings of the Prony method, but it was computationally expensive. The Pencil of Functions method was improved on to form the Generalized Pencil of Function method also known as the Matrix Pencil Algorithm. This improvement was brought about by incorporating the ESPRIT approach into the existing Pencil of Functions method. The resulting MPM algorithm relies on matrix algebra and is considered computationally expensive, especially when dealing with large data samples. The algorithm and its related formulas are explained in Section 2.2.1 of this thesis.

4.4.2 BY1-LC

The BY1-LC algorithm analyzes the parameters of signals in the frequency domain using the Inverse Fast Fourier Transform (IFFT) method [58]. It works on the premise that the mode energy centralizes at its frequency and forms a spectral peak. Each of these spectral peaks contains information needed to accurately determine the unknown signal parameters. The BY1-LC algorithm performs interpolation using the three DFT bins with the highest magnitude. The signal to be processed is of finite duration and as a result, there is spectral leakage in both the negative and positive spectra. The spectral leakage from the positive frequencies spills into the negative frequencies' spectrum and vice versa. Leakage correction is achieved through an iterative process. DFT interpolation is applied on the frequency bins after which, the signal parameters are estimated. The effects of spectral leakage from the negative frequencies of other spectral lines are then subtracted from the three frequency bins and a new estimate of the signal parameters is found. The entire process, as explained in Section 2.1.5, is repeated until the error value falls below a certain threshold.

4.4.3 STMB

The STMB method is an iterative algorithm used to reduce the influence of noise on the signal parameter estimates of an input signal. It achieves this by reducing the mean-square error between the input signal and the model's output. According to [70], the method is able to provide accurate signal parameter estimates even at low SNR < 1 levels. In the first iteration, the input signal samples are denoised by the application of a filter to give a second set of samples. The coefficient values for the filter are obtained from the input signal which consists of noisy data samples. A new estimate of the signal parameters is then found using the denoised samples as explained in Section 2.2.2 and this process is repeated until the error value falls below a certain threshold. It is important to note that each iteration of the STMB method is like the Kalman linear regression method with the difference being in the filtering of the output and input values.

4.4.4 Results and Observations

To evaluate the performance of the SPED algorithm in comparison to existing signal processing methods, the algorithms were tested on three sets of signals: multi-sinusoidal signals generated from synthetic data, signals generated by Multisim simulation, and signals from lab measurements with the passive wireless sensor. The waveforms from synthetic data were generated according to (4.1) for damped sinusoidal signals. The frequency values and damping factor values were chosen such that the generated synthetic signals closely resembled the waveform samples obtained from the lab measurements using the wireless passive sensor. The idea behind this was to train the SPED algorithm using data samples that closely resemble experimental data so that the trained model could be used to estimate the frequency values of damped sinusoidal signals collected from lab tests or field tests in the real world. The results of the test on signals generated using synthetic data are shown in Fig. 4.8. The input frequencies in the synthetic data are 248 MHz, 238 MHz, 288 MHz, 286 MHz and 383 MHz. From the graph, we notice that across the four algorithms, for SNR values $\geq$ 20 dB, the frequency estimates appear to be relatively constant for each algorithm. The MPM algorithm performed similar to the BY1-LC algorithm for signals with SNR values between 0 dB and 10 dB. At higher SNR levels, we see that the MPM algorithm is able to estimate the actual frequency of the signal with an error value between 13 kHz to 23 kHz. Depending on the application where the algorithm is used and the level of accuracy needed, this error range could be acceptable. The BY1-LC algorithm has a similar performance to

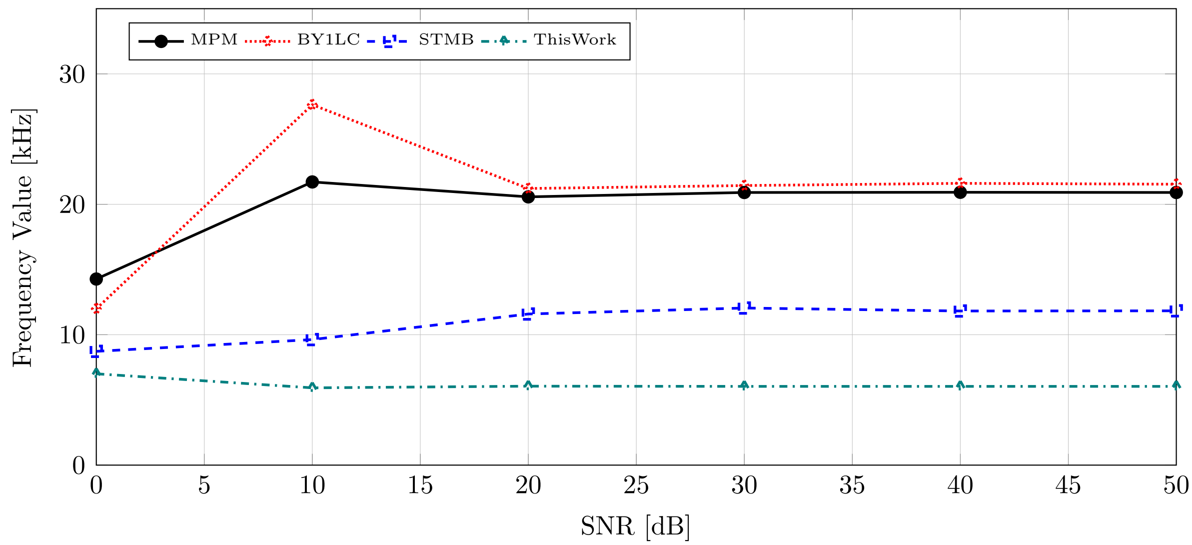


Fig. 4.8: Comparison of the frequency estimation accuracy among the four different algorithms.

that of the MPM algorithm as its error value is between 20 kHz to 30 kHz with an exception when the SNR value is 0 dB. The error value at that point is 12 kHz. The STMB algorithm has error values ranging between 9 kHz, and 12 kHz, which is considerably low in comparison to the two algorithms previously discussed. The SPED algorithm outperforms the other methods. It has the lowest error value ranging between 6 kHz, and 7 kHz as shown on the graph. The algorithm's estimate of frequency values is relatively constant over the range of SNR values.

The slight difference noticed could be attributed to precision errors because of floating-point representation. This means that the algorithm can estimate frequency values at very low SNR levels. The SPED algorithm has the advantage of being able to estimate multiple frequencies within a damped sinusoid while the other methods are only able to estimate single frequencies.

The simulations in Multisim model a wireless passive sensor with a resonant frequency

Table 4.5: Comparison of the SPED algorithm with other signal processing methods when tested on signals from the Multisim simulation

Signal Frequency	Frequency Estimate (This Work)	Frequency Estimate (MPM)	Frequency Estimate (STMB)	Frequency Estimate (BY1LC)
200 MHz	165 MHz 185 MHz	200 MHz	200 MHz	200 MHz
310 MHz	282 MHz 305 MHz	313 MHz	313 MHz	313 MHz
1.56 GHz	1.54 GHz	1.562 GHz	1.561 GHz	1.561 GHz
2.58 GHz	2.62 GHz	2.584 MHz	2.584 MHz	2.583 MHz
3 GHz	3.06 GHz 3.2 GHz	3 GHz	3 GHz	3 GHz

of 2.57 GHz [40]. The waveforms from this simulation represent the ringback signals received from the wireless passive sensor using an antenna. The algorithm was tested on a range of frequencies and the results are shown in Table 4.5. For the signal with an actual frequency of 200 MHz, we see that the SPED algorithm provided two estimates for the frequency both of which are between a 35 MHz difference of the actual frequency. From the table, we see that the other algorithms are able to accurately estimate the actual frequency of 200 MHz. In the case of the 310 MHz actual frequency, we see that the algorithm once again provides two signal estimates that are between a $\pm$ 28 MHz difference of the actual frequency. The other algorithms are able to estimate the actual frequency with a 3 MHz difference.

For higher frequencies in the GHz range, we see that in the case of 1.5 GHz, the algorithm proposed in this work provides an estimate that is 20 MHz different than the actual frequency. While the other algorithms are able to estimate the frequency to within 2 MHz of the actual frequency. For a signal having an actual frequency of 2.58 GHz, the SPED algorithm is able to estimate the frequency to within 40 MHz of the actual frequency while the existing methods are able to estimate the actual frequency within 4 MHz of the actual frequency. In the case of a signal with a frequency of 3 GHz, we notice that the SPED algorithm gives two estimates. The first estimate is within 60 MHz of the actual frequency while the other estimate is 200 MHz over the actual frequency. It is important to note that the algorithm was trained on frequencies within 0 Hz to 2.65 GHz with a resolution of 20 MHz and the large error in the frequency estimation value could be attributed to the fact that the actual frequency is out of the range of frequencies that the algorithm was trained on.

Overall, the performance of the SPED algorithm suggests that it can estimate the frequency contained in a damped sinusoidal signal within ± 40 MHz, and as such the algorithm is viable for real-world applications depending on the level of accuracy needed. This estimation accuracy could be further improved by increasing the precision of the database grid during the training process of the deep learning model.

Several lab tests were carried out and the received ringback signals received from the wireless passive resonator after interrogation were down-converted using a local oscillator. The block diagram of the interrogation system showing its various components is given in Section A.2 of the appendix. The algorithm was tested on these down-converted ringback signals and the results are shown in Table 4.6. For the first signal with a frequency of 258.6 MHz, we see that the SPED algorithm performs better than the other existing algorithms. It provides two signal estimates, the first estimate of 258.6 MHz is very close to the actual frequency with a difference of 600 kHz while the second estimate has a difference of 27 MHz. The other algorithms MPM, STMB, and BY1-LC have frequency estimates that differ from the actual frequency by 14 MHz, 8 MHz, and 9 MHz respectively. The SPED model proposed in this work performs better than the other methods. For the second signal

Table 4.6: Comparison of the SPED algorithm with other signal processing methods when tested on ringback signals from the wireless passive sensor

Interrogator Frequency	LO Frequency	Downconverted Frequency	Frequency Estimate (This Work)	Frequency Estimate (MPM)	Frequency Estimate (STMB)	Frequency Estimate (BY1LC)
2571.4 MHz	2830 MHz	258.6 MHz	258 MHz 285 MHz	244 MHz	250 MHz	249 MHz
2574.4 MHz	2830 MHz	255.6 MHz	228 MHz 243 MHz	243 MHz	250 MHz	249 MHz
2574.5 MHz	2752 MHz	177.5 MHz	163 MHz 178 MHz	172 MHz	181 MHz	178 MHz
2579.4 GHz	2830 MHz	250.6 MHz	248 MHz	243 MHz	250 MHz	249 MHz

with a frequency value of 255.6 MHz, the algorithm proposed in this work provides two estimates that are within a 27 MHz difference from the actual frequency. The MPM, STMB, and BY1-LC algorithms provide estimates that vary from the actual frequency by 12 MHz, 5 MHz, and 6 MHz respectively. For the third signal with a frequency value of 177.5 MHz, the proposed method provides two estimates. The first estimate varies from the actual frequency by a difference of 500 kHz, while the second estimate varies with a difference of 6 MHz. The MPM, STMB, and BY1-LC algorithm vary from the actual frequency by 5 MHz, 4 MHz, and 500 kHz, respectively. In the case of the fourth signal with a frequency value of 250.6 MHz, the proposed method provides an estimate that is within a 2 MHz difference from the actual frequency while the MPM, STMB, and BY1-LC method provide estimates that differ from the actual frequency by 7 MHz, 600 kHz, and 1 MHz, respectively.

Overall, the results show that the algorithm proposed in this work is comparable to the existing signal processing methods as the frequency estimates across all four algorithms are close in value. The SPED algorithm is more advantageous because once the model is trained it takes less time than the other signal processing methods to estimate the frequencies in the damped sinusoidal signals. Also, the SPED model once it is trained can provide frequency estimates of signals without knowing the exact model of the signal or its disturbances.

4.5 Summary

In this chapter, the performance of the SPED algorithm was evaluated using waveforms from three different scenarios. The first evaluation was performed on waveforms generated from synthetic data. In the evaluations, the trained model was evaluated on $1,000$ test samples that were not part of the training or validation dataset and the results showed that the model has a variance score of 0.89 and an R^2 score of 0.88. The algorithm was trained on a different signal model and the result of this training is referred to as the Modified model. The modified model is trained using a double exponential signal waveform that has a gradual amplitude slope from 0 when compared with the amplitude slope of the single exponential signal waveform. The results show that this model has a variance score of 0.69 and an R^2 score of 0.64. The reduction in variance and R^2 scores can be attributed to the increasing complexity of the signal model used to train the modified model. It is important to note that the reduction in the values of these metrics does not mean that the model's performance is bad, it just shows that the model can be improved to achieve better values. The algorithm was also modified to estimate the damping factors of frequencies contained in a multi-sinusoidal signal and the resulting model is referred to as the Damping factor model. The results show that the damping factor model has a variance score of 0.84 and an R^2 score of 0.83.

The second evaluation was carried out using signals gotten from a Multisim simulation.

The results of this evaluation showed that at low frequencies, 0 Hz to 2.65 GHz, the algorithm was able to provide frequency estimations that were accurate up to 40 MHz. In the case where the signal was used to estimate the frequency for a 3 GHz waveform, the frequency estimates given by the model were 60 Hz and 200 MHz above the actual frequency. It is important to note that although the algorithm was trained on frequencies ranging from 0 Hz to 2.65 GHz, it was able to estimate the frequency of a damped sinusoidal signal outside its training boundaries.

The third evaluation was carried out using signals gotten from a wireless passive interrogator system and the results showed that the trained model was able to provide frequency estimations that were accurate up to 40 MHz. Overall, the performance of the algorithm based on the evaluation metrics shows that Deep Learning is a suitable tool for estimating signal parameters of damped sinusoidal signals.

Also, the SPED algorithm was compared with existing signal processing methods for multi-sinusoidal waveforms and the results were presented. The algorithm was compared against three existing signal processing methods namely, STMB, MPM, and BY1-LC algorithm. The results of the comparison show that the SPED algorithm outperforms the other methods when tested on synthetic data. When tested on data from the Multisim simulations, the SPED algorithms' performance is below that of the other algorithms as its estimated values are less accurate than the estimated values of the other algorithms. When tested on data from the wireless passive sensor, the SPED algorithms' performance is comparable to the performance of the other algorithms. The results show that the SPED algorithm is a viable signal processing method for damped sinusoidal signals in real-life applications. It has the advantage of being used over a wide range of frequencies depending on its training dataset and once trained it has a faster estimation time than the other methods.

Chapter 5

Concluding Remarks

5.1 Conclusions

In this thesis, the use of deep learning for signal parameter estimation of damped sinusoidal signals was investigated. The deep learning model referred to as SPED is a convolutional neural network and can be broken down into two modules, a discretization module, and a frequency estimation module.

The SPED model was trained using two different signal definitions and the resulting models are referred to as Conventional model and Modified model. The modified model was trained using a signal definition that modelled a double exponential waveform which has a more pronounced envelope for the damped sinusoidal signal. The modified and conventional model were evaluated on the test dataset using regression metrics and their variance scores were 0.69 and 0.89 respectively, while their R^2 scores were 0.64 and 0.88. The difference is variance scores and R^2 scores between the two models can be attributed to the varying complexity of the signal definition waveforms. The signal waveform for the modified model is noticeably more complex than that of the conventional waveform and as a result, the variance and R^2 scores of the model are noticeably lower than that of the conventional model. This variance and R^2 score of the modified model could be improved by using a deep learning architecture with more layers which would be capable of learning more complex features than the existing SPED model.. The conventional model was tested on different signal waveforms and the results were discussed. The signal waveforms used in the test were of three types: synthetic data, data from a Multisim simulation of a wireless passive sensor and data obtained from tests using the wireless passive sensor. The frequency range for the waveforms ranged from 0 GHz to 2.65 GHz. A wide range of frequencies was chosen for the training data to ensure that the algorithm was trained on a robust dataset allowing it to be used to estimate frequencies over a wide range. The results of the tests on the Multisim simulation data and the data from tests using the wireless passive sensor showed that the deep learning model is able to estimate the frequencies contained in damped multi-sinusoidal signals up to an accuracy of $\pm$ 40 MHz. The deep learning model was also modified and used to estimate the damping factors of different frequencies contained in damped multi-sinusoidal signals and the performance of the model was evaluated using the test dataset. The results show that the Damping factor model has a variance score of 0.84 and R^2 score of 0.83 which implies that the model can fit the test data quite well.

The conventional model was then compared to existing signal processing methods and the results showed that the estimated values of frequencies given by the deep learning model is comparable to the estimated frequency values given by the other existing methods. The deep learning model has the advantage of being able to estimate multiple frequencies while the existing signal processing methods are limited to providing frequency estimate values for single sinusoidal signals. The results from the tests and evaluations carried out in this work show that deep learning is a viable artificial intelligence technique that can be applied in signal processing for parameter estimation.

5.2 Future Work

In this thesis, a signal parameter estimation algorithm consisting of convolutional neural networks was developed and tested. Waveform samples of damped multi-sinusoidal signals were generated, and the performance of the algorithm was evaluated using these samples. The algorithm was also modified to estimate the damping factors of damped multi-sinusoidal signals and the models performance was evaluated and the results were presented. The proposed algorithm can be improved on to enhance its accuracy and computational efficiency.

In this section, possible improvements and further studies are suggested to improve on the existing algorithm.

- Fifteen layers were used in the development of this algorithm, using more layers to increase the number of fine details that can be learnt during training could be explored as an increase in the number of layers could possibly lead to an increase in the accuracy of the parameters estimated by the algorithm.

- The ReLu activation function was used in the proposed algorithm. The use of different activation functions could be explored to determine how they influence the computational efficiency and accuracy of the model.

- The concept of prediction could be further explored such that over a time period the model is able to accurately predict the frequency of signals at a particular location. This could be useful in monitoring and detecting changes in signals at a location.

- The possibility of using the algorithm for real time signal processing could also be explored. Right now, the algorithm works offline, it would be helpful to explore how it would work for real time signal analysis.

- The concept of using reinforced learning or unsupervised learning to train a signal parameter estimation model could be explored.

- The algorithm currently uses the Adam optimizer to update its network attributes during its training process. The use of other optimization methods could be explored as this could lead to an improvement in the accuracy of the models signal parameter estimates.

- The performance of the algorithm can be further investigated alongside other existing signal processing algorithms not explored in this thesis.

- A more robust assortment of data samples can be generated and used to train the algorithm such that it can be applied on different types of signals and still have a good accuracy estimate of the frequencies and damping factors.

- The effect of signal amplitude on the performance of the deep learning algorithm can also be explored to determine if the algorithm performs better when trained on signals within a specific amplitude range.

- Alternative deep-learning methods for estimating the parameters of the signal could also be explored.

References

[1] G. Izacard, S. Mohan, and C. Fernandez-Granda, "Data-driven estimation of sinusoid frequencies," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[2] C. Baum, "The singularity expansion method: Background and developments," *IEEE Antennas and Propagation Society Newsletter*, vol. 28, no. 4, pp. 14–23, 1986.

[3] D. Riley, W. Davis, and I. Besieris, "The singularity expansion method and multiple scattering," *Radio Science - RADIO SCI*, vol. 20, pp. 20–24, 01 1985.

[4] *Singularity Expansion Method for Data Extraction for Chipless RFID.* John Wiley and Sons, Ltd, 2016, ch. 4, pp. 71–92.

[5] L. Palmer, "Coarse frequency estimation using the discrete fourier transform," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 104–109, 1974.

[6] C. Candan, M. Kutay, and H. Ozaktas, "The discrete fractional fourier transform," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 3, 1999, pp. 1713–1716 vol.3.

[7] E. Robles, S. Ceballos, J. Pou, J. L. Martín, J. Zaragoza, and P. Ibañez, "Variable-frequency grid-sequence detector based on a quasi-ideal low-pass filter stage and a phase-locked loop," *IEEE Transactions on Power Electronics*, vol. 25, no. 10, pp. 2552–2563, 2010.

[8] L. Wang, Q. Jiang, L. Hong, C. Zhang, and Y. Wei, "A novel phase-locked loop based on frequency detector and initial phase angle detector," *IEEE Transactions on Power Electronics*, vol. 28, no. 10, pp. 4538–4549, 2013.

[9] M. Rahman and K.-B. Yu, "Total least squares approach for frequency estimation using linear prediction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1440–1454, 1987.

[10] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source localization and sensing: A nonparametric iterative adaptive approach based on weighted least squares," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 425–443, 2010.

[11] Y. Hua and T. Sarkar, "Generalized pencil-of-function method for extracting poles of an EM system from its transient response," *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 2, pp. 229–234, 1989.

[12] ——, "Matrix pencil method and its performance," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 2476–2479 vol.4.

[13] A. Kuritsyn, M. Kharlamov, S. Prokhorov, and D. Shcherbinin, "Application of artificial intelligence systems in the process of crew training," in *2018 International Conference on Artificial Intelligence Applications and Innovations (IC-AIAI)*, 2018, pp. 55–59.

[14] K. Nikolskaia and V. Naumov, "Artificial intelligence in law," in *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2020, pp. 1–4.

[15] S. Aggarwal and A. Kumar, "A smart irrigation system to automate irrigation process using IOT and artificial neural network," in *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, 2019, pp. 310–314.

[16] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 4–15, 2021.

[17] A. Nayak and K. Dutta, "Impacts of machine learning and artificial intelligence on mankind," in *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017, pp. 1–3.

[18] S. Matzka, "Using process quality prediction to increase resource efficiency in manufacturing processes," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, 2018, pp. 110–111.

[19] J. M. Baker, L. Deng, J. Glass, S. Khudanpur, C.-h. Lee, N. Morgan, and D. O'Shaughnessy, "Developments and directions in speech recognition and understanding, part 1 [dsp education]," *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 75–80, 2009.

[20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 10, p. 436–444, 2015.

[21] A. Barnard, “The nursing profession: Implications for AI and natural language processing,” in *2007 International Conference on Natural Language Processing and Knowledge Engineering*, 2007, pp. 497–501.

[22] C. S. Pereira, R. Morais, and M. J. C. S. Reis, “Recent advances in image processing techniques for automated harvesting purposes: A review,” in *2017 Intelligent Systems Conference (IntelliSys)*, 2017, pp. 566–575.

[23] D. Ufuah, “A data augmentation approach using style-based generative adversarial networks for date fruit classification,” Master’s thesis, University Of Manitoba, May 2022.

[24] M. S. Fuentes, N. A. L. Zelaya, and J. L. O. Avila, “Coffee fruit recognition using artificial vision and neural networks,” in *2020 5th International Conference on Control and Robotics Engineering (ICCRE)*, 2020, pp. 224–228.

[25] S. Puttemans, Y. Vanbrabant, L. Tits, and T. Goedemé, “Automated visual fruit detection for harvest estimation and robotic harvesting,” in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2016, pp. 1–6.

[26] H. Altaheri, M. Alsulaiman, and G. Muhammad, “Date fruit classification for robotic harvesting in a natural environment using deep learning,” *IEEE Access*, vol. 7, pp. 117 115–117 133, 2019.

[27] A. Kausar, M. Sharif, J. Park, and D. R. Shin, “Pure-CNN: A framework for fruit images classification,” in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 404–408.

[28] J. Zhang, Y. Xie, Y. Xia, and C. Shen, “Attention residual learning for skin lesion classification,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 9, pp. 2092–2103, 2019.

[29] F. Ercal, A. Chawla, W. Stoecker, H.-C. Lee, and R. Moss, “Neural network diagnosis of malignant melanoma from color images,” *IEEE Transactions on Biomedical Engineering*, vol. 41, no. 9, pp. 837–845, 1994.

[30] H. H. Sultan, N. M. Salem, and W. Al-Atabany, “Multi-classification of brain tumor images using deep neural network,” *IEEE Access*, vol. 7, pp. 69 215–69 225, 2019.

[31] Y. Liu, Y. Li, Y. Zhu, Y. Niu, and P. Jia, “A brief review on deep learning in application of communication signal processing,” in *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, 2020, pp. 51–54.

[32] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, “A learning-based approach to direction of arrival estimation in noisy and reverberant environments,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 2814–2818.

[33] K. Terabayashi, R. Natsuaki, and A. Hirose, “Ultrawideband direction-of-arrival estimation using complex-valued spatiotemporal neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1727–1732, 2014.

[34] M. Agatonovic, Z. Stanković, and B. Milovanović, “High resolution two-dimensional DOA estimation using artificial neural networks,” in *2012 6th European Conference on Antennas and Propagation (EUCAP)*, 2012, pp. 1–5.

[35] Z.-M. Liu, C. Zhang, and P. S. Yu, “Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.

[36] Y. Chen, K.-L. Xiong, and Z.-T. Huang, “Robust direction-of-arrival estimation via sparse representation and deep residual convolutional network for co-prime arrays,” in *2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)*, 2020, pp. 514–519.

[37] G. Izacard, B. Bernstein, and C. Fernandez-Granda, “A learning-based framework for line-spectra super-resolution,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3632–3636.

[38] I. Sajedian and J. Rho, “Accurate and instant frequency estimation from noisy sinusoidal waves by deep learning,” *Nano Convergence*, vol. 6, no. 1, pp. 1–5, 08 2019.

[39] P. Pan, Y. Zhang, Z. Deng, and W. Qi, “Deep learning-based 2-D frequency estimation of multiple sinusoidals,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5429–5440, 2022.

[40] A. Amirkabiri, D. Idoko, B. Kordi, and G. E. Bridges, “High-Q contactless air-filled substrate-integrated waveguide (CLAF-SIW) resonator for wireless sensing applications,” in *2021 IEEE MTT-S International Microwave Symposium (IMS)*, 2021, pp. 577–580.

[41] *Digital Spectral Analysis: Parametric, Non-Parametric and Advanced Methods.* John Wiley and Sons, Ltd, 2011, ch. 1, pp. 1–22.

[42] K. Duda and T. P. Zielinski, “Efficacy of the frequency and damping estimation of a real-value sinusoid part 44 in a series of tutorials on instrumentation and measurement,” *IEEE Instrumentation Measurement Magazine*, vol. 16, no. 2, pp. 48–58, 2013.

[43] K. Wang, H. Wen, and G. Li, "Accurate frequency estimation by using three-point interpolated discrete fourier transform based on rectangular window," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 73–81, 2021.

[44] V. K. Jain, W. L. Collins, and D. C. Davis, "High-accuracy analog measurements via interpolated FFT," *IEEE Transactions on Instrumentation and Measurement*, vol. 28, no. 2, pp. 113–122, 1979.

[45] T. Grandke, "Interpolation algorithms for discrete fourier transforms of weighted signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 32, no. 2, pp. 350–355, 1983.

[46] W. Cochran, J. Cooley, D. Favin, H. Helms, R. Kaenel, W. Lang, G. Maling, D. Nelson, C. Rader, and P. Welch, "What is the fast fourier transform?" *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 45–55, 1967.

[47] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[48] K. Duda, "Analysis of systematic errors of frequency and damping estimation by interpolated DFT BY1 algorithm," *2012 International Conference on Signals and Electronic Systems, ICSES 2012 - The Conference Proceedings*, pp. 1–4, 2012.

[49] K. Duda, L. B. Magalas, M. Majewski, and T. P. Zielinski, "DFT-based estimation of damped oscillation parameters in low-frequency mechanical spectroscopy," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3608–3618, 2011.

[50] G. Andria, M. Savino, and A. Trotta, "Windows and interpolation algorithms to improve electrical measurement accuracy," *IEEE Transactions on Instrumentation and Measurement*, vol. 38, no. 4, pp. 856–863, 1989.

[51] D. C. Rife and G. A. Vincent, "Use of the discrete fourier transform in the measurement of frequencies and levels of tones," *The Bell System Technical Journal*, vol. 49, no. 2, pp. 197–228, 1970.

[52] I. Yoshida, T. Sugai, S. Tani, M. Motegi, K. Minamida, and H. Hayakawa, "Automation of internal friction measurement apparatus of inverted torsion pendulum type," *Journal de Physique Colloques*, vol. 42, no. C5, pp. C5–1123–C5–1128, 1981.

[53] M. Bertocco, C. Offelli, and D. Petri, "Analysis of damped sinusoidal signals via a frequency-domain interpolation algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 43, no. 2, pp. 245–250, 1994.

[54] J. Schoukens, R. Pintelon, and H. Van Hamme, "The interpolated fast fourier transform: a comparative study," *IEEE Transactions on Instrumentation and Measurement*, vol. 41, no. 2, pp. 226–232, 1992.

[55] W. Jenkins and M. Desai, "The discrete frequency fourier transform," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 7, pp. 732–734, 1986.

[56] R. Diao and Q. Meng, "An interpolation algorithm for discrete fourier transforms of weighted damped sinusoidal signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 6, pp. 1505–1513, 2014.

[57] T. P. Zielinski and K. Ostrowska, "Application of Bertocco-Yoshida interpolated DFT algorithm to nmr data analysis," in *2016 International Conference on Signals and Electronic Systems (ICSES)*, 2016, pp. 63–67.

[58] R.-C. Wu and C.-T. Chiang, "Analysis of the exponential signal by the interpolated DFT algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3306–3317, 2010.

[59] D. Belega, D. Petri, and D. Dallet, "Frequency estimation of a sinusoidal signal via a three-point interpolated DFT method with high image component interference rejection capability," *Digital Signal Processing*, vol. 24, p. 162–169, 01 2014.

[60] K. Wang, H. Wen, W. Tai, and G. Li, "Estimation of damping factor and signal frequency for damped sinusoidal signal by three points interpolated DFT," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1927–1930, 2019.

[61] K. Wang, H. Wen, L. Xu, and L. Wang, "Two points interpolated DFT algorithm for accurate estimation of damping factor and frequency," *IEEE Signal Processing Letters*, vol. 28, pp. 499–502, 2021.

[62] R. Prony, "Essai experimental et analytique," *Journal de l'école Polytechnique de Paris*, vol. 1, pp. 24–76, 1795.

[63] M. Van Blaricum and R. Mittra, "A technique for extracting the poles and residues of a system directly from its transient response," *IEEE Transactions on Antennas and Propagation*, vol. 23, no. 6, pp. 777–781, 1975.

[64] Y. Hua and T. Sarkar, "A discussion of E-pulse method and prony's method for radar target resonance retrieval from scattered field," *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 7, pp. 944–946, 1989.

[65] T. K. Sarkar, M. Salazar-Palma, M. D. Zhu, and H. Chen, *"Matrix Pencil Method (MPM)," in Modern Characterization of Electromagnetic Systems and its Associated Metrology*, 2021, pp. 21–106.

[66] T. Sarkar, J. Nebat, D. Weiner, and V. Jain, "Suboptimal approximation/identification of transient waveforms from electromagnetic systems by pencil-of-function method," *IEEE Transactions on Antennas and Propagation*, vol. 28, no. 6, pp. 928–933, 1980.

[67] V. Jain, "Filter analysis by use of pencil of functions: Part I," *IEEE Transactions on Circuits and Systems*, vol. 21, no. 5, pp. 574–579, 1974.

[68] D. Pradhan and R. Bera, "Direction of arrival estimation via ESPRIT algorithm for smart antenna system," *International Journal of Computer Applications*, vol. 118, pp. 5–7, 05 2015.

[69] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[70] K. Steiglitz and L. McBride, "A technique for the identification of linear systems," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 461–464, 1965.

[71] R. E. Kalman, "Design of a self-optimizing control system," *Transactions of ACME*, vol. 80, no. 2, pp. 468–478, 1958.

[72] P. Stoica and T. Soderstrom, "The Steiglitz-McBride identification algorithm revisited–convergence analysis and accuracy aspects," *IEEE Transactions on Automatic Control*, vol. 26, no. 3, pp. 712–717, 1981.

[73] J. McClellan and D. Lee, "Exact equivalence of the Steiglitz-McBride iteration and IQML," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 509–512, 1991.

[74] P. Tomasz, T. Zielinski, and K. Duda, "Frequency and damping estimation methods - an overview," *Metrology and Measurement Systems*, vol. 18, 01 2011.

[75] J. Guo, "Research on artificial intelligence: Deep learning to identify plant species," in *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE)*, 2022, pp. 59–66.

[76] K. C. Kirana, S. Wibawanto, N. Hidayah, G. P. Cahyono, and K. Asfani, "Improved neural network using integral-relu based prevention activation for face detection," in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, vol. 6, 2019, pp. 260–263.

[77] K. Bai, "Gaussian kernels," February 2019. [Online]. Available: https://towardsdatascience.com/669281e58215

[78] J. Johnson, "Kernels," May 2013. [Online]. Available: https://shapeofdata.wordpress.com/2013/05/27/kernels/

[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[80] A. D'Andrea, U. Mengali, and R. Reggiannini, "The modified cramer-rao bound and its application to synchronization problems," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 1391–1399, 1994.

[81] M. Yazdani, D. J. Thomson, and B. Kordi, "Passive wireless sensor for measuring AC electric field in the vicinity of high-voltage apparatus," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4432–4441, 2016.

Appendix A

A.1 Cramér-Rao Bound

This is also referred to as the Cramér-Rao Lower Bound (CRLB). It expresses a lower bound on the variance of an unbiased estimator [80] i.e., it gives information on how well the estimator would perform. An estimator that achieves this lower bound is considered fully efficient as its solution achieves the lowest possible MSE among all unbiased methods. The CRLB is calculated according to [80]

$$CRB(\eta) = \frac{1}{E_y\Big\{\Big[\frac{\partial \ln p(y|\eta)}{\partial \eta}\Big]^2\Big\}}, \tag{A.1}$$

where η denotes the estimation of a single element, E_y denotes the the statistical expectation of y which is a finite-dimensional vector and $p(y|\eta)$ is the probability density function of r given η.

A.2 Interrogator System Components

The components used to construct the interrogation system are shown in the block diagram of Fig. A.1.

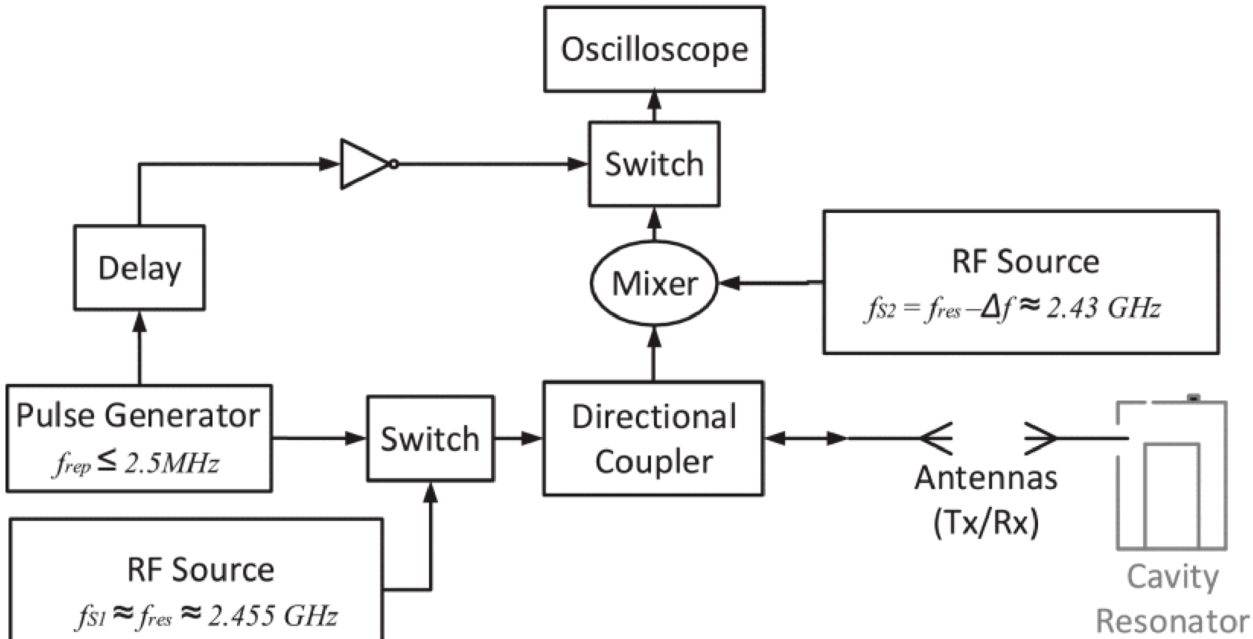


Fig. A.1: Block diagram showing components used in the setup for interrogation system. Adapted from [81] with permission.