SIGNAL EXTRAPOLATION WITH LINEAR PROLATE FUNCTIONS

by

Amal Devasia

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
March 2015

To my dear brother, parents and close friends.

Table of Contents

List of Tables

List of Figures

Abstract

A simple, efficient and reliable method for bandlimited signal extrapolation valid up to basically an arbitrarily high range of frequencies is proposed. The orthogonal properties of linear prolate functions (LPFs) are exploited to form an orthogonal basis set needed for synthesis. A significant step in the process is the higher order piecewise polynomial approximation of the overlap integral required for obtaining the expansion coefficients accurately with very high precision. Two sets of LPFs, one relatively with lower Slepian frequency and the other with higher Slepian frequency, are considered. Numerical results of extrapolation of some standard test signals using our algorithm for the two sets are discussed in detail. Further comparisons show that the proposed method performs optimally over other recent techniques used. The possibility of using our algorithm for extrapolating real experimental signals is also explored. Though it is not fully usable for extrapolating real signals as such, there were some interesting observations made during this study which could be of some help for the future works in this field.

List of Symbols Used

$f(t)$	Continuous function of time
t	Continuous time
t_0	Interval limits for time
ω	Angular frequency
$F(\omega)$	Fourier transform of $f(t)$
Ω	Bandwidth of continuous signal
$\Psi_n(c,t)$	Prolate Spheroidal Wave Function
m,n	Order
c	Slepian frequency
$\lambda_n(c)$	Eigenvalue of *sinc* kernel with PSWFs as the eigenfunction
Ω_0	Finite bandwidth of PSWF
$S_{m,n}(c,\eta)$	Angular solution
x	Continuous time shift
$R_{m,n}(c,\varepsilon)$	Radial solution
N	Truncation values of the Slepian series
(x_k, y_k)	(Discrete instances of time, discrete instances of continuous PSWFs)
i,k	Index variables
$\beta_0 \ldots \beta_{k-1}$	Scalar coefficients of the $(k-1)^{\text{th}}$ order polynomial
η	Angular domain
ε	Radial domain
ϑ	Azimuthal domain
$\gamma_n(c)$	Scalar expansion coefficients of Slepian series
$\Phi_n(t)$	General orthogonal basis function

γ_n	General analysis coefficient
M	Number of time samples of the function for calculating DFT
$f[i]$	Time sampled signal
$ReF[k]$	Frequency domain signals obtained for the cosine waves during analysis of DFT
$ImF[k]$	Frequency domain signals obtained for the sine waves during analysis of DFT
$Re\bar{F}[k]$	Contain amplitudes of the cosine waves for the calculation of inverse DFT
$Im\bar{F}[k]$	Contain amplitudes of the sine waves for the calculation of inverse DFT
j	Imaginary unit
∇^2	Laplace operator
ν	$\frac{2\pi}{\lambda\prime}$
r'	Curvilinear coordinate system
l	Inter-focal distance
λ'	Wavelength
f_e	Extrapolated signal
f	Original known signal

Acknowledgements

I would like to thank Michael Cada, my supervisor for this research, for providing me an opportunity to work on this topic. His valuable insights of the subject have provided me the clarity needed to work on this research. His guidance and continuous support is much appreciated.

The helpful suggestions of Jose Gonzalez-Cueto, whose firsthand knowledge of the signal processing techniques have also helped me a great deal in comprehending certain aspects or ideas and contributing towards this research on the whole. It was he who provided some real medical experimental data on which I tested my algorithm. I would also like to thank William Phillips for being part of my examining committee and for his support whenever required.

I would also like to thank each and every member of my research group in the Photonics Applications Laboratory for their productive discussions and sharing their views, not only on this subject but on a broader range of topics. A special thanks goes to Raunaq Singh Kainth who was with me during the initial stages of my research. His contributions to formulate a better numerical integration model are much appreciated. I am also thankful to Mohammed Eldlio for encouraging me to publish my work during the course of this research. Partial support by Applied Science in Photonics and Innovative Research in Engineering (ASPIRE), a program under Collaborative Research and Training Experience (CREATE) program funded by Natural Sciences and Engineering Research Council (NSERC) of Canada, is also acknowledged.

I am also indebted to Ajin Joy and Gregary Paul whose constant encouragement and wise words have helped me see through tough times that occurred during my research years.

Of course, I am grateful to my brother, my relatives (especially those living in Halifax) and my parents for their patience, understanding and support.

Halifax, NS Amal Devasia

February 12, 2015

Chapter 1

Introduction

A remarkable discovery was made about half a century ago by David Slepian, an American mathematician, and his colleagues on a special set of functions called prolate spheroidal wave functions (PSWFs). These functions, also known as Slepian prolate functions, were bandlimited and exhibited interesting orthogonality relations. They are normalized versions of the solutions to Helmholtz wave equation [1] in spheroidal coordinates. In his paper [2-5], Slepian proposed the idea of bandlimited signal extrapolation using PSWFs. Generating this set of functions practically seemed difficult because of the complexity involved and limited computational capabilities existed. Hence, there hasn't been any significant interest in this field up until very recently.

Signal extrapolation is an extension of a signal, $f(t)$, beyond the interval in which it is known to the observer. Symbolically, we can represent it as:

$$f(t) = \begin{cases} known, & |t| \leq t_0 \\ unknown\ (extrapolation), & |t| > t_0 \end{cases} . \tag{1.1}$$

The region in which the signal is known is called the observation interval (here, $[-t_0, t_0]$). Bandlimited signals are bound in the frequency domain; their Fourier transform, $F(\omega)$, vanishes beyond a particular finite frequency interval. Thus if,

$$F(\omega) = 0, \quad |\omega| > \Omega, \tag{1.2}$$

where ω is the angular frequency, then $f(t)$ is said to be Ω-bandlimited.

1.1 Recent Advancements in Signal Extrapolation

With advanced numerical techniques and superior computational power, there has been noticeable activity in the field of signal extrapolation within the past decade [6,7,8,9,39]. In [10], Senay *et al.* proposed sampling and reconstruction of bandlimited as well as non-bandlimited signals using Slepian functions. They discussed the idea of modifying the Whittaker-Shannon sampling theory by replacing the *sinc* basis by Slepian functions for reconstruction of signals. Further to this, in [11,12], they showed signal reconstruction using non-uniform sampling and level-crossing sampling with Slepian functions.

Since the primary focus of this research is on bandlimited signal extrapolation and not just reconstruction or interpolation, we will concentrate more on the recent advancements in this regard. While we consider the signals to be bandlimited in the Fourier transform domain, much attention has recently been on extrapolation of signals bandlimited in linear canonical transform (LCT) domain; this being a four-parameter family of linear integral transform [13,14] that generalizes Fourier transform as one of its special cases. For extrapolation of LCT bandlimited signals, several iterative and non-iterative algorithms have been proposed [15-18]. Most of the iterative algorithms are centered on modifying the Gerchberg-Papoulis (GP) algorithm [19,20] that relies on successive reduction of error energy. Although theoretical convergence of the result has been shown, there is still some uncertainty associated with the swiftness with which this is achieved. With respect to the non-iterative algorithms proposed by Zhao *et al.* in [16], the authors

themselves admit that the extrapolation could become unstable with an increase in the number of observations. A comparison of the extrapolation of an LCT bandlimited signal, using an iterative GP algorithm and another algorithm based on signal expansion into a series of generalized PSWFs [18] is presented in [17]. The comparison showed better results for the iterative method proposed in [17] over the one described in [18], in terms of the normalized mean square error (NMSE).

In another significant study [21], Gosse, an Italian mathematician performed Fourier bandlimited signal extrapolation by handling lower and higher frequencies of the signal separately. He used PSWFs for extrapolating lower frequency components while the higher frequencies were dealt with compressive sampling [22,23] algorithms. The efficiency of the proposed method was highly dependent on the correlation between low and high frequencies in the signal (it should be weak for better results), the existence of a sparse representation of higher frequencies in the Fourier basis, and on a reasonable choice of extrapolation domain.

1.2 Motivation for Research

Signal extrapolation can be used for various applications pertaining to the characterization of entities in the physical world. Prediction or forecasting of events is not just confined to any particular discipline but commonly to almost all the scientific fields. In relation to photonics and nanotechnology, it can be used in the analysis of distribution of light in optical fibers varying in their dimensions. It would be a convenient tool for developing mathematical models to predict light fluctuation in various lasers (for example, due to mode-hopping noise). Due to its excellent property of invariance to the

finite Fourier transform, it would be apt to use PSWFs for solving some of the differential equations that arise in the field of photonics. Extrapolation can also be applied in the field of image communication for implementing error concealment techniques as well as for image compression. In the field of signal processing, it could play a key role in signal recovery (optical, audio) as well as noise elimination. Forecasting or prediction of the behavior of financial signals is another important area where this can be applied.

The mere idea of predicting a signal behavior accurately from a finite observation range, combined with the numerous applications it can directly have an impact on, excited us and galvanized our minds into doing this research. It was also our objective to come up with an efficient implementation of Slepian's proposed method of bandlimited signal extrapolation.

1.3 Features of the Proposed Algorithm

In this book, we propose a non-iterative and simple method for extrapolating bandlimited signals varying from relatively lower to higher frequencies using linear prolate functions (one-dimensional PSWFs, henceforth abbreviated as LPFs). Although we concentrate mainly on Fourier bandlimited signals, it however might also be applied for LCT bandlimited cases as is shown in one of the results. Several comparisons are made with the results obtained in earlier related publications. They show that, within the prescribed bandwidth, the proposed method is far superior over several other methods referenced in this book.

LPFs for analysis purposes need to be computed accurately and with rather high precision. Here, we rely on a proprietary algorithm developed theoretically and

implemented numerically by Cada [24], for accurately generating the linear prolate functions set with desired high precision. Once the LPFs set is obtained with the corresponding eigenvalues (discussed in the following chapters of this book), they are employed in our algorithm for extrapolation. Here, we do not consider the storage of LPFs set as an issue (as put forward by Shi et al. in [17]) to be addressed, as it is not the primary objective of our work.

Cada's algorithm exploits robust properties of certain formulae derived that are efficient, accurate and suitable for fast numerical evaluations of LPFs and their eigenvalues. Previous methods (Slepian, Flammer [25], etc.) required lengthy cumbersome calculations with slowly converging series and necessary approximations that led to insurmountable numerical problems and/or failing when higher orders were concerned. Prolate functions and the eigenvalues change their properties drastically at certain parameter values (see chapter 2 and 3), which has caused described problems. Even standard professional high-quality commercial packages such as Mathematica or Matlab fail to compute these functions and eigenvalues correctly or at all for such a combination of parameters that is critical for extrapolation applications. His algorithm enables to break through this numerical barrier and makes it possible to calculate linear prolate functions and their eigenvalues correctly for basically any parameters.

1.4 Scope of the Thesis

The material presented in the book can also be found partly in [28,34,35] which were published during the course of this research. The following chapters of the book are organized as follows. In chapter 2, LPFs and their relevant properties are discussed. The

various steps involved in our proposed extrapolation algorithm are described throughout in chapter 3. Chapter 4 is devoted to presenting the actual extrapolated results of various test functions, their comparison and error analysis. Here we also look into the possibility of extrapolating some real experimental signals. Our algorithm is shown to work optimally for ideal signals which are highly precise (preferably with an accuracy of 100 to 200 digits after the decimal point) and noise-free. These tests have helped us to characterize our model, to understand the optimal input conditions needed for the signal, for it to be extrapolated efficiently. Finally, in chapter 5, we conclude with our inferences and possible future prospects.

Chapter 2

Linear Prolate Functions

The theoretical treatment of bandlimited signal extrapolation using PSWFs was first discussed by Slepian and his colleagues in [2]. They explained the use of PSWFs, or more precisely, linear prolate functions (LPFs) as an orthogonal basis set for decomposition and reconstruction of the signal using analysis and synthesis equations. Linear prolate functions are one-dimensional PSWFs denoted by $\Psi_n(c,t)$, where n is the order of LPF (non-negative integer), t is the time parameter and c is the bandwidth parameter also known as Slepian frequency. Following the notations of Moore in [8,33], LPFs can be evaluated using (2.1) as:

$$\Psi_n(c,t) = \sqrt{\frac{\lambda_n(c)/t_0}{\int_{-1}^{1}(S_{0n}(c,t))^2 dt}}\, S_{0n}\left(c,\frac{t}{t_0}\right), \tag{2.1}$$

$$\int_{-t_0}^{t_0} \Psi_n(c,t)\frac{\sin(\Omega_0(x-t))}{\pi(x-t)}\, dt = \Psi_n(c,x)\,\lambda_n(c) \tag{2.2}$$

where $\lambda_n(c)$ is the eigenvalue (a measure of concentration of signal in the observation interval [$-t_0$, t_0]) of *sinc* kernel system as shown in (2.2), with $\Psi_n(c,t)$ as eigenfunction, and t_0 is the observation boundary of the interval in which the function is known. $S_{m,n}(c,\eta)$ are the angular solutions of the first kind to Helmholtz wave equation [1] given by:

$$(\nabla^2 + v^2)U(r', c) = 0 \tag{2.3}$$

where,

$$v = \frac{2\pi}{\lambda'} \tag{2.4}$$

Here, λ' is the wavelength and $U(r', c)$ is the wave function expressed in the curvilinear coordinate system r' where

$$r' = (\eta, \varepsilon, \vartheta) \tag{2.5}$$

and

$$c = \frac{\pi l}{\lambda'} \tag{2.6}$$

l is the inter-focal distance within the ellipsoid.

The eigenvalue $\lambda_n(c)$ is given by,

$$\lambda_n(c) = \frac{2c}{\pi}[R_{0,n}(c, 1)^2], \tag{2.7}$$

where $R_{m,n}(c, \varepsilon)$ are the radial solutions of the first kind to Helmholtz wave equation.

Numerical evaluation of the LPFs set along with their corresponding eigenvalues practically seemed very difficult as it involved finding precise numerical values of the angular ($S_{m,n}(c, \eta)$) and radial ($R_{m,n}(c, \varepsilon)$) solutions. For obtaining this, a typical power series expansion was used which was predetermined by the association of Legendre and spherical Bessel functions to the angular and radial solutions respectively. Interested readers are referred to [8,9,26] for more details on LPFs derivation. This study was more about working with the LPFs set along with their eigenvalues for signal extrapolation

than generating these set of functions itself. These LPFs along with their corresponding eigenvalues, obtained by Cada's proprietary algorithm, have been used in our extrapolation algorithm. It should be noted that since extrapolation relies heavily on values of Ψ_n's and λ_n's for $n > 2c/\pi$, it is of paramount importance that one computes them accurately and with high precision. Ours is the first algorithm that offers such a capability.

2.1 Properties of LPFs

LPFs have many interesting properties of which some of the relevant ones related to this study are discussed below.

2.1.1 Bandlimited

Bandlimiting property of LPFs is denoted by a free bandwidth parameter (Slepian frequency) c given by:

$$c = \Omega_0 t_0, \tag{2.8}$$

where Ω_0 is the finite bandwidth of $\Psi_n(c,t)$ for a given order n.

2.1.2 Symmetry

LPFs exhibit even and odd symmetries based on their integer order n. If n is even, $\Psi_n(c,t)$ is even symmetric. If n is odd, then it is odd symmetric (figure 2.1, 2.2).

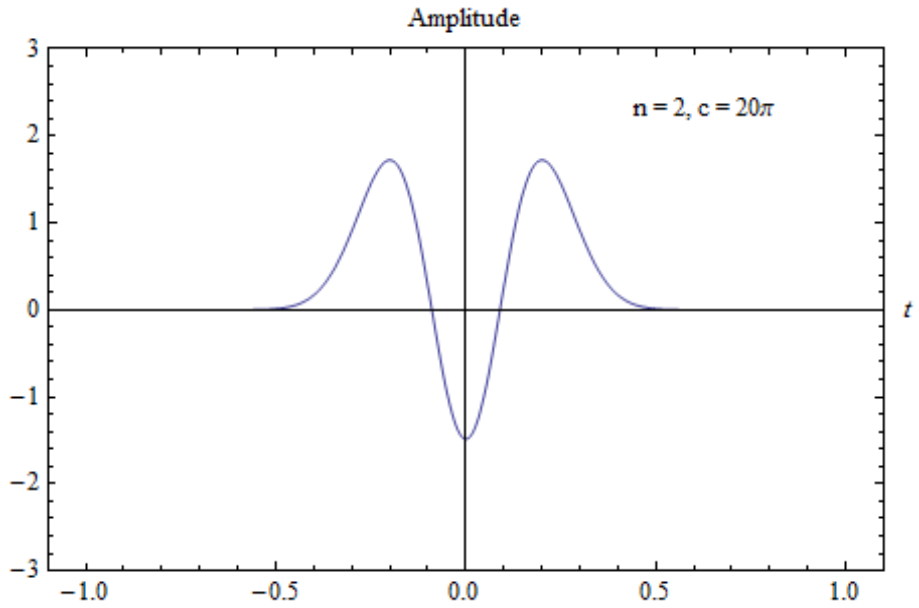

Figure 2.1: Even symmetry of $\Psi_n(c,t)$ for $c = 20\pi$

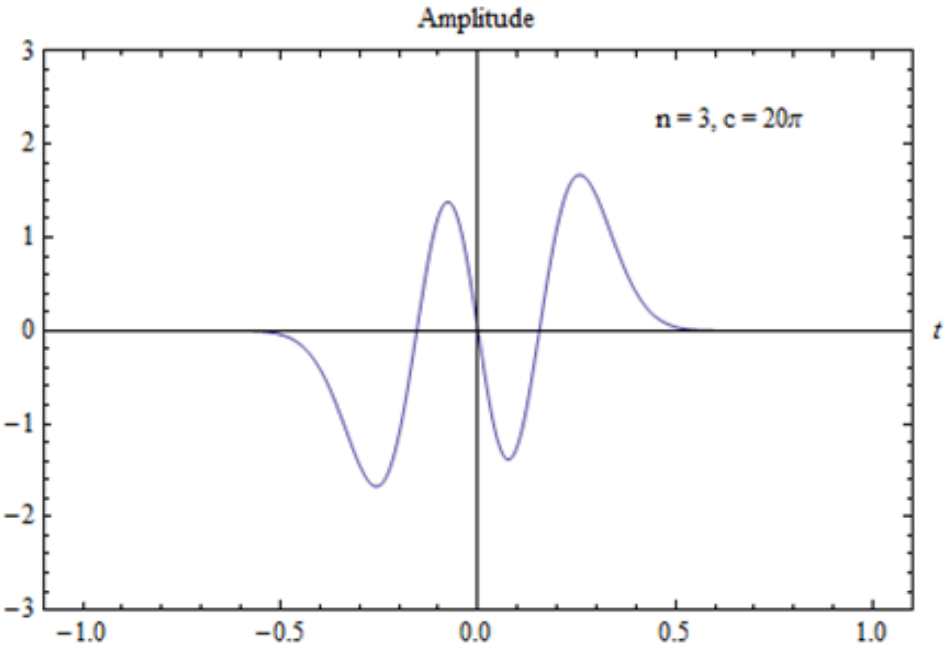

Figure 2.2: Odd symmetry of $\Psi_n(c,t)$ for $c = 20\pi$

2.1.3 Orthogonality

LPFs are linearly independent and orthogonal over finite as well as infinite intervals, unlike, for example, trigonometric functions that are orthogonal only over a finite domain.

$$\int_{-\infty}^{\infty} \Psi_n(c,t)\,\Psi_m(c,t)\,dt = \begin{cases} 1 & for\ n = m, \\ 0 & otherwise \end{cases} \quad \text{(infinite domain)} \qquad (2.9)$$

$$\int_{-t_0}^{t_0} \Psi_n(c,t)\,\Psi_m(c,t)\,dt = \begin{cases} \lambda_n(c)\ for\ n = m, \\ 0 \quad otherwise \end{cases} \quad \text{(finite domain)}, \qquad (2.10)$$

where n, m are non-negative integers. This property is greatly exploited for signal extrapolation especially in the analysis (3.3) and synthesis (3.4), explained in chapter 3.

2.1.4 Invariance to Fourier Transforms

Fourier transforms of LPFs over both finite and infinite intervals are simply scaled versions of themselves.

$$\int_{-t_0}^{t_0} \Psi_n(c,t)\mathrm{e}^{j\omega t}\,dt = j^n\left(\frac{2\pi\lambda_n(c)t_0}{\Omega_0}\right)^{1/2} \Psi_n\left(c,\frac{\omega t_0}{\Omega_0}\right) \quad \text{(finite domain)} \qquad (2.11)$$

$$\int_{-\infty}^{\infty} \Psi_n(c,t)\mathrm{e}^{j\omega t}\,dt = j^n\left(\frac{2\pi t_0}{\Omega_0}\right)^{1/2} \Psi_n\left(c,\frac{\omega t_0}{\Omega_0}\right) \quad \text{(infinite domain)} \qquad (2.12)$$

Expressions (2.11) and (2.12) show LPFs' invariance to Fourier transforms and are a further proof of their bandlimiting property.

2.2 Fourier Transforms and Bandlimiting

The Fourier transform $F(\omega)$ of a function $f(t)$ is defined [2,31] as,

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \mathrm{e}^{-j\omega t}\, dt \tag{2.13}$$

and, its inverse Fourier transform is given by,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \mathrm{e}^{j\omega t}\, d\omega \tag{2.14}$$

While the bandlimiting nature of continuous analog signals depend on their Fourier transforms, we can, for sure, say that any discrete-in-time (time sampled) signal or even the discrete counterparts of those analog signals mentioned above are inherently bandlimited due to sampling. The functions, even though continuous, are discretized in time for numerical computations on a digital computer. Hence, these functions would be bandlimited with frequency ranging from zero hertz to half of its sampling frequency [31,32]. While an analog signal could be made bandlimited by using an anti-aliasing filter, there is not a clear cut way of knowing whether a discrete-in-time signal (which is basically unknown) is bandlimited with or without aliasing. And hence, if such signals contain aliased frequencies (higher frequency components) in their spectrum, then treating them as bandlimited essentially may not be helpful for implementing Slepian's bandlimited extrapolation (discussed further in chapter 4).

The numerical implementation of the discrete Fourier transform (DFT) for analyzing the frequency domain characteristics of the time sampled signal can be formulated [32] as,

$$ReF[k] = \sum_{i=0}^{M-1} f[i] \cos(2\pi ki/M) \tag{2.15}$$

$$ImF[k] = -\sum_{i=0}^{M-1} f[i] \sin(2\pi ki/M) \tag{2.16}$$

where $f[i]$ is the time sampled signal being analyzed, $ReF[k]$ and $ImF[k]$ are the frequency domain signals obtained which gives a measure of the amplitudes of the cosine and sine waves, the index k runs from 0 to $M/2$, and the sampling index i runs from 0 to $M-1$. Thus, the total number of samples of the signal $f[i]$ will be M. Expressions (2.15) and (2.16) can be treated as the analysis equations for computing the DFT.

The synthesis equation, otherwise referred to as, the inverse DFT is given by,

$$f[i] = \sum_{k=0}^{\frac{M}{2}} Re\bar{F}[k] \cos(2\pi ki/M) + \sum_{k=0}^{\frac{M}{2}} Im\bar{F}[k] \sin(2\pi ki/M) \tag{2.17}$$

where $f[i]$ is the M- point signal being synthesized, with the index i running from 0 to $M-1$. $Re\bar{F}[k]$ and $Im\bar{F}[k]$ contain the amplitudes of the cosine and sine waves respectively, with the index k running from 0 to $M/2$. These amplitude functions, $Re\bar{F}[k]$ and $Im\bar{F}[k]$, are scaled versions of (2.15) and (2.16) obtained as per the formulae:

$$Re\bar{F}[k] = \frac{ReF[k]}{M/2} \tag{2.18}$$

$$Im\bar{F}[k] = -\frac{ImF[k]}{M/2} \tag{2.19}$$

with the exceptions for $Re\bar{F}[k]$ when k equals 0 or $M/2$, given as,

$$Re\bar{F}[0] = \frac{ReF[0]}{M} \tag{2.20}$$

$$Re\bar{F}[M/2] = \frac{ReF[M/2]}{M} \tag{2.21}$$

Here, generally, we consider the case when the signal $f[i]$ is real and not complex. The equations for the numerical calculation of the DFT and inverse DFT presented here, from (2.15) to (2.21), are valid only for the case when the signal being considered is real.

Chapter 3

The Extrapolation Method

Starting from the theoretical background and going through various numerical techniques involved, this chapter explains the steps with which we implemented our extrapolation algorithm. It should be noted, however, that though the LPFs considered here are functions continuous in the time domain, their numerical implementations on a digital computer requires it to be discretized or sampled. The functions were derived from Cada's algorithm which is rooted on their basic generation formulae described in (2.1) and (2.3).

3.1 Analysis and Synthesis

Generally speaking, any bandlimited signal can be decomposed into a linear combination of weighted orthogonal basis functions using the relation:

$$f(t) = \sum_{n=-\infty}^{\infty} \gamma_n \Phi_n(t) \qquad \text{(synthesis)}, \qquad (3.1)$$

where $f(t)$ is the signal, γ_n is a set of scalar coefficients, and $\Phi_n(t)$ is the orthogonal basis set.

The set of scalar coefficients is found from:

$$\gamma_n = \frac{\int_{-\infty}^{\infty} f(t)\Phi_n(t)dt}{\int_{-\infty}^{\infty} \Phi_n(t)\Phi_n(t)dt} \qquad \text{(analysis)} \qquad (3.2)$$

Employing LPFs as the orthogonal basis set for a fixed Slepian frequency c yields:

$$\gamma_n(c) = \lambda_n^{-1}(c) \int_{-t_0}^{t_0} f(t)\, \Psi_n(c,t)\, dt \qquad \text{(analysis)}, \tag{3.3}$$

where $\int_{-t_0}^{t_0} f(t)\, \Psi_n(c,t)\, dt$ is known as the overlap integral, and $\gamma_n(c)$ are the scalar expansion coefficient for a given order n of the LPFs set.

The synthesis equation, used for signal extrapolation, is given by:

$$f(t) \cong \sum_{n=0}^{N} \gamma_n(c)\, \Psi_n(c,t) \qquad \text{(synthesis)}, \tag{3.4}$$

where N is the truncation value for the order n.

The above equations (3.3) and (3.4) form the backbone of our signal extrapolation algorithm.

3.2 LPFs Set

Two distinct sets of LPFs were used as potential orthogonal basis sets for the proposed algorithm. These functions were discretized in time for numerical implementation. Each of these sampled data has very high numerical precision of about 200 digits. The two sets vary in their Slepian frequency; the specifics are as follows:

($\Psi_n(c,t)$ *Set 1)* $\quad c = 20\pi, \quad n: 0 \rightarrow 101, \quad t \rightarrow [-1.900, 1.900]$ (see figure 3.1).

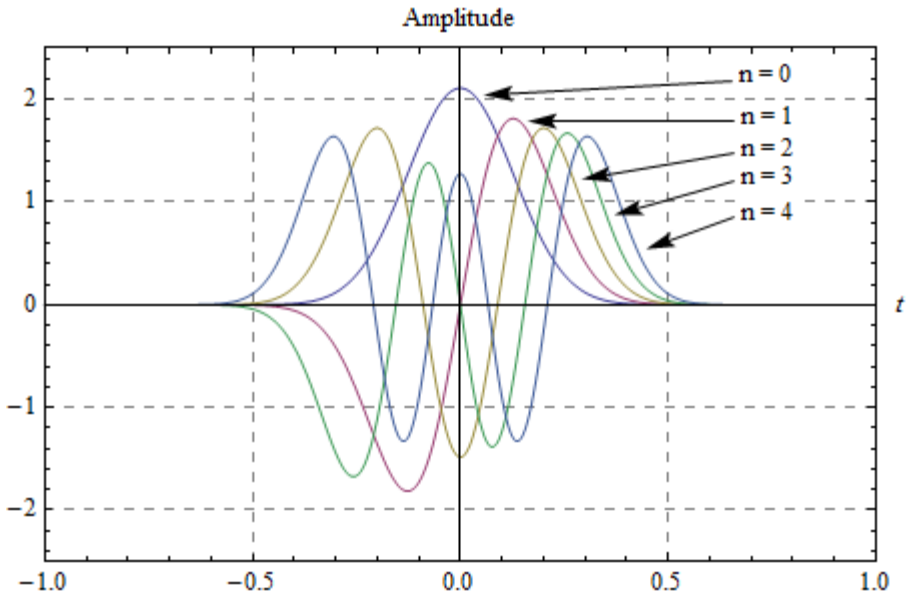

Figure 3.1: LPFs set 1 ($c = 20\pi,\ n: 0 \to 4$)

($\Psi_n(c,t)$ *Set 2)* $c = 100\pi,\quad n: 0 \to 601,\quad t \to [-1.900, 1.900]$ (see figure 3.2).

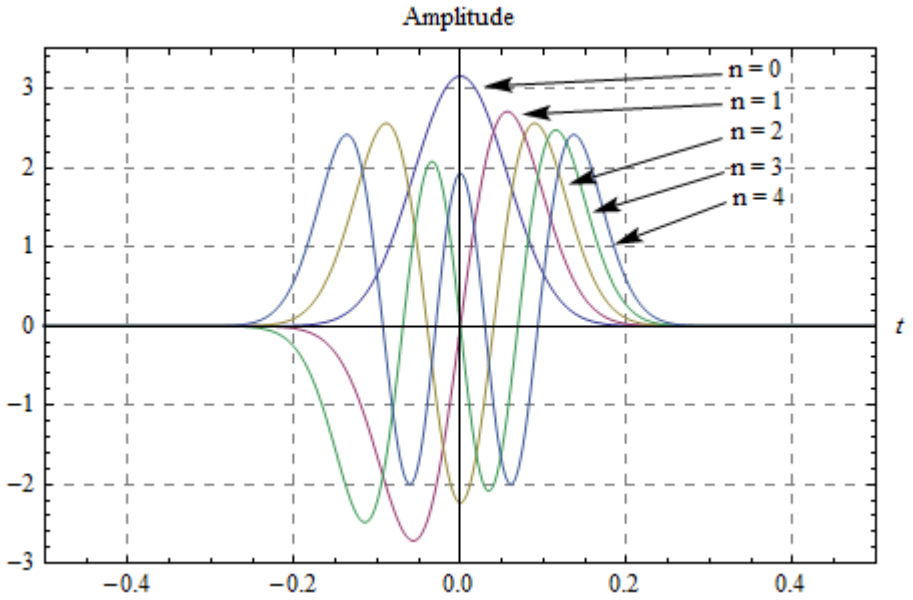

Figure 3.2: LPFs set 2 ($c = 100\pi, n: 0 \to 4$)

The time parameter t is specified with 3 digits of precision after the decimal point as the sampling period in the time axis is 0.001s. Hence the LPFs are sampled at a rate of 1kHz.

Generally, for any LPFs set with a fixed c, as the order n increases, the concentration of the LPFs within the observation interval $[-t_0, t_0]$ decreases. For $n = 2c/\pi$, the signal's maximum concentration reaches the boundary of the observation interval which, in our case, is $[-1,1]$ (see figures 3.3, 3.4).

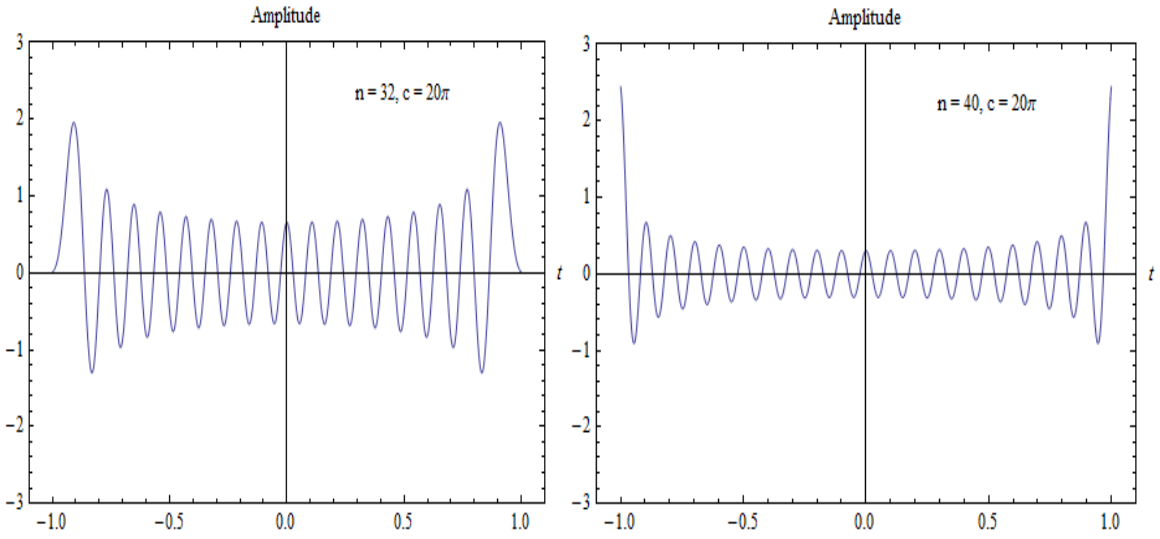

Figure 3.3: Dependency of signal concentration on n for LPFs set 1

As discussed in chapter 2, this can also be seen by observing the eigenvalue (a measure of concentration of signal in the observation interval [-t_0, t_0]) plots for increasing orders of n. As the order n approaches and crosses the critical limit needed for extrapolation, which is $2c/\pi$ (which in our case corresponds to $n = 40$ for LPF set 1 and $n = 200$ for LPF set 2), we can see that eigenvalues suddenly drops from near unity to values close to zero as shown in figures 3.5 and 3.6.

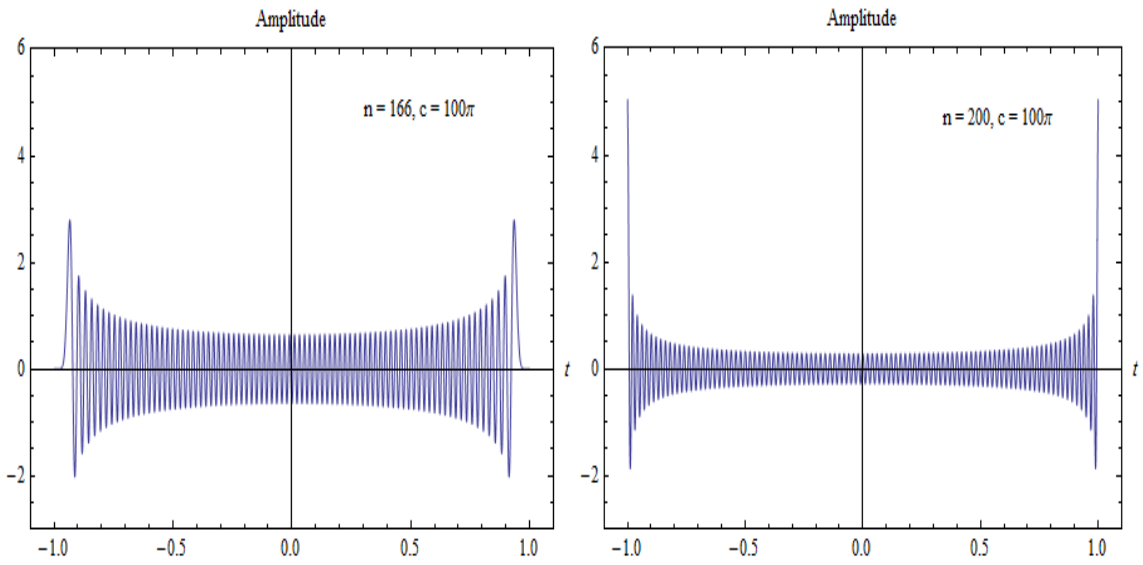

Figure 3.4: Dependency of signal concentration on n for LPFs set 2

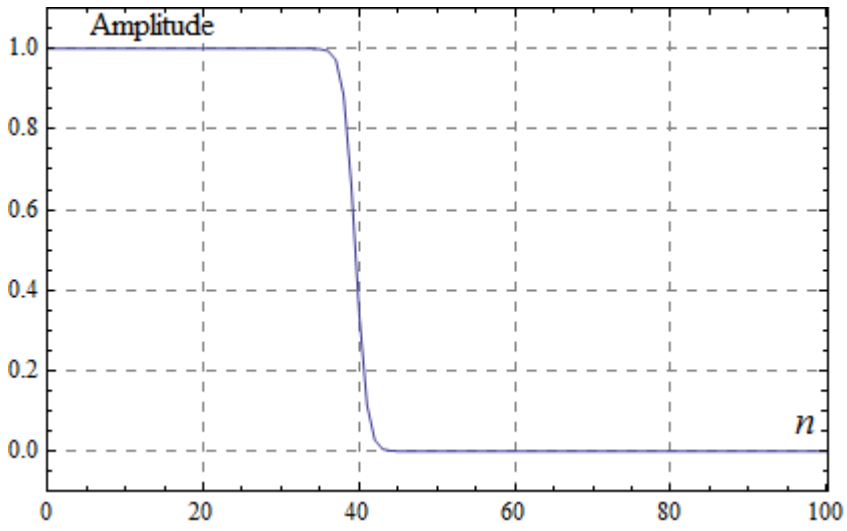

Figure 3.5: Eigenvalues $\lambda_n(c)$ Vs Order n for LPF set 1

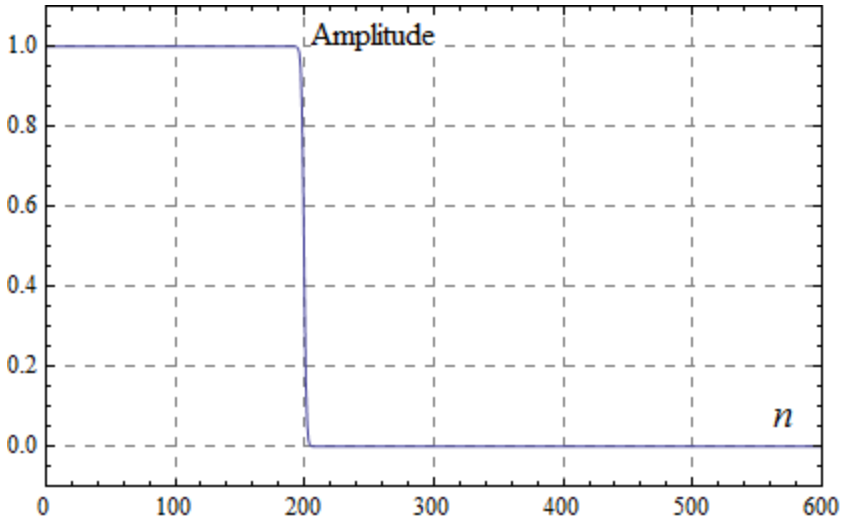

Figure 3.6: Eigenvalues $\lambda_n(c)$ Vs Order n for LPF set 2

3.3 Overlap Integral

By taking a closer look at the analysis (3.3) and synthesis (3.4) equations using LPFs, one can notice that $f(t)$, the function to be extrapolated, is well defined in $[-t_0, t_0]$, i.e. $[-1,1]$ (for sake of simplicity, t_0 was chosen to be 1 for our study). Numerical values of $\lambda_n(c)$ and $\Psi_n(c,t)$ as a set for a given c are also known (as discussed in chapter 2). The only unknown factor is an efficient method to calculate the overlap integral given by $\int_{-1}^{1} f(t)\,\Psi_n(c,t)\,dt$.

Efficient estimation of overlap integral is of paramount importance in obtaining accurate results for extrapolation. As LPFs, $\Psi_n(c,t)$, and eigenvalues, $\lambda_n(c)$, are required to be of high precision for high orders of n, the eigenvalues tend to become extremely small, close to zero, which makes this essentially a problem of high precision numerical integration. If and when computed inaccurately, the coefficients of expansion, $\gamma_n(c)$, of the synthesis equation (3.4) assume extremely large values for such $n's$ that are crucial

and irreplaceable for extrapolation purposes. This, in turn, causes enormous numerical errors that thus render extrapolated signals completely incorrect and useless.

Thus, the method called for very less amount of tolerance for approximation or round of errors that can be allowed when working with these function sets. The original precision with which these functions are obtained has to be preserved throughout our calculations in order to make use of its advantages that helps in containing the expansion coefficients to a desired limit which in turn helps in the synthesis of the bandlimited functions and thus extrapolation itself.

3.4 Calculation of Overlap Integral

A simple and efficient way to compute overlap integral is proposed and implemented to obtain satisfactory results. To make it simple and generic, polynomial approximation of the discrete samples of the scalar product $f(t)\ \Psi_n(c,t)$ was chosen. Our primary focus was to obtain the right polynomial approximation for the underlying common function, $\Psi_n(c,t)$ (for a given c), in any scalar product, irrespective of the bandlimited function $f(t)$. Another major task was to choose the right truncation value N for synthesis (3.4), which we determined by examining the behavior of the scalar expansion coefficients $\gamma_n(c)$ obtained in analysis (3.3). After a series of thorough investigations using different kinds of polynomial interpolation and numerical integration techniques, it was found that piecewise polynomial approximation is best suited for the particular LPF sets that were used.

Direct method of piecewise polynomial interpolation [27] is used. Given k discrete data points, (x_0, y_0) to (x_{k-1}, y_{k-1}), it can be approximated to a polynomial of order $k-1$ as:

$$y = \beta_0 + \beta_1 x + \cdots + \beta_{k-1} x^{k-1}, \tag{3.5}$$

where the coefficients (β) can be found by solving this linear system of equations:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \end{pmatrix} = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{k-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{k-1} & x_{k-1}^2 & \cdots & x_{k-1}^{k-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{pmatrix} \tag{3.6}$$

Here, the y values are the sampled values of LPFs itself for varying time instances given as x values. The scalar coefficients (β_0 to β_{k-1}) are the coefficients of the resulting polynomial representation of the sampled LPFs set. The big matrix (comprising of x values) in (3.6) is commonly referred to as a Vandermonde matrix. As each x value varying from x_0 to x_{k-1} is distinct, since they are the varying instances of the time axis, this Vandermonde matrix is nonsingular (determinant cannot be equal to zero as each row is distinct). This non-singularity of Vandermonde matrix guarantees us a unique solution for the system of equations in (3.6).

Relating this approach to the problem at hand, there are 1001 sampled instances of the overlap integral expression $f(t)\ \Psi_n(c,t)$, for a given n, in the closed time interval $[0,1]$ and 2001 samples in the interval $[-1,1]$. The whole interval of $[-1,1]$ is divided into 8 equal segments, thus there are 251 samples in each segment of the samples of the scalar product $f(t)\ \Psi_n(c,t)$. Applying piecewise polynomial approximation to each segment containing 251 samples, one obtains a polynomial of order 250 for each segment. Thus,

the resulting Vandermonde matrix for each segment would be of the order $251\ X\ 251$. Once the corresponding linear system of equations as in (3.6) is solved for each of these 8 segments, we are left with the corresponding polynomial expressions as given in (3.5). Integrating a polynomial expression is a relatively simple and easy task for any decent computational packages available these days. A simple integration of the resulting polynomials in each segment and subsequent addition yields us the desired overlap integral.

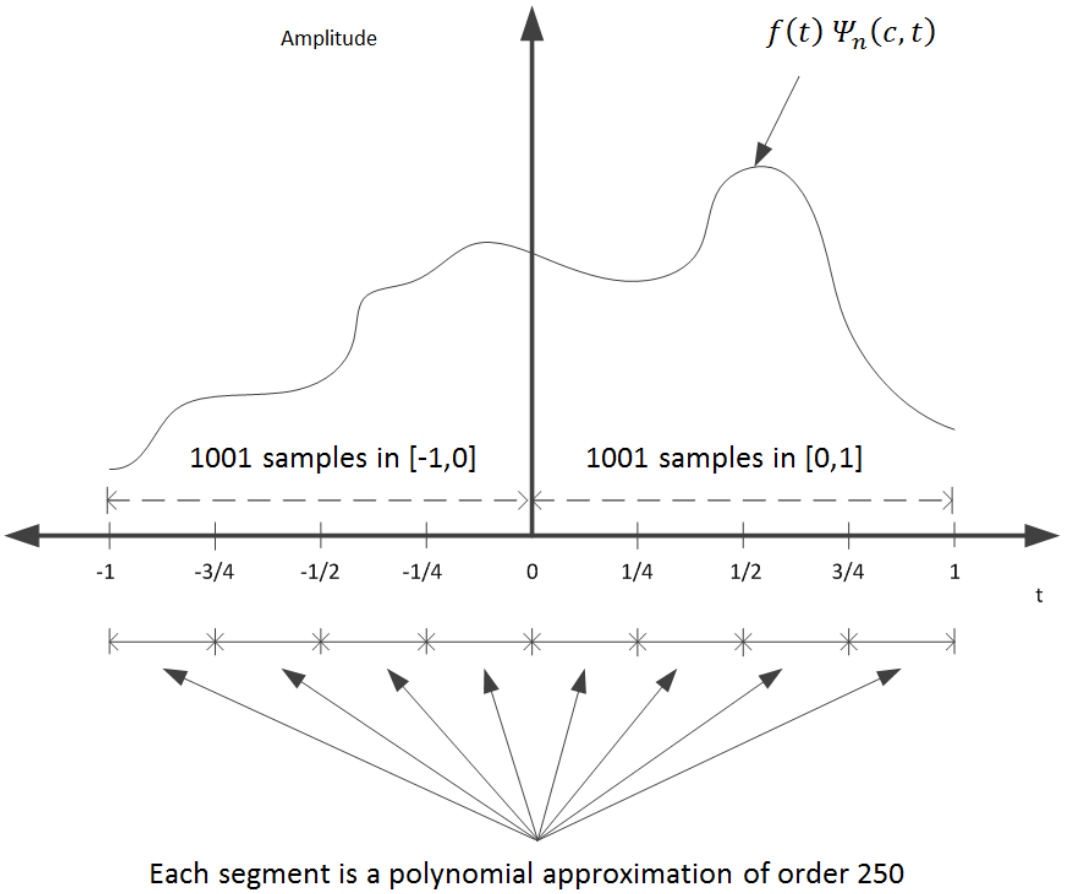

Figure 3.7: Illustration of Overlap Integral Calculation

The key aspect of this approach is the fact that the original precision of the LPFs is maintained throughout in the calculations thus keeping the resulting overlap integral comparable to that of the corresponding eigenvalue λ_n , for high orders of n well and above the critical limit $2c/\pi$. This enables us to have a well contained set of expansion coefficients from (3.3) to be used for extrapolation in (3.4).

Chapter 4

Results

In this chapter, we show the results obtained by employing our extrapolation algorithm on various kinds of signals. Comparisons of the effectiveness of our method to that of other prominent works are also made at appropriate sections. This chapter is further divided into three subsections. The first section is devoted to showing extrapolation results of ideally known test signals; the second section contains results for known test signals that are perturbed in some way or other; and the third section contains our findings of extrapolation attempts on real or experimental signals which are basically unknown to the observer.

Mathematica, a software tool that is well suited for high precision computing, has been used for implementing our algorithm. Extrapolation was carried out for some selected known test functions using both LPFs set 1 and 2; some results of which can also be found in [28,34,35] and is used here for making comparisons and drawing conclusions. The only restriction imposed on the selected signals was that its maximum frequency should be less than or equal to that of the corresponding LPFs set (the bandwidth parameter c) used for their extrapolation. In the results shown below, the same extrapolation formula was employed for both reconstructing the signal within as well as extrapolating it beyond the interval $[-1,1]$. To show the error estimates with respect to

the original signal, common logarithm of the absolute error between the extrapolated and original data is also plotted. Extrapolation errors of the order of up to 1e-03 are considered [21] acceptable. An error norm known as the normalized mean-square error (NMSE) is also used to analyze the effectiveness of our algorithm. Following their [17] notation, it is defined as:

$$NMSE = \frac{\|f_e - f\|^2}{\|f\|^2} \tag{4.1}$$

where f_e is the extrapolated signal and f is the original signal.

4.1 Extrapolation of Ideal Test Signals

Some ideal signals/functions completely free of all kinds of noise is considered here as inputs to test our algorithm. As the signals are known, they can have infinite precision when used in the computations. The algorithm is then basically limited by the precision with which the LPF sets and their corresponding eigenvalues are obtained. In our calculations, this precision comes down to about 200 digits after the decimal point. The necessity of such high precision computing has been pointed out while comparing the results. Extrapolation results for both LPFs set 1 ($c = 20\pi$) and set 2 ($c = 100\pi$) is presented. This section ends with a special case in which the signal is known but is not bandlimited (with respect to LPF set 1), meaning the maximum frequency of the test signal is higher than that of the LPFs set considered.

The test functions extrapolated using our algorithm and presented here as results are listed below.

- $f_1(t) = e^{-2t^2} e^{3t} \sin(\pi t) \cos(3\pi t) + 0.5[\sin(5\pi t) - \cos(7\pi t)]$
- $f_2(t) = 3 \sin\left(17\pi t + \left(\frac{\pi}{5}\right)\right) - 5 \cos\left(13\pi t - \left(\frac{3\pi}{8}\right)\right) - 2e^{-\pi(t+7)^2} + 9e^{-\pi t^2}$
- $f_3(t) = \cos\left(20\pi t - \left(\frac{\pi}{11}\right)\right) + \cos\left(\frac{20\pi t}{7}\right) - \cos\left(\frac{15\pi t}{2}\right)$
- $f_4(t) = \cos\left(100\pi t - \left(\frac{\pi}{11}\right)\right) + \sin\left(\frac{33\pi t}{7}\right) - \cos\left(\frac{75\pi t}{2}\right)$
- $f_5(t) = e^{-\pi(t-1)^2} + e^{-\pi(t+1)^2}$
- $f_6(t) = \cos\left(2\pi t - \left(\frac{\pi}{11}\right)\right) + \cos\left(\frac{2\pi t}{7}\right) - \cos\left(\frac{3\pi t}{2}\right)$
- $f_7(t) = \frac{20}{\pi} e^{j2t^2} \operatorname{sinc}\left(\frac{20t}{\pi}\right)$
- $f_8(t) = \cos(30\pi t)$

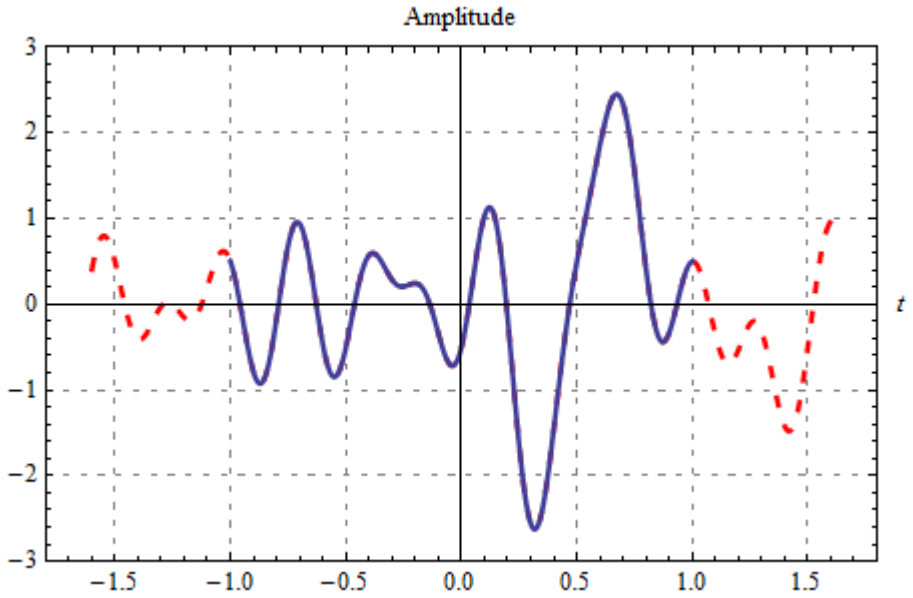

Figure 4.1.1: $f_1(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

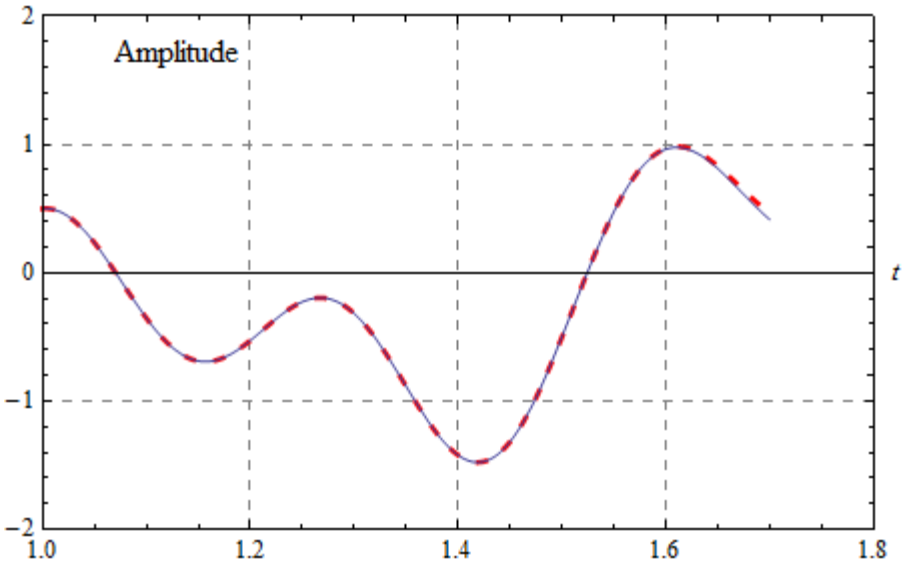

Figure 4.1.2: Magnification of extrapolation of $f_1(t)$ versus time using LPFs set 1

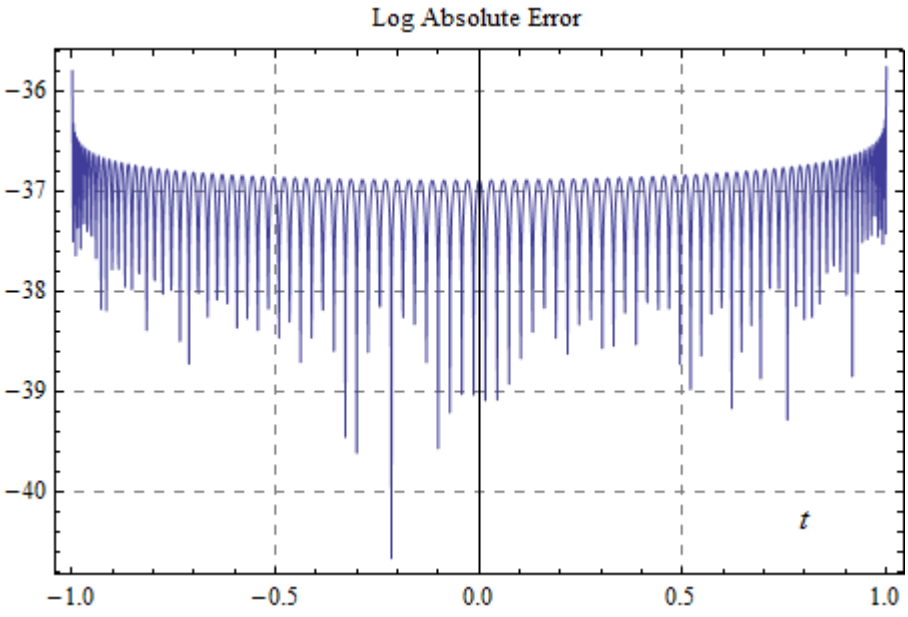

Figure 4.1.3: Logarithm of absolute error for reconstruction of $f_1(t)$ using LPFs set 1

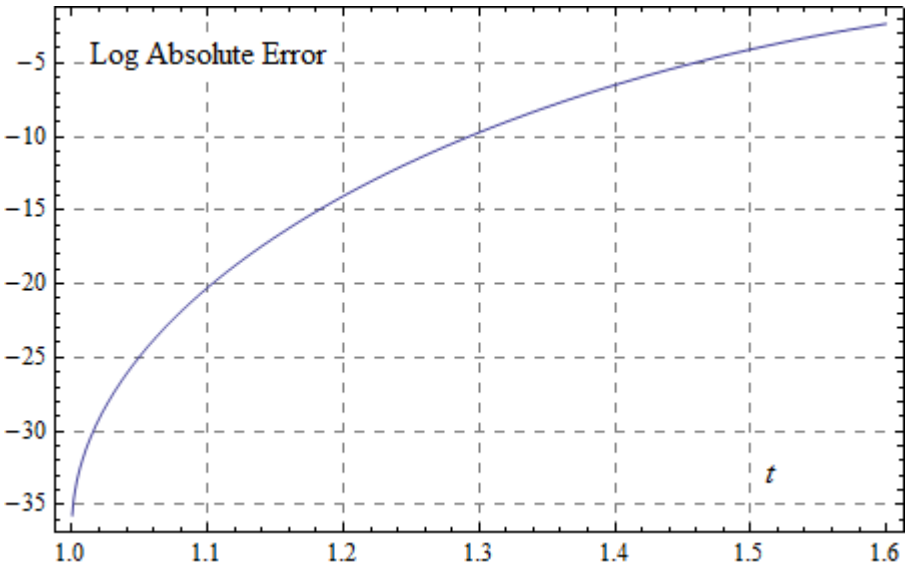

Figure 4.1.4: Logarithm of absolute error for extrapolation of $f_1(t)$ using LPFs set 1

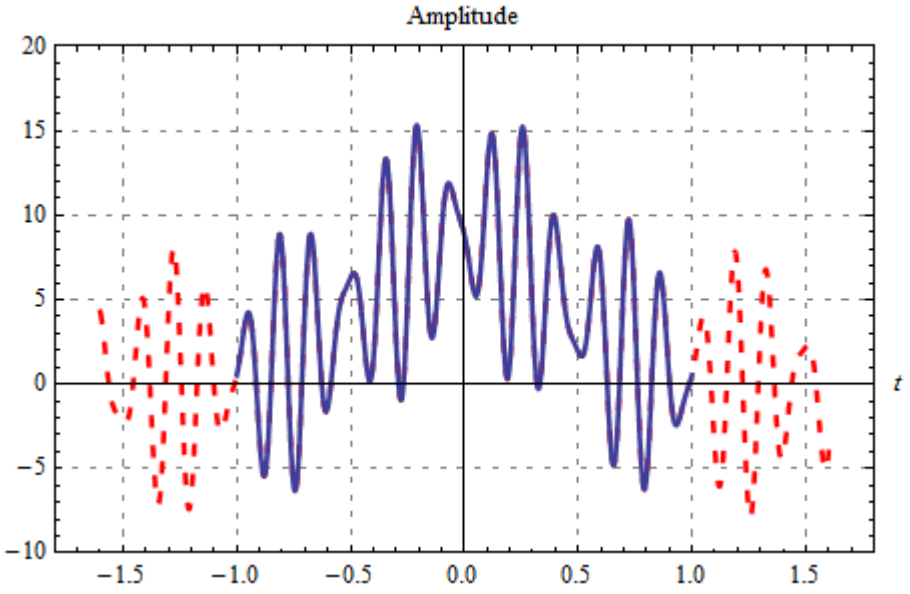

Figure 4.2.1: $f_2(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

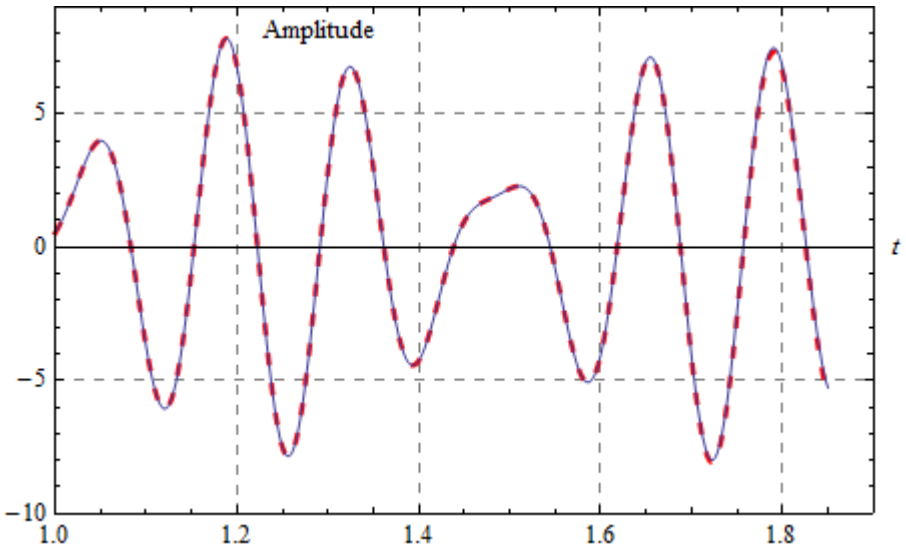

Figure 4.2.2: Magnification of extrapolation of $f_2(t)$ versus time using LPFs set 1

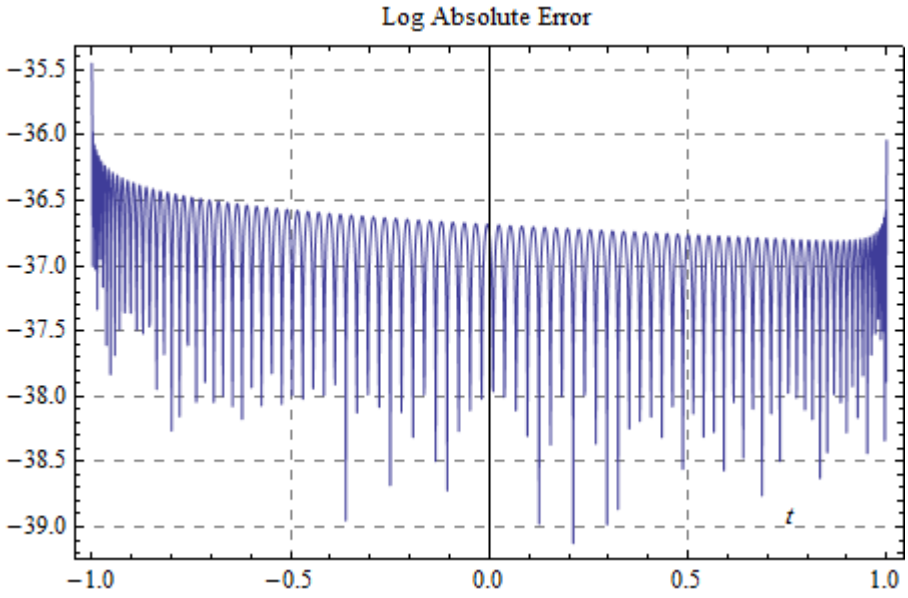

Figure 4.2.3: Logarithm of absolute error for reconstruction of $f_2(t)$ using LPFs set 1

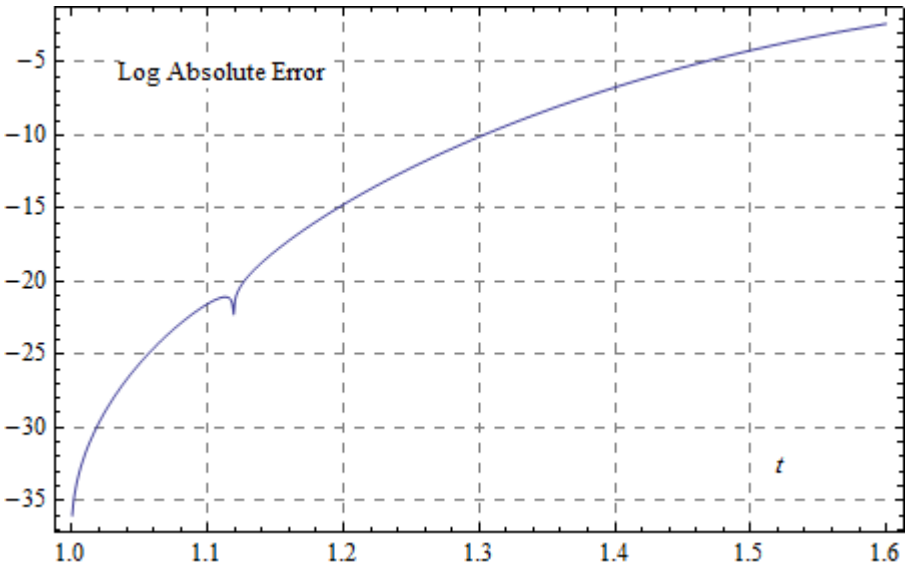

Figure 4.2.4: Logarithm of absolute error for extrapolation of $f_2(t)$ using LPFs set 1

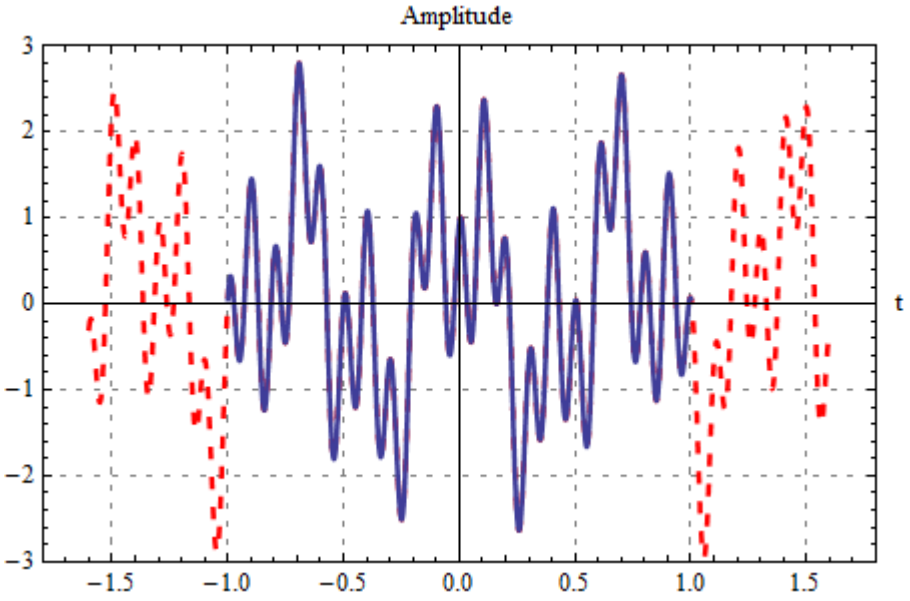

Figure 4.3.1: $f_3(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

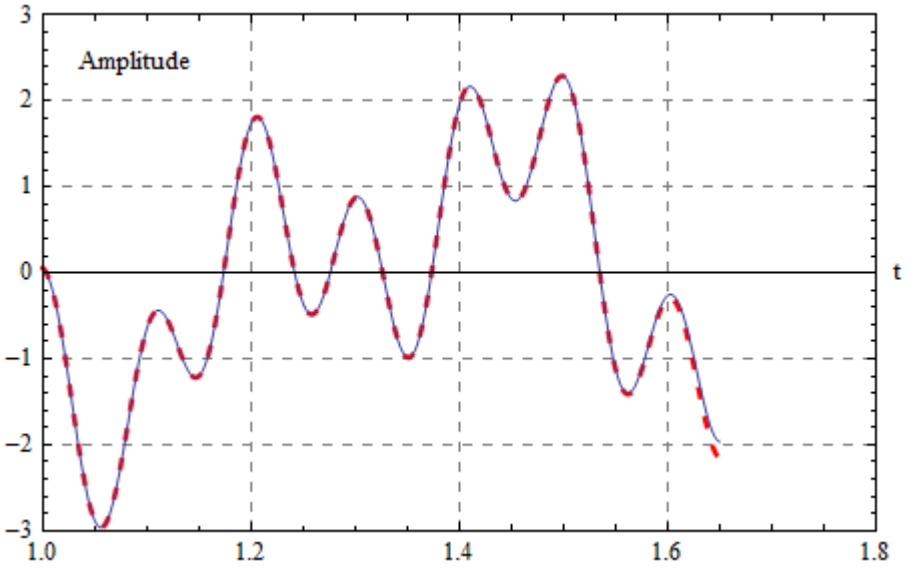

Figure 4.3.2: Magnification of extrapolation of $f_3(t)$ versus time using LPFs set 1

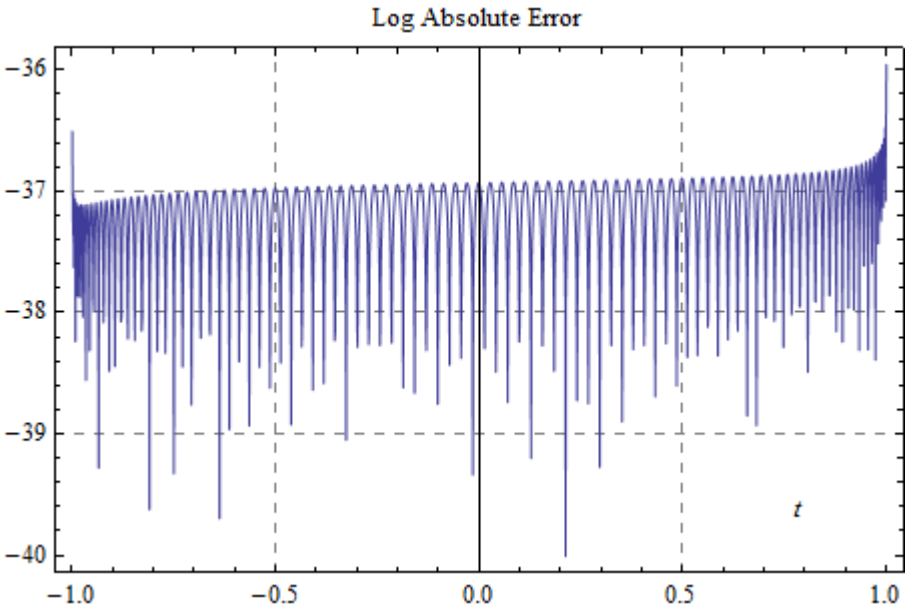

Figure 4.3.3: Logarithm of absolute error for reconstruction of $f_3(t)$ using LPFs set 1

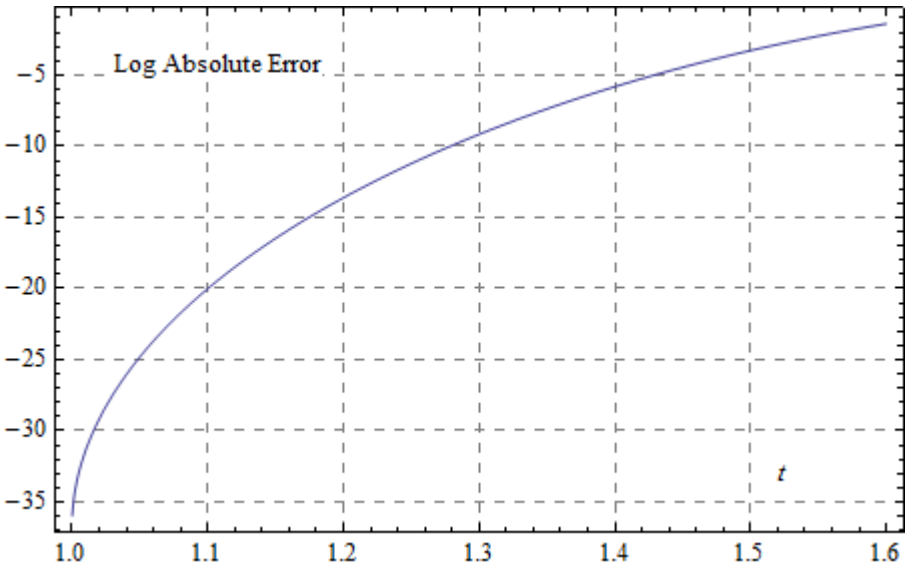

Figure 4.3.4: Logarithm of absolute error for extrapolation of $f_3(t)$ using LPFs set 1

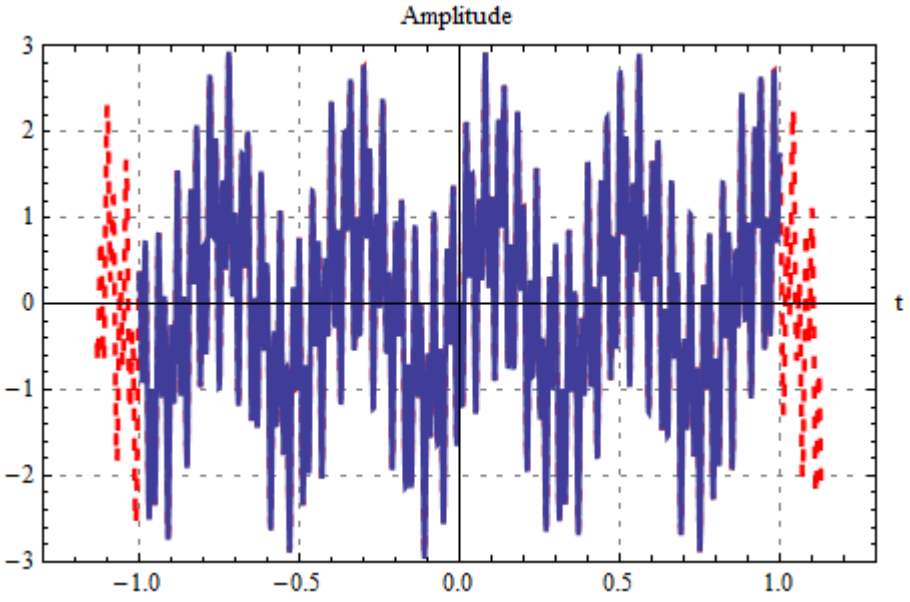

Figure 4.4.1: $f_4(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 2

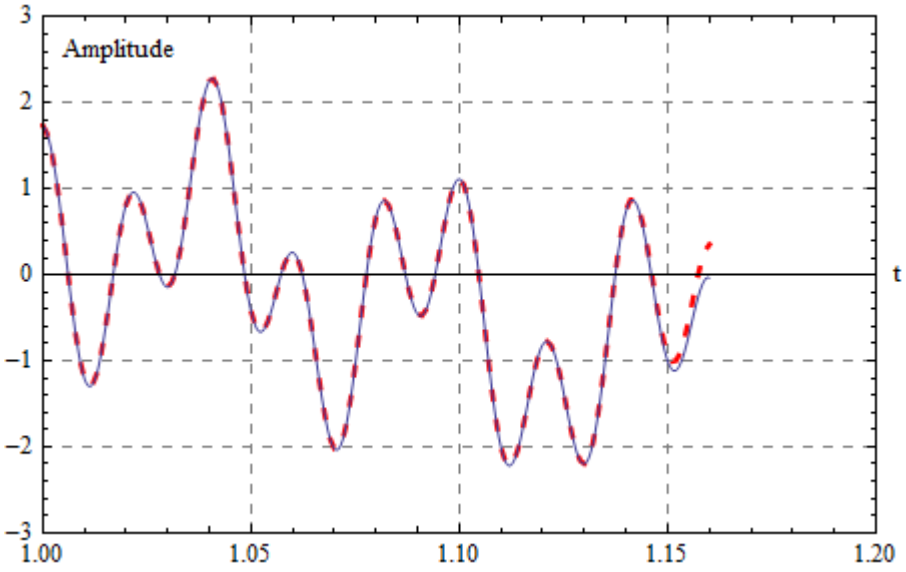

Figure 4.4.2: Magnification of extrapolation of $f_4(t)$ versus time using LPFs set 2

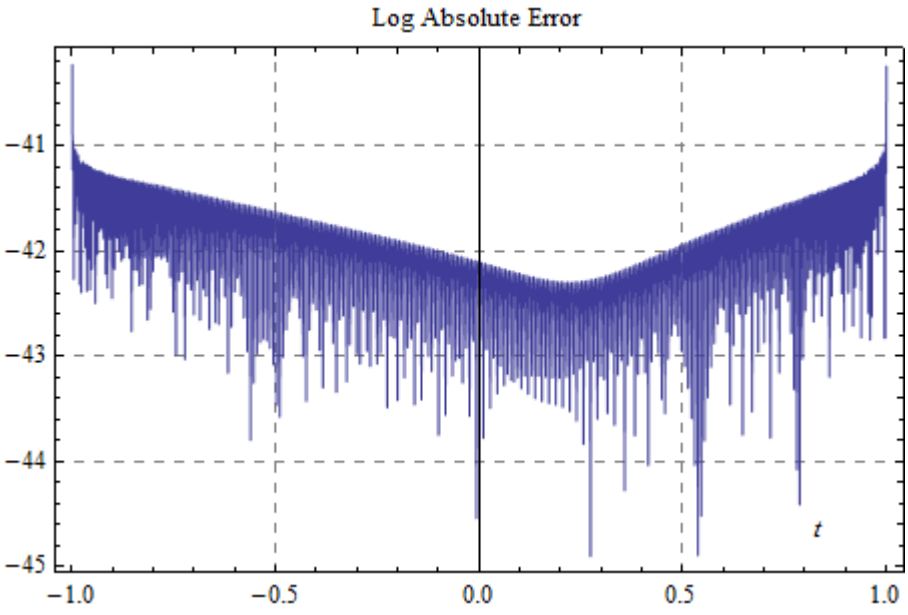

Figure 4.4.3: Logarithm of absolute error for reconstruction of $f_4(t)$ using LPFs set 2

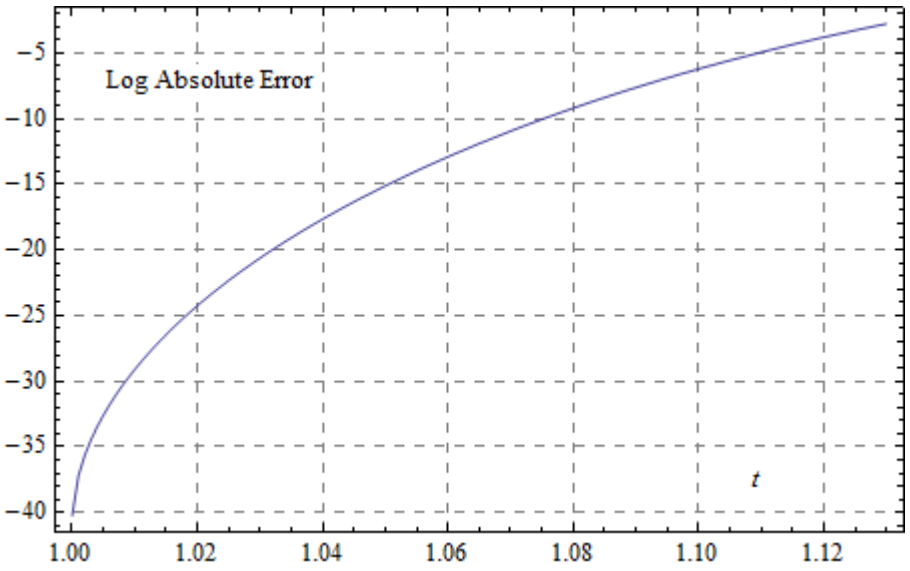

Figure 4.4.4: Logarithm of absolute error for extrapolation of $f_4(t)$ using LPFs set 2

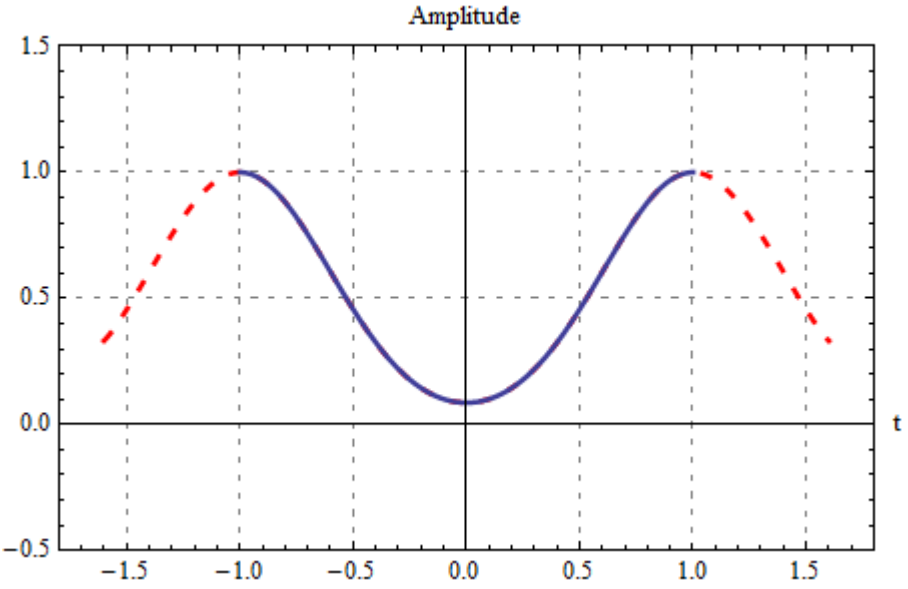

Figure 4.5.1: $f_5(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

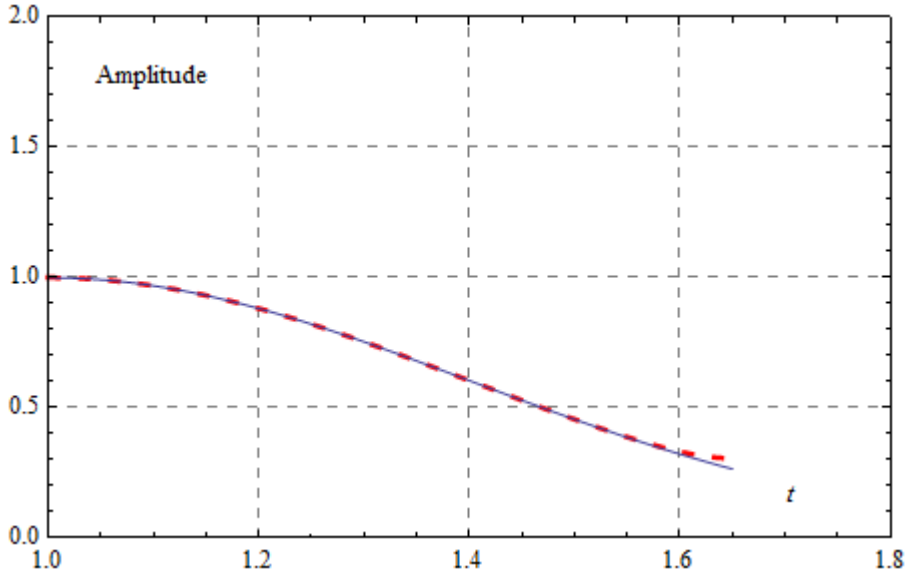

Figure 4.5.2: Magnification of extrapolation of $f_5(t)$ versus time using LPFs set 1

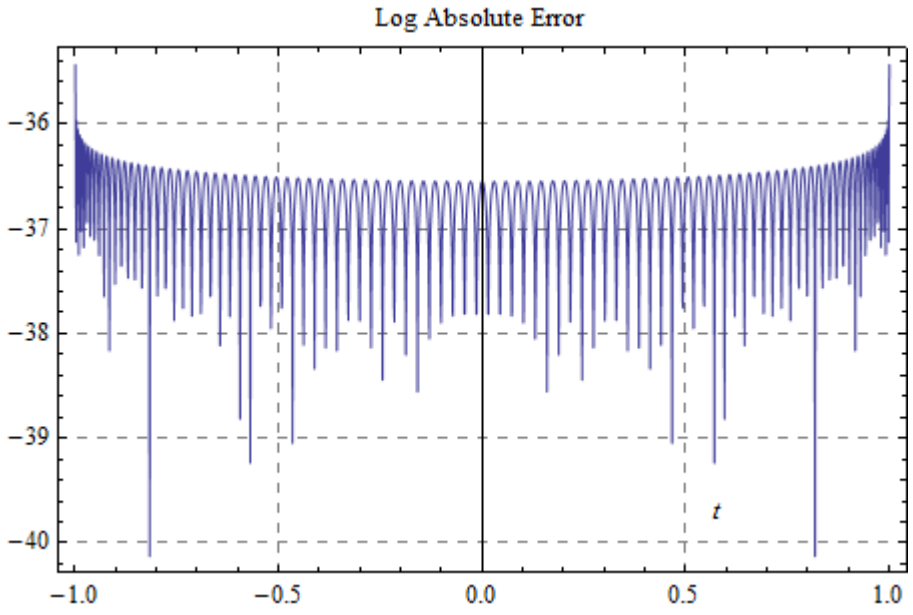

Figure 4.5.3: Logarithm of absolute error for reconstruction of $f_5(t)$ using LPFs set 1

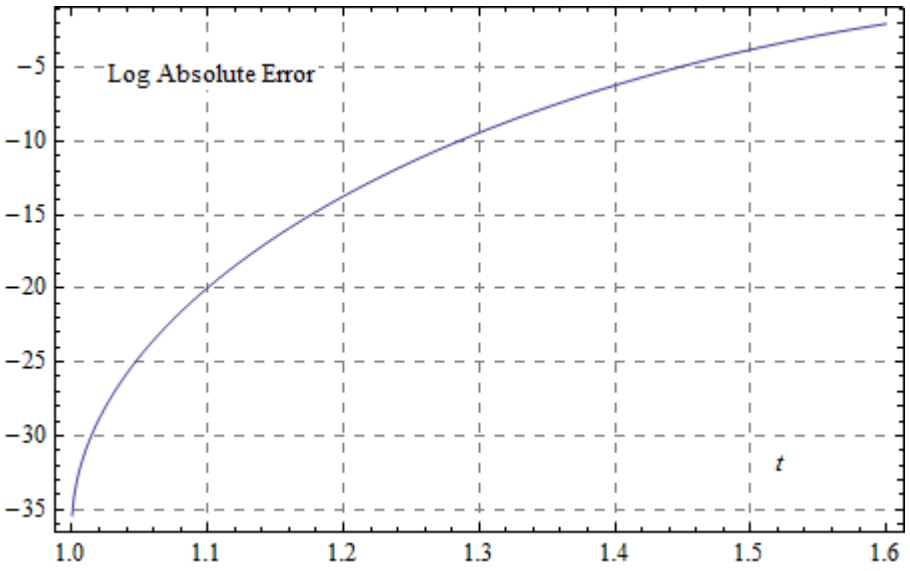

Figure 4.5.4: Logarithm of absolute error for extrapolation of $f_5(t)$ using LPFs set 1

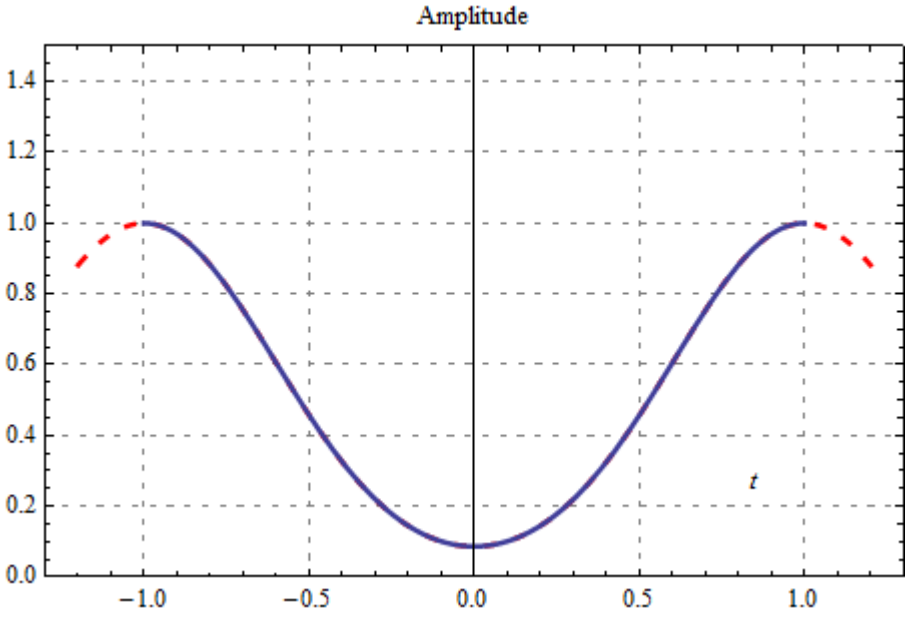

Figure 4.5.5: $f_5(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 2

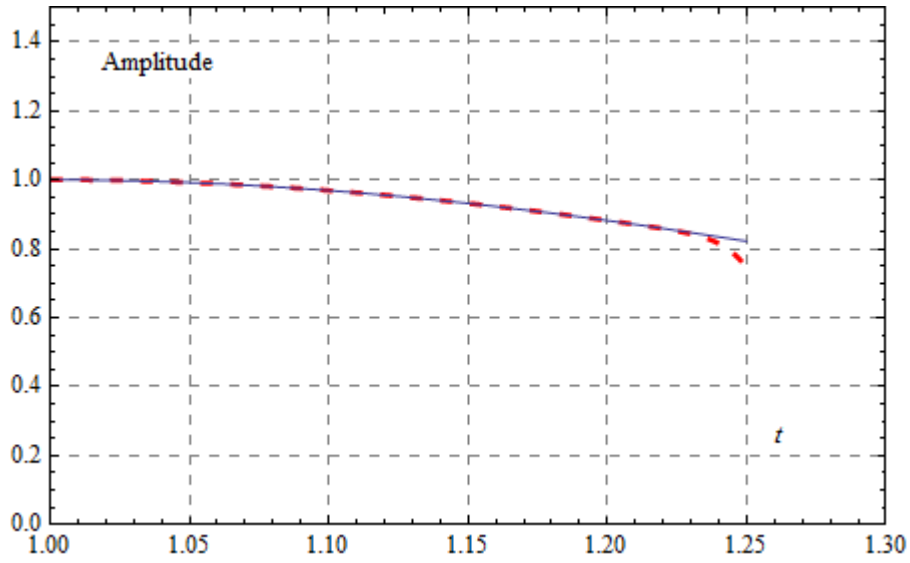

Figure 4.5.6: Magnification of extrapolation of $f_5(t)$ versus time using LPFs set 2

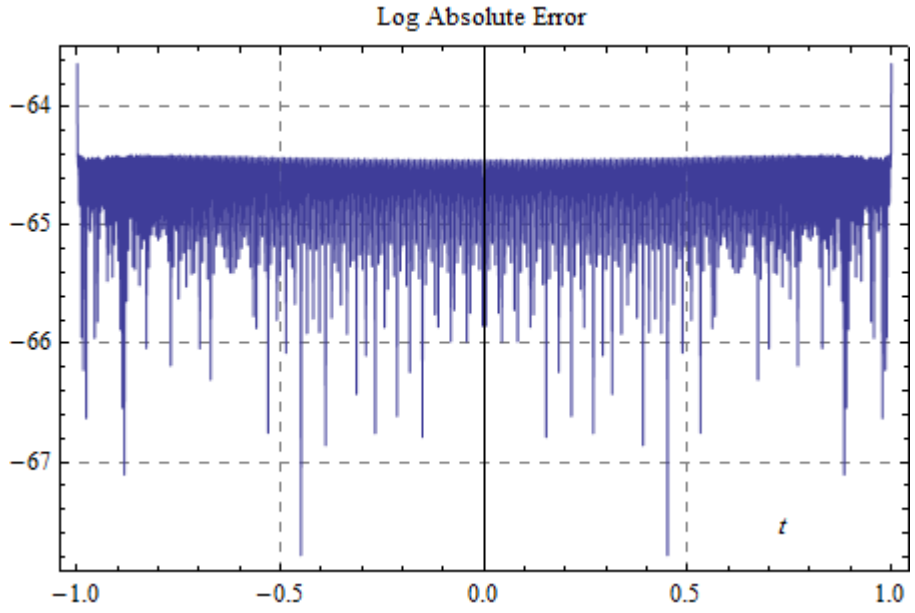

Figure 4.5.7: Logarithm of absolute error for reconstruction of $f_5(t)$ using LPFs set 2

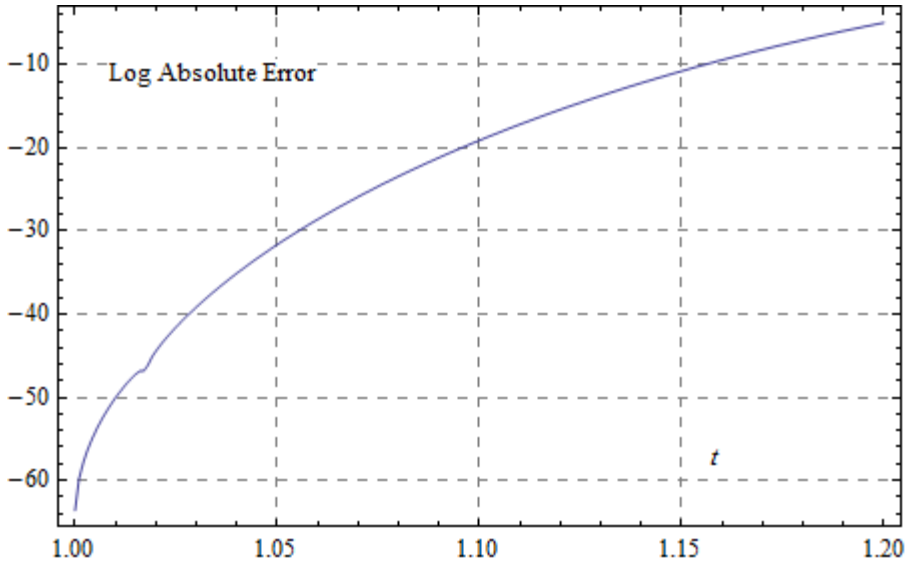

Figure 4.5.8: Logarithm of absolute error for extrapolation of $f_5(t)$ using LPFs set 2

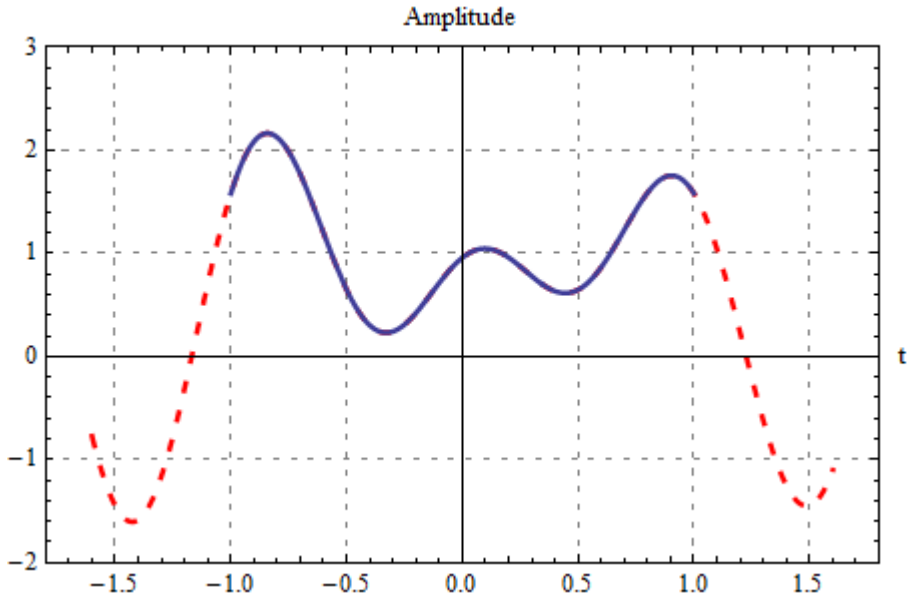

Figure 4.6.1: $f_6(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

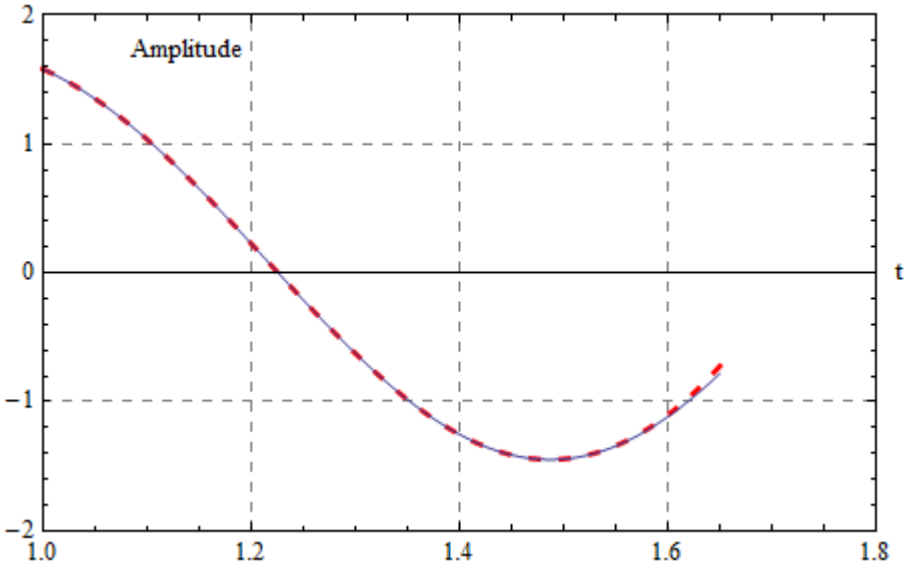

Figure 4.6.2: Magnification of extrapolation of $f_6(t)$ versus time using LPFs set 1

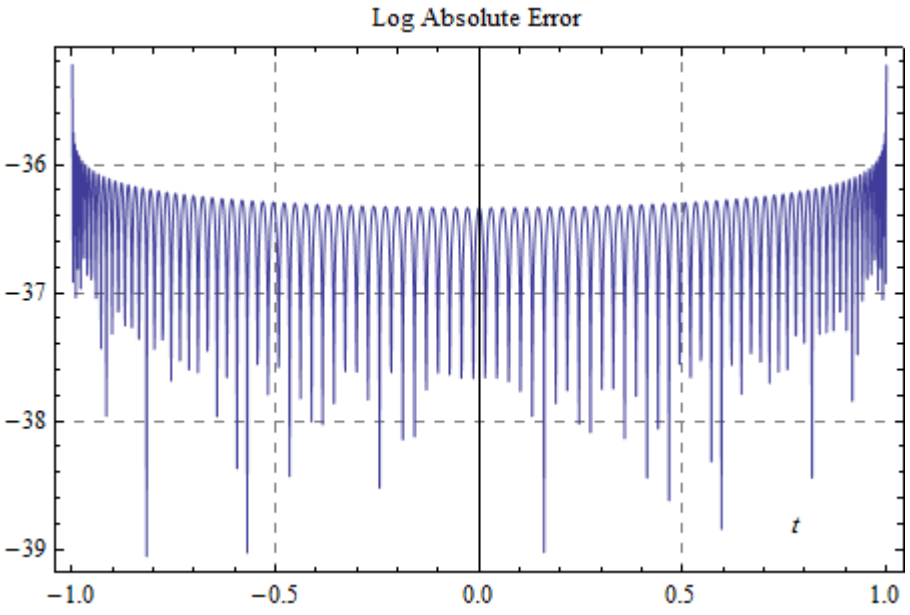

Figure 4.6.3: Logarithm of absolute error for reconstruction of $f_6(t)$ using LPFs set 1

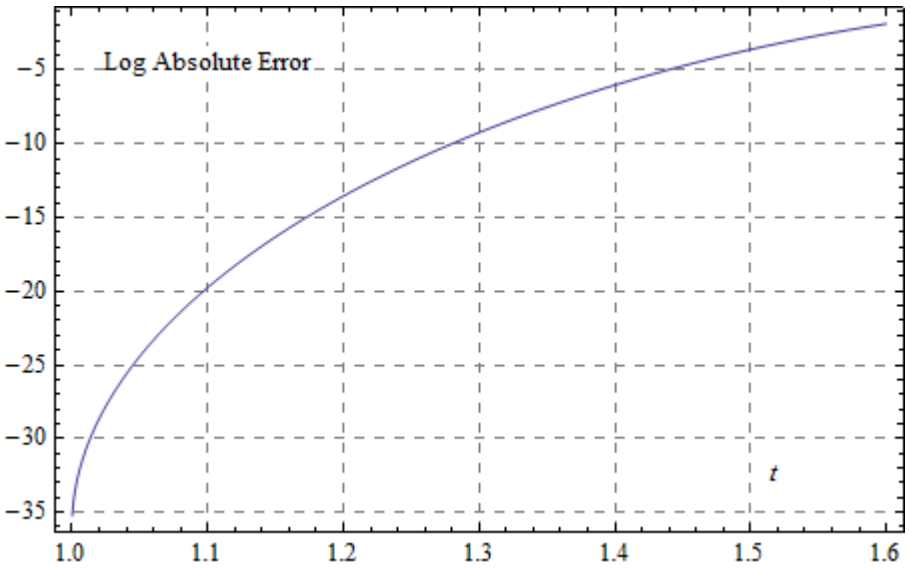

Figure 4.6.4: Logarithm of absolute error for extrapolation of $f_6(t)$ using LPFs set 1

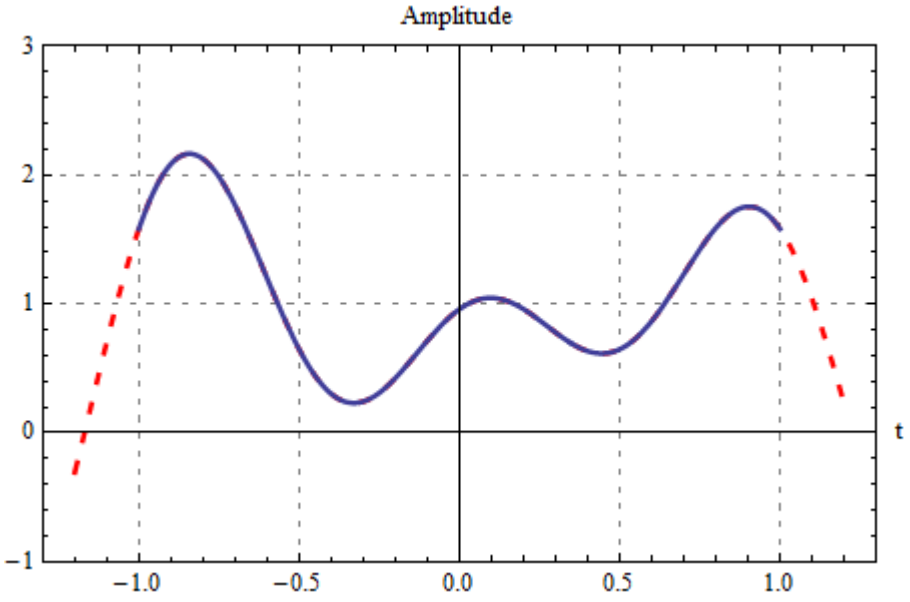

Figure 4.6.5: $f_6(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 2

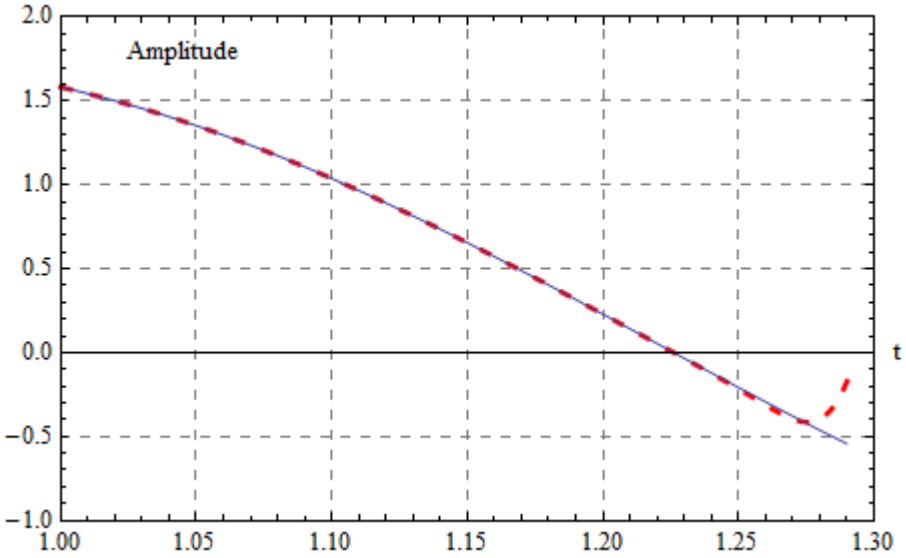

Figure 4.6.6: Magnification of extrapolation of $f_6(t)$ versus time using LPFs set 2

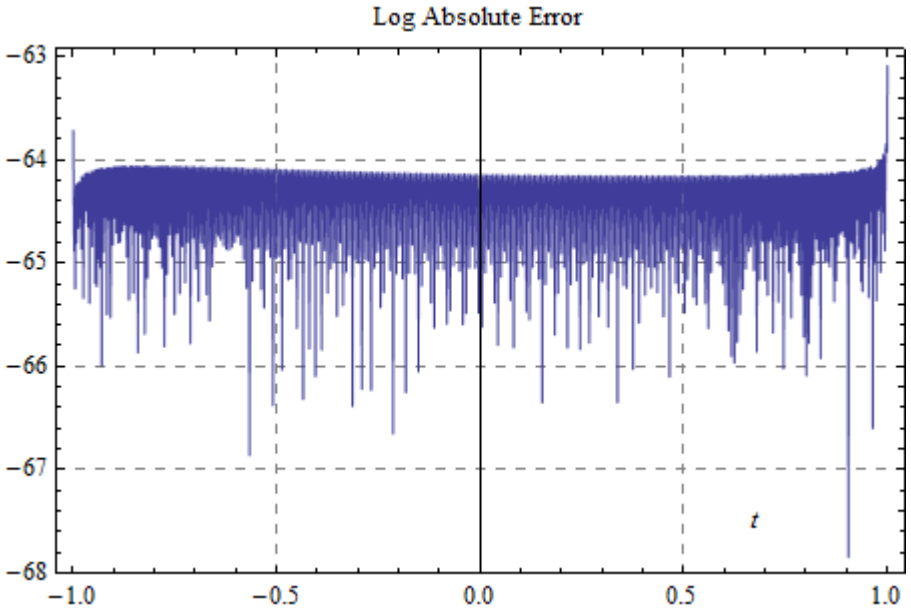

Figure 4.6.7: Logarithm of absolute error for reconstruction of $f_6(t)$ using LPFs set 2

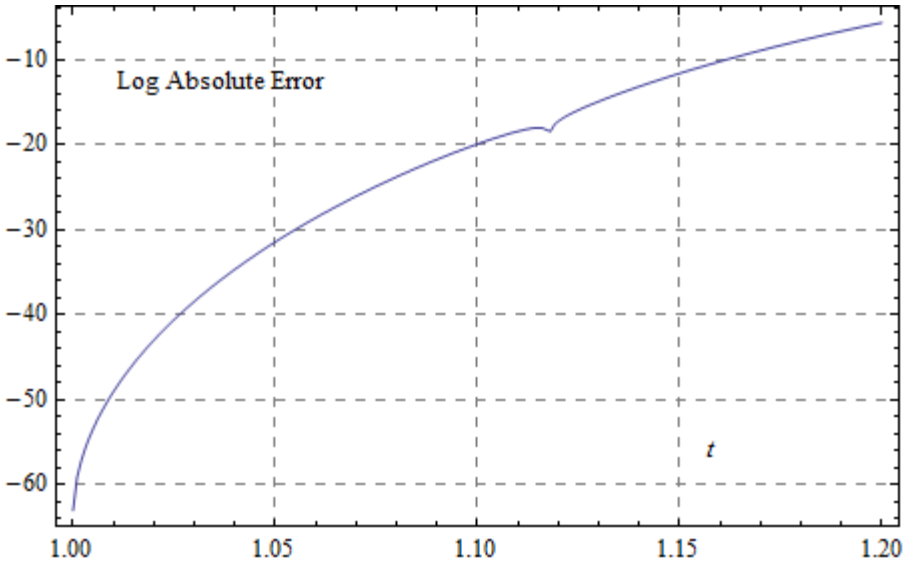

Figure 4.6.8: Logarithm of absolute error for extrapolation of $f_6(t)$ using LPFs set 2

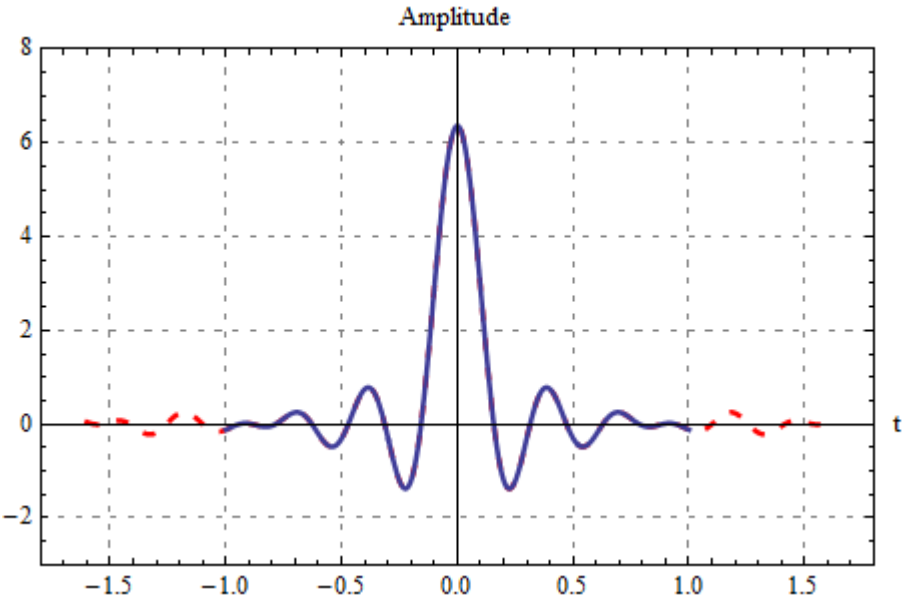

Figure 4.7.1: Real part of $f_7(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

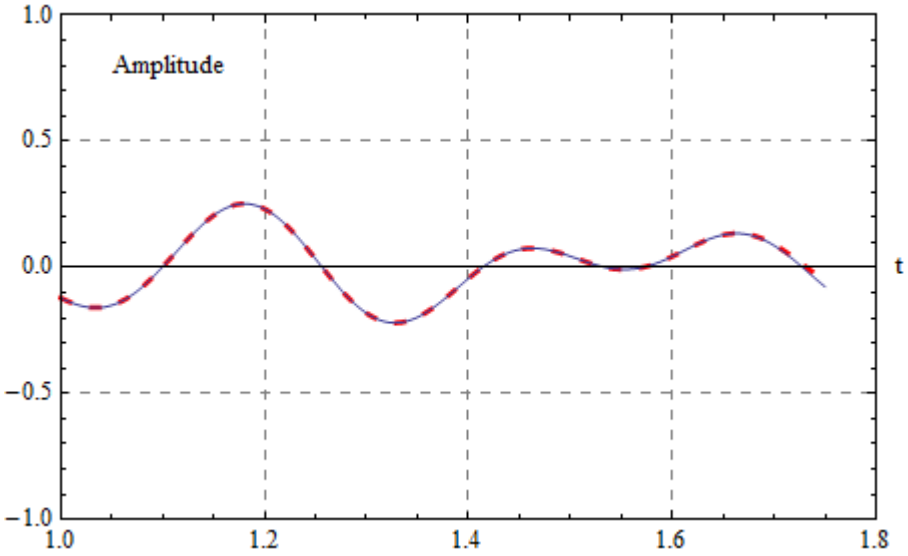

Figure 4.7.2: Magnification of extrapolation of real part of $f_7(t)$ versus time using LPFs set 1

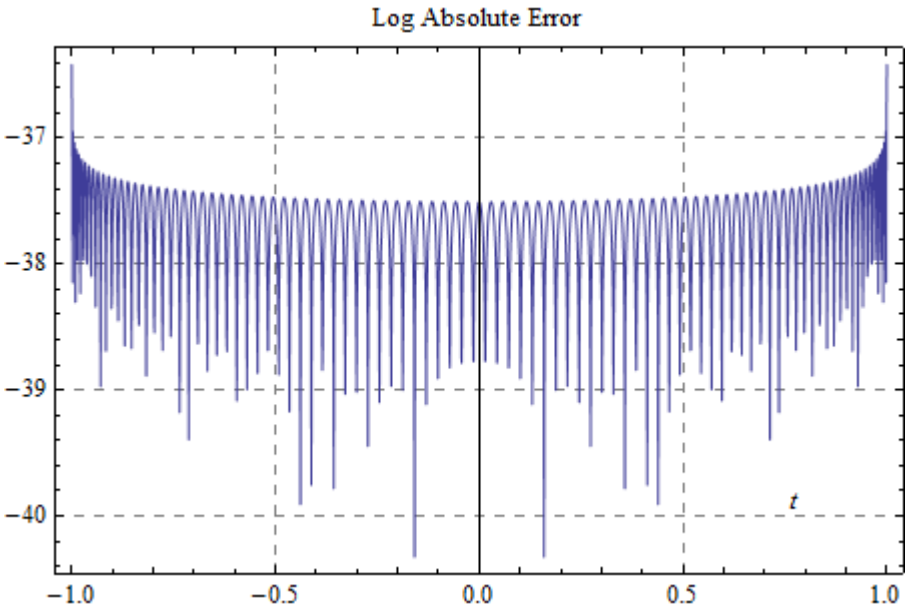

Figure 4.7.3: Logarithm of absolute error for reconstruction of real part of $f_7(t)$ using LPFs set 1

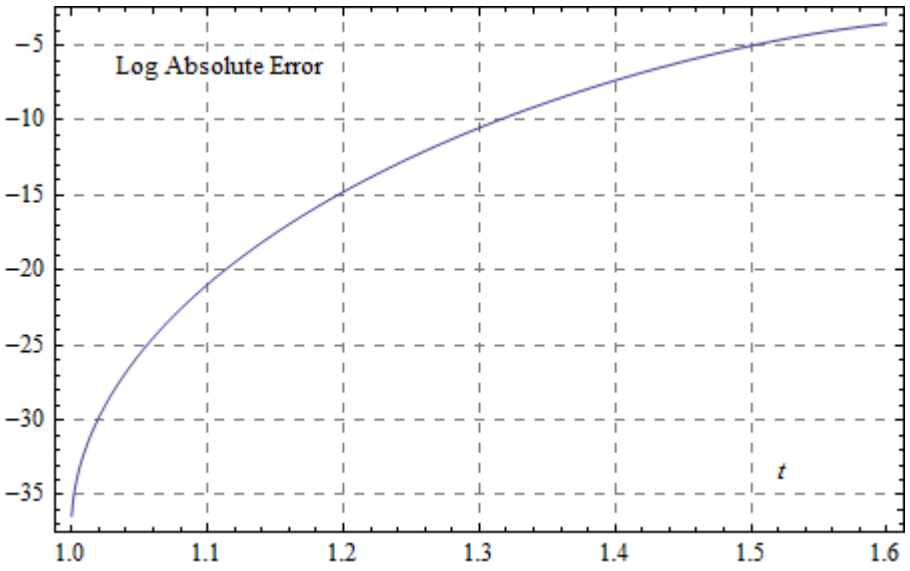

Figure 4.7.4: Logarithm of absolute error for extrapolation of real part of $f_7(t)$ using LPFs set 1

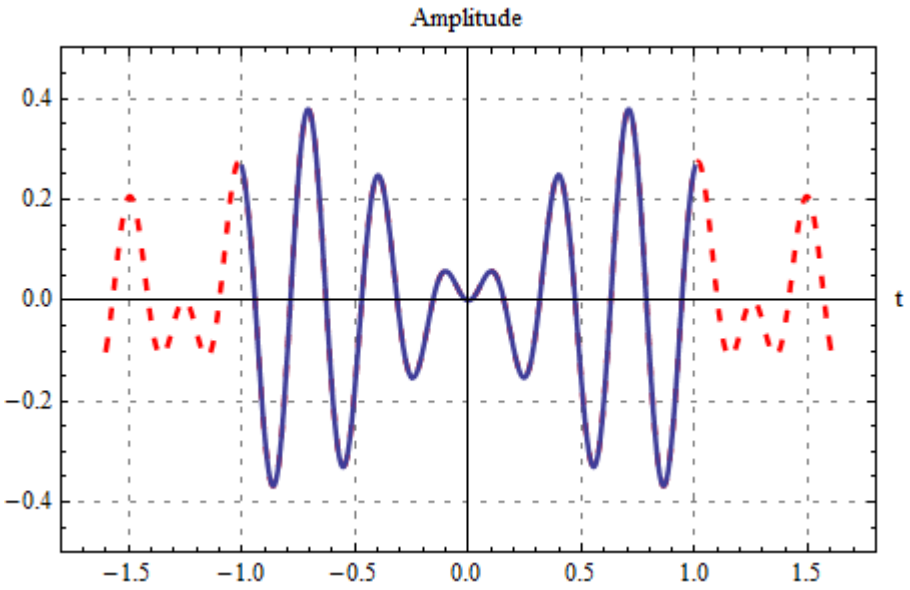

Figure 4.7.5: Imaginary part of $f_7(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

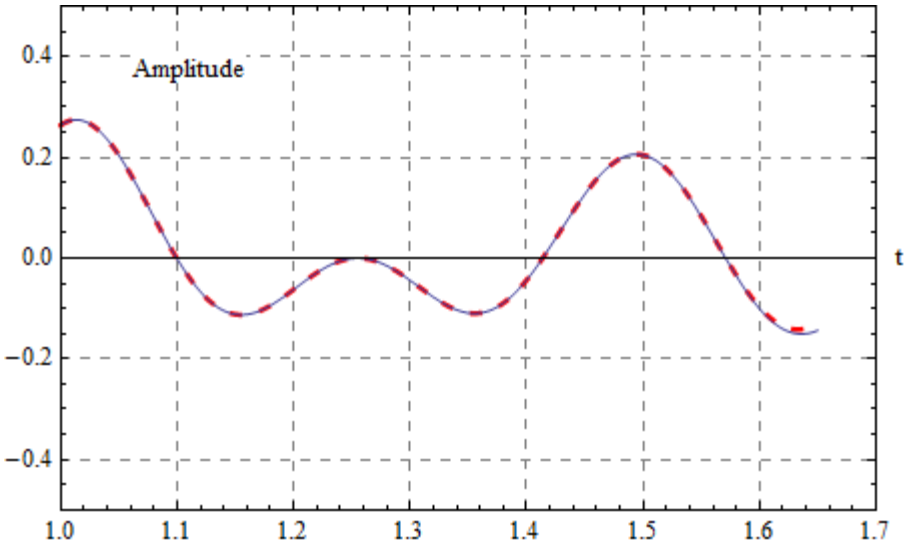

Figure 4.7.6: Magnification of extrapolation of imaginary part of $f_7(t)$ versus time using LPFs set 1

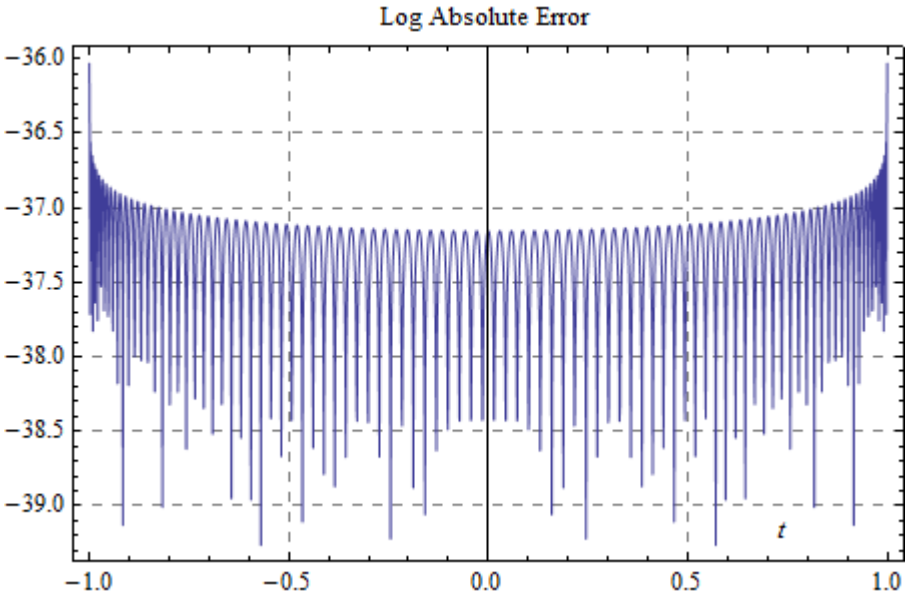

Figure 4.7.7: Logarithm of absolute error for reconstruction of imaginary part of $f_7(t)$ using LPFs set 1

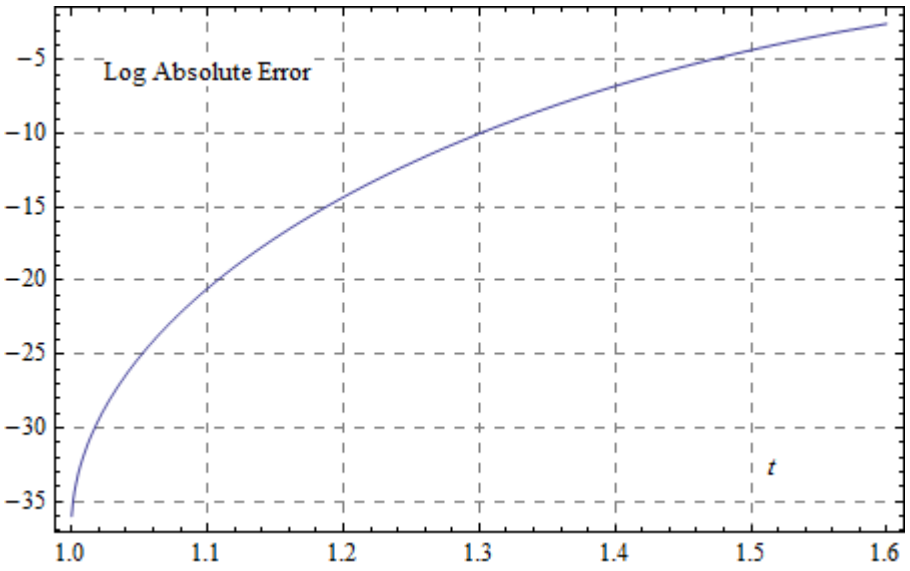

Figure 4.7.8: Logarithm of absolute error for extrapolation of imaginary part of $f_7(t)$ using LPFs set 1

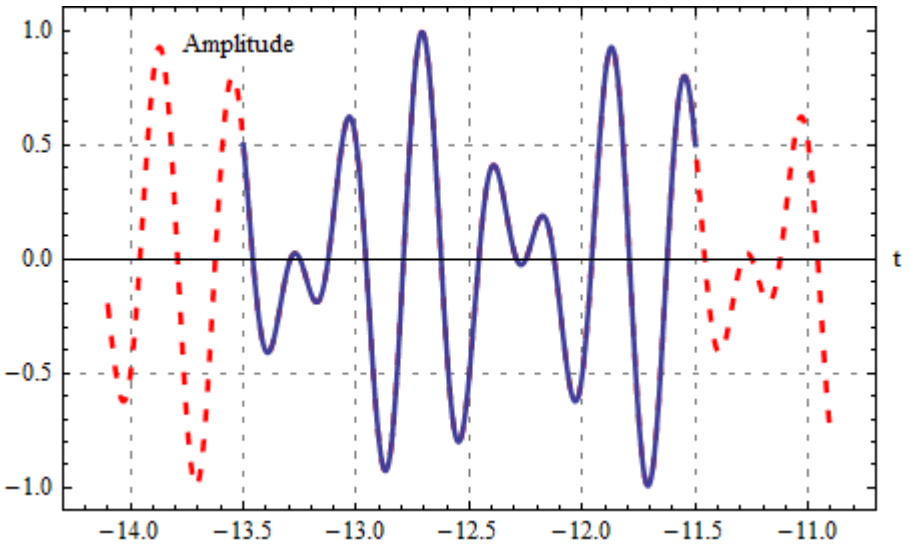

Figure 4.8.1: Time-shifted $f_1(t)$ original [solid] and extrapolated [dashed] versus time using LPFs set 1

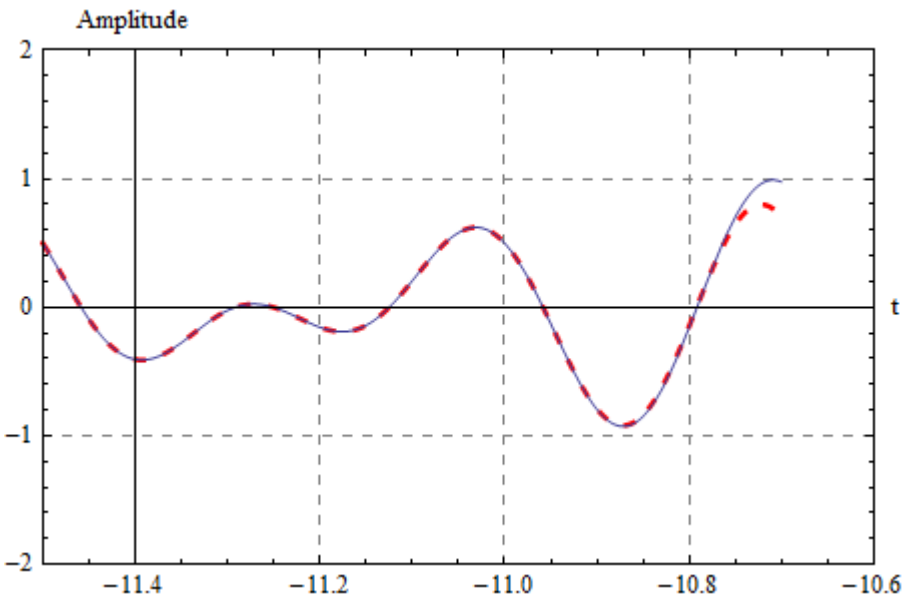

Figure 4.8.2: Magnification of extrapolation of time-shifted $f_1(t)$ versus time using LPFs set 1

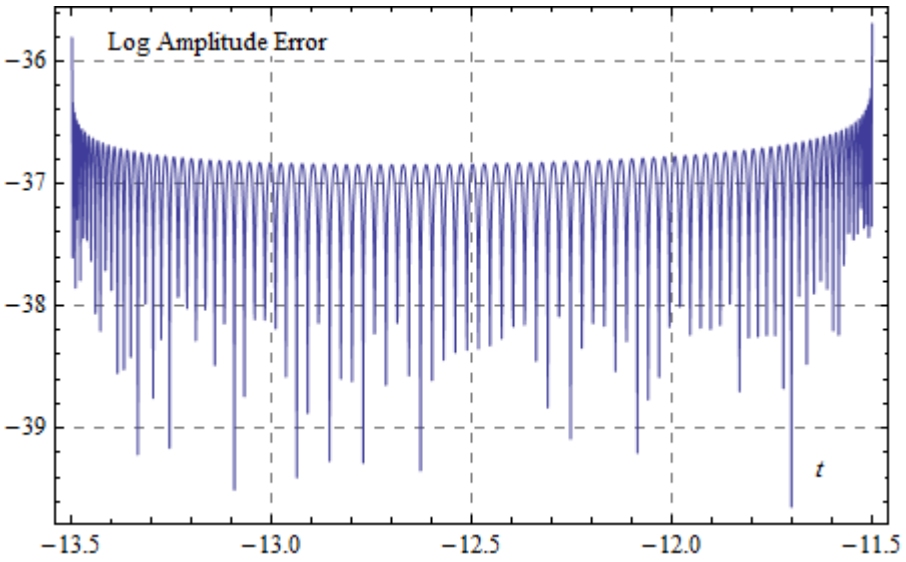

Figure 4.8.3: Logarithm of absolute error for reconstruction of time-shifted $f_1(t)$ using LPFs set 1

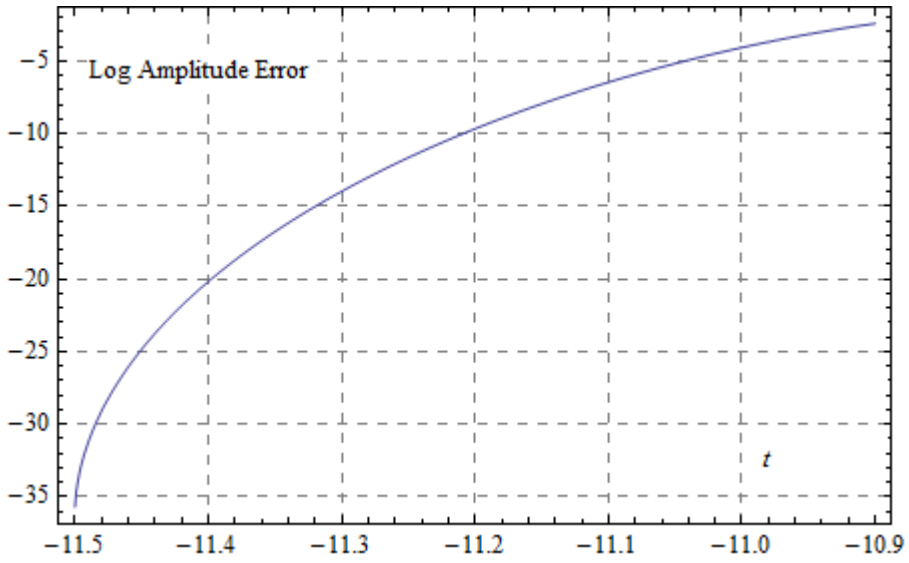

Figure 4.8.4: Logarithm of absolute error for extrapolation of time-shifted $f_1(t)$ using LPFs set 1

Table 4.1: List of expansion coefficients obtained for various test signals for the order $n = 0$ and $n > 2c/\pi$ using LPFs set 1.

Signal	$n = 0$	$n = 60$ $(> 2c/\pi)$
$f_1(t)$	-0.020656950289993160770520 9E+00	-0.048701665204667687887964 1E+00
$f_2(t)$	5.732737117763425105417159 3E+00	0.212465358167601684201325 5E+00
$f_3(t)$	0.346046382565123850445460 8E+00	2.496503313079186462728394 0E+00
$f_5(t)$	0.073023992462274286657212 2E+00	0.192397709588267645217099 0E+00
$f_6(t)$	0.572251244129688447641771 8E+00	-0.028350381550402300260729 3E+00
$f_7(t)$ Real	2.089544859346410457050963 9E+00	0.042106591929182892245544 9E+00
$f_7(t)$ Imaginary	0.005595127153522055925873 1E+00	-0.003842976996485399604059 0E+00
$f_1(t)$ Time Shifted	-0.046516101290455434962727 1E+00	-0.110941324727856769795951 2E+00
$f_8(t)$	7.483141181529866426719823 9E-28	-3.115649139513237398963396 8E+08

Table 4.2: List of expansion coefficients obtained for various test signals for the order $n = 0$ and $n > 2c/\pi$ using LPFs set 2.

Signal	$n = 0$	$n = 279$ $(> 2c/\pi)$
$f_4(t)$	-5.22839855528172649832185 53E-11	-0.74217428774063851045636 90E+00
$f_5(t)$	0.04068537375118342653823 43E+00	-1.30512157917238061572382 55E-68
$f_6(t)$	0.41782219030813452704753 10E+00	0.02237176252569657009994 27E+00

Upon careful inspection of the plots in figures 4.5.1 to 4.5.8 and figures 4.6.1 to 4.6.8, one could notice that when the same signal was extrapolated using the two LPF sets, the higher Slepian frequency set (set 2) was more accurate in their approximations (see the logarithm of absolute error plots) within their effective extrapolation range. The signals for which the extrapolation is shown in figures 4.1.1 to 4.1.4 and figures 4.2.1 to 4.2.4, although not exactly bandlimited because of the Gaussian functions involved, also gave good results (as seen in figures 4.2.3, 4.2.4), which are comparable to what was achieved using strictly bandlimited cases (figures 4.3.1 to 4.3.4, 4.4.1 to 4.4.4 and 4.6.1 to 4.6.8).

As expected, the effective extrapolation range was higher for the lower c LPFs set (set 1) as compared to the higher c LPFs set (set 2). The range of extrapolation is limited mainly due to the series truncation, N, the value of which should be at least greater than $2c/\pi$ for performing extrapolation. We also verified the effective extrapolation range that can be achieved analytically by using simple sinusoidal signals, for which the analytical expressions are known [8]. We found that, using LPFs set 1, the truncation value N was mostly equal to 100 (out of the total order of $n = 101$ being considered), while using LPFs set 2 it was more or less close to 300 (out of $n = 601$); both greater than their corresponding $2c/\pi$ values, thus allowing signal extrapolation.

However, it is worth mentioning at this point that what we considered here as the lower frequency set has been regarded as, or included in, the higher frequency group by Gosse in [21]. The effective extrapolation range in our case is significantly improved when compared with their results for the same signal (see page 1277 of [21]; and figures 4.1.1 to 4.1.4 in this chapter). In the same context, the error analysis also shows better results as our method has absolute error magnitude around 1e-38 (while it is 1e-02 in [21]) within the reconstruction interval $[-1,1]$. We also obtained better ratios of error magnitudes (varying smoothly from the order of 1e-36 to 1e-03 using our algorithm, while oscillating between 1e-02 and 1e-01 in [21]) in the effective extrapolation region, i.e. outside $[-1,1]$ (see absolute error plots in figures 4.1.3 and 4.1.4).

Table 4.3: Normalized mean-square error (NMSE) values obtained for various test signals.

Signal <LPF Set #>	**NMSE**
$f_1(t)$ <LPF Set 1>	0.00228449E+00
$f_2(t)$ <LPF Set 1>	8.58712915E-08
$f_3(t)$ <LPF Set 1>	0.00190264E+00
$f_4(t)$ <LPF Set 2>	0.00011602E+00
$f_5(t)$ <LPF Set 1>	2.43831466E-06
$f_5(t)$ <LPF Set 2>	1.00703838E-12
$f_6(t)$ <LPF Set 1>	6.77877025E-06
$f_6(t)$ <LPF Set 2>	1.89254260E-12
$f_7(t)$ <LPF Set 1>	8.43003145E-07
$f_1(t)$ Time Shifted <LPF Set 1>	3.21497874E+00

As mentioned earlier, we also performed extrapolation on an LCT bandlimited signal to compare our method with existing ones. We chose the same signal used by Shi *et al.* in [17]. It is given by the function $f_7(t)$ (the *sinc* function used is a normalized *sinc* function) in the results. The extrapolation and error analysis for the real and imaginary parts of the signal are shown in figures 4.7.1 to 4.7.4 and 4.7.5 to 4.7.8, respectively. The performance was measured by calculating the NMSE of the actual and extrapolated signals. The NMSE for the aforementioned signal, $f_7(t)$, using our algorithm is 8.430e-07, as shown in table 4.3 which also has the NMSE listed for other test signals considered here as well. The corresponding NMSE using the iterative algorithm proposed in [17] is 1.037e-04, and NMSE using the generalized PSWFs expansion method proposed in [18] is only 0.685e+00. Thus our algorithm is shown performs superiorly even for LCT bandlimited signals despite the fact that it is not the primary objective of our work.

To check the robustness of our algorithm, within the ideal noise-free domain of the test signals, we considered a time-shifted version of the signal $f_1(t)$ also for extrapolation. The observation interval in this case was from $[-13.5, -11.5]$ instead of $[-1,1]$ and the extrapolation was done using LPFs set 1. The signal was extrapolated as expected; the error analysis and extrapolation plots can be seen in figures from 4.8.1 to 4.8.4.

Tables 4.1 and 4.2 hold some of the relevant scalar expansion coefficients obtained for extrapolation of the various test signals using LPF set 1 and LPF set2 respectively. Barring one exception (for the signal $f_8(t)$), we can see that these coefficients are all well-contained, not running away to assume huge values, thus enabling us to use them effectively for extrapolation.

A non-bandlimited test case was also considered, out of curiosity, to confirm with the theory established [2,3] that it is not extrapolatable. For examining this, we tried extrapolating the signal $f_8(t)$ using the LPFs set 1 which has a lower bandwidth parameter, $c = 20\pi$, which is less than the maximum frequency 30π of the test signal. Confirming what was studied from the theory, the scalar expansion coefficients obtained from the analysis (3.3) assumed huge values for $n > 2c/\pi$, as evident from table 4.1 for the signal $f_8(t)$, thus creating numerical errors rendering the extrapolation incorrect.

4.2 Extrapolation of Perturbed Signals

The known ideal signals were perturbed or altered in some way to create some disturbance to study if the algorithm is tolerable at all to noise. Two types of perturbations were studied on an otherwise known ideal test signal:

- Less precision
- Induced error

As the proposed algorithm demands very high precision of the signal being studied, we decided to test how our model responded to varying precisions of an input signal (ideally known). The precision of the input test signal alone was changed, while that of the LPFs set were preserved during computations. For the second type of perturbation, the induced error type, we were looking into the possibility of our test signal getting corrupted; the least case of which would be that affecting only one of the samples of the input signal. To be precise, we induced an error to the original test signal by changing only one significant digit at varying positions (as given in table 4.5) after the decimal point of the sample at the equivalent continuous time $t = 0.009$ seconds, which falls within the closed

observation interval of $[-1,1]$ and used such a signal for extrapolation. For both of these test cases, we chose the signal $f_1(t)$ as the original test input. The results for carrying out analysis (3.3) to obtain the expansion coefficients are shown in tables 4.4 and 4.5 along with those obtained using the original ideal signal $f_1(t)$ itself.

A careful analysis of table 4.4 shows that for input signal precisions of 5, 30 and 50, the scalar coefficients of expansion were not contained and hence were not suitable for extrapolation. As the precision increased from 70, 80, 100 and finally 200, the coefficients start to get contained, and for a precision of 100 and 200 they were almost the same for roughly up to around 25 decimal digits. It is worth remembering now that these coefficients still had around 200 digits of precision as obtained from the LPFs set used. For ease of presentation, only 8 significant digits after the decimal point are shown here in the tables.

The extrapolation and error analysis plots for varying precisions of the input signal $f_1(t)$ is shown in figures from 4.9.1 to 4.9.4, 4.10.1 to 4.10.4 and 4.11.1 to 4.11.4. The error plots both for reconstruction (figures 4.9.3, 4.10.3 and 4.11.3) and extrapolation (figures 4.9.4, 4.10.4 and 4.11.4) clearly show that for increasing precision of the input signal, the algorithm performs better with lesser (absolute) errors in its approximation of the original signal. The results for the case shown in figures 4.11.3 and 4.11.4 which used a precision of 100 digits is even comparable with that of the ideal signal test case without any change in its actual precision (see figures 4.1.3 and 4.1.4).

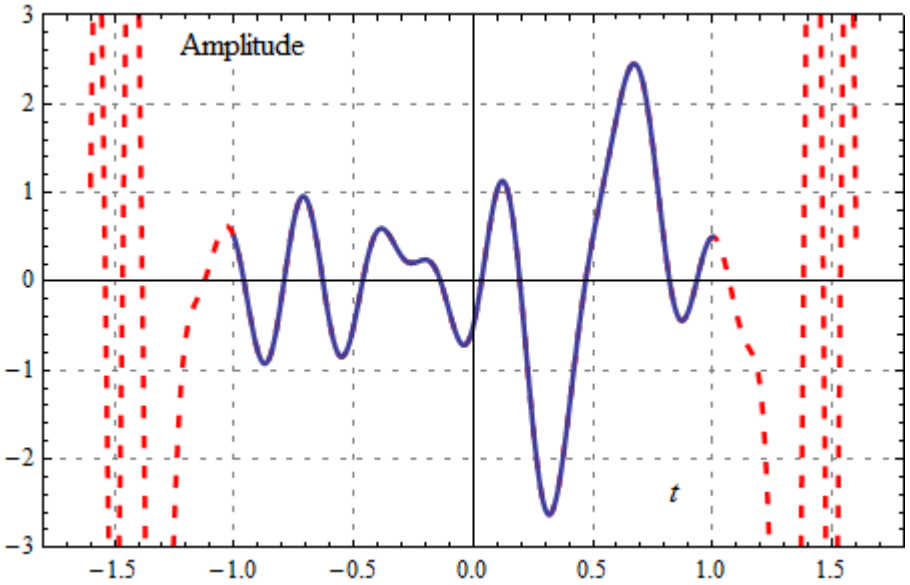

Figure 4.9.1: $f_1(t)$ with 70 digits of precision original [solid] and extrapolated [dashed] versus time using LPFs set 1

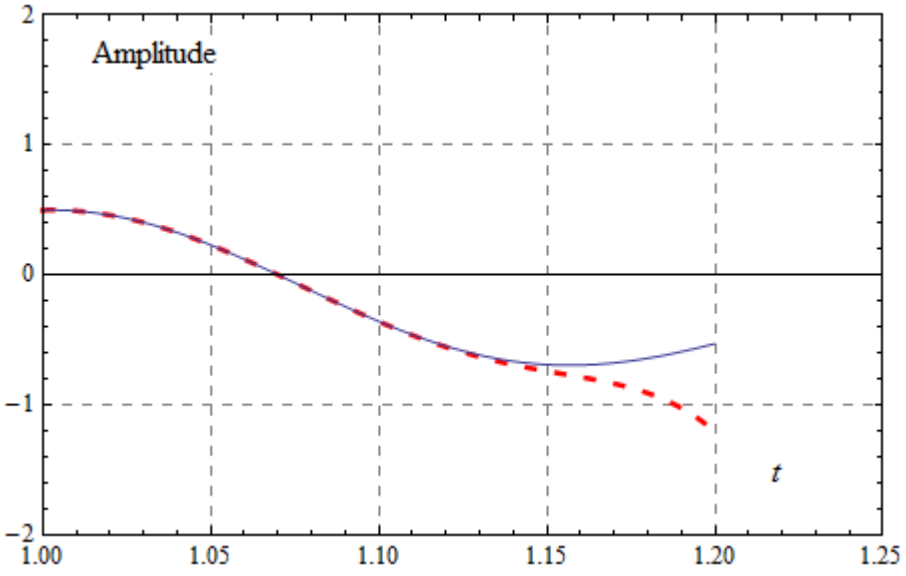

Figure 4.9.2: Magnification of extrapolation of $f_1(t)$ with 70 digits of precision versus time using LPFs set 1

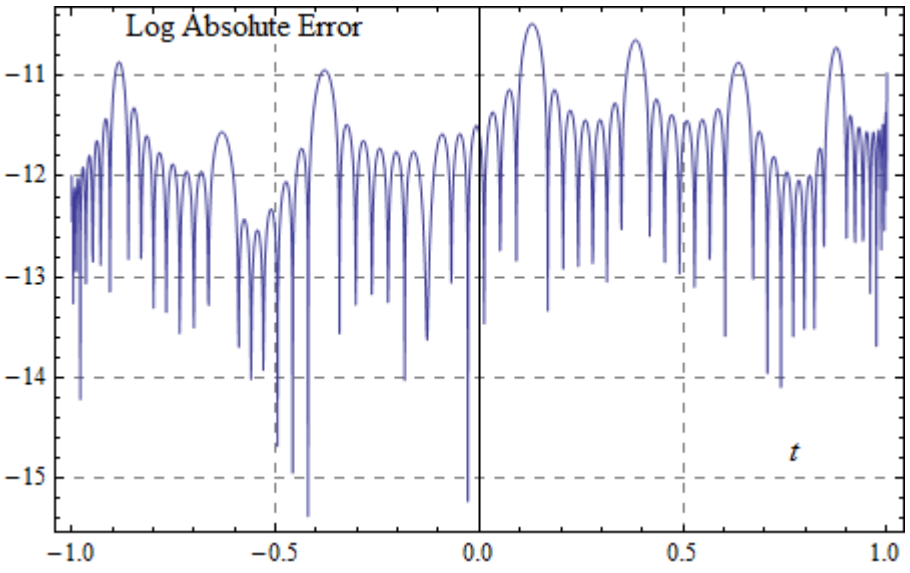

Figure 4.9.3: Logarithm of absolute error for reconstruction of $f_1(t)$ with 70 digits of precision using LPFs set 1

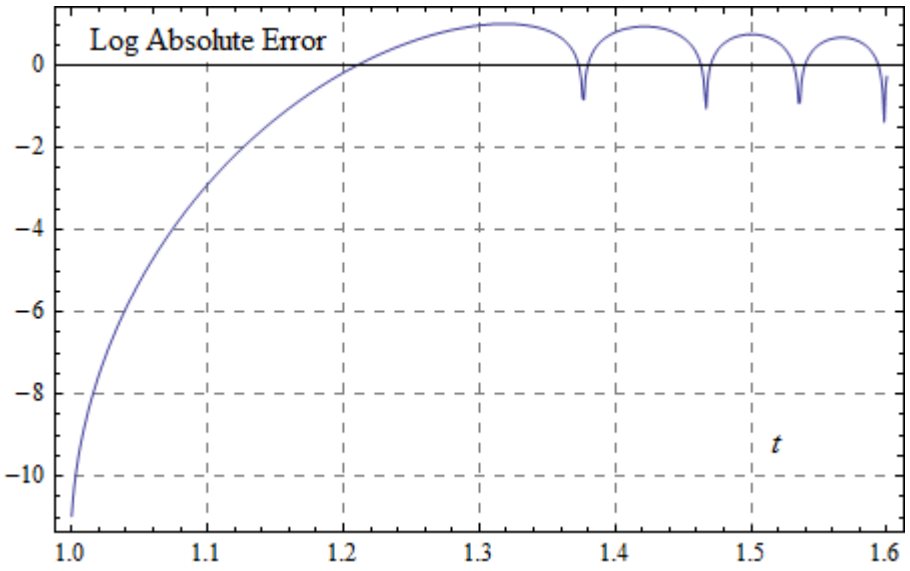

Figure 4.9.4: Logarithm of absolute error for extrapolation of $f_1(t)$ with 70 digits of precision using LPFs set 1

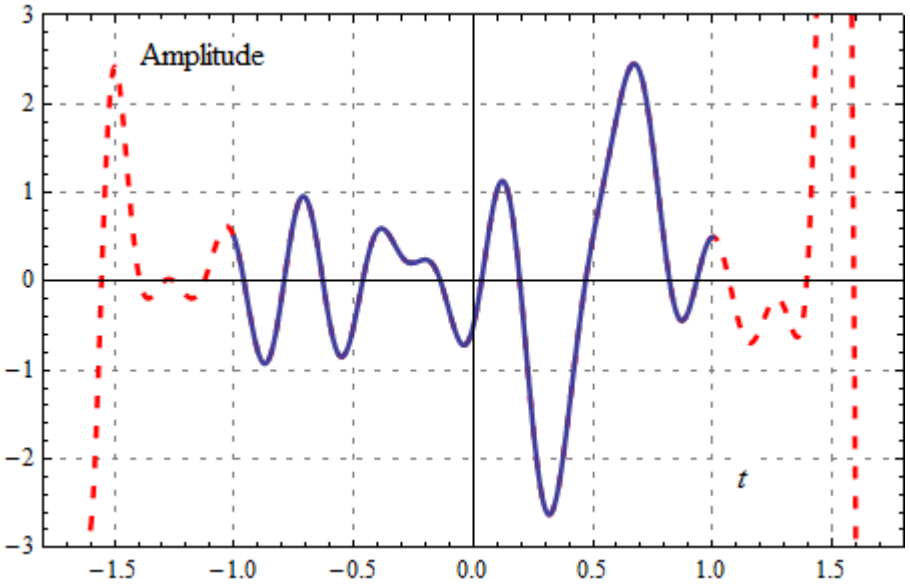

Figure 4.10.1: $f_1(t)$ with 80 digits of precision original [solid] and extrapolated [dashed] versus time using LPFs set 1

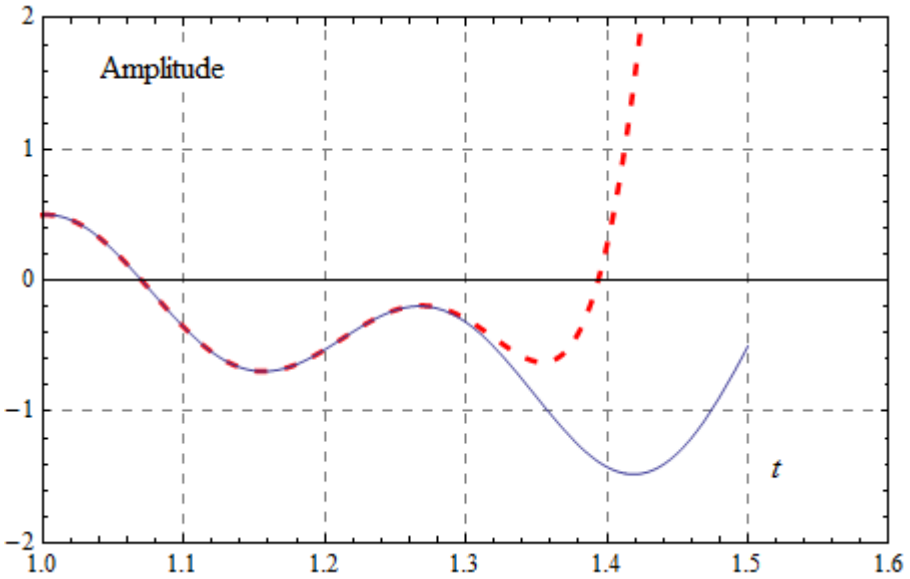

Figure 4.10.2: Magnification of extrapolation of $f_1(t)$ with 80 digits of precision versus time using LPFs set 1

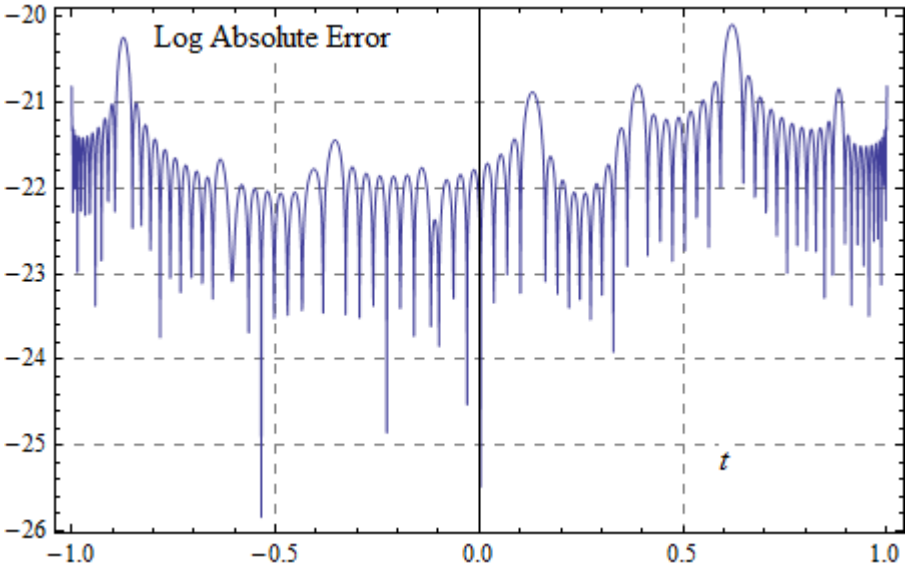

Figure 4.10.3: Logarithm of absolute error for reconstruction of $f_1(t)$ with 80 digits of precision using LPFs set 1

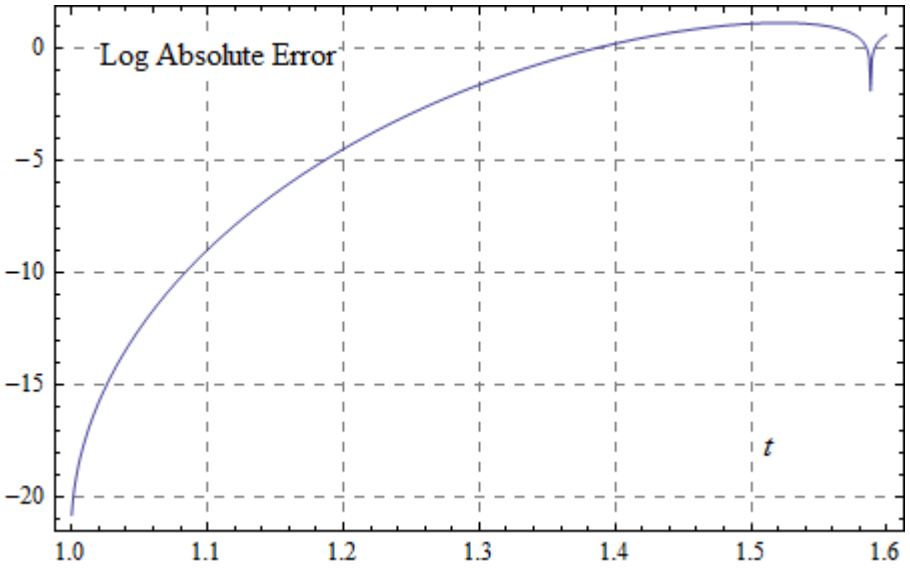

Figure 4.10.4: Logarithm of absolute error for extrapolation of $f_1(t)$ with 80 digits of precision using LPFs set 1

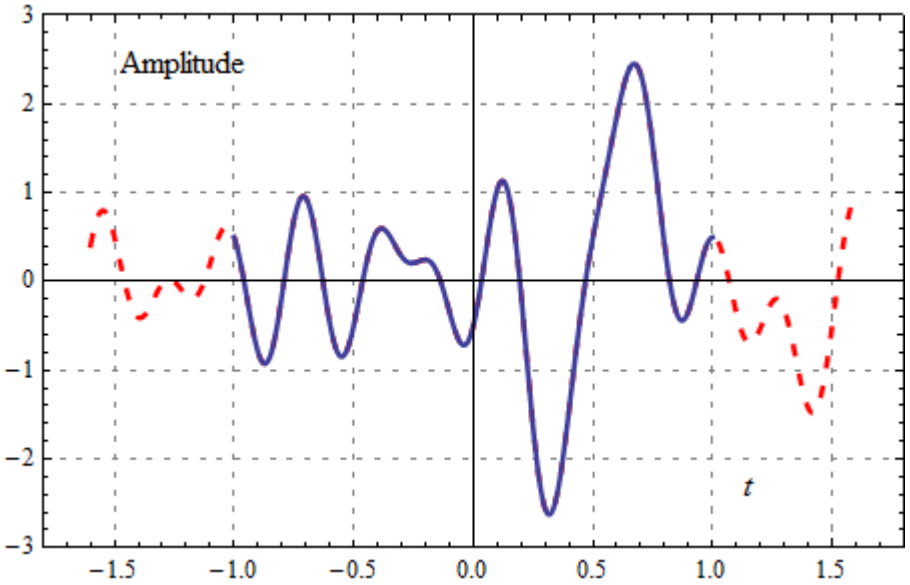

Figure 4.11.1: $f_1(t)$ with 100 digits of precision original [solid] and extrapolated [dashed] versus time using LPFs set 1

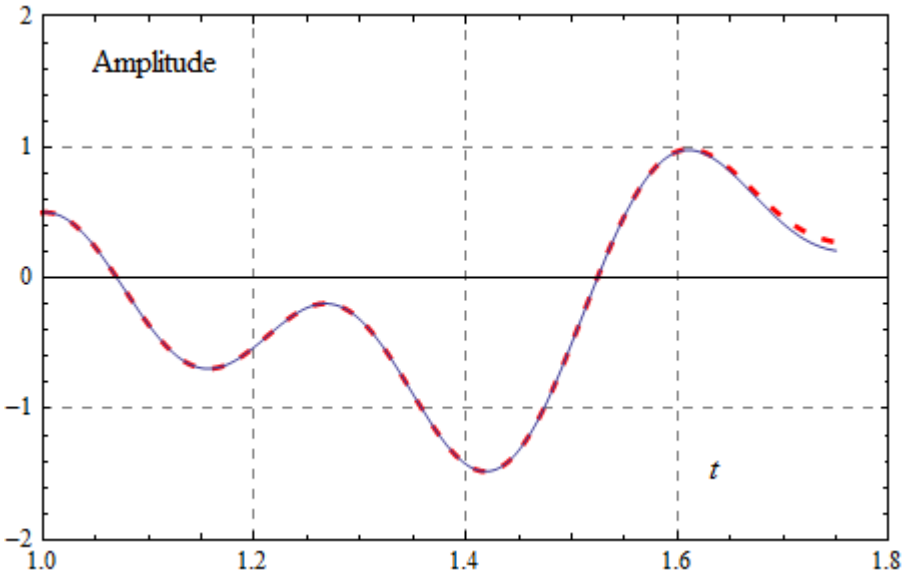

Figure 4.11.2: Magnification of extrapolation of $f_1(t)$ with 100 digits of precision versus time using LPFs set 1

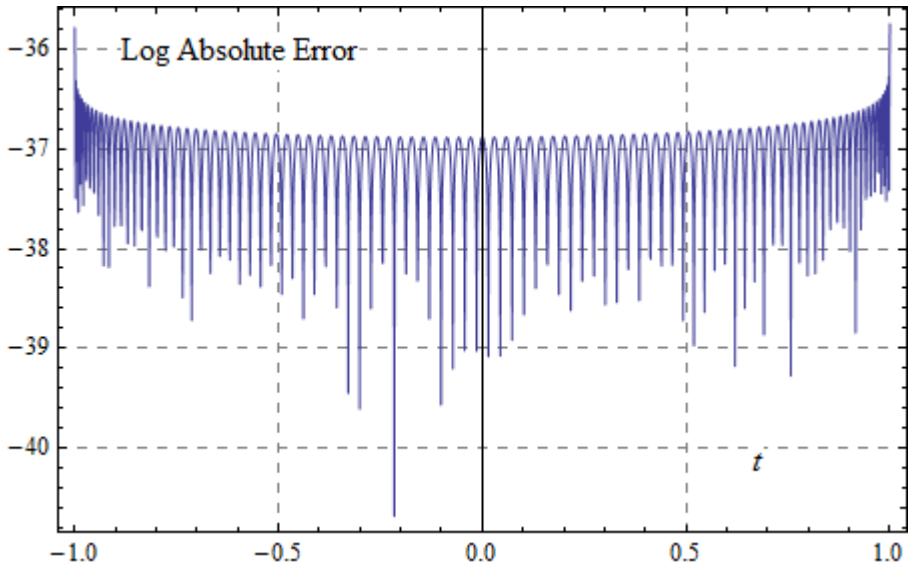

Figure 4.11.3: Logarithm of absolute error for reconstruction of $f_1(t)$ with 100 digits of precision using LPFs set 1

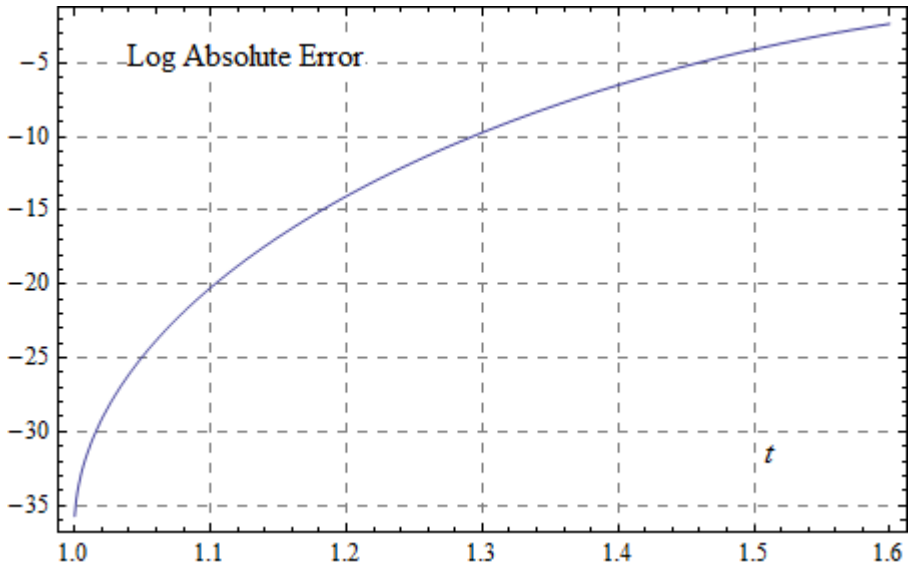

Figure 4.11.4: Logarithm of absolute error for extrapolation of $f_1(t)$ with 100 digits of precision using LPFs set 1

Table 4.4: List of expansion coefficients obtained for varying precisions (less-precision case) of the input signal $f_1(t)$ for the order $n = 0$ and $n = 2c/\pi$ using LPFs set 1.

Precision	$n = 0$	$n = 40$ $(=2c/\pi)$	$n = 60$ $(>2c/\pi)$
5	-7.71169588E+60	1.51970776E+61	-3.02186849E+70
30	-1.65894258E+28	-5.11665568E+27	1.58270745E+37
50	1.68985568E+08	-3.24908355E+07	-2.63426837E+17
70	-0.02065695E+00	0.09008843E+00	-0.04854151E+00
80	-0.02065695E+00	0.09008843E+00	-0.04870166E+00
100	-0.02065695E+00	0.09008843E+00	-0.04870166E+00
200	-0.02065695E+00	0.09008843E+00	-0.04870166E+00
$f_1(t)$ (original)	**-0.02065695E+00**	**0.09008843E+00**	**-0.04870166E+00**

Table 4.5: List of expansion coefficients obtained for the order $n = 0$ and $n = 2c/\pi$ after changing 1 significant digit of the input signal sample at $t = 0.009s$ for varying positions after the decimal point (induced-error case). The input signal is $f_1(t)$ and LPFs set 1 was used.

Position	$n = 0$	$n = 40$ $(=2c/\pi)$	$n = 60$ $(>2c/\pi)$
1	1.03057427E+10	3.67374224E+09	6.74566692E+18
5	1.03057424E+06	3.67374314E+05	6.74566692E+14
10	41.20231386E+00	14.78505740E+00	2.69826676E+10
20	-0.02065695E+00	0.09008843E+00	-2.74696843E+00
$f_1(t)$ (original)	**-0.02065695E+00**	**0.09008843E+00**	**-0.04870166E+00**

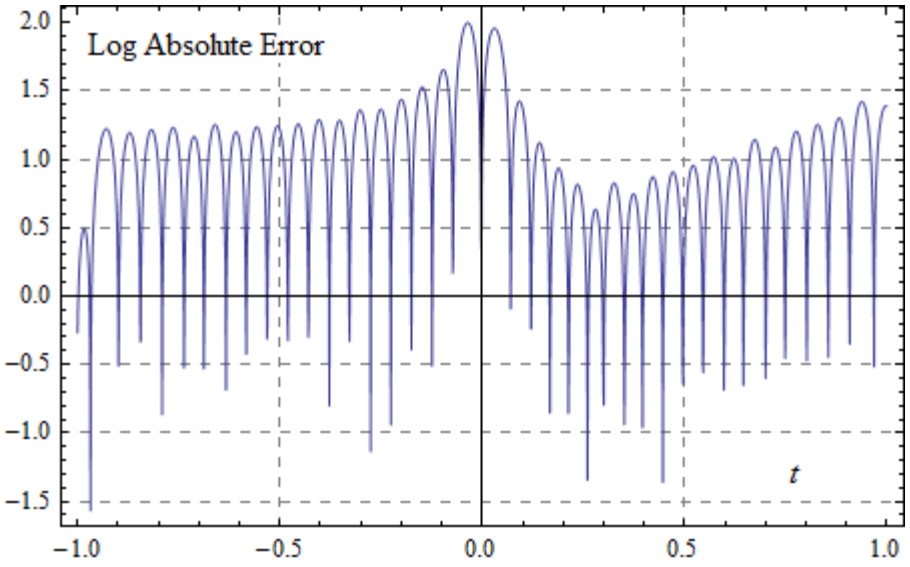

Figure 4.12.1: Logarithm of absolute error for reconstruction of $f_1(t)$ with error introduced at 10th position after the decimal point of the sample at $t = 0.009s$; LPFs set 1 used.

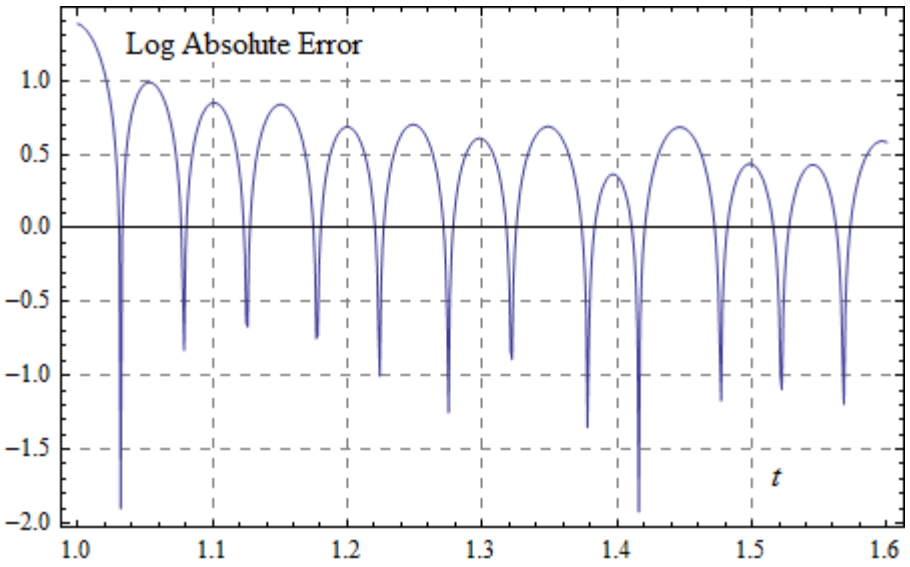

Figure 4.12.2: Logarithm of absolute error for extrapolation of $f_1(t)$ with error introduced at 10th position after the decimal point of the sample at $t = 0.009s$; LPFs set 1 used.

The expansion coefficients obtained for the induced error case (table 4.5) shows that the values only start to get contained and usable when the error induced was at a position of 20 digits after the decimal point. Though the coefficients seem to be contained for both the positions 10 and (even more so for) 20, the absolute error plots (figures 4.12.1 and 4.12.2) for the case when such a signal (for position 10) was extrapolated shows that the algorithm is greatly intolerable to noise introduced of any kind within the precision range of around 100 to 200 digits. These observations reinforce the fact that the signal being studied should not only be bandlimited and precise up to 100-200 digits, but also be free from all kinds of noise that can affect their discrete time values within this specified precision range.

4.3 Extrapolation of Real Signals

Extrapolation attempts on real or experimental signals are presented in this section. In general, two types of signals were considered; financial and medical related. Forex data was the financial signal which consisted of minute-by-minute, hourly and daily data of various currency pairs. The medical signals were acquired from Jose Gonzalez-Cueto which he collected as part of his doctoral thesis [36]. It consisted of one compound nerve action potential (CNAP) signal and one heart-rate signal. The forex signals had a precision of around 5 digits whereas the medical signals had it around 8 (see figures 4.13.1, 4.14.1 and appendix for plots). Both LPF sets 1 and 2 were used in our attempts to extrapolate these signals. The results were not so satisfactory, but were, in a way, expected as these signals did not have enough precision in their approximations. The

expansion coefficients obtained up on analyzing (3.3) these signals, shown in tables 4.6 to 4.9, sums up our expectations.

Table 4.6: List of expansion coefficients obtained by analyzing forex signals for the order $n = 0$ and $n = 2c/\pi$ using LPFs set 1.

Currency Pair	$n = 0$	$n = 40$ $(=2c/\pi)$
AUDUSD (Minute by minute)	-1.456742267568970571656107 8E+64	-1.012316131573940518973512 3E+64
AUDCAD (Hourly)	-3.268776823157823024796478 2E+64	-2.946017561561202110356123 6E+64
USDJPY (Hourly)	7.725111374871867722540950 4E+66	5.246899560417632240400870 2E+66
USDJPY (Daily)	-4.775640550735935999236048 4E+67	-9.735954691265813344325445 9E+67

Table 4.7: List of expansion coefficients obtained by analyzing various forex signals for the order $n = 0$ and $n = 2c/\pi$ using LPFs set 2.

Currency Pair	$n = 0$	$n = 200$ $(=2c/\pi)$
AUDUSD (Minute by minute)	-3.938064963652056035582894 0E+63	-2.848503460989933110374694 4E+64
AUDCAD (Hourly)	-7.213942243079008297630912 7E+63	8.720768734492087156391635 4E+64
USDJPY (Hourly)	1.371392685037530117278180 7E+66	1.655773947387909213496499 6E+67
USDJPY (Daily)	-9.629117879065289295914531 6E+66	-2.620961929125456662663171 1E+68

Table 4.8: List of expansion coefficients obtained by analyzing medical signals for the order $n = 0$ and $n = 2c/\pi$ using LPFs set 1.

Signal	$n = 0$	$n = 40$ (=$2c/\pi$)
CNAP	1.25883195375853763328929 48E+66	-2.40298984498008785604240 30E+66
Heart Rate	-5.72847222690587036574328 94E+67	-1.13824146703650823117838 22E+67

Table 4.9: List of expansion coefficients obtained by analyzing medical signals for the order $n = 0$ and $n = 2c/\pi$ using LPFs set 2.

Signal	$n = 0$	$n = 200$ (=$2c/\pi$)
CNAP	4.06076824671116145003781 11E+65	-5.96956340217465482884866 08E+66
Heart Rate	-1.16573267812643180622637 17E+67	-1.53516589362225018582982 55E+67

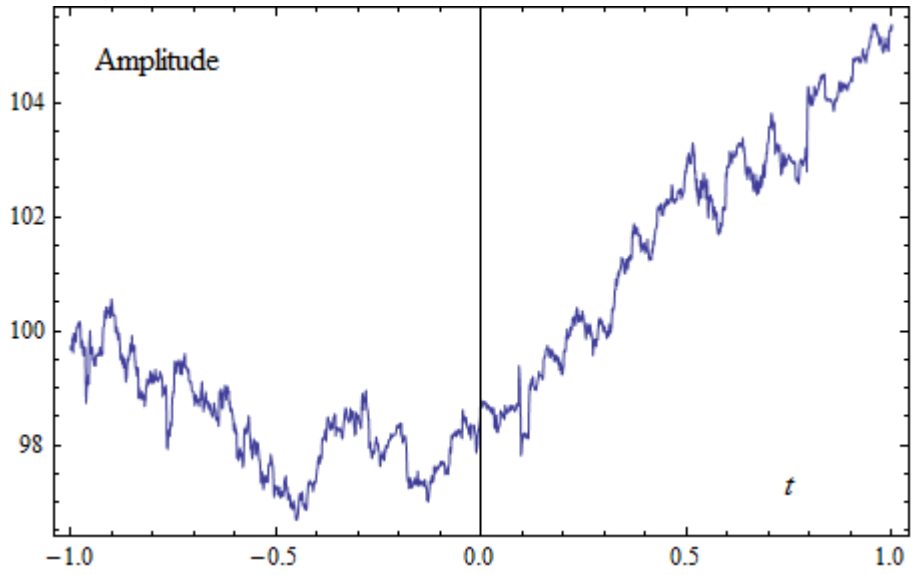

Figure 4.13.1: Forex signal of USDJPY currency pair (hourly) observed from 2013 to 2014

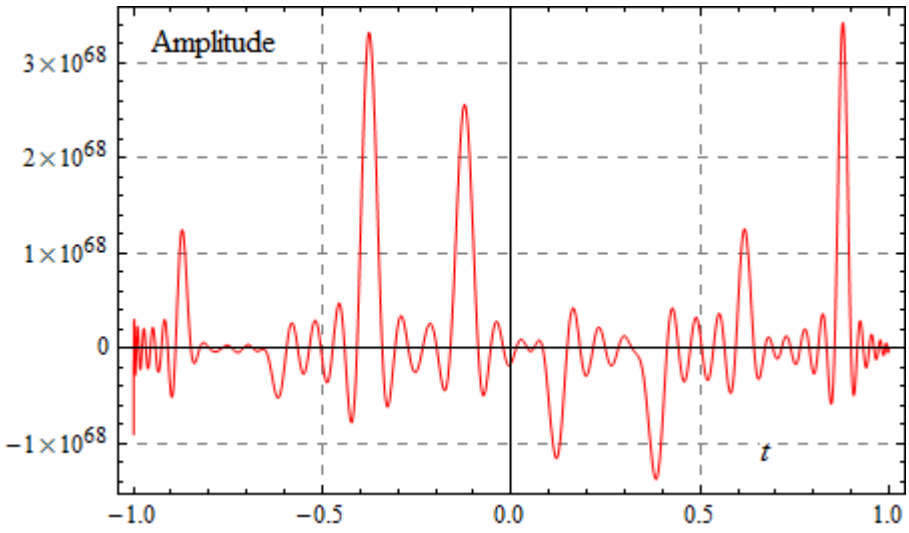

Figure 4.13.2: Reconstruction plot of USDJPY (hourly) signal using LPFs set 1

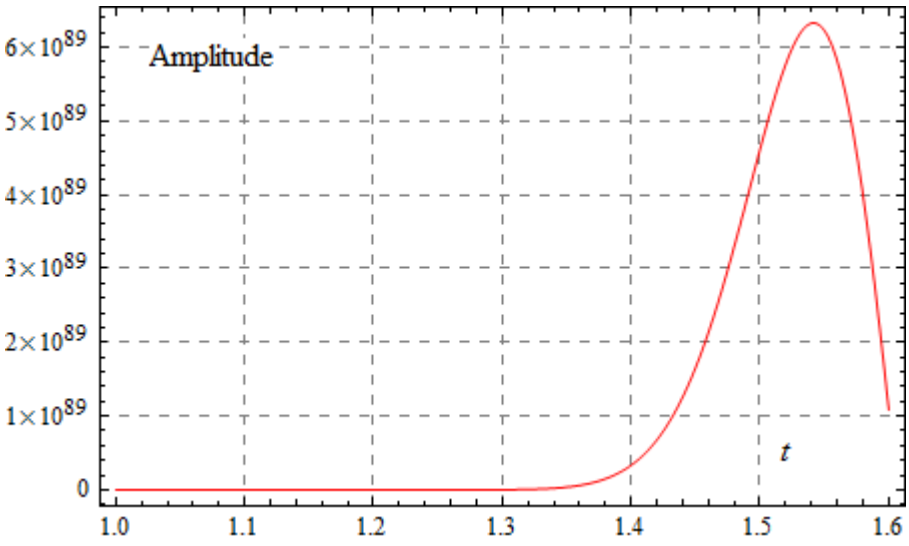

Figure 4.13.3: Extrapolation plot of USDJPY (hourly) signal using LPFs set 1

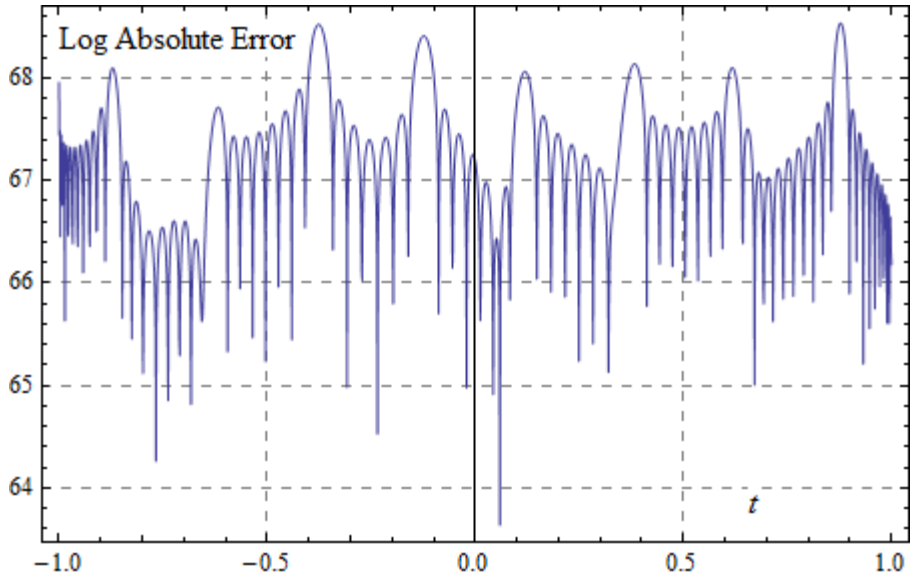

Figure 4.13.4: Logarithm of absolute error for reconstruction of USDJPY (hourly) signal using LPFs set 1

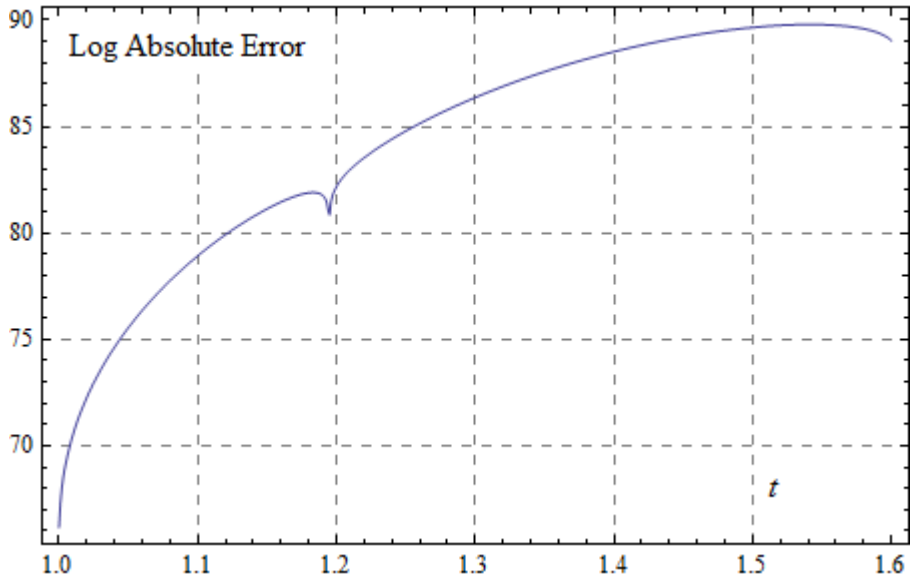

Figure 4.13.5: Logarithm of absolute error for extrapolation of USDJPY (hourly) signal using LPFs set 1

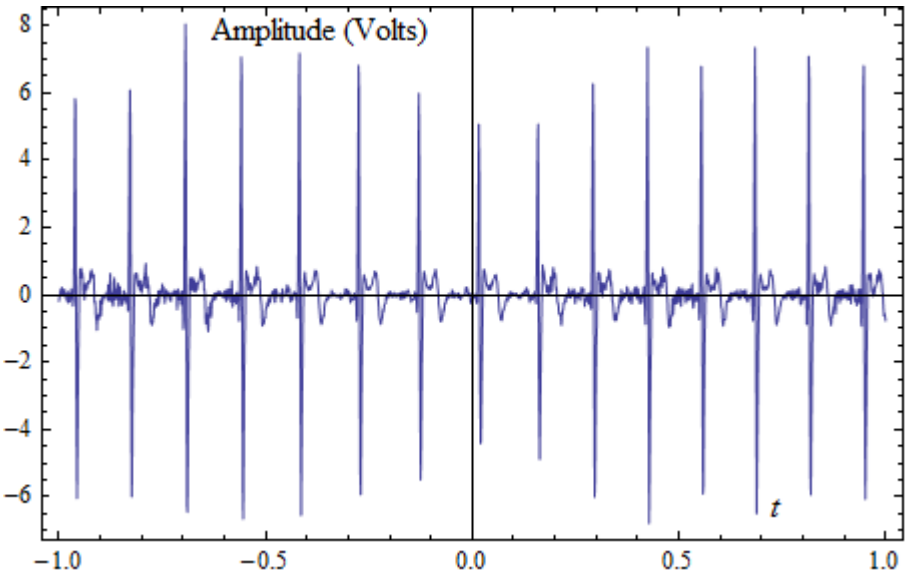

Figure 4.14.1: Heart-rate signal obtained from Gonzalez-Cueto's study [36]

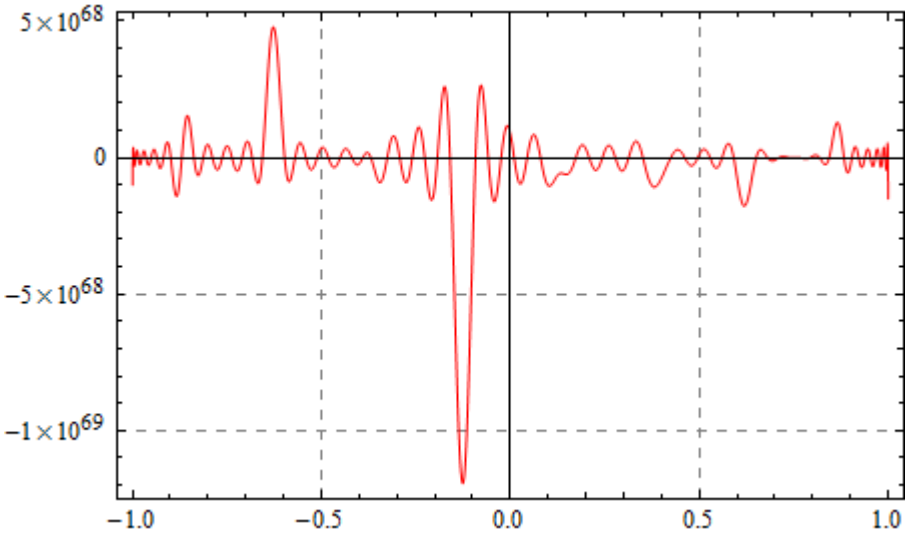

Figure 4.14.2: Reconstruction plot of heart-rate signal using LPFs set 1

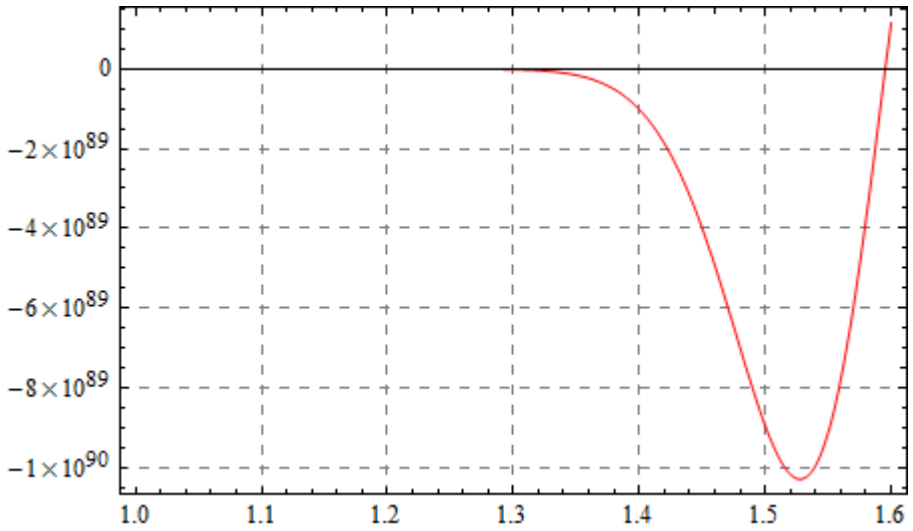

Figure 4.14.3: Extrapolation plot of heart-rate signal using LPFs set 1

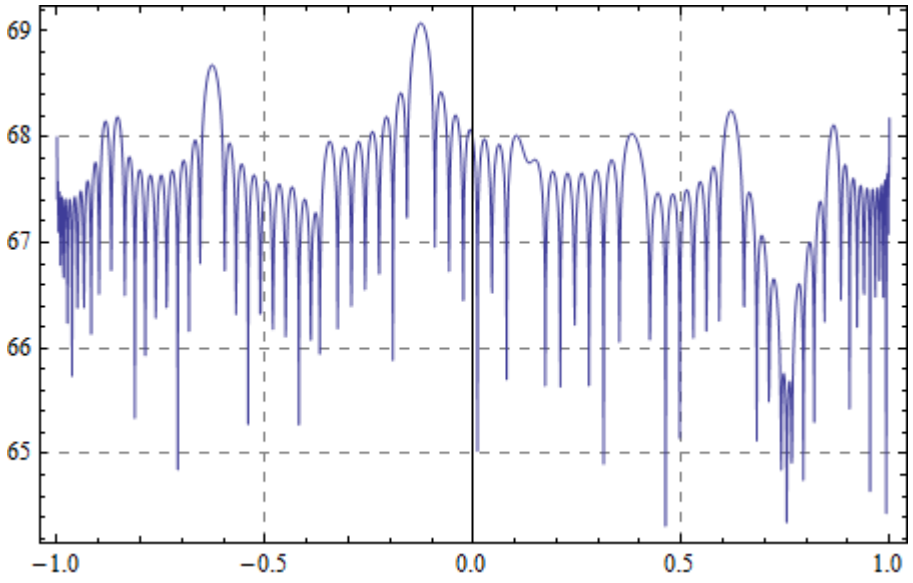

Figure 4.14.4: Logarithm of absolute error for reconstruction of heart-rate signal using LPFs set 1

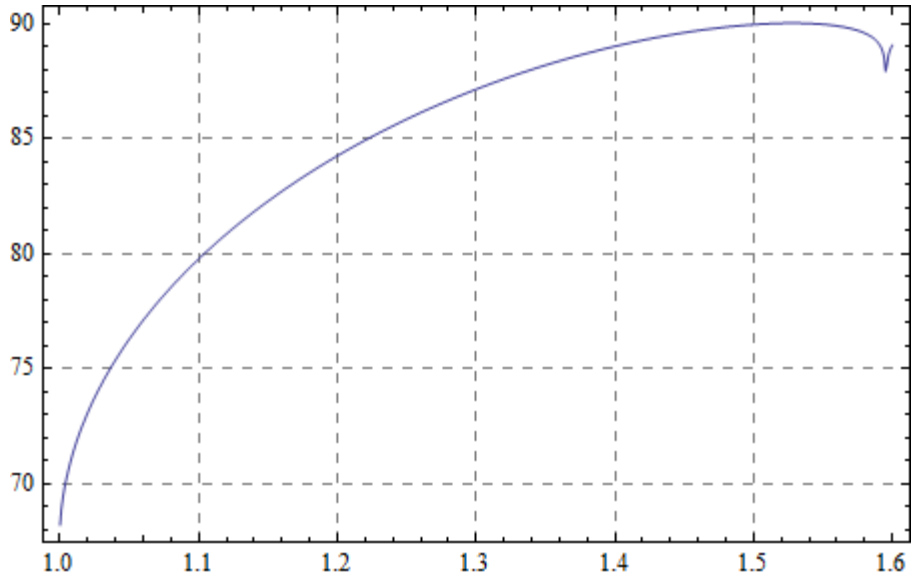

Figure 4.14.5: Logarithm of absolute error for extrapolation of heart-rate signal using LPFs set 1

As evident from the tables (4.6 to 4.9) and figures (4.13.1 to 4.13.5 and 4.14.1 to 4.14.5), the expansion coefficients were not contained at all for the real experimental signals and hence were not suitable for use in extrapolation. Some of the possible reasons for this behavior of the scalar coefficients are discussed below. The signal might have been inadequately precise, with the precision only reaching up to around 5-8 digits. As evident from the experiments done in the previous sections, our algorithm requires very high precision of the input signal; with the signal being precisely known up to 100-200 digits after the decimal point. Another possible reason would be the presence of aliasing effects in the discretely bandlimited signal. Though the signal is inherently bandlimited due to sampling/discretization-in-time, there might be higher frequency components present in the signal that exceeds the limit imposed by the LPFs set used. Comparing the expansion

coefficients for the bandlimited and non-bandlimited test cases as shown in tables 4.1 and 4.2, we can see that the coefficients were well contained for orders $n < 2c/\pi$ for both cases, but assuming bigger numbers beyond the critical value, $2c/\pi$, for the non-bandlimited case (table 4.1, signal $f_8(t)$). When it came to corrupted test signals (tables 4.4 and 4.5), either due to being less precise or because of having induced errors, the expansion coefficients were not contained for the cases of lower precision and error induced at lower positions after the decimal point (error at greater significant positions of the sample). The coefficients assumed these large ill-suited values irrespective of the order, n, whether they were within or beyond the critical value, $2c/\pi$. As the precision of the signal was increased, and also when the induced-error position was changed to lower significant digits (higher positions after the decimal point, as shown in table 4.5), the coefficients started to get contained and thus becoming suitable for use in extrapolation. Presence of noise is another factor that can adversely affect our extrapolation algorithm. This may occur due to improper or inconsistent data acquisition methods used in connection with these experiments. As we saw in the induced-error test case, even a small change in the sampled values can impart significant changes in the way these signals get analyzed. Even known test signals will not be working satisfactorily, if there is some noise or if the precision is reduced. As our algorithm uses a more general method of numerical integration, less precise signals may not be even interpolatable or cannot be reconstructed using the analysis (3.3) and synthesis (3.4) equations.

Chapter 5

Conclusion

We have presented an implemented robust and efficient algorithm for bandlimited signal extrapolation valid up to basically an arbitrarily high range of frequencies. Even though the algorithm is complex in the sense that it involves time consuming calculations and tedious computations with big matrices of very high precision, the overall idea is simple and easy to execute, thanks to the current computational speeds and available system memory. We believe that the accuracy with which the LPFs and their corresponding eigenvalues were computed with very high precision allowed this method to work efficiently thus making it suitable for extrapolating signals within the prescribed bandwidth.

The model chosen was a noise-free model that performed extremely well under ideal conditions. Extrapolation of known test signals showed that the effective range of extrapolation is dependent on the truncation value N and also, in some way, to the Slepian frequency, c. As we increased N (incorporating more high orders of n), the effective extrapolation range also increased. We were able to achieve relatively higher truncation values for the lower c LPF set 1 and thus able to attain better extrapolation range as compared to what was achieved using the higher c LPF set 2. When it comes to the accuracy of the results within the effective extrapolation range, the higher c LPF

set 2 performed better than that of their lower c counterparts.

When subjected to noisy conditions the algorithm almost collapsed with very high scalar expansion coefficients, hindering proper extrapolation. Inadequate precision of the original signal, and, the signal getting corrupted by some sort of noise, are the two main reasons attributing to this behavior. We found that the complexity of running our algorithm is mainly dependent on two factors – the Slepian frequency, c, used and, the overall precision used for computation. When both these factors were increased, the overall computation time of our algorithm also increased. On the lab computer, which had an Intel® Core™ i5-2400 CPU @ 3.10 GHz processor and 8GB of RAM, it took around 2.5 hours for extrapolating using LPF set 1 ($c = 20\pi$) and around 10 hours using LPF set 2 ($c = 100\pi$). We believe that a qualitative analysis of the behavior of our algorithm to noisy signals (provided in the last chapter) may serve as a guide or reference to someone who is interested in taking this study a step further, to design a better model, in the future.

5.1 Suggestions for Further Study

This research can be regarded as a preliminary fundamental work for implementing effective bandlimited signal extrapolation using linear prolate functions as proposed by Slepian and his colleagues in [2]. This work was mainly concentrated on noise-free signals. However, similar studies [38] have been going on in parallel within our research group for interpolation using inbuilt Mathematica functions for applications related to digital filters. This abovementioned work focuses on reducing intersymbol interference using LPFs. Using this technique, the interpolation is satisfactorily achieved for some real

signals but fails in their extrapolation. For this, the interpolation order needs to be tailored according to the order of the LPFs used and it also depends on the signal being studied as well. We believe there is scope for a lot more to be done in this field. Some of them are listed below.

- Modify the algorithm to incorporate higher c LPF sets.
- Include noise-affected signals as well in the design, which may be more applicable in a practical sense.
- An extensive study to formulate a relationship between the Slepian frequency, c, the truncation value, N, the observation interval, [-t_0, t_0], and the effective extrapolation range, for achieving satisfactory signal extrapolation, is highly recommendable and would be of great use for application in the future models.
- Implement 2-D extrapolation, that can be applied in areas of signal processing like image processing, by studying further on circular prolate functions (two dimensional PSWFs) [33].
- Investigate in detail the reasons for the ill-behavior of the expansion coefficients especially for values of n above the critical value, $2c/\pi$.
- Study the relationship between the position of the induced error (noise) and the highest possible order n for which useful expansion coefficients are obtained.
- Formulate other efficient and more general ways of high precision numerical implementation of the overlap integral.
- Examine the relationship between the sampling rate of the discretized signal and the order of the resulting piecewise polynomial for finding the overlap integral.

Characterizing the Slepian functions (LPFs set) finely and precisely into an appropriate polynomial expression is the key with which, this method could be extended to other LPF sets. During the characterization process, emphasis should be on incorporating more higher order (n) terms in the synthesis equation (3.4) thus making N sufficiently large for extrapolation. This is a promising development in the field of signal processing [15-17,29,30] and will be helpful in the characterization of both known and random bandlimited observations. It should be stressed, however, that the key to the successful better-than-others results of our extrapolation algorithm is, indeed, the accurate numerical evaluations of linear prolate functions and their eigenvalues employing our proprietary robust algorithm for computing them.

Appendix

A sample Mathematica code used in our algorithm for signal extrapolation, its explanation and some relevant figures are presented here in this section.

A.1 Sample Mathematica Code

The code for implementing bandlimited signal extrapolation was written in Mathematica version 9.0.1.0. The sample provided here is that written for the LPFs set 1 with Slepian frequency, $c = 20\pi$. This code assumes that the "*sample_code.nb*" Mathematica notebook file (containing the code) is run from a folder named "*Mathematica*" located in the "*My Documents*" folder in a Windows 7 (64-bit) operating system. The relevant parts and purpose of the code are explained there itself. Interested readers are advised to refer to [37] for a more detailed explanation of the commands used.

Comment: This code adds the directory in which the files are located for our program.

```
AppendTo[$Path,ToFileName[{$HomeDirectory,                "My
Documents\\Mathematica\\"}]];
```

Comment: This code is for importing the high precision LPFs set 1.

```
ko = << ksi0-20pi100c;
```

```
lo = << lms-20pi100c;

ksi = << ksi-20pi2000;
```

Comment: Initialization of a bunch of variables to work with and hold relevant results. The variable 'expval' is initialized here as a list of length 2001 containing zeros. This variable is actually supposed to be holding the 2001 samples of the signal to be analyzed within the observation interval $[-1,1]$.

```
yvalue = Table[0, {101}, {2001}];

expval = Table[0, {2001}];

nval = cresult = lovalue = expCoeff = Table[0, {101}];

functionlist = Table[0, {101}, {8}];
```

Comment: Segment wise piecewise polynomial formulation and subsequent numerical implementation of the calculation of overlap integral is done in this block. In essence, the analysis (3.3) equation is implemented here. The key functions to be noted here are *LinearSolve* and *Integrate*. *LinearSolve* solves for x that satisfies the equation $Mx = b$. *Integrate* is used here to compute the definite integral obtained in each segment of the piecewise-polynomial form of the scalar product in the overlap integral (see section 3.4). The resulting scalar expansion coefficients for various orders of n are stored in the variable 'expCoeff'.

```
Do[pans = res1 = 0;

  yplus = Rationalize[ksi[[nvalue, Range[1, 1001]]], 0];
```

```
  yminus =

   If[OddQ[nvalue],

    Reverse[yplus[[2  ;;  1001]]],  -1*Reverse[yplus[[2  ;;
1001]]]];

  yfull = Join[yminus, yplus];

  ynew = expval*yfull;

  y = N[ynew, 500];

  nval[[nvalue]] = nvalue;

  Do[xdata = Table[x, {x, i/1000, (i + 250)/1000, 1/1000}];

   n = Dimensions[xdata][[1]];

   M = Table[xdata[[i + 1]]^j, {i, 0, n - 1}, {j, 0, n -
1}];

   A = LinearSolve[M, y[[i ;; i + 250]]];

   fp[z_] := Sum[A[[j]]*z^(j - 1), {j, 1, n}];

   functionlist[[nvalue, 1 + (i - 1)/250]] = fp[z];

   Do[yvalue[[nvalue, j]] = fp[j/1000];

    If[j  ==  2000,  yvalue[[nvalue,  j  +  1]]  =  fp[(j  +
1)/1000]], {j, i,
```

```
      i + 249}];

    res1 = Integrate[fp[z], {z, i/1000, (i + 250)/1000}];

    pans += res1, {i, 1, 1751, 250}];

  cresult[[nvalue]] = pans;

  lovalue[[nvalue]] = lo[[nvalue, 100]];

  expCoeff[[nvalue]] =

   Divide[cresult[[nvalue]],  lovalue[[nvalue]]],  {nvalue,
1, 41, 40}];

finaltable = {nval, cresult, lovalue, expCoeff}; Grid[

 Prepend[Transpose[finaltable],  {"'n'",  "Computational",
"lo value",

   "Expansion Coefficient"}], Frame -> All]
```

Comment: Synthesis (3.4) of the analyzed signal is performed here. The synthesized signal in 'newexFunction[]' is used for both reconstruction and extrapolation of the signal.

```
newexFunction[time_] := (sum = 0;

  Do[If[MantissaExponent[expCoeff[[nvalue]]][[2]] <= 90,
```

```
    yplus = ksi[[nvalue, Range[1, 2000]]];

    yminus =

     If[OddQ[nvalue],

      Reverse[yplus[[2 ;; 2000]]], -1*Reverse[yplus[[2 ;;
2000]]]];

    yfull = Join[yminus, yplus];

    sum += (expCoeff[[nvalue]]*

       yfull[[Rationalize[time,    0]*1000    +    2000]])],
{nvalue, 1, 101}];

   sum)
```

Comment: This code plots the resulting extrapolated signal.

```
ListLinePlot[Table[{t,   newexFunction[t]},   {t,   -16/10,
16/10, 1/1000}],

  PlotRange -> All, PlotStyle -> Red, Frame -> True,

 GridLines -> Automatic, GridLinesStyle -> Directive[Gray,
Dashed]]
```

A.2 Figures of Experimental Signals

This section contains the plots for the experimental real signals used for extrapolation using our algorithm.

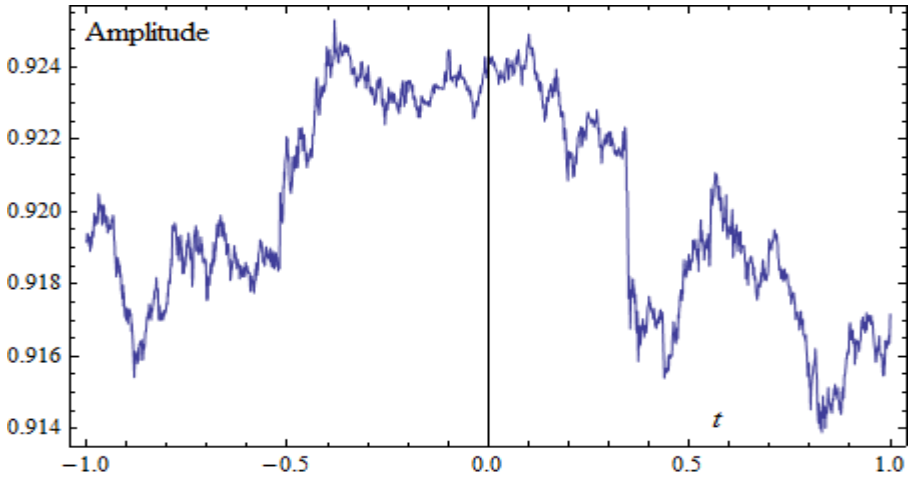

Figure A.1: Forex signal of AUDUSD currency pair (minute-by-minute) observed in 2013

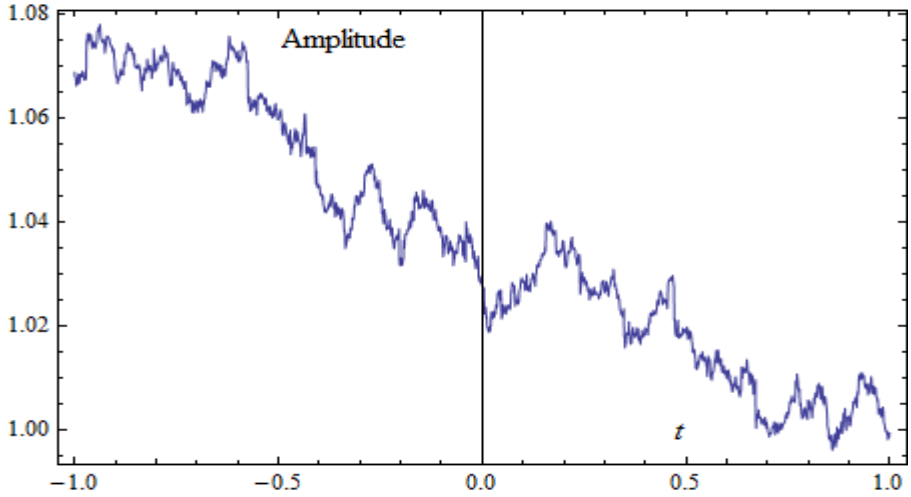

Figure A.2: Forex signal of AUDCAD currency pair (hourly) observed from 2012 to 2013

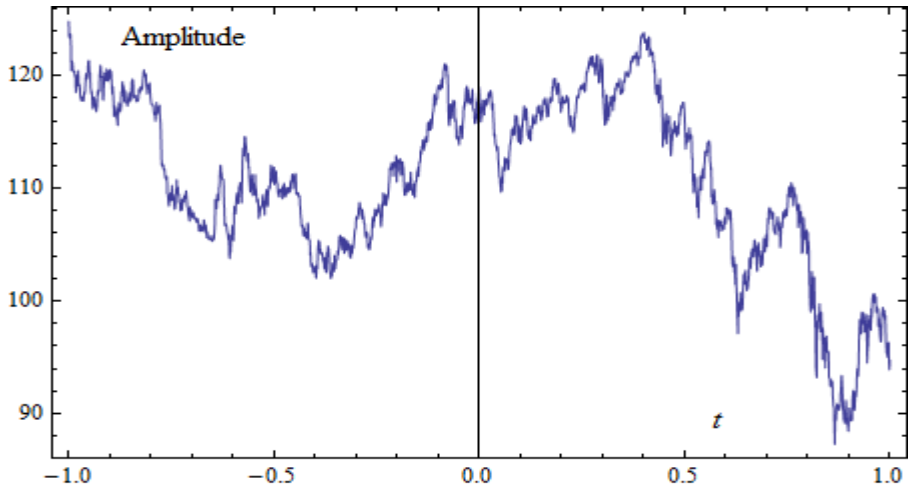

Figure A.3: Forex signal of USDJPY currency pair (daily) observed from 2000 to 2014

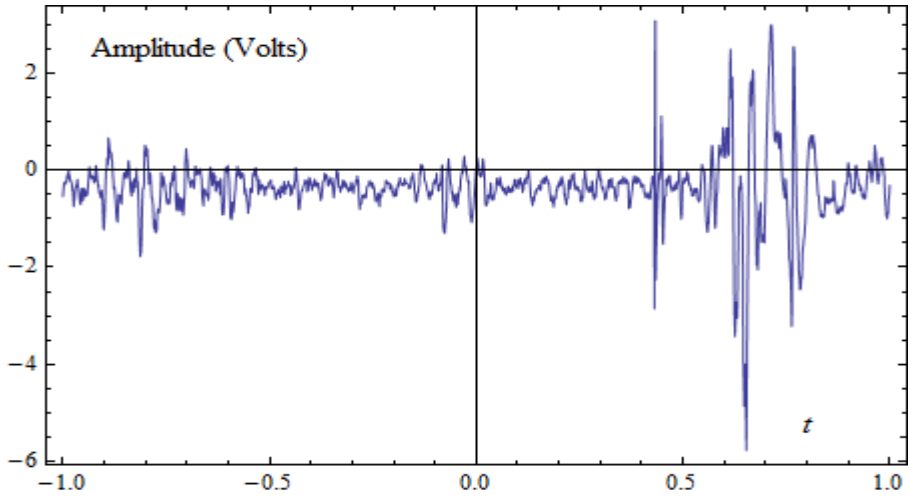

Figure A.4: Compound nerve action potential (CNAP) signal obtained from Gonzalez-Cueto's study [36]

Bibliography

[1] M. Abramowitz, I. Stegun, Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, Addison–Wesley, New York, 1972.

[2] D. Slepian, H.O. Pollack, "Prolate spheroidal wave functions, Fourier analysis, and uncertainty-I", Bell System Technical Journal, Vol 40, 43-63, 1961.

[3] D. Slepian, Prolate spheroidal wave functions, Fourier analysis and uncertainty—IV: extensions to many dimensions; generalized prolate spheroidal functions, Bell Syst. Techn. J. 43 (1962) 3009–3057.

[4] D. Slepian, Some asymptotic expansions for prolate spheroidal wave functions, J. Math. Phys. 44 (1965) 99–140.

[5] D. Slepian, Some comments on Fourier analysis, uncertainty and modeling, SIAM Rev. 25 (1983) 379_393.

[6] K. Khare, N. George, "Sampling theory approach to prolate spheroidal wave functions", J. Phys. A: Math. Gen. 36 (2003) 10011_10021.

[7] P. Kirby, "Calculation of spheroidal wave functions", Comput. Phys. Comm. 175 (7) (2006) 465_472.

[8] I.C. Moore, M. Cada, “Prolate spheroidal wave functions, an introduction to the Slepian series and its properties”, Appl. Comput. Harmon. Anal. 16 (2004) 208-230.

[9] H. Xiao, V. Rokhlin, N. Yarvin, “Prolate spheroidal wave functions, quadrature and interpolation”, Inverse Problems, 17: 805_838, 2001.

[10] S. Senay, L.F. Chaparro, A. Akan, Sampling and reconstruction of non-bandlimited signals using Slepian functions, in: EUSIPCO 2008, Lousanne, Switzerland, August 25–29, 2008.

[11] S. Senay, L.F. Chaparro, L. Durak, Reconstruction of non-uniformly sampled time-limited signals using prolate spheroidal wave functions, Signal Processing 89 (December) (2009) 2585–2595.

[12] S. Senay, J. Oh, L.F. Chaparro, Regularized signal reconstruction for level-crossing sampling using Slepian functions, Signal Processing 92 (2012) 1157–1165.

[13] M. Moshinsky and C. Quesne, “Linear canonical transformations and their unitary representations,” J. Math. Phys., vol. 12, pp. 1772–1783, Aug. 1971.

[14] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, The Fractional Fourier Transform With Applications in Optics and Signal Processing. New York: Wiley, 2000.

[15] H. Zhao, R.Y.Wang, D.P.Song, D.P.Wu, An extrapolation algorithm for M-bandlimited signals, IEEE Signal Processing Letters 18 (12) (2011) 745–748.

[16] H. Zhao, et al., Extrapolation of discrete bandlimited signals in linear canonical transform domain, Signal Processing, vol. 94, pp. 212-218, January 2014.

[17] J. Shi, X. Sha, Q. Zhang, N. Zhang, Extrapolation of bandlimited signals in linear canonical transform domain, IEEE Transactions on Signal Processing, Vol. 60, No. 3, March 2012.

[18] H. Zhao, Q. W. Ran, J. Ma, and L. Y. Tan, "Generalized prolate spheroidal wave functions associated with linear canonical transform", IEEE Trans. Signal Process., vol. 58, no. 6, pp. 3032–3041, June 2010.

[19] R. Gerchberg, "Super-resolution through error energy reduction," Opt. Acta., vol. 12, no. 9, pp. 709–720, Sept. 1974.

[20] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," IEEE Trans. Circuit Syst., vol. CAS-22, no. 9, pp. 735–742, Sep. 1975.

[21] L. Gosse, "Effective band-limited extrapolation relying on Slepian series and ℓ^1 regularization", Computers and Mathematics with Applications 60 (2010) 1259-1279.

[22] E.J. Candès, Compressive sampling, in: Proceedings of the International Congress of Mathematicians, Madrid, Spain, 2006.

[23] E.J. Candès, The restricted isometry property and its implications for compressed sensing, C. R. Acad. Sci. Paris, Ser. I 346 (2008) 589-592.

[24] M. Cada, Private communication, Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS, B3H 4R2, Canada, 2012.

[25] C. Flammer, Spheroidal Wave Functions, Stanford Univ. Press, Stanford, CA, 1956.

[26] V. Rokhlin, H. Xiao, Approximate formulae for certain prolate spheroidal wave functions valid for large values of both order and band-limit, Appl. Comput. Harmon. Anal. 22 (2007) 105–123.

[27] A. Kaw, E. Kalu, and D. Nguyen, Numerical Methods With Applications: Abridged, Lulu, 2nd ed., 2011.

[28] A. Devasia, M. Cada, “Extrapolation of bandlimited signals using Slepian functions”, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2013, WCECS 2013, 23-25 October, 2013, San Francisco, USA, pp492-497.

[29] Kauppinen, K. Roth, “Audio signal extrapolation-theory and applications”, in proc. (DAFx-02), Hamburg, Germany, September 26-28, 2002.

[30] A. Kaup, K. Meisinger, T. Aach, “Frequency selective signal extrapolation with applications to error concealment in image communication”, Int. J. Electron. Commun., Vol 59, 147-156, 2005.

[31] A.V. Oppenheim & R.W. Schafer w/ J.R. Buck, "Discrete-time Signal Processing", 3rd Ed, Prentice Hall, 2010.

[32] S.W. Smith, "The Scientist & Engineer's Guide to Digital Signal Processing", 2nd Ed, California Technical Publishing, 1999.

[33] I.C. Moore, *An introduction to the Slepian series and its applications*, Master's Thesis, Dalhousie University, Halifax, Nova Scotia, December 2003.

[34] A. Devasia, M. Cada, "Bandlimited Signal Extrapolation Using Prolate Spheroidal Wave Functions", IAENG International Journal of Computer Science, Volume 40 Issue 4, Pages 291-300.

[35] A. Devasia, M. Cada, "High Precision Numerical Implementation of Bandlimited Signal Extrapolation Using Prolate Spheroidal Wave Functions", Transactions on Engineering Technologies - Special Issue of the World Congress on Engineering and Computer Science 2013: 387-402, 2014.

[36] Gonzalez-Cueto, Jose A., "*An investigation of non-invasive techniques for the estimation of conduction velocity distributions in skeletal muscles and nerve bundles*", Ph.D. Thesis Report, Department of Electrical and Computer Engineering, University of New Brunswick, June 2001.

[37] Wolfram Language & System Documentation Center (2015). Retrieved from *http://reference.wolfram.com/language/*.

[38] Soman, Sagar, "*Reduction in Intersymbol Interference using Linear Prolate Functions*", Master's Thesis, Dalhousie University, Halifax, Nova Scotia, 2015.

[39] H. Xiao, Prolate spheroidal wave functions, quadrature, interpolation, and asymptotic formulae, Ph.D. Thesis, Yale University, 2001.