T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

STEADY STATE ANALYSIS OF NONLINEAR CIRCUITS WITH HARMONIC BALANCE METHOD AND APPLICATIONS

ELİF BETÜL ŞEN ÖZEN
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF ELECTRONIC ENGINEERING

GEBZE
2018

T.R.

GEBZE TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

STEADY STATE ANALYSIS OF NONLINEAR CIRCUITS WITH HARMONIC BALANCE METHOD AND APPLICATIONS

ELİF BETÜL ŞEN ÖZEN

A THESIS SUBMITTED FOR THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF ELECTRONIC ENGINEERING

THESIS SUPERVISOR

ASSIST. PROF. DR. ÖNDER ŞUVAK

GEBZE

2018

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOĞRUSAL OLMAYAN DEVRELERİN HARMONİK DENGE METODU İLE KARARLI DURUM ANALİZİ VE UYGULAMALARI

ELİF BETÜL ŞEN ÖZEN
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
YRD. DOÇ. DR.ÖNDER ŞUVAK

GEBZE
2018

GEBZE TEKNİK ÜNİVERSİTESİ	YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 20/12/2017 tarih ve 2017/63 sayılı kararıyla oluşturulan jüri tarafından 27/12/2017 tarihinde tez savunma sınavı yapılan Elif Betül Şen ÖZEN'in tez çalışması Elektronik Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI)	:Yrd. Doç. Dr. Önder ŞUVAK
ÜYE	:Doç. Dr. Koray KAYABOL
ÜYE	:Doç. Dr. Ender Mete EKŞİOĞLU

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve ...../..... sayılı kararı.

İMZA/MÜHÜR

.....

Gebze Teknik Üniversitesi
Fen Bilimleri Enstitüsü Müdürü

SUMMARY

The requirement of high performance simulators is increasing with growing circuit sizes and complex electronic structures. Depending on this situation, the demand for simulation algorithms to analyze circuits is increasing. The periodic steady state analysis of circuits, which is a prerequisite for small signal modeling, is an indispensable step in circuit design. Through transient solution, obtaining periodic steady state takes hundreds of periods which both increases computational cost and causes loss of time. Towards handling with this problem, algorithms have been developed to analyze periodic steady state of circuits. Among these algorithms, the most common used methods in the related literature are the shooting method and harmonic balance method. In this study, periodic steady state analysis was performed by using harmonic balance method. As an iterative method for solving nonlinear harmonic balance equations, the Newton's method was utilized. In addition to this, since the initial values of the Newton's method are highly crucial to work properly and to achieve the desired solution, a new software was designed to calculate the Fourier coefficients. The considered algorithms in this thesis have been implemented with object-oriented programming in MATLAB environment by taking into consideration the rules of an efficient software design. Periodic steady state analysis of various nonlinear systems was substantiated with the developed software in the scope of this thesis and obtained results were presented.

Key Words: periodic steady state, harmonic balance method, Fourier coefficients, circuit simulation.

ÖZET

Her geçen gün büyüyen devre boyutları ve karmaşıklaşan elektronik yapılar yüksek performanslı simülatörlere olan ihtiyacı artırmaktadır. Bu duruma bağlı olarak devreleri analiz edecek simülasyon algoritmalarına olan talep artmaktadır. Devrelerin periyodik kararlı durum analizi küçük sinyal modellemesinin yapılabilmesi için ön şart olup devre tasarımının vazgeçilmez bir aşamasıdır. Geçici hal çözümü ile periyodik kararlı duruma gelinmesi beklendiği takdirde yüzlerce periyot geçmesini beklemek ve bu süre zarfında simulasyonu koşturmak hem hesaplama yükünü çok artırmakta hem de zaman israfına sebep olmaktadır. Bu sebeple devrelerin periyodik kararlı durum analizini gerçekleyen algoritmalar geliştirilmiştir. Bu algoritmalardan literatürde en yaygın kullanılanları shooting metodu ve harmonik balans metodudur. Bu çalışmada da harmonik balans metodu ile periyodik kararlı durum analizi yapılmıştır. Harmonik balans denklemlerini çözmek için iteratif yöntem olarak Newton metodu kullanılmıştır. Newton metoduna verilen giriş değerleri metodun düzgün çalışması ve çözüme ulaşmak için çok önemli olduğundan üzerinde çalışılan sinyallerin Fourier katsayılarını hesaplayacak bir yazılım tasarlanmıştır. Tüm bunlar MATLAB ortamında nesne yönelimli programlama ile gerçeklenmiş olup, efektif bir yazılım tasarımında uyulması gereken kurallar dikkate alınmıştır. Tez kapsamında geliştirilen yazılım ile çeşitli nonlineer sistemlerin periyodik kararlı durum analizi gerçeklenmiş ve ilgili sonuçlar sunulmuştur.

Anahtar Kelimeler: periyodik kararlı durum, harmonik denge metodu, Fourier katsayıları, devre simülasyonu.

ACKNOWLEDGEMENTS

My experience in graduate school has been both challenging and rewarding.I want to thank my supervisor Asst.Prof. Önder Şuvak for his unfailing support and encouragement through this journey. I would also like to thank the committee members Assoc. Prof. Ender Mete Ekşioğlu and Asst. Prof. Koray Kayabol, for evaluating this thesis and for their suggestions.

I would like to express my deep gratitute to my parents Ayşe Şen and Orhan Şen for supporting me throughout my life with their hearts and souls and believing in my abilities. Without their encouragement I would never be able to discover myself to pursue an academic career.

I would also like to thank my dear husband Ahkemin for the all love and support he gave me throughout this process. Lastly, I would like to thank my little sunshine, my son Muhsin Aziz for being such a cheerful kid and an endless source of energy for me.

TABLE of CONTENTS

.

LIST of ABBREVIATIONS and ACRONYMS

Abbreviations and Acronyms		**Explanations**
PSS	:	Periodic Steady State
CTFT	:	Continuous Time Fourier Transform
DFT	:	Discrete Fourier Transform
FFT	:	Fast Fourier Transform
MNA	:	Modal Nodal Analysis
IDFT	:	Inverse Discrete Fourier Transform
IFFT	:	Inverse Fast Fourier Transform
DAE	:	Differential Algebraic Equation
KCL	:	Kirchoff Current Law
PWL	:	Piecewise Linear

LIST of FIGURES

LIST of TABLES

1. INTRODUCTION

It is very important to be able to simulate very large circuits with developing electronic materials and growing circuit sizes. Because of this reason computer-aided design is an important subject in literature for decades. Computer-aided design of nonlinear systems has several problems. An important one of them is the steady state analysis of nonlinear circuits with periodic responses.

For nonlinear circuits which governed by time constants that are very much larger than the period of the input signals[1], it takes hundreds of periods for the transients to die where the periodic steady state(PSS) occurs. To compute these hundreds of periods by integration increase the computational cost for computers and spends time, which is the most precious existence of mankind. therefore, this problem has been tried to be solved for decades and algorithms have been developed to find PSS. One of these algorithms is harmonic balance method that has become a important topic in literature.

Authors of [2] propose a time domain Newton Method which converge very fast to the PSS but this method works for only forced circuits and is not suitable for oscillators.

A method is proposed to make the harmonic balance method reliable even for large systems[1]. This reliability is obtained by reducing the number of variables to be optimized based on the fact that considerable part of the network is usually linear.

Kundert and Vincentelli draw attention that shooting method is an iterative method which applied in time domain on one period but works well only with non-strongly nonlinear forced circuits which it is known that it has a periodic solution. They also remark that time domain simulators need a lot of time to reach PSS so for this reason it is necessary to use spectral methods. They propose a method to accelerate PSS analysis and with this way it is possible to analyze large circuits. This method also permits to use more harmonics and this increases the accuracy of solution dramatically[3].

Authors of [4] remark that growing circuit size dramatically increase the computational complexity but matrix-implicit Krylov-subspace iterative methods change this situation. The preconditioned iterative methods use harmonic balance, time domain shooting methods to find PSS.

Zhang and El-Moselhy propose a uncertainty quantification method for PSS analysis and they point out that proposed stochastic testing formulation is superior than Monte Carlo methods. They used harmonic balance method with Gaussian parameters to analyze forced circuits and used stochastic Galerkin with non-Gaussian parameters to simulate oscillators.[5]

It is important to understand periodic flows of nonlinear dynamical systems because it enables to analyze dynamical behaviors of dynamical systems. The authors, in [6], propose a generalized harmonic balance method to find periodic motions of nonlinear dynamic systems.

Wu and Roychowdhury point out that, designers would like to know each component's contribution to distortion to optimize distortion performance of a circuit. With the purpose of finding per-element distortion contribution a metric which, allows nonlinear elements to be linearized separately, proposed[7].

Authors of [8] propound an harmonic balance approach which use wavelets as basis instead of Fourier series. They draw attention that this approach is compatible for large scale simulations. Because, sparsity of wavelet Jacobian reduces the computational complexity. Same authors proposed a steady state analysis method which require nonlinear component's dependency to frequency in the system representation[9]. If this system trying to be solved in wavelet basis, there is a possibility to make a big matrix sparse with a threshold. With this approach computational complexity is reduced.

Wavelet transform which is superior to Fourier transform due to fact that it can analyze for different regions on the time axis. As a consequence of this situation Jacobian matrix of wavelets is sparser and it is desirable because of reducing computational complexity. Based on this information, a method to reduce the simulation cost for the analysis of nonlinear circuits with widely separated time scales is presented in [10]. Last, it must be remarked that wavelet has dramatically advantages on highly nonlinear circuits under multitone excitations.

Authors, in [11], introduces a domain partitioning technique which allow to represent a big nonlinear system with connect of small nonlinear blocks. Instead of a common spectrum to be used for an entire system, solutions are implemented using a special constant spectrum for each small nonlinear part. With this approach the solution is found faster. However, there are subjects which requires attention in

design, such as nonlinear interactions between interconnected blocks and linear and nonlinear effects of reflections at the connection ports. A cautious approach to these issues is suggested in the proposed method.

In this study, it is aimed to achieve an effective numeric approach to solve harmonic balance equation. For this purpose Newton's Method is used as an iterative method with quadratic convergence which makes it suitable for large data sets. In order to calculating inputs of harmonic balance equation correctly, detailed and deep software has been implemented to calculate the Fourier coefficients.

The rest of this thesis is organized as follows. In Chapter 2, designed Fourier Coefficients calculation tool is introduced and the numeric approach used in this tool is explained in detail. In Chapter 3, harmonic balance method is expounded. In Chapter 4, it is explained what is paid attention to construct an effective software. In Chapter 5, performance of our software is tested with various nonlinear systems and represented as results. Finally, in Chapter 6, the conclusions are drawn.

2. A TOOL FOR CALCULATION OF FOURIER SERIES COEFFICENTS

The main idea of Fourier Transform is to express a function with a linear combination of trigonometric functions. Orthogonality property of sine function allows doing such an expression. Formulas of Continuous-time Fourier Transform (CTFT) and Inverse Fourier Transform are below.

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \exp(-j\omega t)\, dt \tag{2.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(\omega) \exp(j\omega x)\, d\omega \tag{2.2}$$

Formulas of Discrete-time Fourier Transform (DFT) and Inverse Fourier Transform are below.

$$X(n) = \sum_{k=0}^{N-1} x(k) \exp(-j2\pi nk/N) \tag{2.3}$$

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) \exp(j2\pi nk/N) \tag{2.4}$$

In this study a calculation tool designed for evaluating Fourier Series Coefficients of a signal. The purpose of designing this tool is to calculate Fourier Series Coefficients efficiently for input signals with different properties. This tool can decide to use CTFT or DFT according to two different property of signal that can be known by two flags in code. Considering that the user knows properties of the input signal, these flags are wanted from user as input parameters. The first flag checks if the signal is uniformly sampled in time domain. The second flag checks if the sample number is enough to represent data. According to these flags, before calculating coefficients a preliminary preparation may be needed. This preparation is interpolation and this tool has three different options as interpolation types. This choice is left to the user and it is wanted from user as a string input parameter.

In case of uniformly sampled input signal, tool decides to use DFT. But due to the superiority of Fast Fourier Transform (FFT) to DFT, this faster algorithm is used instead of DFT. In this study MATLAB ‘fft’ function is used for this purpose. In case of sample number is enough to represent data, tool only apply the ‘fft’ command to input signal. If the sample number is not enough to represent data, tool increases the sample number by interpolation method that is choice of user. Thus the representation of signal is improved and then coefficients are estimated with ‘fft’ command.

In case of non-uniform sampling with interpolation method the piecewise polynomial form of signal is generated. Then if the sample number is enough for these pieces, the integration in equation (2.1) is evaluated. By this way the coefficients are acquired. If the sample number is not enough, the sample number is increased with interpolation method. Then the coefficients are acquired with integration in equation (2.1).

Acquired complex coefficients are shifted and stored in the properties of belonging class. Thus they are more suitable for operations in frequency domain.

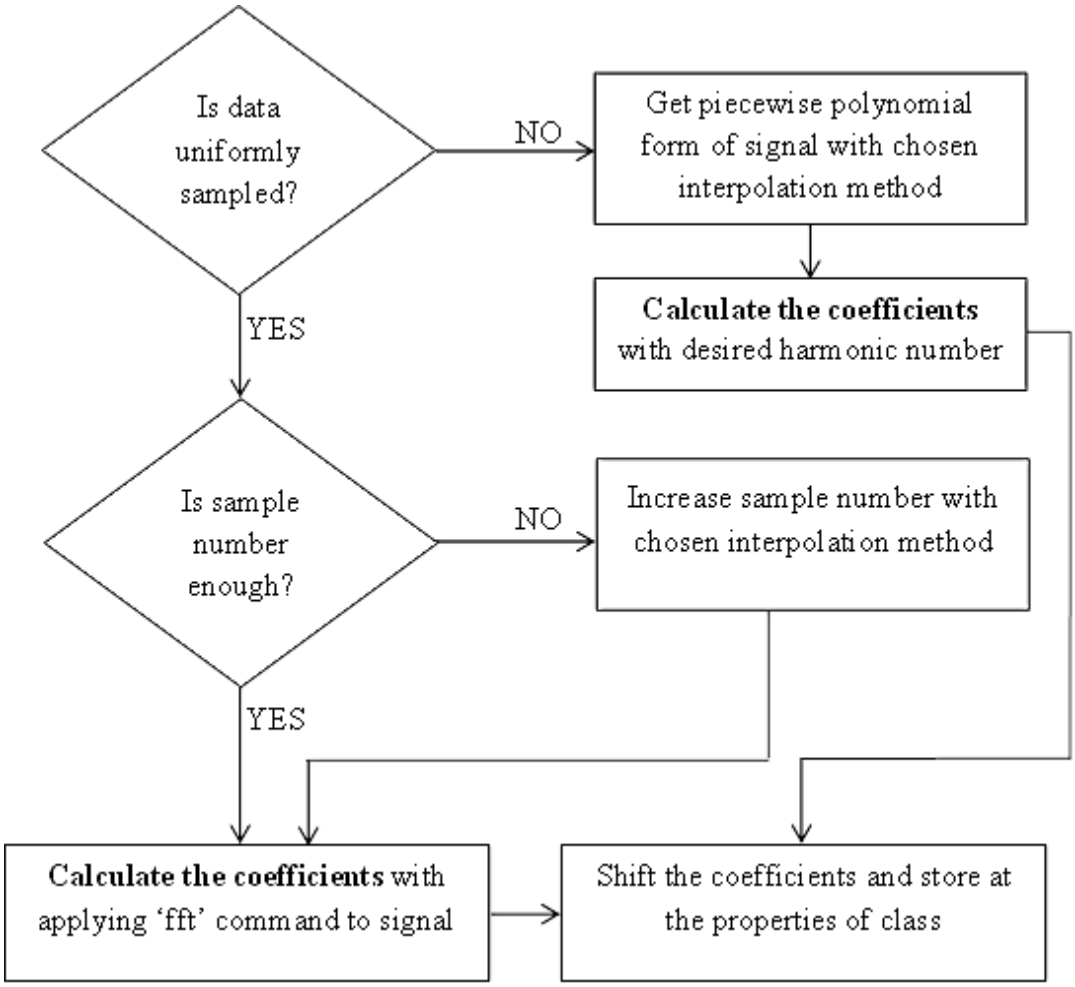

Figure 2.1: Flow chart of Fourier Series Coefficient calculation tool algorithm.

In this section, mathematical background, which is used in the design of this tool, will be explained in subheadings as interpolation methods, Fourier transform and operations which performed in frequency domain.

2.1 Interpolation Methods

Interpolation is a method in numerical analysis field. This method estimates unknown values that are in a range of known values. In this study the purpose of using interpolation is to make non-uniform sampled data uniformly sampled, to increase sample number if it is not enough to represent data and to get piecewise polynomial form of signal.

2.1.1 Stair Step Interpolation

In this interpolation type until a new data shows up the output of the function remains stable at the last output. This interpolated signal is independent from time and can be represented with a constant number as shown in equation (2.5).

$$f_s(x) = a \tag{2.5}$$

$$f(x) = f(x_k) \; if \; x_k < x < x_{k+1} \tag{2.6}$$

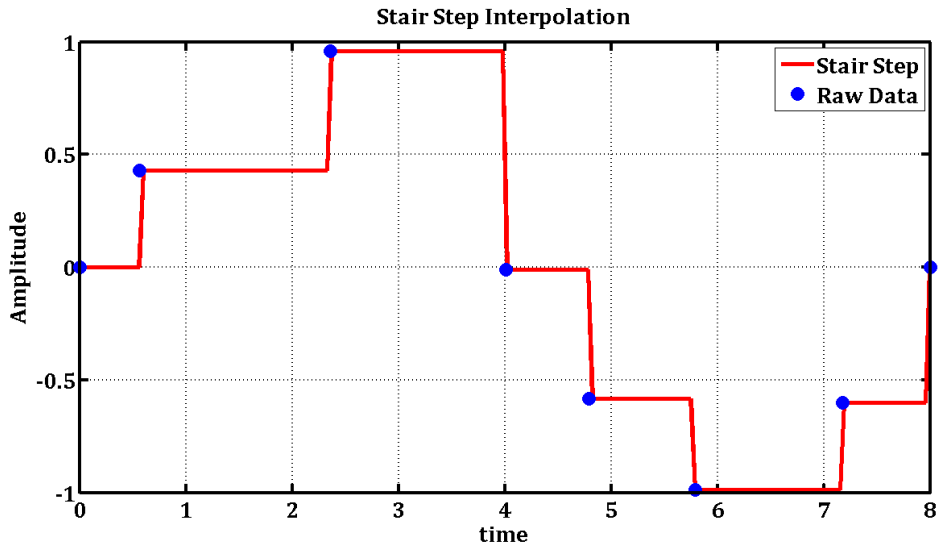

Figure 2.2: Stair step interpolation.

2.1.2 Linear Interpolation

In this interpolation type, the shortest distance between two different points is expressed by a first order equation as shown in equation (2.7). a and b coefficients of this formula are estimated for each range in x domain. This linear equation can be expressed in a way as seen in equation (2.8) by easily calculating the slope between two points.

$$f_l(x) = ax + b \tag{2.7}$$

$$f(x) = f(x_k) + [f(x_{k+1}) - f(x_k)]\frac{x - x_k}{x_{k+1} - x_k} \tag{2.8}$$

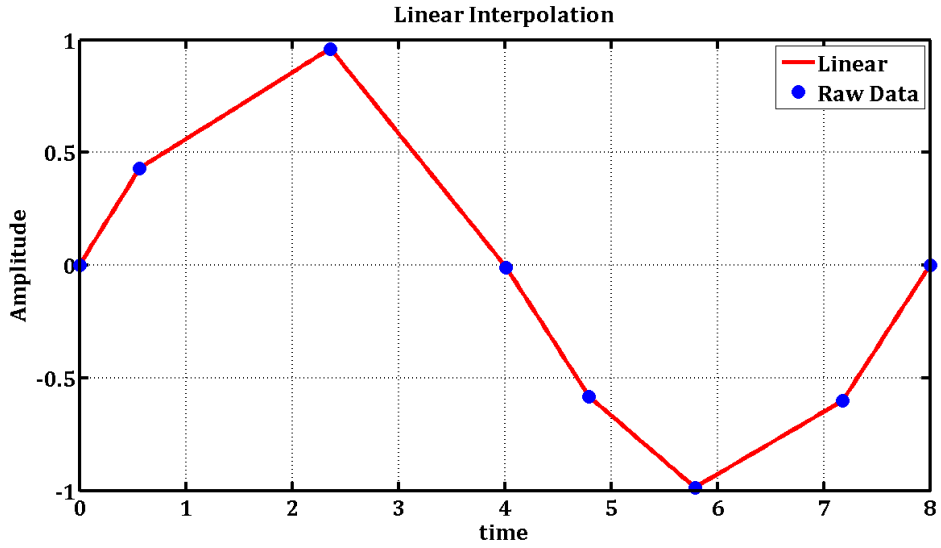

Figure 2.3: Linear interpolation.

2.1.3 Cubic Spline Interpolation

In this section, cubic spline interpolation type is used which constitutes a third order equation as shown in equation (2.9) for each interval.

$$f_{cs}(x) = ax^3 + bx^2 + cx + d \tag{2.9}$$

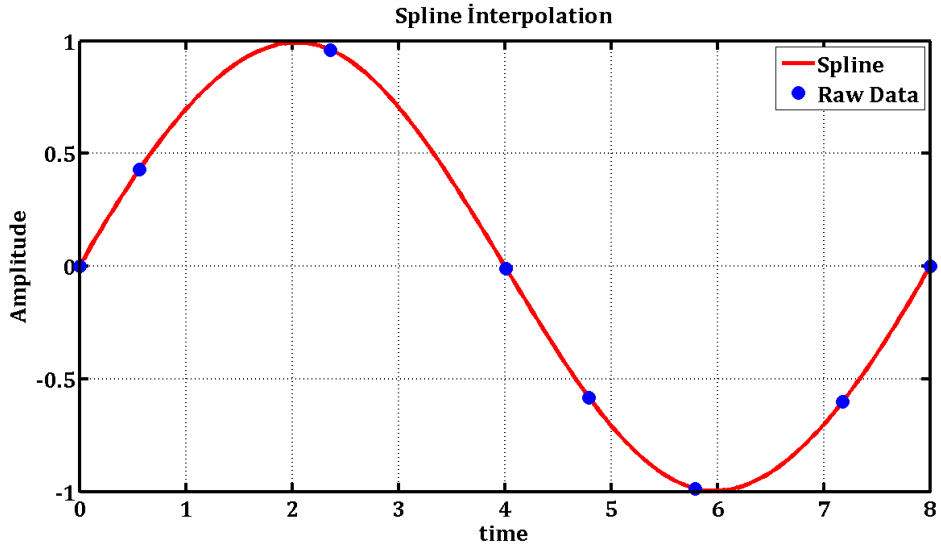

Figure 2.4:Cubic Spline Interpolation.

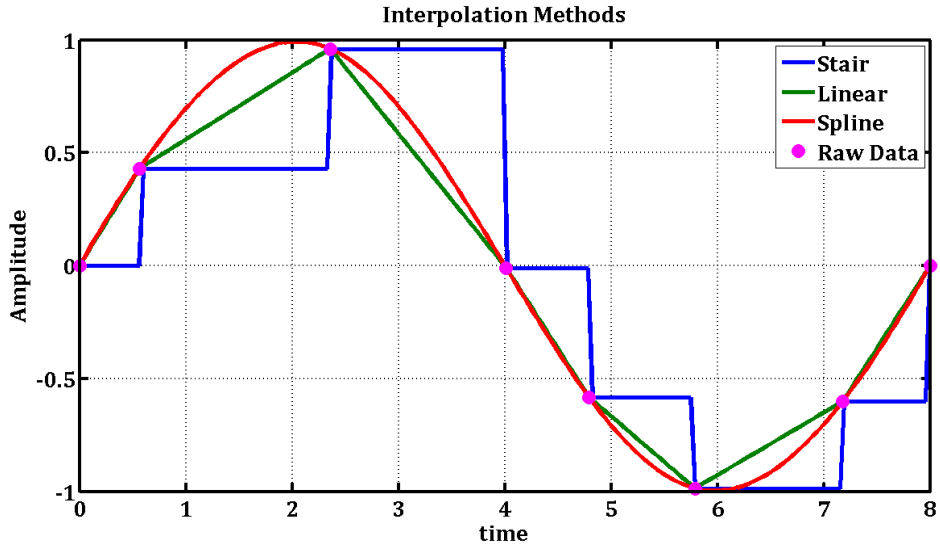

Figure 2.5: All interpolation methods altogether.

Only eight samples are selected from one period of a sinus signal via non-uniform sampling. Then the signal is constructed from these samples with different interpolation methods. The performances of the methods are obvious as seen in Figure 2.5.

2.2 Fourier Transform

Transfer of the signals to the frequency domain by the Fourier transform was performed in two different ways in this study. These two different methods will be explained in this section.

2.2.1 FFT Superiority to DFT

In this part of tool, DFT computation of the uniform sampled signals in time domain was implemented with the FFT. Undoubted one of the most important algorithms in signal processing and data analysis is FFT. In this section, it is tried to understand via what kind of approach FFT can be implemented as a superior algorithm to DFT.

For an input vector of length N, classic DFT scales as $O(N^2)$ while the FFT algorithm scales as O(NlogN). Cooley and Tuckey outlined this superior computational trick in their classic paper.[12]

"*...one must multiply an N-vector by an N X N matrix which can be factored into m sparse matrices, where m is proportional to log N. This results in a procedure requiring a number of operations proportional to N log N rather than* N^2*.* "

First the value of $X(N+k)$ in equation (2.10) must be asked to understand how Cooley-Tukey used symmetries in the DFT. The calculated value of $X(N+k)$ is $X(k)$ as seen in equation (2.12). This result can be generalized as in equation (2.13) and shows the symmetry property of DFT.

$$X(N+k) = \sum_{k=0}^{N-1} x(n) \exp\left(-\frac{j2\pi n(N+k)}{N}\right) \tag{2.10}$$

$$= \sum_{k=0}^{N-1} x(n) \exp(j2\pi n) \exp\left(-\frac{j2\pi kn)}{N}\right) \tag{2.11}$$

$$= \sum_{k=0}^{N-1} x(n) \exp\left(-\frac{j2\pi kn}{N}\right) = X(k) \tag{2.12}$$

$$X(i * N + k) = X(k) \tag{2.13}$$

Cooley and Tukey exploit this symmetry property and showed that dividing the DFT computation into smaller parts is possible. In equation (2.15) the DFT equation is divided into two parts.

$$X(k) = \sum_{k=0}^{N-1} x(n) \exp\left(-\frac{j2\pi kn}{N}\right) \tag{2.14}$$

$$= \sum_{m=0}^{N/2-1} x(2m) \exp\left(-\frac{j2\pi k2m}{N}\right) + \sum_{m=0}^{N/2-1} x(2m+1) \exp\left(-\frac{j2\pi k(2m+1)}{N}\right) \tag{2.15}$$

$$= \sum_{m=0}^{N/2-1} x(2m) \exp\left(-\frac{j2\pi k2m}{N}\right) + \sum_{m=0}^{N/2-1} x(2m+1) \exp\left(-\frac{j2\pi k(2m+1)}{N}\right) \tag{2.16}$$

There is a thing to consider that each smaller part in equation (2.16) looks similar to first single DFT expression. Then there is no reason to stop exploiting this symmetry property. As long as the element number of summation symbol is even for these smaller parts of a single DFT, dividing process can be applied for each small summation. For each dividing the computational cost is halved. Basically this recursive approach scales to O(NlogN) in the asymptotic limit with this dividing approach.

2.2.2 A Numerical Approximation for CTFT

In this section, an arbitrary signal $x(t)$ denotes a periodic signal with period T. This signal can be expressed as sum of sines and cosines with coefficients as shown in equation (2.17). a_n and b_n are defined in the equations (2.19) and (2.20) respectively. These equations simply use the orthogonality property of these trigonometric functions. a_n can be represented as cosine coefficients and b_n can be represented as sine coefficients.

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{N} \left[a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right) \right] \tag{2.17}$$

$$= \sum_{n=-N}^{N} \left[c_n \exp\left(j \frac{2\pi nt}{T}\right) \right] \tag{2.18}$$

$$a_n = \frac{2}{T} \int_{t_i}^{t_i+T} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt \tag{2.19}$$

$$b_n = \frac{2}{T} \int_{t_i}^{t_i+T} x(t) \sin\left(\frac{2\pi nt}{T}\right) dt \tag{2.20}$$

By exploiting Taylor expansions of exp(±ix), cos(x) and sin(x) where x is equal to zero, i.e. x=0, the following equations (2.21) and (2.22) can be proven.

$$\exp(jx) = \cos(x) + j\sin(x) \tag{2.21}$$

$$\exp(-jx) = \cos(x) - j\sin(x) \tag{2.22}$$

c_n can be presented as complex coefficients and is defined as in equation (2.23). We should note that this equation is highly similar with the aforementioned equation (2.1). From the formula in equation (2.22) c_n can be derived as in equation (2.24). Then as seen in equation (2.25) complex coefficients can be expressed with sine and cosine coefficients. $c^*_{|n|}$ term equals complex conjugate of c_n where n is positive.

$$c_n = \frac{1}{T} \int_{t_i}^{t_i+T} x(t) \exp\left(-\mathrm{j} \frac{2\pi nt}{T}\right) dt \tag{2.23}$$

$$= \begin{cases} \frac{1}{T}\int_{t_i}^{t_i+T} x(t)\left[\cos\left(\frac{2\pi nt}{T}\right) - j\,\sin(\frac{2\pi nt}{T})\right]dt & (for\ n > 0) \\ \frac{1}{T}\int_{t_i}^{t_i+T} x(t)\left[\cos\left(\frac{2\pi 0t}{T}\right) - j\,\sin(\frac{2\pi 0t}{T})\right]dt & (for\ n = 0) \\ \frac{1}{T}\int_{t_i}^{t_i+T} x(t)\left[\cos\left(\frac{2\pi nt}{T}\right) + j\,\sin(\frac{2\pi nt}{T})\right]dt & (for\ n < 0) \end{cases} \quad (2.24)$$

$$c_n = \begin{cases} \frac{1}{2}(a_n - jb_n) & (for\ n > 0) \\ \frac{1}{2}a_0 & (for\ n = 0) \\ c^*_{|n|} & (for\ n < 0) \end{cases} \quad (2.25)$$

Actually these coefficients are the frequency domain presentation of our signal and somehow they need to be found. For this study the form of $x(t)$ is known as polynomial functions. The degree of these functions is '1' for linear interpolation and '3' for spline interpolation. For a static structure of equation (2.23) a numeric approximation can be developed for this integration.

Because of the simplicity of its equation, polynomial function of linear interpolation is selected to describe this approach. Let write the equation (2.23) with a linear equation $x(t)$ as in (2.7).

$$c_{n,lin} = \frac{1}{T}\int_{t_i}^{t_i+T} (\mathrm{ax} + \mathrm{b})\exp\left(-\mathrm{j}\frac{2\pi nt}{T}\right)dt \quad (2.26)$$

$$= \frac{1}{T}\int_{t_i}^{t_i+T} \left[\mathrm{ax} * \exp\left(-\mathrm{j}\frac{2\pi nt}{T}\right) + \mathrm{b} * \exp\left(-\mathrm{j}\frac{2\pi nt}{T}\right)\right]dt \quad (2.27)$$

$$= a\left[\frac{1}{T}\int_{t_i}^{t_i+T} \mathrm{x} * \exp\left(-\mathrm{j}\frac{2\pi nt}{T}\right)dt\right] + b\left[\frac{1}{T}\int_{t_i}^{t_i+T} \exp\left(-\mathrm{j}\frac{2\pi nt}{T}\right)dt\right] \quad (2.28)$$

The equation mentioned above can be derived into a simple form as in equation (2.28).

If the $G(x)$ is calculated as in equation (2.29), then the integration in equation (2.30) is known from fundamental theorem of calculus. Based on this equation for each interval of the interpolation method, the integration can be calculated easily and numerically if the $G(x)$ is already known.

$$G(x) = \int g(x)dx \tag{2.29}$$

$$\int_{x_n}^{x_{n+1}} g(x)dx = G(x_{n+1}) - G(x_n) \tag{2.30}$$

For linear interpolation the equation (2.28) has two integration in it, where a and b constant multipliers of these integrations. If these integrations are calculated, then they can be used for numerically computing for each interval of interpolation method with known boundary values and constant multipliers.

For a numeric computation purpose of this integration, it is known that all interpolation methods have a static integration expression. If these integration expressions are solved at first, then they can be used to calculate of each interval of interpolation method, which have different constant multipliers, via a numerical method. Sum of integrations belong to these intervals gives the total integration which mentioned in equation (2.26).

2.3 Frequency Domain Operation

The functions of the time domain can also be transferred to the frequency domain as the data can be transformed into the frequency domain. This can often be a good approach to reduce computational complexity. Convolution theorem, undoubtedly which has the most common use in this field, states that convolution in time domain corresponds to multiplication in frequency domain. In this study time derivative, time integral and time shift operations described below, which are meaningful for our project, have been implemented in this tool.

As previously described in this chapter, the Fourier coefficients are calculated corresponding to a main frequency and its multiples. Then they are shifted and with this way the coefficients are ready for frequency domain operations. This is an important detail and will be covered in the following sections.

2.3.1 Time Derivation

Equivalent in the frequency domain of the time domain derivation operation is shown in equation (2.31).

$$\mathcal{F}\left[\frac{d}{dt}x(t)\right] = j\omega X(j\omega) \tag{2.31}$$

Proof of equation (2.31) can easily be derived in three lines as in equation (2.34).

$$\frac{d}{dt}x(t) = \frac{d}{dt}\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)\exp(j\omega t)\,d\omega\right] = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)\frac{d}{dt}\exp(j\omega t)\,d\omega \tag{2.32}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}\big(j\omega X(j\omega)\big)\exp(j\omega t)\,d\omega = \mathcal{F}^{-1}\big(j\omega X(j\omega)\big) \tag{2.33}$$

$$\mathcal{F}\left[\frac{d}{dt}x(t)\right] = j\omega X(j\omega) \tag{2.34}$$

Equation (2.34) can be revised as in equation (2.35) for n which is a positive integer.

$$\mathcal{F}\left[\frac{d}{dt}x(t)\right] = j\omega n X(j\omega n) \tag{2.35}$$

For the Fourier Coefficients that are already computed, shifted and stored, the derivation is as simple as multiplying by $j\omega n$ where $\omega = 2\pi f$ and the f is the main frequency. ωn is represents the n^{th} harmonic. For shifted coefficients with M harmonic this n multiplier is an integer sequence which is increasing one by one from –M to M. Thus, the derivative is obtained by 2M + 1 multiplication.

2.3.2 Time Integration

Equivalent in the frequency domain of the time domain integration operation is shown in equation (2.31).

$$\mathcal{F}\left[\int_{-\infty}^{t} x(\tau)d\tau\right] = \frac{1}{j\omega}X(j\omega) + \pi X(0)\delta(\omega) \tag{2.36}$$

Let suppose that Fourier Transform of unit step function in equation (2.37) is already known before starting to prove the expression in equation (2.36).

$$\mathcal{F}[u(t)] = \frac{1}{j\omega} + \pi\delta(\omega) \tag{2.37}$$

Then it must be noticed that the wanted integration expression in equation (2.36) is equal to the convolution of signal and unit step where * is the convolution operator. As mentioned in chapter 2.3, the convolution of two signals corresponds to the multiplication in the frequency domain. With this way equality (2.37) is proven in equation (2.40).

$$x(t) * u(t) = \int_{-\infty}^{\infty} x(\tau)u(t-\tau)d\tau = \int_{-\infty}^{t} x(\tau)d\tau \tag{2.38}$$

$$\mathcal{F}\left[\int_{-\infty}^{t} x(\tau)d\tau\right] = \mathcal{F}[x(t) * u(t)] = X(j\omega)\left[\frac{1}{j\omega} + \pi\delta(\omega)\right] \tag{2.39}$$

$$= \frac{1}{j\omega}X(j\omega) + \pi X(0)\delta(\omega) \tag{2.40}$$

For $\neq 0$, $\delta(\omega) = 0$, So only for DC component of Fourier Coefficients the Dirac function gives 1 as a multiplier. For harmonics this term can be omitted and the expression in equation (2.40) can be written as below.

$$\mathcal{F}\left[\int_{-\infty}^{t} x(\tau)d\tau\right] = \frac{1}{j\omega n}X(j\omega n) \tag{2.41}$$

As aforementioned $\omega = 2\pi f$ and f is the main frequency. For shifted coefficients with M harmonic this n multiplier is an integer sequence which is increasing one by one from –M to M. In the knowledge that the expression $\frac{1}{j\omega n}X(j\omega n)$ is not defined for $n = 0$, it can be said that the integration is obtained by 2M division for harmonics and a one multiplication for DC component X(0).

2.3.3 Time Shift

Equivalent in the frequency domain of the time shift operation is shown in equation (2.31).

$$\mathcal{F}[x(t \pm t_0)] = X(j\omega)\exp(\pm j\omega t_0) \tag{2.42}$$

In equation (2.43), Fourier Transform of a time shifted signal is presented. This equation turns into equation (2.45) with changing variable as in equation (2.44).

$$\mathcal{F}[x(t \pm t_0)] = \int_{-\infty}^{\infty} x(t \pm t_0)\exp(-j\omega t)\,dt \tag{2.43}$$

$$t' = t \pm t_0 \; and \; so \; t = t' \mp t_0 \tag{2.44}$$

$$= \int_{-\infty}^{\infty} x(t')\exp\left(-j\omega(t' \mp t_{0})\right)dt' = \exp(\pm j\omega t_0)\int_{-\infty}^{\infty} x(t')\exp(-j\omega t')\,dt' \tag{2.45}$$

$$= X(j\omega)\exp(\pm j\omega t_0) \tag{2.46}$$

It is proved that in equation (2.46), time shifting of a signal by an amount t0 is equal to multiplying by $\exp(j\omega t_0)$in the frequency domain. As aforementioned, for other harmonics a revision must be applied to equation (2.46) and so on the final state of our equation becomes as in equation (2.47). It is noticed from equation (2.47) that the DC component of Fourier coefficients X(0) is multiplied by $\exp(0) = 1$ as expected.

$$\mathcal{F}[x(t \pm t_0)] = X(j\omega n)\exp(\pm j\omega n t_0) \tag{2.47}$$

2.4 Results

For testing the software explained in this chapter a matrix is constructed. This software can take matrixes as input, with the purpose of working with multivariate equation systems. The first column of matrix is a single sine that is denoted in equation (2.48) and the second column of matrix is a sum of cosine and sine that is denoted in equation (2.51).

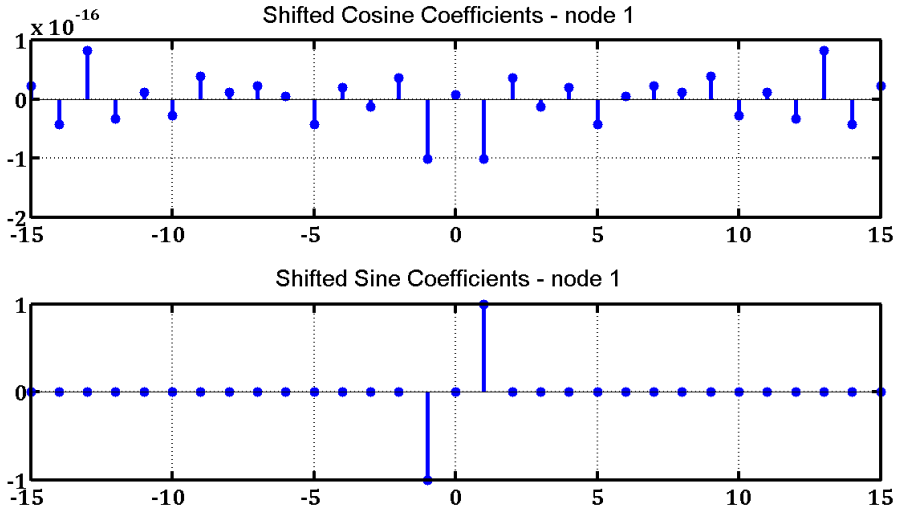

Figure 2.6: Fourier Coefficients of a simple signal.

Fourier coefficients of a single sine with period 1 as in equation (2.48) are shown in figure 2.6. In this figure cosine coefficients can be symbolized as a_n in equation (2.19) and sine coefficients can be symbolized as b_n in equation (2.20). It is observed clearly in this figure, as expected only the sine coefficients for first harmonic have a value other than zero.

$$s(t) = \sin(2\pi * 1 * t) \tag{2.48}$$

$$\frac{ds(t)}{dt} = 2\pi\cos(2\pi * 1 * t) \tag{2.49}$$

$$\int s(t)dt = -\frac{1}{2\pi}\cos(2\pi * 1 * t) \tag{2.50}$$

Frequency domain operations that explained in chapter 2.3 are performed on this simple sine signal to test this software. The result of our software for the derivation of this signal is illustrated in figure 2.7 which is compatible with the expression derived in the equation (2.49).

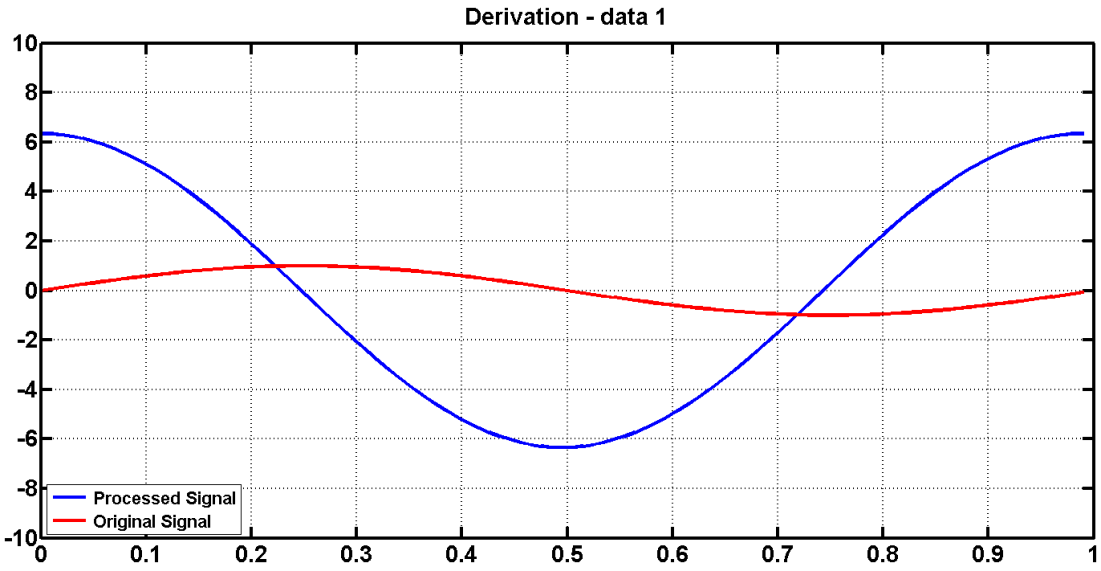

Figure 2.7: Derivation of a single sine.

The result of our software for the integration of this signal is illustrated in figure 2.8 which is compatible with the expression derived in the equation (2.50)(2.49).

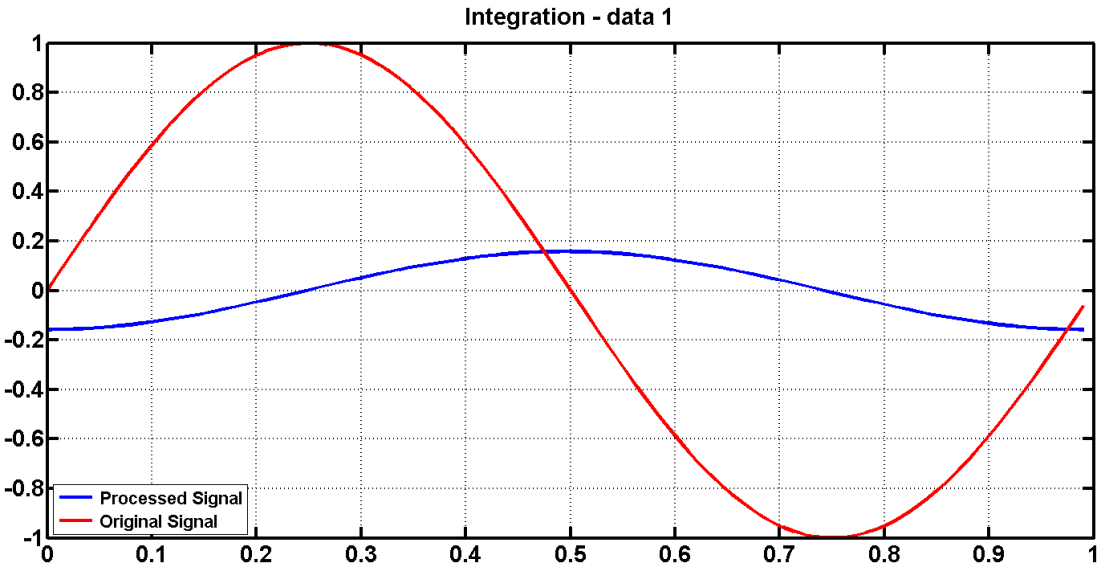

Figure 2.8: Integration of a single sine.

The result of our software for time shifting of this signal by quarter period, which is 0.25 second, is illustrated in figure 2.9.

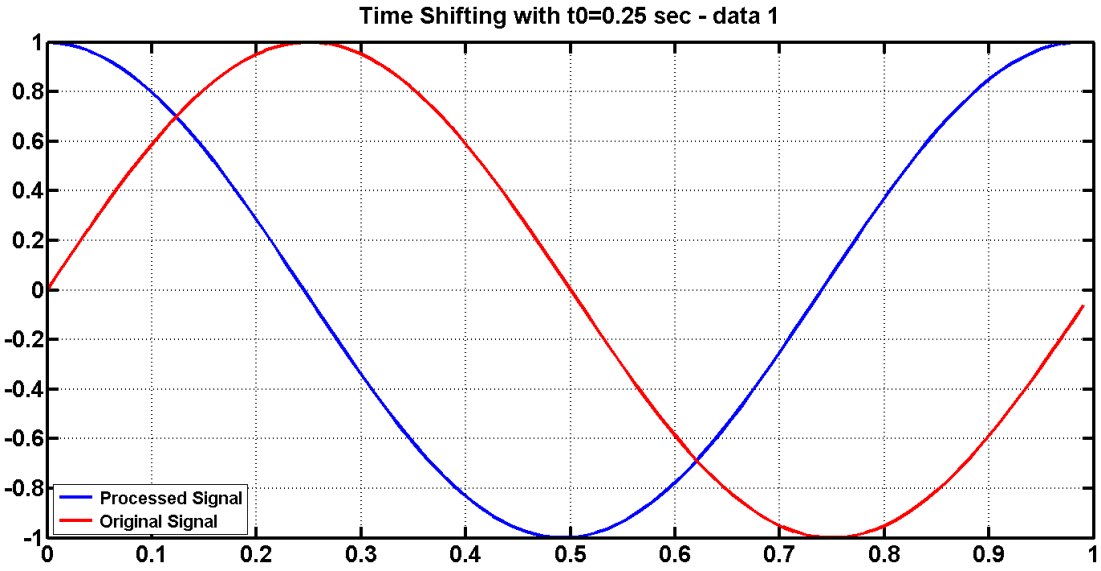

Figure 2.9: Time shifting of a single sine.

$$g(t) = 0.25 * \cos(2\pi * 1 * t) + 0.5 * \sin(2\pi * 4 * t) \quad (2.51)$$

The Fourier coefficients of the signal denoted in equation (2.51) are observed in figure 2.10. As expected for the first harmonics cosine coefficients are appeared with value 0.25 and for the fourth harmonics sine coefficients are appeared with value 0.5.

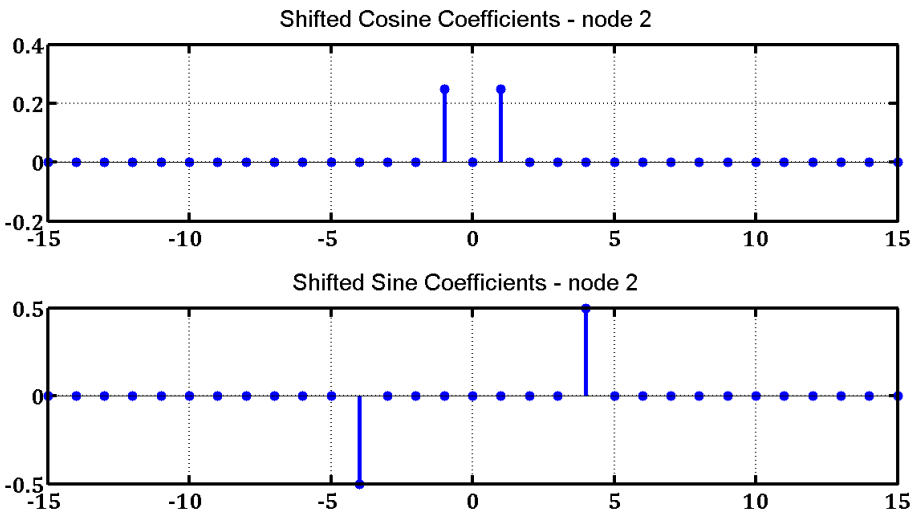

Figure 2.10: Fourier coefficients of sum of sine and cosine.

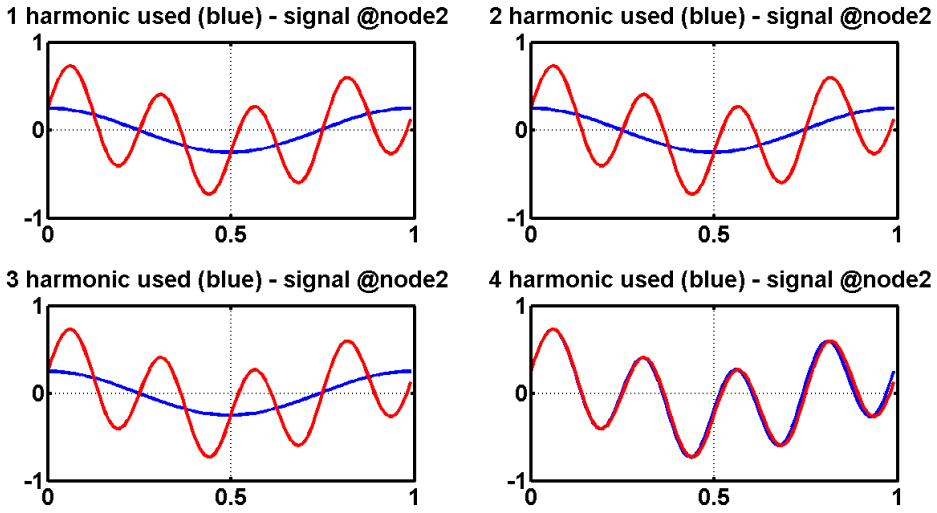

Figure 2.11: Inverse Fourier transform with different harmonic numbers.

The presented signal $g(t)$ by equation (2.51) contains only two elements which contains different harmonics. As can be seen from figure 2.11, four harmonics is enough to represent $g(t)$ signal.

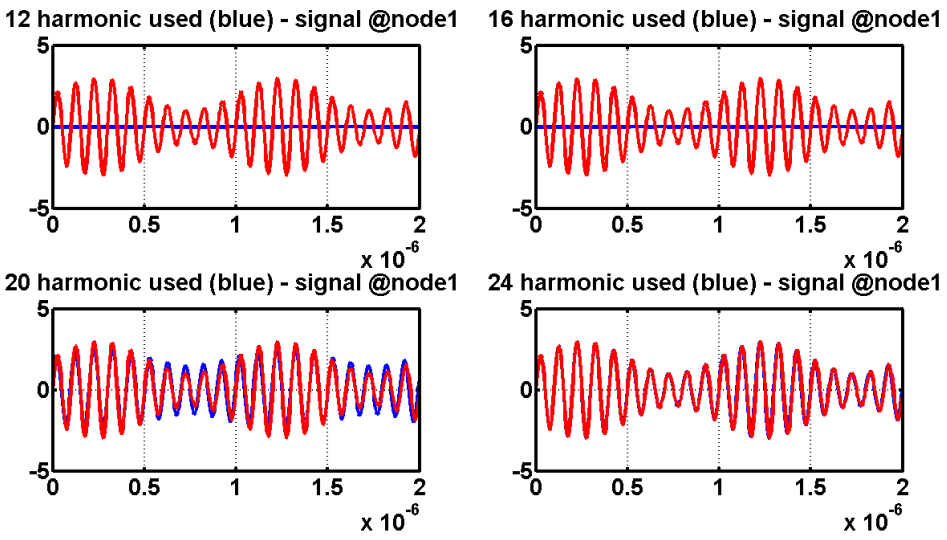

Figure 2.12: Inverse Fourier transform of an amplitude modulation signal according to different number of harmonics.

Here, another example for different harmonic representation is shown in figure 2.12 for an amplitude modulation signal. With the purpose of reducing computational cost it is important to decide the number of harmonics to be used in operations. This presentation in figure 2.12 that appeals to the human perspective can help making this decision properly.

3. HARMONIC BALANCE METHOD

Harmonic Balance is a method for calculating steady-state response of nonlinear differential equations. It is applied mostly on nonlinear electronic circuits to perform PSS analysis. This method can be applied in time domain or frequency domain but in this study it is calculated in frequency domain and the switch between time and frequency domain is performed by Fourier Transform.

The nonlinear equation sets that are trying to be solved in this study, are basic diode circuits equations created with Modal Nodal Analysis(MNA). For these sets of equations there is as much unknown as the number of equations. For our system these unknowns correspond to node voltages and source currents of the analyzed circuit. The calculation of these unknowns for each time instance in the solution domain is the solution of our problem.

$$f(v,t) = \dot{q}\big(v(t)\big) + i\big(v(t)\big) + u(t) = 0 \tag{3.1}$$

Equation (3.1) represents the nonlinear circuit. $v(t)$ is the unknown vector of the circuit, $u(t)$ is the input. $\dot{q}\big(v(t)\big)$ is the memory characteristic of the circuit such as charge or flux. $i\big(v(t)\big)$ is the memoryless characteristic of the circuit such as voltage or current.

$$F(V) = \Omega\Gamma q\big(\Gamma^{-1}V(k)\big) + \Gamma i\big(\Gamma^{-1}V(k)\big) + \Gamma u = 0 \tag{3.2}$$

Equation (3.2) is the harmonic balance equation that represents the nonlinear circuit equations in frequency domain. q and i describe the $q\big(v(t)\big)\ and\ i\big(v(t)\big)$ respectively. Γ and Γ^{-1} describes DFT and IDFT matrices respectively, provide transition of the signal between time and frequency domains. Ω operates the differentiation in frequency domain that is a block diagonal differentiation matrix.

The equation (3.2) accepts unknown vector in the frequency domain and then transfer them to the time domain with Γ^{-1} operator. In time domain the q and i functions are calculated then the results are transferred to the frequency domain with Γ operator. Last, with Ω operator the derivation of q function is calculated. Then the

sum of elements in result of equation (3.2) must be zero in the frequency domain to find accurate solution. This condition must be solved with an iterative method which is Newton' Method. Readers who want to learn more about this topic can refer to chapter 2 of [13] and chapter 6.4.2 of [14].

3.1 Construction of the Harmonic Balance Unknown Vector

In harmonic balance method an unknown vector is constituted that contain all unknowns of an equation set for all time instances in solution domain. For an equation set with N unknown and a solution domain with M sample, harmonic balance unknown vector contains NxM element. For example, let $x(t_m)$, which is shown in equation (3.3), is the unknown vector of problem involving N number equations at time t_m.

$$x(t_m) = [\, x_1(t_m)\; x_2(t_m)\; x_3(t_m) \ldots\; x_N(t_m)] \tag{3.3}$$

Let $v(t)$ is the harmonic balance unknown vector in time domain. Then for all time instances in solution domain $x(t_m)$ are concataneted and struct the $v(t)$ as in equation (3.4).

$$v(t) = [\, x(t_1)\;\; x(t_2)\;\; x(t_3)\;\; \ldots\; x(t_M)] \tag{3.4}$$

To carry this unknown vector to the frequency domain Γ operator is called. For the $k_m{}^{\text{th}}$ harmonic coefficients of all unknowns of an equation set with N unknown can be represented as in equation (3.5).

$$X(k_m) = [\, X_1(k_m)\; X_2(k_m)\; X_3(k_m) \ldots\; X_N(k_m)] \tag{3.5}$$

The number of all harmonics is M that is the time sample number in solution domain, we ensure that is an odd number. M= 2K +1 for some K and then for all harmonics $X(k_m)$ are concataneted and struct the $V(k)$ as in equation (3.6).

$$V(k) = [\, X(-K)\, X(-K+1)\, X(-K+2\,) \ldots X(0)\ \ldots\ X(K-2)\, X(K-1) X(K)] \tag{3.6}$$

Transition between time domain and frequency domain, namely transition between $v(t)$ and $V(k)$ is provided by Γ and Γ^{-1} matrices where $v(t)$ and $V(k)$ are vectors with NMx1 size and Γ and Γ^{-1} are matrices with NMxNM size.

$$\Gamma\, v(t) = V(k) \tag{3.7}$$

$$\Gamma^{-1} V(k) = v(t) \tag{3.8}$$

3.2 Utility Functions to Achieve Harmonic Balance Equation

To calculate harmonic balance equation (3.2), first we need to make sure that operators work properly. For this purpose functions belong to operators are written and checked separately and independently from the main harmonic balance equation.

3.2.1 Γ Operator with FFT Task

As mentioned before in equation (3.7) Γ operator is a DFT matrix but to reduce computation complexity and to use superiority of FFT to DFT this operator implemented in a different approach. For each separate node, the data from the time domain harmonic balance unknown vector was taken and the FFT operation was performed on them separately. Then for frequency domain operations the coefficients shifted. For example the signal at node m for all time instances in solution domain can be represented as in equation (3.9).

$$x_m(t) = [x_m(t_1)\, x_m(t_2)\, x_m(t_3) \ldots\ x_m(t_M)] \tag{3.9}$$

Then with FFT of the signal, the coefficients are calculated and then shifted. The result with K harmonic and total M=2K+1 sample can be represented as in equation (3.10).

$$X_m(k) = [\, X_m(-K)\, X_m(-K+1) \ldots . X_m(-1) X_m(0) X_m(1) \ldots X_m(K-1) X_m(K)\,] \tag{3.10}$$

Then this result is written back to the relevant places in the frequency domain harmonic balance unknown vector. With this way the conversion from the time domain to the frequency domain is achieved.

3.2.2 Γ^{-1} Operator with IFFT Task

As mentioned before in equation (3.8) Γ^{-1} operator is inverse discrete Fourier transform (IDFT) matrix but to reduce computation complexity and to use superiority of FFT to DFT this operator implemented in a different approach. For each separate node, the data from the frequency domain harmonic balance unknown vector was taken and the coefficients shifted back. Then inverse fast Fourier transform (IFFT) operation was performed for each node separately. Then the result is written back to the relevant places in the time domain harmonic balance unknown vector. With this way transformation from equation (3.6) to equation (3.4) is performed.

3.2.3 Ω Operator with Differentiation Task

Ω operator in equation (3.2) is actually a block diagonal matrix and this matrix multiplied on the right by a vector. Each block of this NMxNM size Ω matrix is a NxN size square matrix where N is the equation number in nonlinear equation set. Ω can be represented as in equation (3.11) where I_N is the identity matrix with size NxN and K is the harmonic number.

$$\Omega = \begin{bmatrix} j\omega(-K)I_N & & & \\ & j\omega(-K+1)I_N & & \\ & & \ddots & \\ & & & j\omega(K)I_N \end{bmatrix} \quad (3.11)$$

It is clearly visible that Ω matrix is very large but actually its job can be done with an approach which reduces the computational complexity. This Ω operator can be performed by multiplying the Nx1 size small vectors in the big unknown vector

by their corresponding $j\omega k_t$ instead of multplying the N(2K+1)xN(2K+1) size matrix with the N(2K+1)x1 size vector. In this study, the Ω operator is implemented with this method and it is aimed to reduce the computational cost and speed up to software.

3.3 The Jacobian of Harmonic Balance Equation

Jacobian matrix is a dense matrix with NMxNM size where number of equations is symbolized as N and total harmonic number is symbolized as M=2K+1. Jacobian matrix of the harmonic balance equation which is represented in equation (3.2) can be expressed as in equation (3.12) where $'it'$ denotes the iteration number.

$$J^{(it)} = \Omega\Gamma C^{(it)}\Gamma^{-1} + \Gamma G^{(it)}\Gamma^{-1} \tag{3.12}$$

C and G are block diagonal matrixes which are relevant with q and i respectively can be expressed as in equation (3.13) and (3.14). It should be reminded that these are matrixes with NMxNM size.

$$C = \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_M \end{bmatrix} \tag{3.13}$$

$$G = \begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_M \end{bmatrix} \tag{3.14}$$

C_n and G_n are circuit capacitance and conductance matrixes respectively. These are NxN square matrixes which construct the C and G matrixes. C_n which can be described by equation (3.15) is the Jacobian of q function at t_n time point , G_n which can be described by equation (3.16) is the Jacobian of i function at t_n time point, where $n \in \{1,2,\dots,M\}$. A method similar to applied with the omega operator is used in the processing of these matrices. This method will be explained in detail later in this chapter.

$$C_n(r,s) = \frac{dq(v_r(t_n))}{dv_s} \tag{3.15}$$

$$G_n(r,s) = \frac{di(v_r(t_n))}{dv_s} \tag{3.16}$$

It is pointed out that the operators in equation (3.12) are defined as matrices but as we did with operators in chapter 3.2, different approaches will be applied here to decrease the computational complexity.

First it must be noticed that Γ, Γ^{-1} operators need NMx1 vector as input. Another important point is the fact that $J^{(it)}$ is a big Jacobian matrix with size NMxNM. In order to these facts, an approach can be expressed as in equation (3.17) where I is the identity matrix with size NMxNM. This expression can be derived to (3.18).

$$J^{(it)}I = (\Omega\Gamma C^{(it)}\Gamma^{-1} + \Gamma G^{(it)}\Gamma^{-1})I \tag{3.17}$$

$$J^{(it)} = \Omega\Gamma C^{(it)}\Gamma^{-1}I + \Gamma G^{(it)}\Gamma^{-1}I \tag{3.18}$$

For each column of I which is a vector with size NMx1 equation (3.18) can be calculated and the result can be written to Jacobian matrix's relevant column. In this equation $(c_b,:)$ indices denotes the ${c_b}^{th}$ column of a matrix. So to solve equation (3.18), equation (3.19) must be solved for NM times where NM is the column number of Jacobian matrix.

$$J^{(it)}(c_b,:) = \Omega\Gamma C^{(it)}\Gamma^{-1}I(c_b,:) + \Gamma G^{(it)}\Gamma^{-1}I(c_b,:) \tag{3.19}$$

As aforementioned $C^{(it)}$ and $G^{(it)}$ are block diagonal matrices with NMxNM size and this matrices multiplied on the right by NMx1 size vector as shown in equation (3.20). With the purpose of reducing computational complexity for time point t_m, NxN Jacobian matrix C_m can be multiplied with relevant Nx1 small vector $x(t_m)$ which is part of NMx1 size unknown vector $v(t)$ which is defined in equation (3.4). With this way we will be able to save memory from large matrices and reduce

the computational complexity. A MATLAB structure array can save the NxN square Jacobian matrices, which is represented as C_n and G_n, for any time point instead of constructing a large matrix.

$$\begin{bmatrix} C_1 & & \\ & C_2 & \\ & \ddots & \\ & & C_M \end{bmatrix} \cdot \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ x(t_M) \end{bmatrix}' = \begin{bmatrix} C_1 x(t_1) \\ C_2 x(t_2) \\ \vdots \\ C_M x(t_M) \end{bmatrix} \tag{3.20}$$

3.4 Newton's Method as an Iterative Nonlinear System Solver

Newton's Method is an iterative solver for a function f(x), with the purpose of finding 'x' where f(x)=0. It is assumed that x_k is known to be an approximate solution. Then Newton's Method proposes the equation (3.21) that aims to find a more approximate solution which is denoted by x_{k+1}. It is checked with some tolerances whether the new solution is the exact solution. If not, a new iteration is started by calculating equation (3.21) once more. This process continues until the exact solution is found.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{3.21}$$

Equation (3.21) is written for only one equation but if a equation system is desired to solve, equation (3.21) must be revised as equation (3.22) where $\bar{x}$ is unknown vector, $\bar{f}(\bar{x})$ represents equation set and $D\bar{f}(\bar{x})$ represents the Jacobian of this set.

$$\bar{x}_{k+1} = \bar{x}_k - \frac{\bar{f}(\bar{x}_k)}{D\bar{f}(\bar{x}_k)} \tag{3.22}$$

For this study, equation (3.22) is revised as equation (3.23). Here J must be found with equation (3.12) and F must be found with equation (3.2). V, which is denoted by equation (3.6), is the unknown vector in the frequency domain.

$$J(V_k) * (V_{k+1} - V_k) = -F(V_k) \tag{3.23}$$

For solving harmonic balance equation with Newton's Method, setting the initial values of unknown vector is very important. Because the success of the

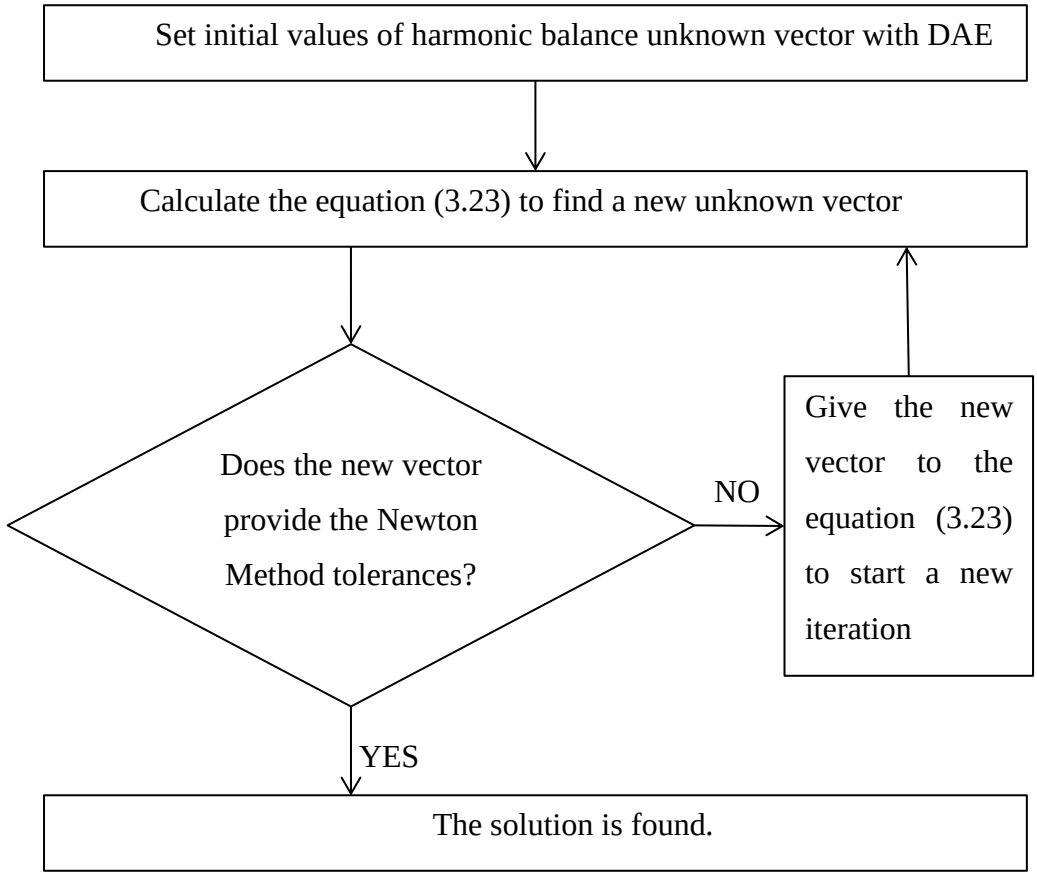

Figure 3.1: Flow chart of Newton's Method for harmonic balance

Newton's Method depends on the initial values. If the initial values are not close to the solution, then the method will not work properly. In order to set initial value properly differential algebraic equation(DAE) solver used on equation (3.2). 'ode15s', a MATLAB function, is used as DAE solver in this study. Then the result given to Newton's Method as initial value. Actually this initial value is the transient solution of our system where the Newton's Method trying to find PSS solution.

Rate of convergences is a important parameter to compare iterative method algorithms. Because this rate gives important ideas about the speed of algorithm. Newton's Method has quadratic convergence which needs less iteration to reach solution. This fact make Newton's Method frequently-used and more desirable.

Finally to implement Newton's Method, 'fsolve' which is a MATLAB function is used in this study.

3.5 Results

In this section, it is aimed to represent that utility functions work properly. For this purpose a square wave signal and a triangular wave signal shown in figure 3.2 and 3.3 are constructed with piecewise linear (PWL) modeling. The harmonic balance unknown vector is constructed from the Fourier coefficients of these two signals and the utility functions are performed on them.

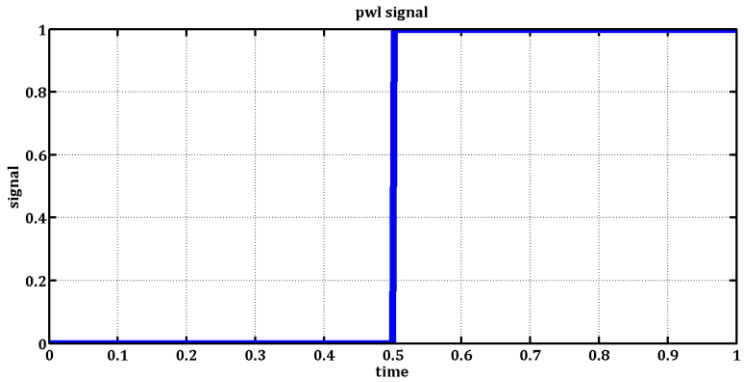

Figure 3.2: Square wave signal.

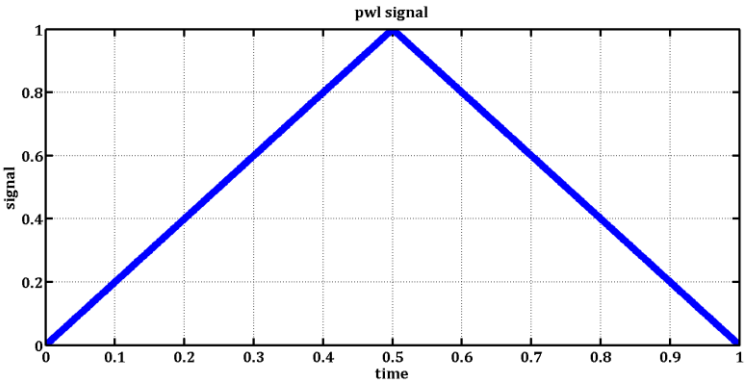

Figure 3.3: Triangular wave signal.

First, Fourier coefficients of these signals are calculated with Fourier coefficient calculation tool. Plotting method of this tool is utilized to obtain results in figure 3.4 and 3.5. By investigating figure 3.4 it can be observed that the inverse transform of square wave with different harmonics. It is clear that even 22 harmonics

is not enough to represent square wave by satisfying human perspective. The same process is performed on triangular wave and illustrated in figure 3.5. It is seen from this figure that 7 harmonics is enough to represent triangular wave and to satisfy human perspective. This representation helps to form an opinion about how many harmonic is needed for denoting a signal. By this means it is possible to reduce the number of calculated variables and the computational cost.

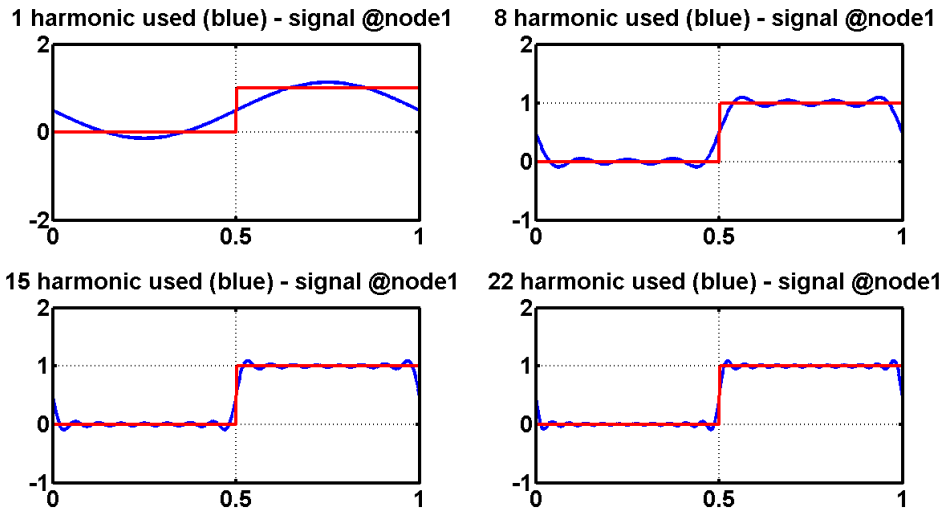

Figure 3.4: Inverse Fourier transform of square wave for various harmonic numbers.

Gibbs phenomenon is observed in figure (3.4), where square wave has a jump discontinuity. It is an artifact of Fourier series which needs infinite coefficients to represent square wave.

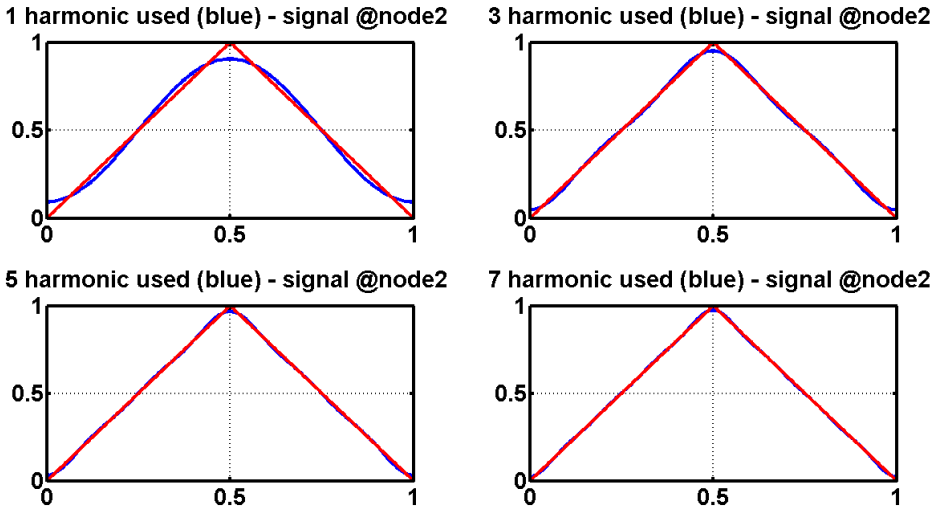

Figure 3.5: Inverse Fourier transform of triangular wave according to various harmonic numbers.

The complex coefficients belong to first forty harmonics of square and triangular wave are illustrated in figure 3.6. By examining this figure it is remarkable that square wave is occurred from sum of sines and triangular wave is occurred from sum of cosine.

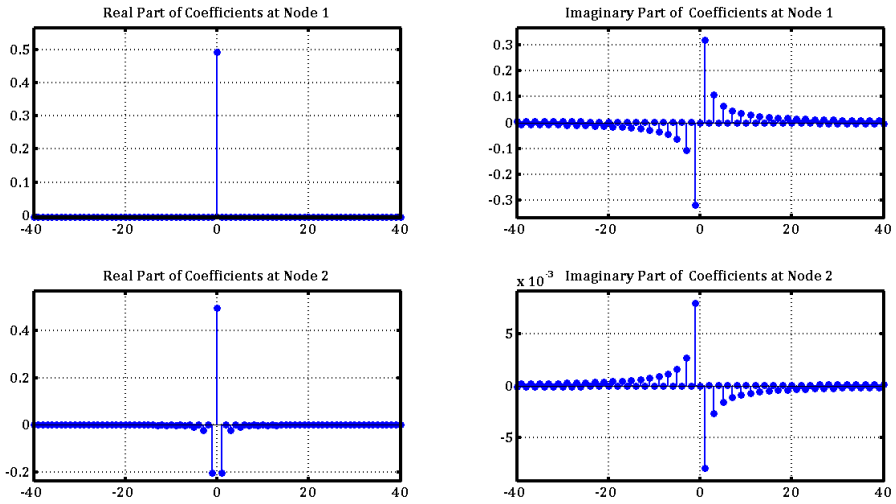

Figure 3.6: Fourier coefficients of square wave at node 1 and triangular wave at node 2.

Then Γ^{-1} operator is performed on these Fourier coefficients and the result, which are as expected, is illustrated in figure 3.7.

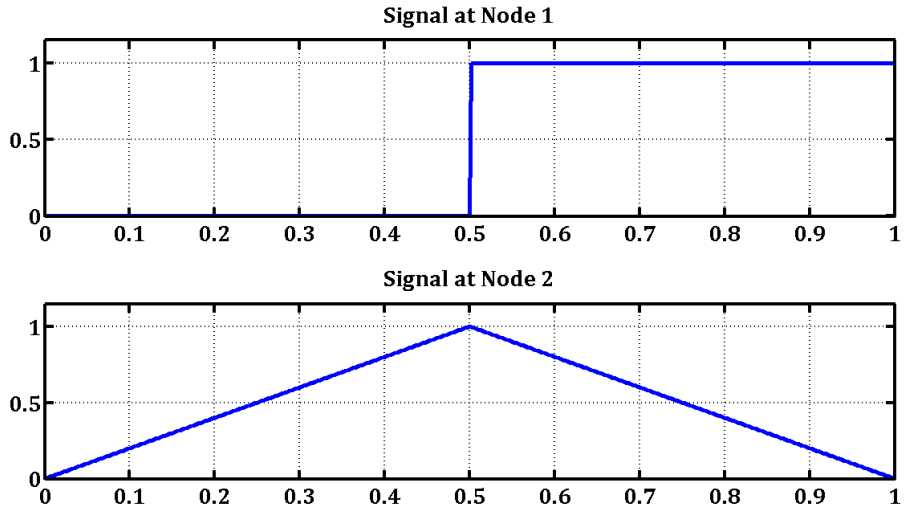

Figure 3.7: Results of Γ^{-1} operator on Fourier coefficients of related signals.

The results in figure 3.7 actually stored in the harmonic balance unknown vector in time domain. Then Γ operator performed on this unknown vector to differentiate signal. With this way harmonic balance unknown vector is transferred to the frequency domain again and differentiation operation is performed on it with $\boldsymbol{\Omega}$. Then for representation this result is transferred to the time domain again with Γ^{-1} operator. The derivation of square wave and triangular wave is illustrated in figure 3.8. This result is expected as in this figure. For square wave, at the jump point a high value is expected. For triangular wave a positive slope is expected for the rising half and a negative slope is expected for the descending half.

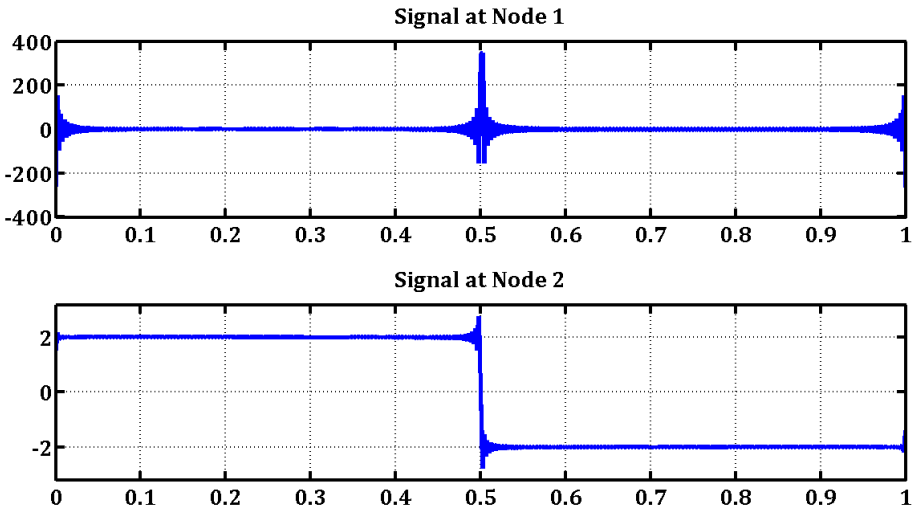

Figure 3.8: Results of Ω and Γ^{-1} operator respectively on Fourier coefficients of related signals.

4. SOFTWARE CONSTRUCTION

It was aimed that the code written in this study will be open source software that everyone whom is interested in this topic can use. For this purpose this software is written comply with fundamental principles and techniques of software development. We intended to write a software which is safe from bugs, easy to understand and ready for change.

4.1 Fourier Coefficient Calculation Tool

This software has two main tasks to do. The first task is interpolation. The second one is calculating Fourier Coefficients. Two classes were created, 'AbsInterpolation' and 'FourierCoefficient', respectively, to perform these tasks.

'AbsInterpolation' is an abstract base class. In MATLAB object-oriented programming, the base class name is used instead of the parent class. It has subclasses named 'stair_interpolation', 'linear_interpolation' and 'spline_interpolation'. As it can be seen, the inheritance concept is used here. Many different types of interpolation carry the same purpose in essence. This purpose is to derive a function that can calculate the value of unknown points between known two points. Each discrete method tries to solve it in its own distinct ways and derives different functions. The tasks expected to be performed jointly by different interpolation types in this study are declared as abstract methods in the 'AbsInterpolation' class. Then in the same named methods of subclasses, the operations specific to the interpolation type are performed separately. The base class is chosen as an abstract class. Because it actually serves as an interface for a group of related subclasses. So it is not desirable to instantiate this class during the run of code. So it would not be wrong to say that this 'AbsInterpolation' class is a common name we can use to call the corresponding subclasses during the execution of the code.

The task of 'FourierCoefficient' class is to select and implement the most effective Fourier Coefficients calculation method for different data types. As explained in detail in Chapter 2, the characteristics of the data are passed to this class via flags. The desired interpolation method is passed to this class via a string. Passing

multiple input parameters to the constructor of a class is not nice in terms of programming and writing the same input parameters repeatedly is also challenging and daunting. Even if all this is left aside, it has become imperative to construct a different class for the control and storage of these parameters. Because the control of these parameters, which will affects the operation to be performed, is a big responsibility. For this purpose 'FourierCoefProperty' class is created. This class's task is to check if flags, string and the data requested to be processed are given properly. If these values are not given properly, a warning should be given to the user and default values should be assigned to these values in order to maintain operation. Here, it is aimed to simplify a class which starts to become complicated and grows excessively. It is hoped to attain user-friendly software with this approach.

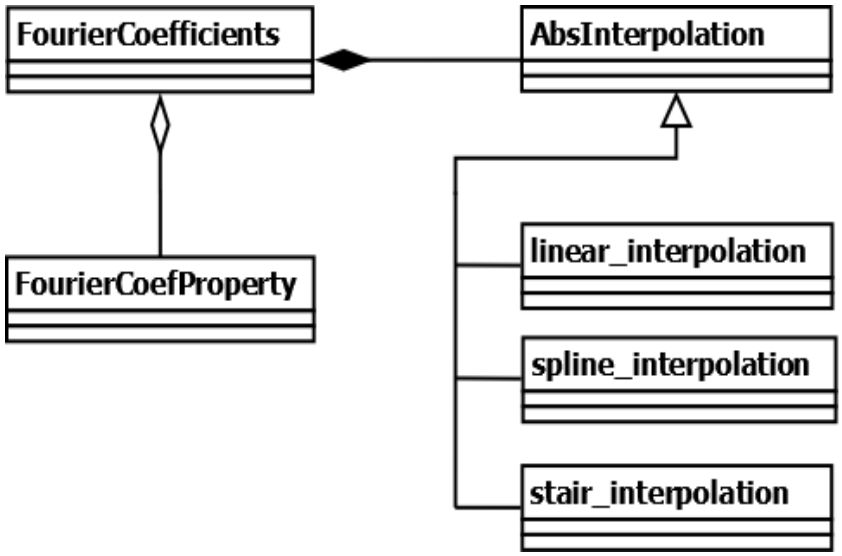

Figure 4.1: Class diagram of Fourier Coefficient calculation tool.

In Figure 4.1, the relation of the classes of this tool to each other is revealed. A line with the non-filled arrow between 'AbsInterpolation' and its subclasses represents inheritance as explained in detail in this section. A line with the filled diamond represents composition concept and a line with the non-filled diamond represents aggregation concept. As it is understood from this figure, it is 'FourierCoefficient' class undertakes the work to be done and it receives help from 'AbsInterpolation' and 'FourierCoefProperty' classes.

According to the composition concept, 'FourierCoefficient' class owns the 'AbsInterpolation' class and 'AbsInterpolation' class seems to have no role in the

system, regardless of ‘FourierCoefficient’ class. Indeed, ‘AbsInterpolation’ is a tool designed to do ‘FourierCoefficient’ class’s work and its methods consist of algorithms specific to Fourier calculation tool. In other words, the use of this ‘AbsInterpolation’ classes seems to be unreasonable apart from the ‘FourierCoefficient’ class. At the same time, in this concept if ‘FourierCoefficient’ will be destroyed then the contained objects that are ‘AbsInterpolation’ for example will be destroyed too.

In compliance with agreegation concept, ‘FourierCoefProperty’ class exists independent from ‘FourierCoefficient’ class. Indeed the ‘FourierCoefProperty’ class does not have much meaning independent of ‘FourierCoefficient’. Because, in fact this class deals preparing the inputs of ‘FourierCoefficient’. But as already explained, it is a necessity to form this class separately from the main structure. This is because the input parameters are crowded and the controls they require do not leave a better solution. In fact, if we think of this ‘FourierCoefProperty’ class as the driver of the ‘FourierCoefficient’ vehicle, it is understandable that it conforms to the concept of aggregation. . If the ‘FourierCoefficient’ class is destroyed in this concept, destruction of the ‘FourierCoefProperty’ class is not absolutely necessary. According to these explanations, the composition seems to be a stronger association than aggregation.

It should be noted that all classes created in this tool are derived from the handle superclass. In short when a handle class object is instantiated, actually a reference to an object is created. When this handle derived object is copied, MATLAB copies only the handle. So the copy only stores the reference and both the copy and the original handle are referring to the same object in the memory. Handle class has important roles in projects like this tool which has to work with large data sets. Because handle class makes a concession working with large data sets without copying it.

4.2 HarmonicBalance Equation Solver

This software has one main purpose which is solving the harmonic balance equation with Newton’s Method. With this purpose ‘HarmonicBalance’ class is created. But aforementioned as in chapter 3 in this thesis, the work to be done is too much and too complicated to fit into one class. In order to reduce this complication a

new class is created with name ‘HB_utilities’ that its purpose is to implement utility functions mentioned in chapter 3.2.

As predicted, it is necessary to instantiate ‘HarmonicBalance’ and ‘HB_utilities’ objects for same specific problems. On the basis of this fact, the relation between ‘HarmonicBalance’ and ‘HB_utilities’ classes is selected as composition which is explained in the previous section. In order to implementing this composition approach ‘HB_utilities’ class is instantiated in the constructor of ‘HarmonicBalance’ class. This relationship between two classes is illustrated by a line with filled diamond in figure 4.2.

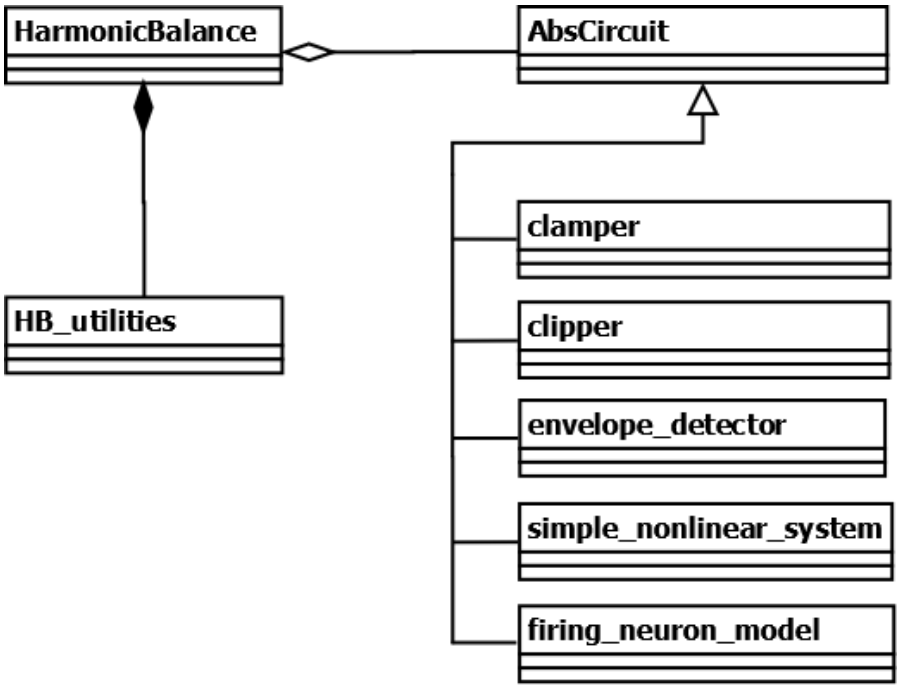

Figure 4.2: Class diagram of harmonic balance equation solver.

In this study a class with name ‘AbsCircuit’ is created for the implementation of nonlinear equation sets. ‘AbsCircuit’ class is also an abstract base class which is defined in the previous section in detail while explaining the ‘AbsInterpolation’ class. Inheritance concept used here and this class has subclasses named ‘clamper’, ‘clipper’, ‘envelope_detector’, ‘simple_nonlinear_system’ and ‘firing_neuron_model’. The common feature of these subclasses is that they all store nonlinear equations which are specific for their relevant problem.

‘AbsCircuit’ class gives the nonlinear equations at time domain to the ‘HarmonicBalance’ class where these nonlinear equations are aimed to solve at

frequency domain with Newton's Method. But this 'AbsCircuit' class is instantiated independently from 'HarmonicBalance' class and can be used with different solvers. So aggregation concept is used here and this is illustrated by a line with un-filled diamond in figure 4.2.

Created classes belong to this software are also derived from the MATLAB 'handle' superclass which explained in detail in the previous section.

5. RESULTS

Nonlinear systems tried to be solved in this study are nonlinear circuits. Nonlinear equation sets of these circuits are obtained by MNA method. Unknowns of these equations are node voltages and source currents. The Kirchoff Current Law (KCL) is applied for each node, the voltage equation for each voltage source is written, and finally these equations form the nonlinear equation sets.

It is desired to observe the success of this software in different nonlinear systems. For this purpose, two systems which are suitable for this software were selected from the systems that Gu and Roychowdhury used in their work.[15] Selected systems resolved and the results are compared to theirs.

5.1 Clamper Circuit

Clamper circuit is used to stabilize the upper or lower extreme of a waveform to a constant DC voltage level. The direction of the diode connection decides whether upper or lower extreme of the signal is to be stabilized.

In figure 5.1, the schematic of the positive clamper circuit is shown. Voltage on the resistor is the output of this circuit. Input signal with an offset addition will appear at the output node. This offset is determined by 'Vdc'. However, as clearly seen in the circuit diagram, the diode voltage will be added to this offset value.

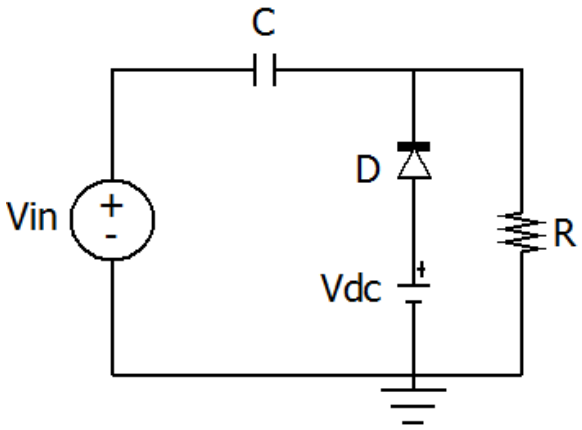

Figure 5.1: Positive clamper circuit.

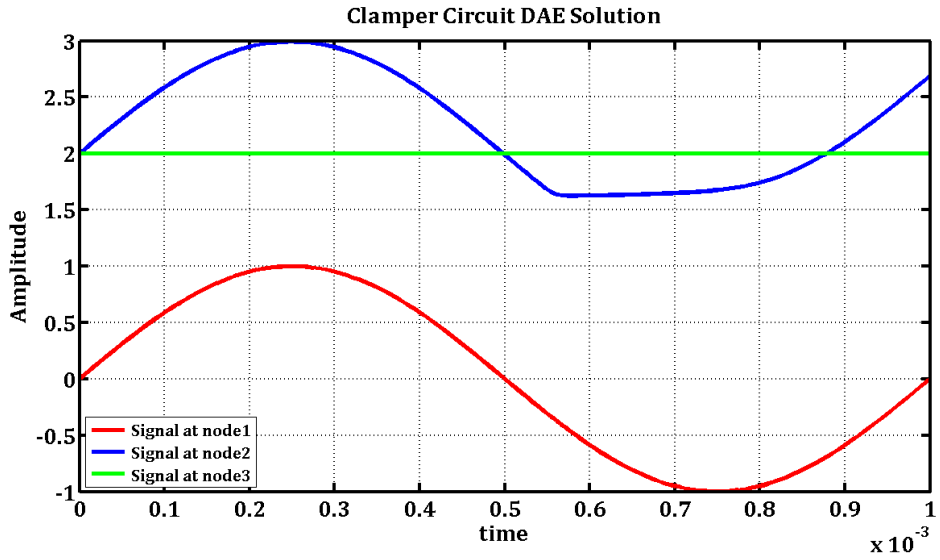

Figure 5.2: Clamper circuit transient solution with ‘ode15s’.

In figure 5.2, the first period of transient solution is illustrated for the clamper circuit. For this simulation the component values and sources are chosen as R=10 kohm, C=10 μF, Vdc=2 V, Vin=sin(2π1000t) V. To represent one period 151 sample is used which means harmonic balance unknown vector involves 151 elements for each node. Nonlinear equation set of this circuit has five equation, which means that three of them from KCL and two of them from voltage sources. It also means that this problem has five unknowns so harmonic balance unknown vector has 755 elements in total.

It can be seen from figure 5.2 signal at node 1 represents Vin, signal at node 3 represents Vdc and signal at node 2 represents the output voltage of this circuit. From the output signal at transient solution the effect of capacitance can be seen.

Then with harmonic balance method the PSS analysis is implemented on this circuit. The results are illustrated in figure 5.3. The change on signal at node 2 is obvious.

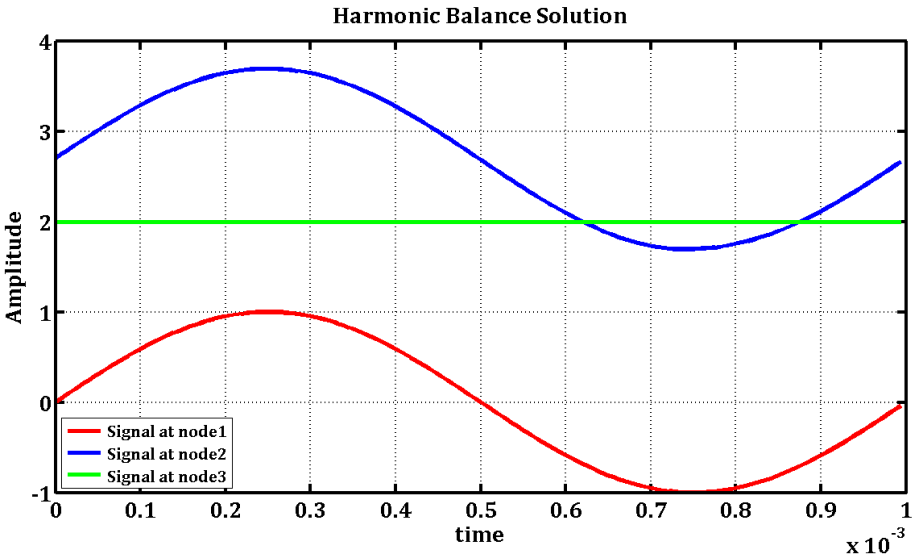

Figure 5.3: Clamper circuit PSS analysis with our software.

5.2 Envelope Detector Circuit

Envelope detector circuits are used to get envelope of the input signal which is generally a signal with high frequency. The schematic of circuit is illustrated in figure 5.4. Capacitor charge with the rising input signal then when the diode cuts-off, capacitor discharges through the resistor. It is briefly explained here how this circuit works.

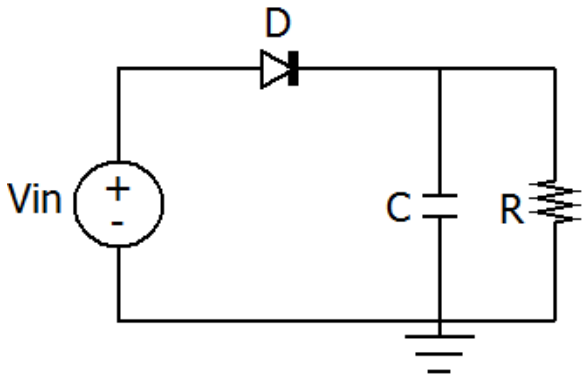

Figure 5.4: Envelope detector circuit.

In figure 5.5, the first four periods of transient solution is illustrated for the envelope detector circuit. For this simulation the component values and source are chosen as R=1 kohm, C=70 pF, Vin=[(sin($2\pi 10^6 t$) + 2)x(sin($2\pi 10^6 t10$))] V. It is seen that Vin is an amplitude modulation signal with a signal (sin($2\pi 10^6 t$) + 2) and a

carrier $\sin(2\pi 10^6 t10)$. It is obvious that to represent one period 151 sample is not enough. In order to represent data properly 301 samples is used per period. That means harmonic balance unknown vector involves 301 elements for each node. Nonlinear equation set of this circuit has three equations, which means that two of them from KCL and one of them from voltage sources. It also means that this problem has three unknowns so harmonic balance unknown vector has 903 elements in total.

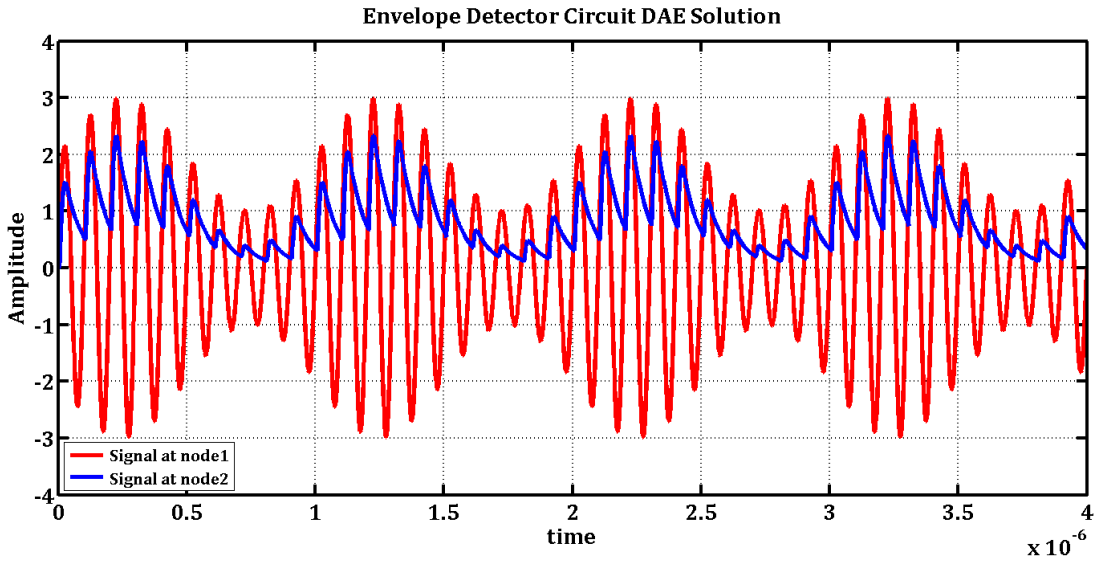

Figure 5.5: Envelope detector circuit transient solution with 'ode15s'.

The results of harmonic balance method is illustrated in figure where signal at node 1 represents Vin, signal at node 2 represents the output voltage of this circuit. By investigating figure 5.6 and 5.5 it can be observed that the falling parts of the output signal of the periodic solution has changed dramatically. Also it can be seen that peak-to-peak voltage of ripples at the rising part of output signal is smaller at PSS solution.

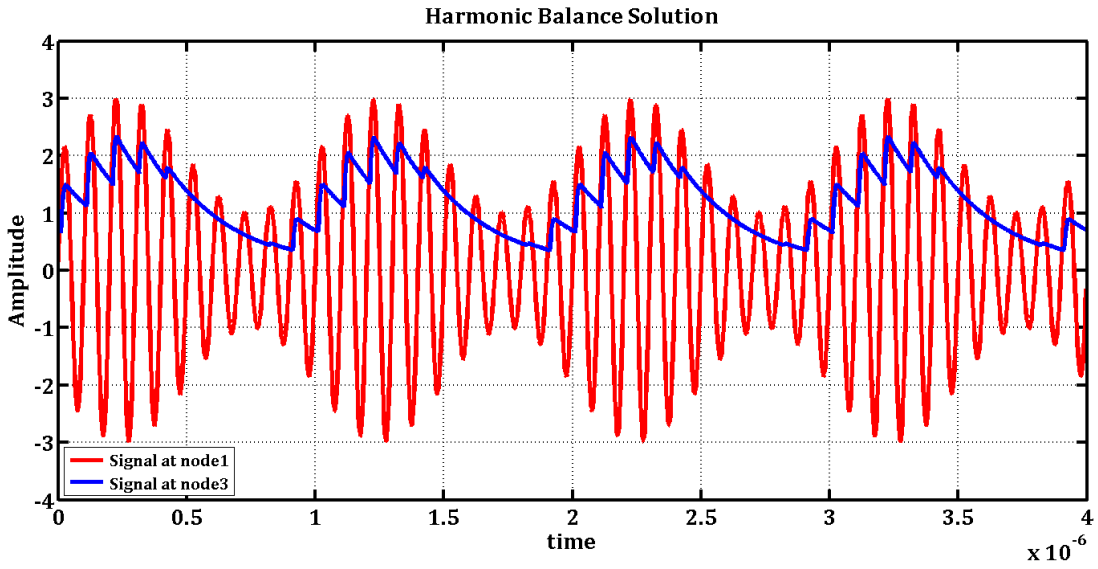

Figure 5.6: Envelope detector circuit PSS analysis with our software.

In table 5.1 the process of Newton's Method is represented with function values and norm of the difference between solutions for each iteration. As aforementioned 'fsolve' function is used for Newton's Method. In this analysis 1e-14 is given to the 'fsolve' as 'Tolfun' and 1e-12 is given to the 'fsolve' as 'Tolx'. 'fsolve' function could calculate the solution after 9 iteration, it also gives the exit flag '1' which means that the solution is accurate.

Table 5.1: Newton's Method process datas for envelope detector circuit PSS analysis.

Iteration No	Function Values	Norm of Step
0	2.7532e-06	-
1	2.7532e-06	1.2505e-01
2	1.5005e-06	3.1263e-02
3	3.7908e-07	6.2527e-02
4	1.1623e-07	6.3644e-02
5	1.6191e-08	1.0814e-02
6	7.2576e-10	1.0267e-02
7	1.3040e-12	3.1136e-03
8	2.8842e-18	9.4542e-05
9	5.2102e-30	6.4449e-08

The quadratic decrease at function values in table 5.1, which refers to quadratic convergence, can be seen clearly. Figure 5.7 represents this quadratic convergence according to function values for each iteration.

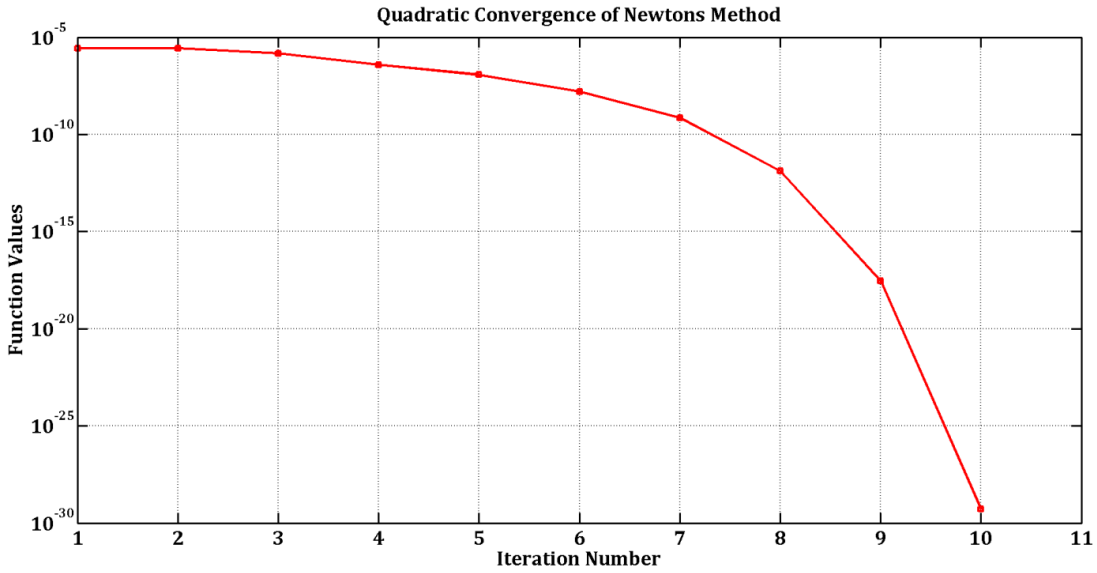

Figure 5.7: Quadratic convergence for envelope detector circuit PSS analysis.

As mentioned before the sample number is highly important for envelope detector circuit which needs more sample to represent high frequency components. According to this fact, the success of Newton's Method can be seen clearly by investigating figure 5.8 and figure 5.9 which are the results of PSS algorithm with 191 samples and 2101 samples respectively. The higher sample number gives accurate results for envelope detector circuit.

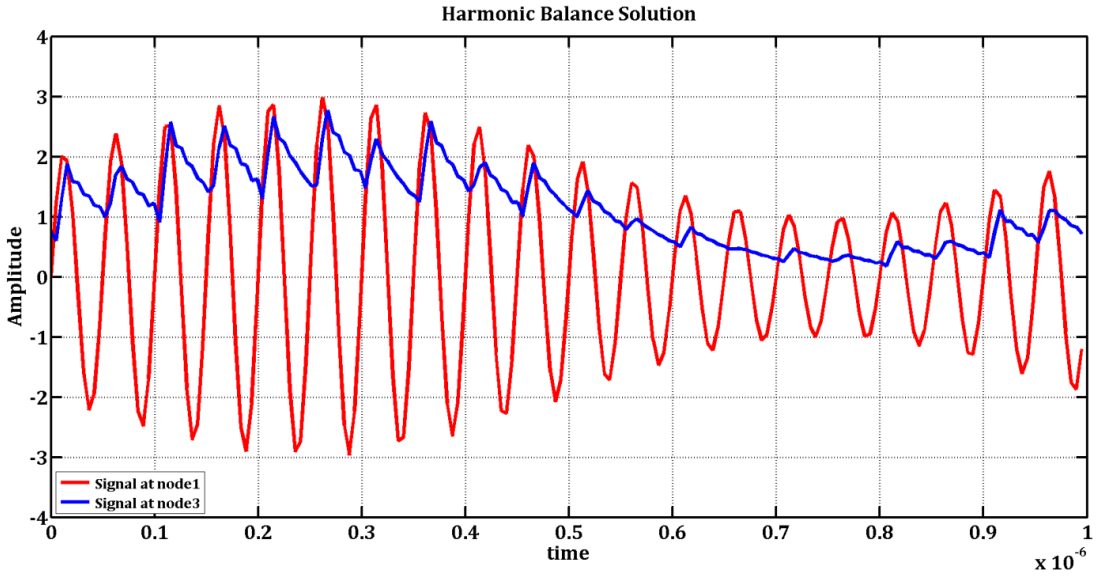

Figure 5.8: Envelope detector circuit PSS analysis with 191 samples for one period.

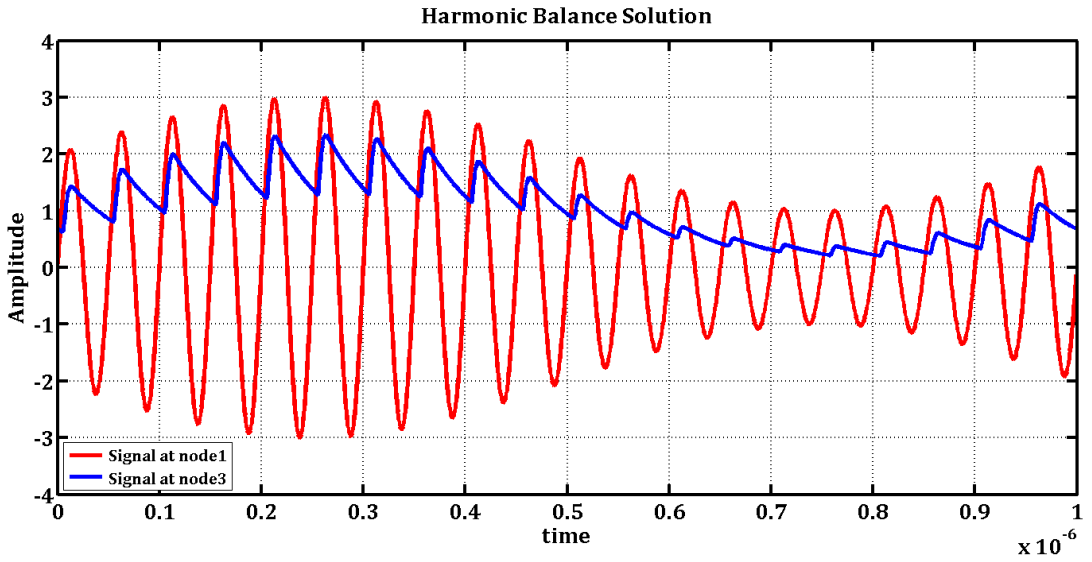

Figure 5.9: Envelope detector circuit PSS analysis with 2101 samples for one period.

With the purpose of observing the effect of different sample numbers at solution, the simulation runs with the highest number of samples allowed by the computer hardware and the solution is accepted as the exact solution. Then for sample numbers smaller then this highest sample number, the PSS algorithm has ran to found each solution. The results of this process is represented in table 5.2. It is clearly seen that the calculation time is increased with the growing sample number. The column ‘norm of difference of vectors’ represents the similarity of each solution

to the exact solution. It is obvious that with the increasing sample number solutions are getting similar to the exact solution.

Table 5.2: Observation of Newton's Method performence according to different sample numbers for envelope detector circuit.

Sample Number	Iteration Number	Calculation Time(sec)	Norm of difference of vectors
109	11	5.19	3.09
191	8	9.84	2.57
281	17	36.71	0.6725
571	12	104.85	0.3062
901	17	345.44	0.1524
1591	19	1257.85	0.0438
2101	21	2609.57	0.0314
2501	20	3283.22	0

5.3 Clipper Circuit

Clipper is used to shaping an input signal by clipping from its tops. It can clip from positive half or negative half or both halves. A clipper circuit, which clips both halves with two diodes that connected to output voltage in opposite direction, is shown in figure 5.10 diode D1 part of this circuit clips the negative parts of input signal and D2 part clips the positive half.

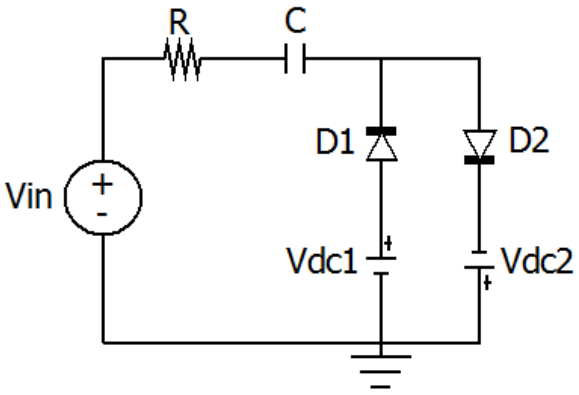

Figure 5.10: Clipper circuit.

In figure 5.11, the first two periods of transient solution is illustrated for the clipper circuit. For this simulation the component values and sources are chosen as R=100 ohm, C=1 μF, Vin=5*sin($2\pi 10^3 t$) V, Vdc1=3 V, Vdc2=3 V. To represent one period 201 sample is used which means harmonic balance unknown vector involves 201 elements for each node. Nonlinear equation set of this circuit has eight equation, which means that five of them from KCL and three of them from voltage sources. It also means that this problem has eight unknowns so harmonic balance unknown vector has 1608 elements in total which means computational cost is expensive for this problem.

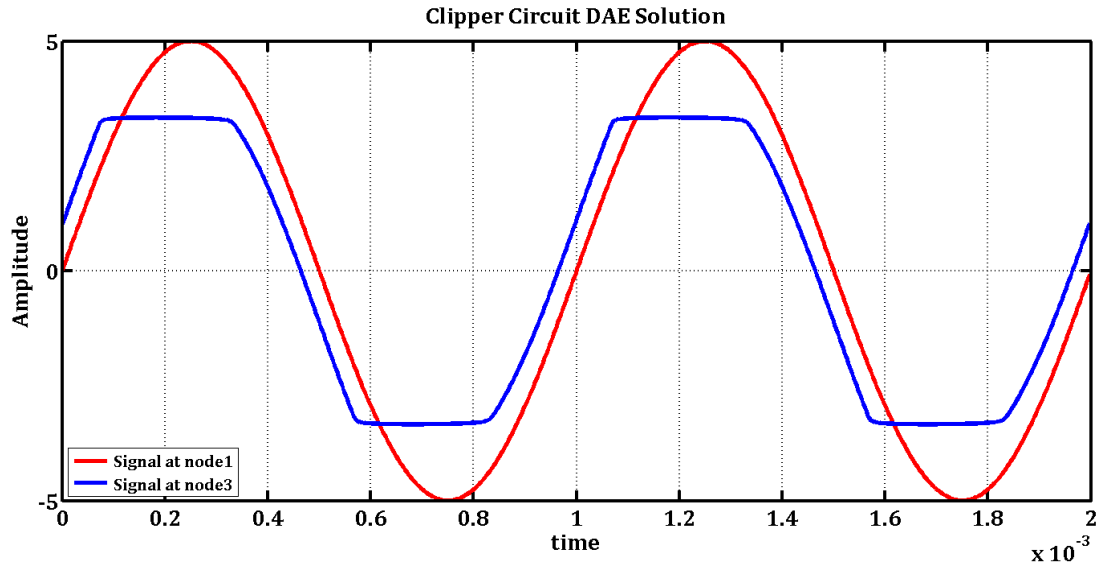

Figure 5.11: Clipper circuit transient solution with 'ode15s'.

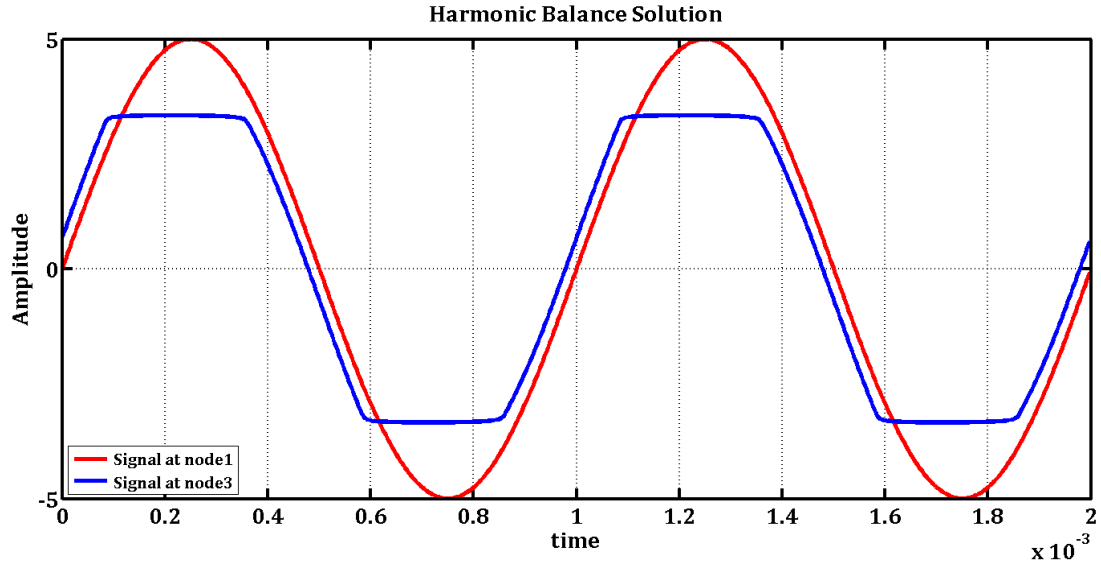

Figure 5.12: Clipper circuit PSS analysis with our software.

The PSS solution of clipper circuit is illustrated in figure 5.12 where signal at node 1 represents Vin, signal at node 3 represents the output voltage of this circuit. By examining figure 5.11 and 5.12 it can be seen that in PSS solution phase shift between input and output signals is smaller than transient solution. It can be said that the exponents, which caused the transient effect on the capacitance, have died in the PSS solution.

In table 5.3 the process of Newton's Method is represented with function values and norm of the difference between solutions for each iteration. In this analysis 1e-14 is given to the 'fsolve' as 'Tolfun' and 1e-12 is given to the 'fsolve' as 'Tolx'. 'fsolve' function could calculate the solution after 12 iteration, it also gives the exit flag '1' which means that the solution is accurate.

Table 5.3: Newton's Method process datas for clipper circuit PSS analysis.

Iteration No	Function Values	Norm of Step
0	1.4416e-04	-
1	1.4416e-04	3.9829e-01
2	5.4231e-05	0.9957e-01
3	5.4230e-05	1.9914e-01
4	3.2517e-05	0.4979e-01
5	1.0160e-05	0.9957e-01
6	6.7817e-07	1.4093e-01
7	6.5784e-08	0.9976e-02
8	3.9013e-09	0.3184e-02
9	5.8434e-11	0.1145e-02
10	3.0208e-14	0.1732e-03
11	1.0327e-20	3.9852e-06
12	4.5860e-30	2.3203e-09

Figure 5.7 represents thequadratic convergence of Newton's Method according to function values for each iteration which are presented in table 5.3.

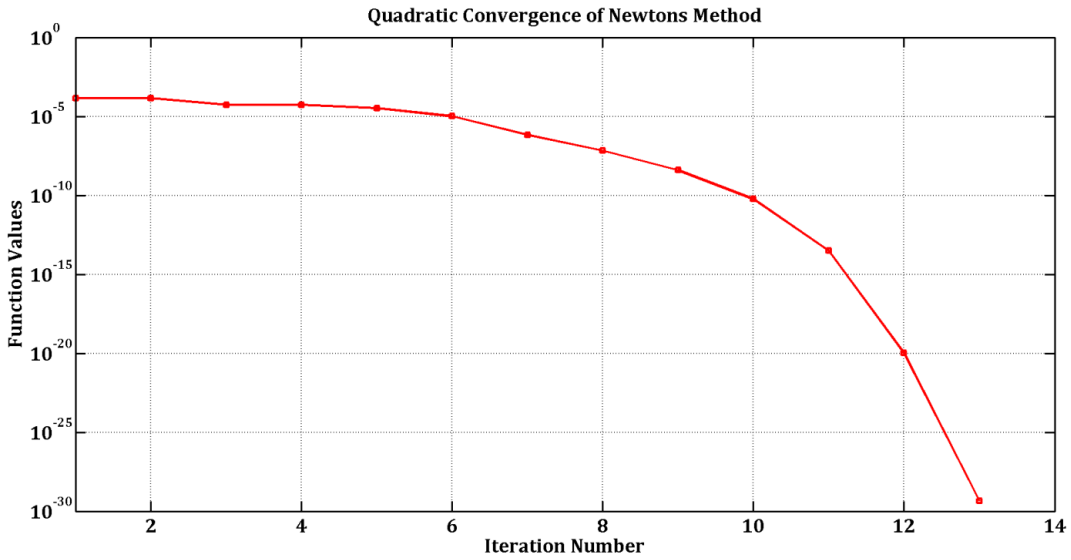

Figure 5.13: Quadratic convergence for clipper circuit PSS analysis.

There is no need big sample numbers to represent signal for clipper circuit unlike the envelope detector circuit. At the same time it must not be missed that the node number of the clipper circuit is 8 and the node number of the envelope detector circuit is 3. These numbers have a considerable effect on the computational cost.

With the purpose of observing the effect of different sample numbers at solution, the simulation runs with a chosen highest number of samples which is 141 for this analysis and the solution is accepted as the exact solution. Then for sample numbers smaller then this highest sample number, the PSS algorithm has ran to found each solution. The results of this process is represented in table 5.4. It is clearly seen that the calculation time is increased with the growing sample number. The column 'norm of difference of vectors' represents the similarity of each solution to the exact solution. It is obvious that with the increasing sample number solutions are getting similar to the exact solution.

Table 5.4: Observation of Newton's Method performence according to different sample numbers for clipper circuit.

Sample Number	Iteration Number	Calculation Time(sec)	Norm of difference of vectors
21	5	1.73	0.0499
33	5	2.67	0.0179
51	8	6.35	0.0157

Table 5.4: Continues

71	7	8.67	0.0091
99	8	15.35	0.0061
141	8	24.76	0

5.4 A Simple Nonlinear System

As aforementioned, nonlinear equation set, which denotes a simple nonlinear system, in equation 5.1 is taken from [15]. This system has two equation and two unknowns that are symbolized as $x_1(t)$ and $x_2(t)$. $b(t)$ is the independent source of system and is a simple cosine function as seen in (5.2).

$$\begin{bmatrix} \frac{d}{dt}x_1(t) + x_1(t) + 1000\big(x_1(t) - x_2(t)\big)^3 - b(t) \\ \frac{d}{dt}x_2(t) + x_2(t) - 1000\big(x_1(t) - x_2(t)\big)^3 \end{bmatrix} = 0 \tag{5.1}$$

$$b(t) = \cos(2\pi t) \tag{5.2}$$

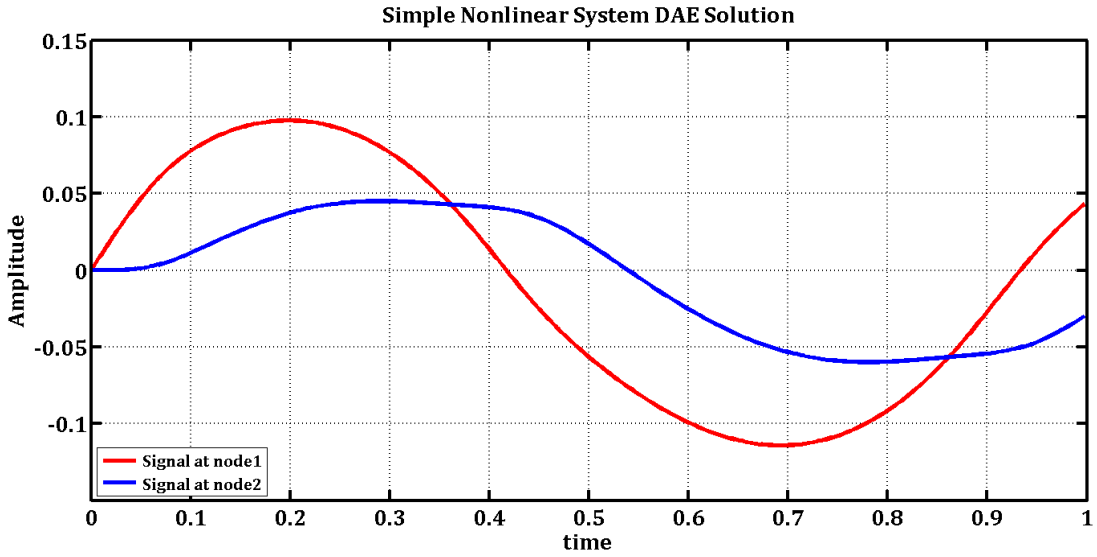

Figure 5.14: Simple nonlinear system transient solution with 'ode15s'.

Transient solution for first period is illustrated in figure 5.14 and PSS solution in figure 5.15. These results are closely resemble with Gu and Rowchudry have found in [15].

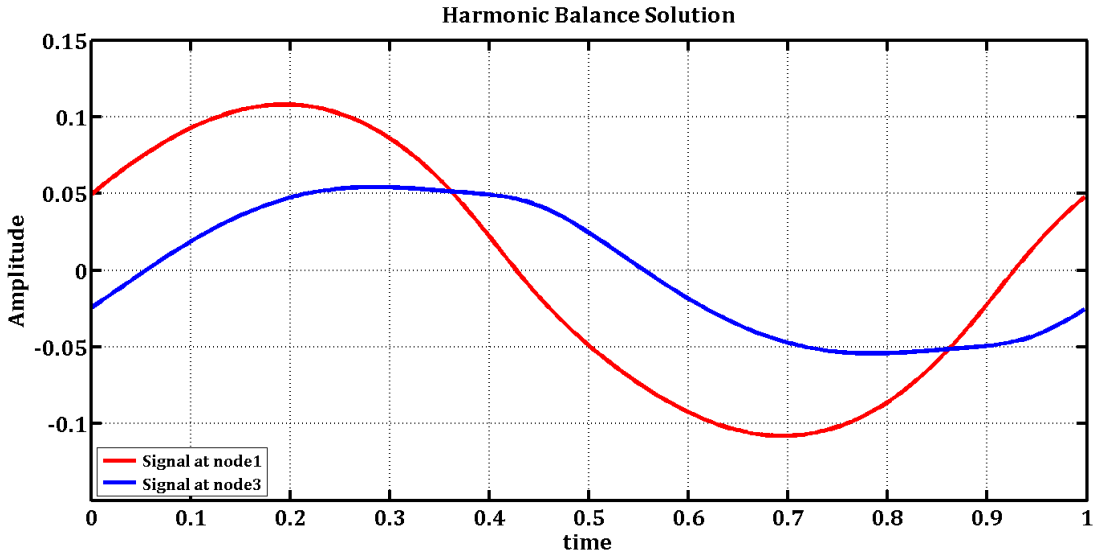

Figure 5.15: Simple nonlinear system PSS analysis with our software.

5.5 A Firing Neuron Model

As aforementioned, nonlinear equation set, which denotes a firing neuron model known as Morris-Lecar model, in equation (5.3) is taken from [15]. This system has two equation and two unknowns that are symbolized as $V(t)$ and $N(t)$. $I_{ext}(t)$, which is an injection current, is the input of this system. For other parameters in equation set (5.3), the reader can refer to [15].

$$\begin{bmatrix} \frac{d}{dt}V(t) + \frac{1}{C_\mu}(\rho_L(V(t)-V_L) + \rho_{Ca}M_\infty(V(t)-V_{Ca}) + \rho_K N(t)(V(t)-V_K) - I_{ext}(t)) \\ N(t) + (N - N_\infty)\frac{1}{\tau_N} \end{bmatrix} = 0 \quad (5.3)$$

$$M_\infty = 0.5(1 + \tanh(\frac{V(t)-V_1}{V_2}) \quad (5.4)$$

$$N_\infty = 0.5(1 + \tanh(\frac{V(t)-V_3}{V_4}) \quad (5.5)$$

$$\frac{1}{\tau_N} = \frac{1}{\Phi \cosh(\frac{V(t)-V_3}{2V_4})} \tag{5.6}$$

M_∞, N_∞ and $\frac{1}{\tau_N}$ is expressed in equation (5.4), (5.5) and (5.6) respectively with the purpose of reducing complexity of equation set.

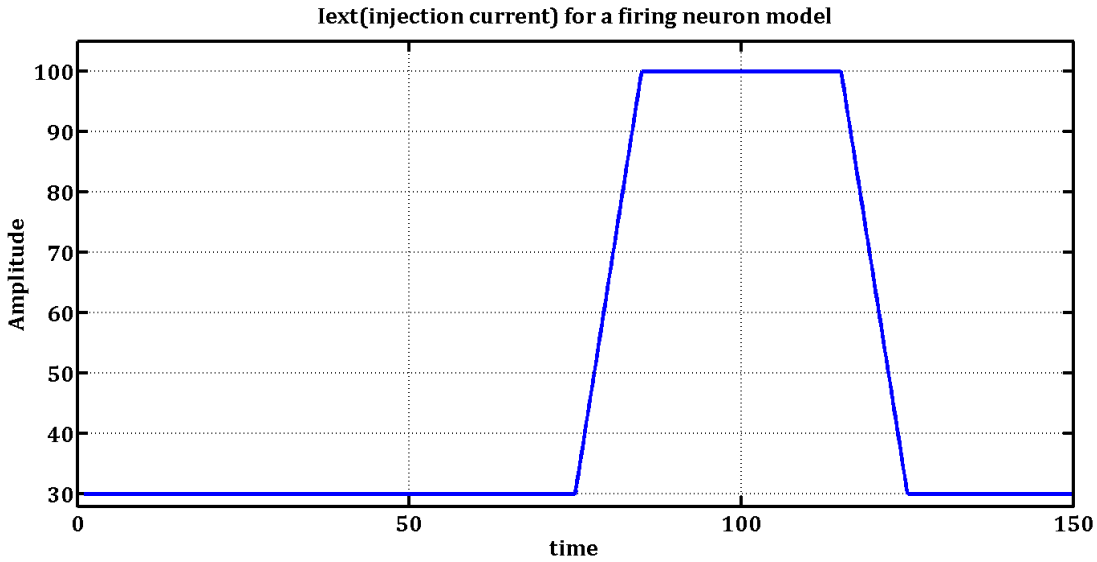

Figure 5.16: Pulse signal as an input to firing neuron model.

Where I_{ext} is the input signal of equation set (5.3) the transient solution and PSS solution are illustrated together in figure 5.17. These results are closely resemble with Gu and Rowchudry have found in [15].

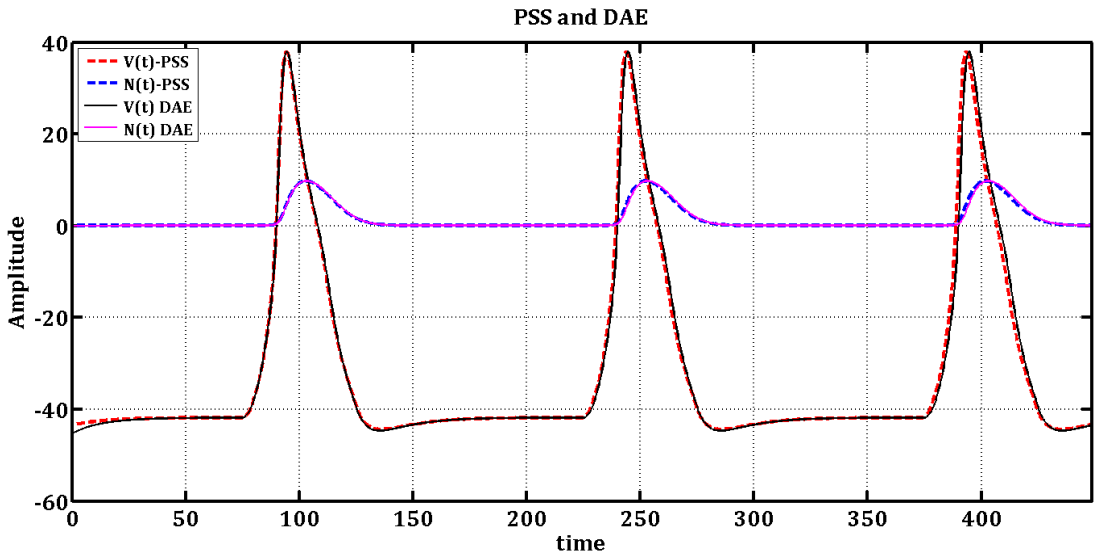

Figure 5.17: PSS and transient solution for firing neuron model.

6. CONCLUSION

In this study mathematical background of harmonic balance method is studied for the purpose of implementing PSS analysis of nonlinear systems. Then algorithms of this mathematical background is tried to be implemented in MATLAB environment according to fundamental principles and techniques of software development.

In this study a system is designed to solve nonlinear systems. For large systems, it is seen that computational complexity is a problem. It is seen that the iteration number is highly dependent on the initial values given to the Newton method. If the initial values are close to solution Newton's Method can easily converge. If the circuit is stiffly nonlinear, this fact also increases the iteration number and calculation time.

For large circuits Jacobian matrix is very large and variable numbers that must be optimized is very large. So this situation increases the computational cost. With the purpose of reducing computational cost this software can be written with C++, which is a faster language, instead of MATLAB. With same purpose better Newton's Method algorithms, which use Krylov subspace methods, can be used as iterative solvers.

- Future Work:

Lawrence Livermore National Laboratory (LNLL), Center for Applied Scientific Computing (CASC) develops software for scientific computing. CASC software includes math libraries and one of them is SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic Equation Solvers). KINSOL is a module of SUNDIALS library and includes a Newton-Krylov solver. My first purpose is using this tool to calculate equation (3.23) with using MEX that is a MATLAB API for C/C++. After this purpose is reached, the second purpose is going towards to Harmonic Balance applications such as distortion analysis and uncertainty quantification analysis for more complicated systems.

REFERENCES

[1] Nakhla M. S., Vlach J., (1976), "A piecewise harmonic balance technique for determination of periodic response of nonlinear systems," IEEE Trans. Circuits Syst., 23 (2), 85–91.

[2] Aprille T. J., Trick T. N., (1972), "Steady-state analysis of nonlinear circuits with periodic inputs," Proc. IEEE, 60 (1), 108–114.

[3] Kundert K. S., Vincentelli A. S., (1986), "Simulation of nonlinear circuits in th frequency domain," IEEE Trans. Comput. Des. Integr. Circuits Syst., 5 (4), 521–535.

[4] Nastov B. O., Telichevesky R., Kundert K., White J., (2007), "Fundamentals of fast simulation algorithms for RF circuits," in Proceedings of the IEEE, 95 (3), 600-621.

[5] Zhang Z., El-Moselhy T. A., Maffezzoni P., Elfadel I. M., Daniel L., (2013), "Efficient uncertainty quantification for the periodic steady state of forced and autonomous circuits," IEEE Trans. Circuits Syst. II Express Briefs, 60 (10), 687–691.

[6] Luo A. C., Huang J., (2012) "Approximate solutions of periodic motions in nonlinear systems via a generalized harmonic balance," J. Vib. Control,.18 (11), 1661–1674.

[7] Wu B., Roychowdhury J., (2014), "Efficient per-element distortion contribution analysis via harmonic balance adjoints," in Proceedings of the IEEE Custom Integrated Circuits Conference, 1–4, San Jose, CA, 15-17 September.

[8] Soveiko N., Nakhla M., (2002), "Wavelet harmonic balance," IEEE Microw. Wirel. Components Lett., 4, 1–11.

[9] Soveiko N., Nakhla M., (2005), "On the steady-state analysis of nonlinear circuits with frequency dependent parameters in wavelet domain," 15 (5), 384–386.

[10] Soveiko N., Gad E., Nakhla M., (2007) "A wavelet-based approach for steady-state analysis of nonlinear circuits with widely separated time scales," IEEE Microw. Wirel. COMPONENTS Lett., 17 (6), 451–453.

[11] Rizzoli V., Masotti D., Mastri F., Montanari E., (2011), "System-oriented harmonic-balance algorithms for circuit-level simulation," IEEE Trans. Comput. Des. Integr. CIRCUITS Syst., 30 (2), 256–269.

[12] Cooley J. W. Tukey J. W., (1965), "An algorithm for the machine calculation of complex fourier series," Math. Comput., 19 (90), 297.

[13] Nastov O. J., (1999), “Spectral Methods for Circuit Analysis,”, Ph. D. Dissertion, Massachusetts Insitute of Technology.

[14] Şuvak Ö., (2008), “Theory and Numerical Methods for the Analysis of Biological and Electronic Oscillators”, Ms. Thesis, Koc University.

[15] Gu C., Roychowdhury J., (2010), “Generalized nonlinear timing/phase macromodeling: theory, numerical methods and applications,” in IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 284–291, San Jose, CA,19-21 September.

BIOGRAPHY

Elif Betül ŞEN ÖZEN was born in 1989 in Samsun. She completed his high school education in Samsun Fen Lisesi in 2007 and graduated from Istanbul Technical University (ITU) Electronics Engineering Department in 2013. She started her M.S. at GTU Graduate School of Natural and Applied Sciences, Electronics Engineering in 2015. She has been working as a research assistant in GTU since 2015.