ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FORMANT-BASED SPEECH SYNTHESIS: A CASE OF AMHARIC WORDS

By: Yibeltal Tafere

A Project report Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

June, 2008

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

FORMANT-BASED SPEECH SYNTHESIS: A CASE OF AMHARIC WORDS

By

Yibeltal Tafere

APPROVED BY

EXAMINING BOARD:

1. Sebsibe H/Mariam, Advisor _____

2. _____ _____

3. _____ _____

Acknowledgements

I would like to take this opportunity to thank all the people who have helped me to complete this project. I would like to thank sincerely my advisor Sebsibe H/Mariam for his invaluable guidance, encouragement and advice throughout this project.

I would like to express my special thanks to Geletaw Sehale, who had put a lot to make this work possible.

I am thankful to Hibrework Yimenu , for his continuous support throughout this work.

I should also thank Tibebe Kassahun for commenting and proof reading the document.

I would like to thank all my colleagues at the Department of Computer Science for their support and knowledge sharing in doing this work.

TABLE OF CONTENTS

List of Tables

List of Figures

Abbreviations

TTS	-	Text-to-Speech
NLP	-	Natural Language Processing
DSP	-	Digital Signal Processing
F0	-	Fundamental Frequency (pitch)
F1	-	First Formant Frequency
F2	-	Second Formant Frequency
F3	-	Third Formant Frequency
LTS	-	Letter-to-Sound
IIR	-	Infinite Impulse Response
LP	-	Linear Predictive
LPC	-	Linear Predictive coefficients
CV	-	Consonant-Vowel
UTF	-	Unicode TransForm

ABSTRACT

Speech synthesis is a process of making artificial speech which mainly requires computers to understand the language speaking rules. Even if there are several techniques of producing synthetic speech, it is still challenging to find one that overcomes all the limitations. One of the speech synthesis techniques is formant synthesis which is based on the well-known source-filter model. This project work has aimed at developing a speech synthesizer for Amharic language words using this technique.

The project started by extracting the resonance which is the most important parameter in formant synthesis and other parameters like formant bandwidth, fundamental frequency (pitch), etc from speech file. At the same time the acoustic parameters of the file were passed to the formant synthesizer to synthesize voiced sounds. The unvoiced sounds were segmented from all Amharic syllables and stored in appropriate place.

In order to develop the system, there are phases that were performed. These are speech analysis, text analysis and synthesis. In the speech analysis phase, recording of the speech and extraction of the acoustic features from the wave files was made. Segmentation process was conducted before extracting the features of the speech. This process was performed using wavesurfer which is a speech analysis tool used for studies of acoustic phonetics.

Therefore, every input text that comes to the system had to be transcribed and produced phonemes from input text. With these phonetic strings, we could generate an artificial speech for the voiced sounds and selecting the consonants with its co-existing context and concatenate these units to synthesize a word. The system provided flexibility of a speech with low memory and data requirements.

CHAPTER ONE

1. INTRODUCTION

Language is a fundamental part of everyday life. Whether we are using speech, sign language, or a coding system that conveys meaning through touch, we use language to express our thoughts, intentions, reactions, and experiences [1].

Speech is one of the oldest and the most widely used means of communication between people and it plays a great role especially in a human-machine interaction. People have extensively studied it and often tried to build machines to handle in an automatic way.

The idea that a machine could generate speech has been with us for some time, but the realization of such machines has only really been practical within the last 50 years. Even more recently, it is in the last 20 years that we have seen practical examples of text-to-speech systems.

Text-to-speech (TTS) synthesis technology gives machines the ability to convert arbitrary text into audible speech, with the goal of being able to provide textual information to people via voice messages. The system takes as input a sequence of words and converts them to speech. Speech synthesis systems are often called text-to-speech (TTS) systems in reference to their ability to convert text into speech. The ultimate goal is to have the best human-like speech quality from the overall system [2].

When an application produces synthesized speech, however, it communicates with users in human terms, in a natural and efficient way. Using speech, an application can communicate an almost infinite range of information to the user. Because it is not limited to producing a small set of sounds, users must learn to associate with specific conditions or actions.

The creation of synthetic speech covers a whole range of processes, and though often they are all lumped under the general term text-to-speech, a good deal of work has gone into generating speech from sequences of speech sounds. This would be a speech sound (phoneme) to audio waveform synthesis, rather than going all the way from text to phonemes, and then to sound [6].

The realization that the speech signal could be decomposed as a source-and-filter model, with the glottis acting as a sound source and the oral tract being a filter, was used to build analog electronic devices that could be used to mimic human speech [1].

The prediction of parameters that compactly represent the signal, without the loss of any information critical for reconstruction, has always been, and still is, difficult. Early versions of formant synthesis allowed these to be specified by hand, with automatic modeling as a goal.

Today, formant synthesizers can produce high quality, recognizable speech if the parameters are properly adjusted, and these systems can work very well for some applications. It is still hard to get fully natural sounding speech from these when the process is fully automatic as it is for all synthesis methods.

Even if there is a great advancement on the tools used for developing speech synthesizer for different languages in the global world, very few works were tried for local languages like Amharic. Therefore our target will be to build speech synthesizer for Amharic words that will produce a recognizable speech using formant synthesis technique.

1.1. Statement of the Problem

In the past decade, the performance of automatic speech processing systems like speech synthesizers have improved dramatically, resulting in an increasingly widespread use of speech technology in real-world scenario. Whenever we are choosing to develop a speech synthesizer for

Amharic language words, we are considering the following most important predictors [6].

- the current and future number of Amharic language speakers
- the economic and political potential of the countries where this language is spoken
- the information technology needs of the population, etc.

Even though the technology like TTS is growing from time to time to make things easy, people who are living in developing countries like Ethiopia are unable to utilize it. This is because of the difficulty to find and use these products easily and lack of knowledge about the foreign language. Therefore, Amharic is one of the local languages that require language analysis so as to allow easy access to these TTS applications. For example, it is very hard for visually impaired people to navigate computer prompts, unless they are supported with some sort of Amharic voice.

Therefore, this project focused to develop Amharic word synthesizer that will play a great role in supporting human-machine interaction in the area of TTS and solve the problems that exhibit the teaching of the language.

The reason for choosing a word level synthesis and employing formant-based technique is described as follow.

It is impossible to create a synthesized speech for a phrase, clause or a sentence of any language, unless words are synthesized. This is because, words:

- are units of the language that carry meaning
- have phonetical value and consist of one or more morphemes
- used to create phrases , clauses and sentences

Formant speech synthesis techniques is employed because

- it is not computationally intensive process compared to other synthesis techniques.
- it needs very small memory as compared to other techniques.
- it has flexible nature.

1.2. Significance of the Project

The project would have a lot of significance in different areas. Some of them are listed below.

- Automatic telephone-based inquiry systems for every organization

  TTS applications play a great role in communication services. In these systems, textual information can be accessed over the telephone. Mostly, they are used when the requirement of interactivity is little and texts range from a word to simple messages.

- Computer based language teaching

  Speech Synthesizers are helpful to learn a new language known as computer aided learning system. In addition, they are very useful to teach kids with new languages.

- Automatic document reading for the blind

  Blind people can benefit from the synthesizer since it gives them access to written information.

- To synthesize different types of voices in animated movies and self-learn multimedia education packages in Amharic

1.3. Objective of the Project

The general objective of this project is to develop a speech synthesizer for Amharic words using formant based technique.

The specific objectives are:

- Collecting acoustic data about Amharic syllables from a given utterance and extract features appropriate for parameterization.
- Identifying and extracting phonetic feature of syllables.
- Synthesizing Amharic words using the selected parameters.

1.4. Scope of the Project

The focus of this project is to develop a speech synthesizer for Amharic words only using formant based technique.

1.5. Methods and Tools

Different speech synthesis techniques have been studied to identify features that would be applicable for Amharic and other languages in order to choose the technique for this project.

The tools used and main activities that were conducted to achieve the objectives of this project were the followings.

- Appropriate and representative parallel Amharic corpus (text versus acoustic) data were collected to analyze the Amharic phonemes.
- Speech files were segmented into phonemes in order to extract the acoustic parameters to all voiced sounds from the wave file.

- Wavesurfer and colea, a speech analysis tools, were used to analyze the speech and extract the formants and other relevant information from the speech file.
- Matlab 7.0 software was used to develop the speech synthesizer.

1.6. Organization of the Report

This project report contains seven chapters including this chapter. Chapter two presents review of related literature and research works which were conducted previously about speech synthesis on Amharic language. In chapter three, analysis of Amharic language is presented. In chapter four, the requirement analysis of the system is presented. In chapter five, the design and system development environment of the system is discussed. In chapter six, the implementation of the system i.e., presentation of the speech synthesizer is described and finally conclusion and future works are stated.

CHAPTER TWO

2. LITERATURE REVIEW

Speech Technology, one of the major areas in language technology mainly focuses in speech synthesis and speech recognition. Speech synthesis can be described in simple words as a machine speaking to people. This mainly requires computers to understand the language speaking rules. TTS synthesizers belong to this category. Speech recognition can be considered as people speaking to machine. This requires computers to understand the speech. Speech-to-text systems belong to this category [10].

In this project work, the speech synthesis component of speech technology has been considered with reference to Amharic language as required.

2.1. Speech Synthesis

Speech synthesis is the process of converting a written text into speech and this technology gives machines the ability to convert arbitrary text into audible speech, with the goal of being able to provide textual information to people via voice messages.

The major purposes of speech synthesis techniques are to convert a chain of phonetic symbols into artificial speech, to transform a given linguistic representation and to generate speech automatically with information about intonation and stress i.e. prosody [8].

In TTS systems, the process of converting written text into speech contains a number of steps. In general, TTS system contains two components: the Natural Language Processing (NLP) and the Digital Signal Processing (DSP) [7].

NLP is targeted to produce phonetic transcription of the text, together with the desired intonation and rhythm. The DSP transforms the symbolic information it receives from the NLP module into speech. These components are discussed in the following section.

2.1.1. The Natural Language Processing component

The NLP component consists of three processing stages: text analysis, automatic phonetization and prosody generation [7].

Text Analysis

The first stage of natural language processing, the text analysis, consists of four modules: pre-processing module, morphological analysis module, contextual analysis module and syntactic-prosodic parser. Each of these modules will be discussed in the following section.

In a pre-processing module, the input sentences are organized into lists of words. The system must first identify these words or tokens in order to find their pronunciations. In other words, the first step in the text analysis is to make chunks out of the input text - tokenizing it. There are many tokens in a text that appear in a way where their pronunciation has no obvious relationship with their appearance such as abbreviations, acronyms and numbers. Apart from tokenization, normalization is needed where a transformation of these tokens into full text is done.

In a morphological analysis module, all possible part-of-speech categories for each word are proposed on the basis of their spelling. For example inflected, derived and compound words are decomposed into their morphs by simple grammar rules.

A contextual analysis module considers word in their contexts. This is important to be able to reduce possible part-of-speech categories of the word by simple regular grammars by using lexicons of stems and affixes.

In a syntactic-prosodic parser, the remaining search space is examined and the text structure is found. The parser organizes the text into clause and phrase like constituents. After that the parser tries to relate these into their expected prosodic realization [8].

Phonetization

The second module is the Letter-to-Sound module (LTS), where the words are phonetically transcribed. In this stage, the module also maps sequences of grapheme into sequences of phoneme with possible diacritic information, such as stress and other prosodic features that are important to fluency in naturally sounding speech.

Prosody Generator

The last module, the prosody generator, is where certain properties of the speech signal such as pitch, loudness and syllable length are processed. Prosodic features create segmentation of the speech chain into groups of syllables. This gives rise to the grouping of syllables and words in larger chunks [7].

2.1.2. The Digital Signal Processing Component

Intuitively, the operations involved in the DSP component are the computer analogue of dynamically controlling the articulatory muscles and the vibratory frequency of the vocal folds so that the output signal matches the input requirements [8].

In order to do it properly, the DSP component should obviously, in some way, take articulatory constraints into account, since it has been known for a long time that phonetic transitions are more important than stable states for the understanding of speech [9].

Computers made it possible to utilize speech synthesis for practical purposes, and several systems

with the function of converting text to speech were developed. TTS systems perform a range of processes, from text normalization, pronunciation, and several aspects on symbolic and acoustic prosody, finally generating speech at the last step [2].

This, in turn, can be basically achieved in two ways.

- Explicitly, in the form of a series of rules which formally describe the influence of phonemes on one another;
- Implicitly, by storing examples of phonetic transitions and co-articulations into a speech segment database, and using them just as they are, as ultimate acoustic units i.e. in place of phonemes.

2.2. Speech Synthesis techniques

Even if it is hard to find the best method that satisfies the naturalness of concatenative technique and intelligibility of the formant method, there exist several methods to synthesize speech. Each method falls into one of the following categories: articulatory synthesis, concatenative synthesis, and formant synthesis [3].

2.2.1. Articulatory Synthesis

Articulatory synthesis also mathematically models speech production, but models the speech production mechanism itself using a complex physical model of the human vocal tract. Articulatory synthesizers model human speech production mechanisms directly rather than the sounds generated; in some cases they might give more natural sounding speech than formant synthesis. They classify speech in terms of movements of the articulators, the tongue, lips and velum, and the vibrations of the vocal cord. Text to be synthesized is converted from a phonetic

and prosodic description into a sequence of such movements and the synchronizations between their movements calculated. A complex computational model of the physics of a human vocal tract is then used to generate a speech signal. Articulatory synthesis is a computationally intensive process and is not widely available outside the laboratory [2].

2.2.2. Concatenative Synthesis

This speech synthesis technique uses actual snippets of recorded speech that were cut from recordings and stored in an inventory called voice database, either as waveforms (uncoded), or encoded by a suitable speech coding method. Elementary units i.e., speech segments are, for example, phones (a vowel or a consonant), or phone-to- phone transitions (diphones) that encompass the second half of one phone plus the first half of the next phone (e.g., a vowel-to-consonant transition). Concatenative synthesis itself then strings together (concatenates) units selected from the voice database, and, after optional decoding, outputs the resulting speech signal. Because concatenative systems use snippets of recorded speech, they have the highest potential for sounding "natural"[3].

This method is very suitable for some announcing and information system applications. However, it is quite clear that we can not create a database of all words and common names in the world. Most literatures suggested that it is even inappropriate to call this as a method of speech synthesis, as it contains only recordings. This approach usually needs larger memory to store the recorded sounds. The problem is not only in the size of the database, but the output from such system is also limited to one speaker and one voice [21].

2.2.3. Formant Synthesis

Formant synthesis systems synthesize speech using acoustic models of speech production. This

means that they model the speech spectrum and its changes in time as we speak, rather than model the production mechanisms themselves. Parameters such as pitch, voicing, and noise levels are varied over time to synthesize speech waveforms. This method is also called parametric based synthesis. Formant synthesis uses a relatively simple system to select from a small number of parameters, with which to control a mathematical model of the speech sounds. A set of parameters is picked for each speech sound and they are then joined up to make the speech. This stream of parameters is so turned into synthetic speech using the model [2].

Formant synthesis systems are sometimes referred to as synthesis-by-rule systems or more usually formant synthesizers. Commercial TTS engines using formant synthesis have been around for many years. DecTalk, Apollo, Orpheus and Eloquence are well known TTS engines that use formant synthesis [4].

The strength of formant synthesis is its relative simplicity and the small memory footprint needed for the engine and its voice data. This can be important for embedded and mobile computing applications. Formant synthesis generates highly intelligible, but not completely natural sounding speech [4].

2.2.4. Mathematical Description of formant Synthesizers

There are two basic structures in formant synthesis: cascade and parallel. But, for better performance some kind of combination of these is usually used. A cascade formant synthesizer, as shown in Figure 2.1, consists of band-pass resonators connected in series and the output of each formant resonator is applied to the input of the following one [21].

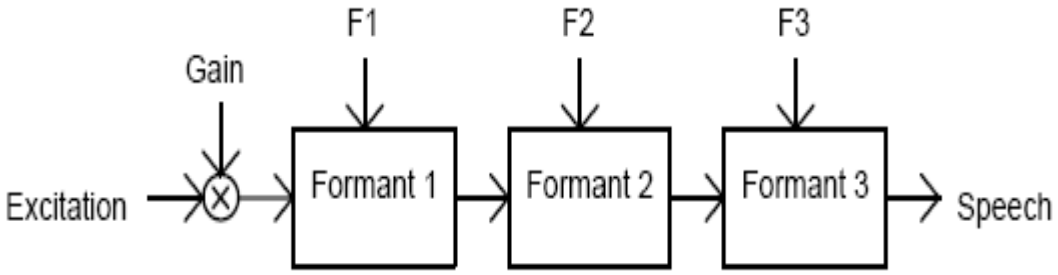

Figure 2.1: Basic structure of the cascade formant synthesizer [21].

A parallel formant synthesizer as shown in Figure 2.2 consists of resonators connected in parallel. Sometimes extra resonators for nasals are used. The excitation signal is applied to all formants simultaneously and their outputs are summed. Adjacent outputs of formant resonators must be summed in opposite phase to avoid unwanted zeros or anti-resonances in the frequency response. The parallel structure enables controlling of bandwidth and gains for each formant individually and thus needs also more control information [21].

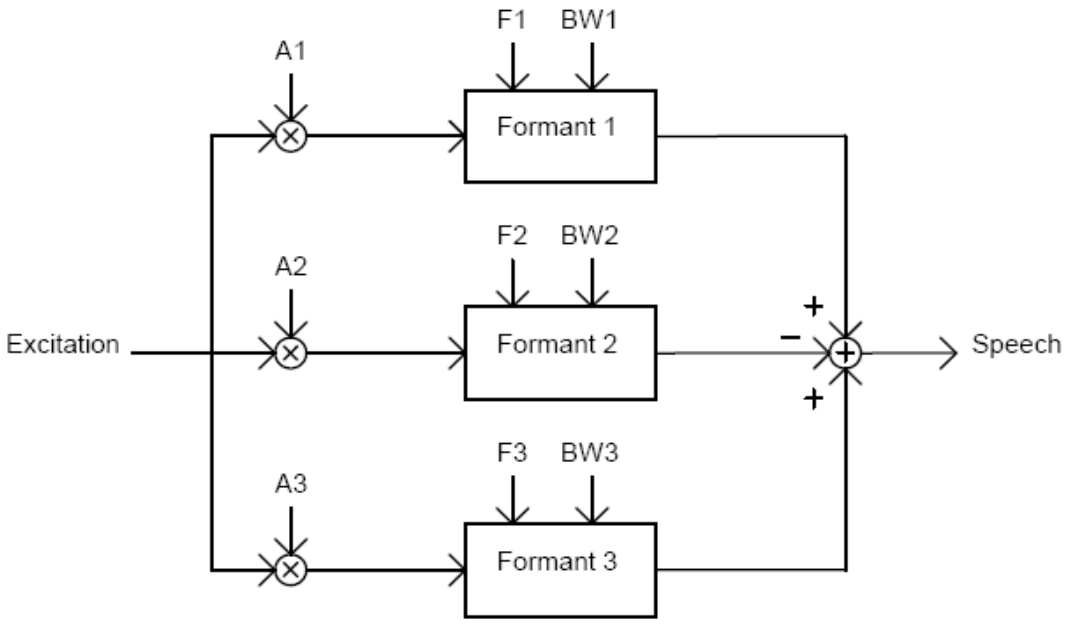

Figure 2.2: Basic structure of the parallel formant synthesizer [21]

A parallel formant synthesizer allows for the direct control over formant amplitudes and sums the outputs of the simultaneously excited formant resonators. But, it is not as good as accurate acoustic imitation of vocal tract behavior in speech as cascade. Parallel synthesizers are better adapted at producing consonants, but some vowels can not be modeled with parallel formant synthesizer [21].

Formant synthesis uses a set of parameters to synthesize the speech. The model explicitly represents a number of formant resonances (usually from two to six) and this formant resonance can be implemented with a second-order IIR (Infinite Impulse Response) filter[1] as shown in equation (2.1). The H_i are the formant resonance where $1 < i < 6$.

$$H_i(z) = \frac{1}{1 - 2e^{-\pi b_i}\cos(2\pi f_i)z^{-1} + e^{-2\pi b_i}z^{-2}} \qquad (2.1)$$

With $f_{i=}F_i/F_s$ and $b_{i=}B_i/F_{s,}$

Where Fi *is* the formant's center frequency

B_i *is* formant's bandwidth,

F_s is sampling frequency

Z is a point in a complex plane (z-transform plane).

The point where H(z) becomes zero is called the poles of the transfer function and can be represented as $z=re^{j\theta}$. Poles and zeros exist in complex conjugate and hence $z^*=re^{-j\theta}$ is also a pole. θ defines the formant frequency and r defines the bandwidth of the transfer function.

[1] IIR is a property of signal processing systems and has an impulse response function which is non-zero over an infinite length of time.

Note that, $re^{+j\theta}$ is within the unit circle for all stable systems[2], which is shown in figure 2.3.

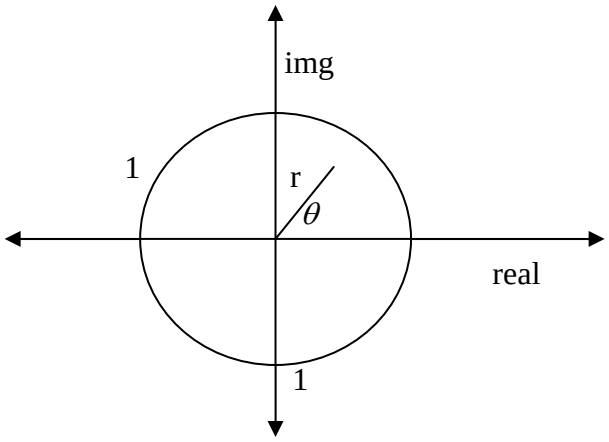

Figure 2.3: unit circle

The formants obtained from this equation are important input in the synthesis process. A filter with several resonances can be constructed by cascading several such second-order sections (cascade model) or by adding several such sections together (parallel model) [22].

Unlike the cascade model, the parallel model requires gains (G) to be specified for each second-order section, which are chosen proportional to the formant's frequency and inversely proportional to the formant's bandwidth. So equation (2.1) will be represented in the following manner.

$$H_i(z) = \frac{G}{1 - 2e^{-\pi b_i}\cos(2\pi f_i)z^{-1} + e^{-2\pi b_i}z^{-2}} \qquad (2.2)$$

2.2.5. Formant Extraction Techniques

Formants, the resonance of the vocal tract, are the key parameters in the synthesis of the speech. The frequency at which they occur is said to be the formant frequency. The equation of the

[2] Stable system is a system that all nearby initial conditions remains nearby of an equilibrium point.

transfer function that defines formant frequencies and its bandwidth $H_i(z)$ is shown below.

$$H_i(z) = \frac{G_i}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

$$H_i(z) = \frac{G_i}{1 - \sum_{k=1}^{2} a_k z^{-k}} = \frac{G_i}{(1 - re^{j\theta} z^{-1})(1 - re^{-j\theta} z^{-1})} \qquad 2.3$$

This equation describes a single pole equation ($z = re^{j\theta}$). However, a speech signal has multiple formants. An all model of such a formant can be represented by a transfer function

$$H_i(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \qquad 2.4$$

This equation is exactly identical to the transfer function of linear predictive coding which states the n^{th} speech sample is a linear combination of past p speech signals plus the current input, where p is the order of the prediction.

Mathematically

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Gx[n]$$

$$y[z] = \sum_{k=1}^{p} a_k y[z] z^{-k} + GX[z]$$

$$y[z] - \sum_{k=1}^{p} a_k y[z] z^{-k} = GX[z]$$

$$y[z](1-\sum a_k y[z]z^{-k}) = GX[z]$$

$$\frac{y[z]}{x[z]} = h[z] = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} \qquad (2.5)$$

Where, G is the gain factor and a_k are the LP (Linear Prediction) coefficients and p is the order of the linear prediction, where $0 \leq k \leq P$. In Linear Prediction, we assume a signal s[z] to be "predicted" from a linear combination of its past values and an input x[n].

The Linear Predictor is used to separate speaker dependent information such as vocal-tract length, pitch, formant frequencies from speech. These information's are extracted by using different methods and two of them are discussed in the coming section [23].

Spectral peak picking method

The spectral peak picking methods and its variants have been widely used for a long time in formant extraction. Usually it requires low computational complexity, but they often seriously suffer from the peak merger problems [23], where two adjoining formants are identified into a single one. From the LP coefficients it is possible to construct the inverse filter of H(z) (prediction error) as follows:

$$H^{-1}(z) = A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \qquad (2.6)$$

This inverse filter provides the spectral shape (carrier signal) given unit impulse as an input.

The vocal tract filter shape in Eq. (2.1) can be obtained from the prediction error using filter Eq.

(2.6). The spectrum obtained by the above-mentioned procedure, usually called LP spectrum. As the name suggests, this type of formant extractors tries to find resonances on the spectrum. Spectral peak picking methods are advantageous in that they show relatively reliable results and do not require much computation.

Root extraction method

In this method, like the spectral peak picking method, we first compute linear prediction coefficients and obtain the prediction-error filter A(z). Comparing with Eq. (2.3), we can easily find that the roots of this polynomial A(z) correspond to the poles of the vocal tract system. Thus, we can obtain candidates for formants by solving A(z) = 0, using numerical methods.

When poles are kept sufficiently apart, and one of these poles, $z = r_0 e^{j\phi_0}$ forms a formant, the formant frequency F, and the formant bandwidth B can be represented by the following equations:

$$F = \frac{f_s}{2\pi}\phi_0, \tag{2.7}$$

$$B = -\frac{f_s}{\pi}\ln(r_0), \tag{2.8}$$

Where r_0 is the magnitude of the pole, φ_0 is the phase of the pole, f_s is the sampling frequency, F is the formant frequency, and B is the 3-dB formant bandwidth. Thus, if we find the roots of the prediction-error polynomial, we can obtain the formant frequencies using Eq. (2.7). In addition, we can get the bandwidth information from Eq. (2.8).

Studies show that, root extraction method is not as reliable as the spectral peak picking method and obtaining roots of A(z) requires very high computational complexity [23]. So, in most cases, this method is not used in real-time implementation, except for research purposes.

2.2.6. Formant-based Speech Synthesis on Amharic Language

Many formant synthesizers were done for different languages like English, French, Finnish, etc and it shows promising result. When we consider the Amharic language, very few works were done on speech synthesis before and some of them are the following. A thesis work which were done by Laine Birhane [20], TTS system for Amharic language, and a project is also done by Habtamu Taye[19] on speech synthesis for Amharic language using the same technique., i.e., concatenative technique.

Even if these are some of the works done using concatenative approach, there is no significant research work conducted using formant based speech synthesis techniques except one i.e., formant based speech synthesis for Amharic vowels by Nadew Tademe. He has introduced a method of formant based speech synthesis for Amharic vowels. A synthesizer is developed that produces vowels according to their context in a given word. The technique models the human speech production system in the form of source and filter, in which the source is completely independent from the filter. The source is identified by the air flow from the lung to vocal cord and the filter represents the resonance of the vocal and nasal tracts, which are also called the formant that changes from time to time. The resonance is due to the constriction of the vocal tract while generating different sounds. Finally, the formant synthesizer provides high flexibility, due to the potential of adjusting any of the acoustic parameters during run time. This potential has enabled us to produce vowels from different contexts [5].

CHAPTER THREE

3. ANALYSIS OF AMHARIC LANGUAGE

3.1. Introduction

Amharic is the official language of Ethiopia. It belongs to the Semitic language family that has the largest number of speakers after Arabic and is characterized by a quite homogeneous phonology distinguishing between 234 distinct Consonant-Vowel (CV) syllables [11].

Amharic uses a unique script, which has originated from ancient language, the Ge'ez alphabet, which is the liturgical language of the Ethiopian Orthodox Church. Written Ge'ez can be traced back to at least the 4th century AD. The first versions of the language included consonants only, while the characters in later versions represent consonant-vowel (CV) phoneme pairs [18].

The script of Amharic language is phonetic in nature. It has 32 consonants and 7 vowels. The orthographic representation of the language is organized into orders. Each of the 32 consonants has seven orders (derivatives). Six of them are CV combinations while the seventh is the consonant itself. For each consonant C, the orthographic ordering is as follows [17]:

C/e/ C/u/ C/i/ C/a/ C/ie/ C C/o/.

Unlike the orthographic representation, Amharic language has one special property in its spoken form (CV sequence of the acoustic form of the orthographic representation). The sixth order orthographic symbols, which do not have any vowel unit associated to it in the written form (CV transcription of the orthographic form), may associate the vowel /ix/ in its spoken form which has important role during syllabification of the word in the language which allows splitting impermissible consonant clusters [18].

Like other languages, Amharic has its own characterizing properties. For example, Amharic has a set of speech sounds that are not found in other languages, for example English. These are the glottalized plosives (å,Ø, p, ß and ê) which have a sharp click-like character [13].

3.2. Amharic Language Syllable and Phoneme

Amharic words use consonantal roots with vowel variation expressing difference in interpretation. In modern written Amharic, each syllable pattern comes in seven different forms (called orders), reflecting the seven vowel sounds. The first order is the basic form; the other orders are derived from it by more or less regular modifications indicating the different vowels. There are 33 basic forms, giving 7 * 33 syllable patterns (syllographs), or fidels [18].

Although Amharic is a Semitic language like Hebrew and Arabic, its writing is a syllabic left-to-right script [18]. The basic Amharic alphabets are shown in Table 3.1 with their derived forms. They are organized in the form of 33 rows by 7 columns table. Characters in the first column are called First (First order), the remaining columns are also labelled as: Second, Third, Fourth, Fifth, Sixth and Seventh according to their position in the table. Another naming system is still available. The first column characters are called Ge'ez (Ó°´) that is to say first in Ge'ez. Similarly the remaining characters are named as Ka'ib (Ÿ®w), Sali's (dMe), Rab'i (^w°), Hami's (HUe), Sadi's (dÉe) and Sab'i (dw°) respectively [14].

Table 3.1: Amharic alphabets with their seven orders

First	Second	Third	Fourth	Fifth	Sixth	Seventh
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ደ	ዱ	ዲ	ዳ	ዶ	ድ	ዶ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

A set of 39 phones, seven vowels and thirty-two consonants, makes up the complete inventory of sounds for the Amharic language. A brief overview of each of these major categories of Amharic phonemes is given in the following section [13].

3.2.1. Consonants

Amharic consonants are generally classified as stops, fricatives, nasals, liquids, and semi-vowels [4]. Table 3.2 shows the phonetic representation of the consonants of Amharic as to their manner of articulation, voicing, and place of articulation.

Table 3.2: phonetic representation of the Amharic consonants [17]

		Labials		*Alveolar*		*Palatals*		*Velars*		*Labio-Velar*		*Glottals*	
Stops	Voiceless	p	ፕ	t	ት			k	ክ	kx	ኳ	ax	ዕ
	Voiced	b	ብ	d	ድ			g	ግ	gx	ጓ		
	Glottalized	px	ጵ	tx	ጥ			q	ቅ	qx	ቋ		
Fricatives	Voiceless	f	ፍ	s	ስ	sx	ሽ					h	ህ
	Voiced	v	ቭ	z	ዝ	zx	ዥ						
	Glottalized			xx	ጽ							hx	ኀ
Africatives	Voiceless					c	ች						
	Voiced					j	ጅ						
	Glottalized					cx	ቈ						
Nasals	Voiced	m	ም	n	ን	nx	ኝ						
Liquids	Voiced			l	ል								
				r	ር								
Glides		w	ው			y	ይ						

3.2.2. Vowels

The vowels (· ›< ›= œ ›? and *) are categorized as rounded (›< and *) and un rounded (·, œ ›= and ›?). Another categorization according to the place of articulation is given in table 3.3.

Table 3.3: Categories of Amharic Vowels.

	front	centre	back
high	›=[ii]	›[ix]	›<[u]
mid	›?[ie]	[e]	*[o]
low		œ[a]	

In general, there are about 39 phonemes of Amharic alphabets. These phonemes have been used to contain the acoustics units of the formants and speech file, i.e., (acoustic inventory data) which are then used to get the synthesized speech corresponding to the given word. For this project, we have used synthesis units consisting of CV clusters along with consonants (C) and vowels (V).

3.3. Transcription and Transliteration of Amharic Scripts

For Amharic language, before transcription is made the transliteration (representation of an alphabet with letters from a different alphabet) of each character is made by using the ASCII value of each of the character.

Transliteration is the practice of transcribing a word or text written in one writing system into another writing system or system of rules for such practice. For this work, the transliteration scheme proposed by Sebsibe [17] is adopted. The complete list of this transliteration scheme is show in table 3.4[17].

Transcription is the process of producing phonetic representation of a given word. For example the word Petros / â?Øae / is transcribed as *px ie tx ix r o s*.

Table 3.4: Amharic Phonetic List, IPA Equivalence and its ASCII Transliteration

IPA	Transcription	Amharic equivalence
Consonants		
[p]	[p]	ፐ
[t]	[t]	ተ
[k]	[k]	ከ
[ʔ]	[ax]	ዐ
[b]	[b]	በ
[d]	[d]	ደ
[g]	[g]	ገ
[pʾ]	[px]	ጰ
[tʾ]	[tx]	ጠ
[cʾ]	[cx]	ፈ
[q]	[q]	ቀ
[f]	[f]	ፍ
[s]	[s]	ሰ
[ʃ]	[sx]	ሸ
[h]	[h]	ሀ
[sʾ]	[xx]	ጸ
[tʃ]	[c]	ቸ
[gʾ]	[j]	ጀ
[m]	[m]	መ
[n]	[n]	ነ
[nʾ]	[nx]	ኘ
[l]	[l]	ለ
[r]	[r]	ረ
[j]	[y]	ደ
[w]	[w]	ወ
[v]	[v]	ቨ
[z]	[z]	ዘ
[zʾ]	[zx]	ዠ
Vowels		
[ɛ]	[e]	ኧ
[ʊ]	[u]	ኡ
[ɪ]	[ii]	ኢ
[ɑ]	[a]	ኣ
[e]	[ie]	ኤ
[ɨ]	[ix]	እ
[o]	[o]	ኦ

CHAPTER FOUR

4. SYSTEM REQUIREMENT ANALYSIS

4.1 Introduction

Requirements are descriptions of how a system should behave or a description of system properties. It can alternatively be a statement of ‘what’ a proposed system is expected to do. In the following section, the requirement analysis and different analysis models will be produced and discussed.

4.2 Requirement Analysis

Requirements analysis is the process of understanding the needs and expectations from a proposed system. The Requirement analysis contains both functional and non-functional requirements and those are described in the following section.

4.2.1 Functional Requirements

The developed system is expected to provide different functionalities. First, the system should be able to accept text as an input, and this input text should be transliterated, i.e., changing of a text written in one writing system into another writing system as discussed in section 3.3. The system should also be able to perform phonetiization, i.e., identifying the phonemes (consonants and vowel) of the given text, the system should also be able to retrieve the parameters from the inventory data and finally the system should be able to perform synthesis and produces speech.

4.2.2 Non-Functional Requirement

When designing the system, the following constraints must be taken into account. Those non-functional requirements are discussed below.

The system should take into consideration the time and storage complexity of the system, time/space bounds, i.e., the system should need a very small amount of space and time to process.

The system must be expandable, i.e., the system should support the addition of new functionality without too much effort.

The system must be easy to use, i.e., the system should have very easy interface and easy to learn.

4.3 Analysis Model

To produce a model of the system which is correct, complete and consistent, we need to construct the analysis model which focuses on structuring and formalizing the requirements of the system. Analysis model contains functional, object and dynamic models. The functional model can be described by use case diagrams. Class diagrams can describe the object model. Dynamic model can also be described in terms of sequence, state chart and activity diagrams. For the purpose of this project we have described the analysis model in terms of the functional model and dynamic models using use case, sequence and activity diagrams.

4.3.1 Use case Diagram

Use cases of the system are identified to be “Phonetize”, “FeatureExtract” , “Transliterate”, “selectFeature” and “synthesize”.

“FeatureExtract” and“Transliterate” Use cases are initiated by the user (Actor) of the system. “SelectFeature” initiate the “featuerExtract” and “synthesize” use cases. The “transliteriate” use case initiates the “phonetize” use case. Figure 4.1 depicts the use case diagram of the system.

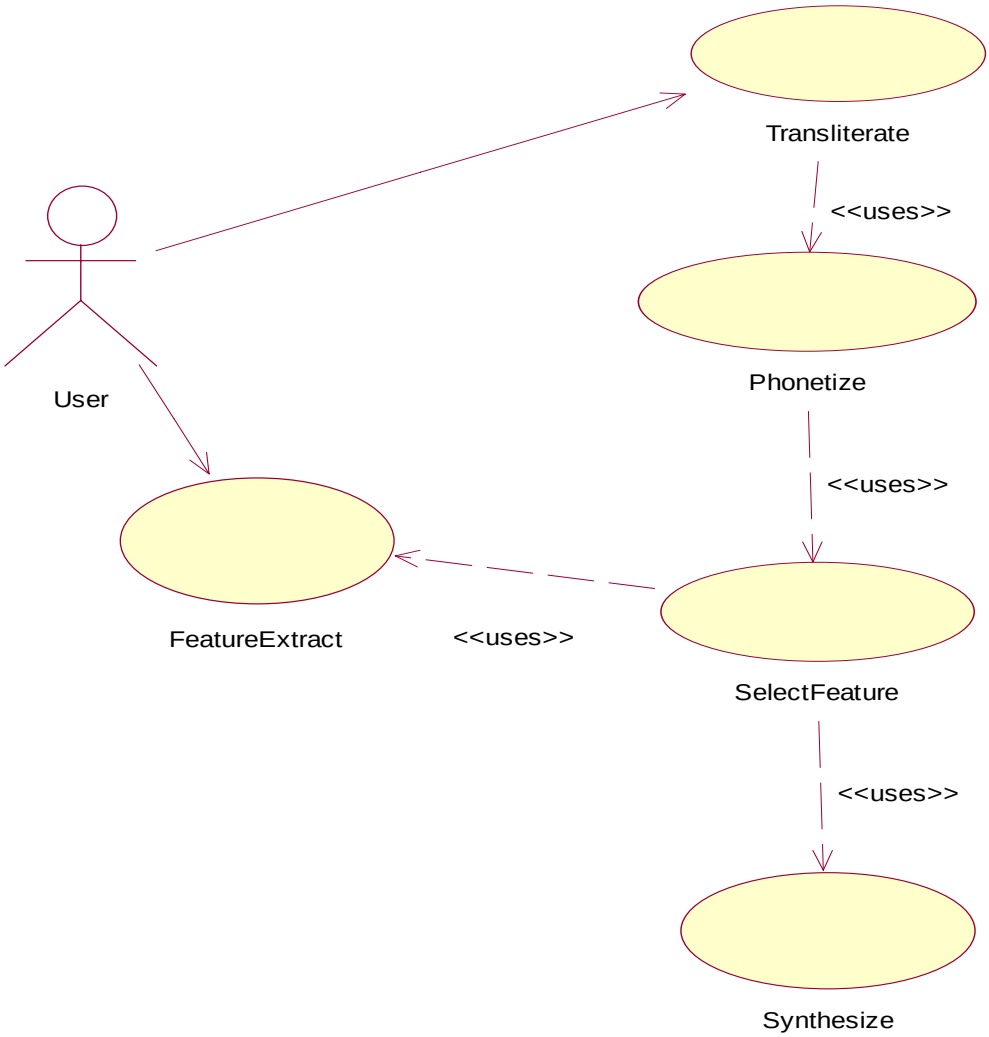

Figure 4.1: Use Case Diagram of the system

4.3.2 Sequence Diagram

Sequence diagrams show the interaction between participating objects in a given use case. They are helpful to identify the missing objects that are not identified in the analysis object model. To see the interaction between objects, the following describe the sequence diagram of each identified use cases.

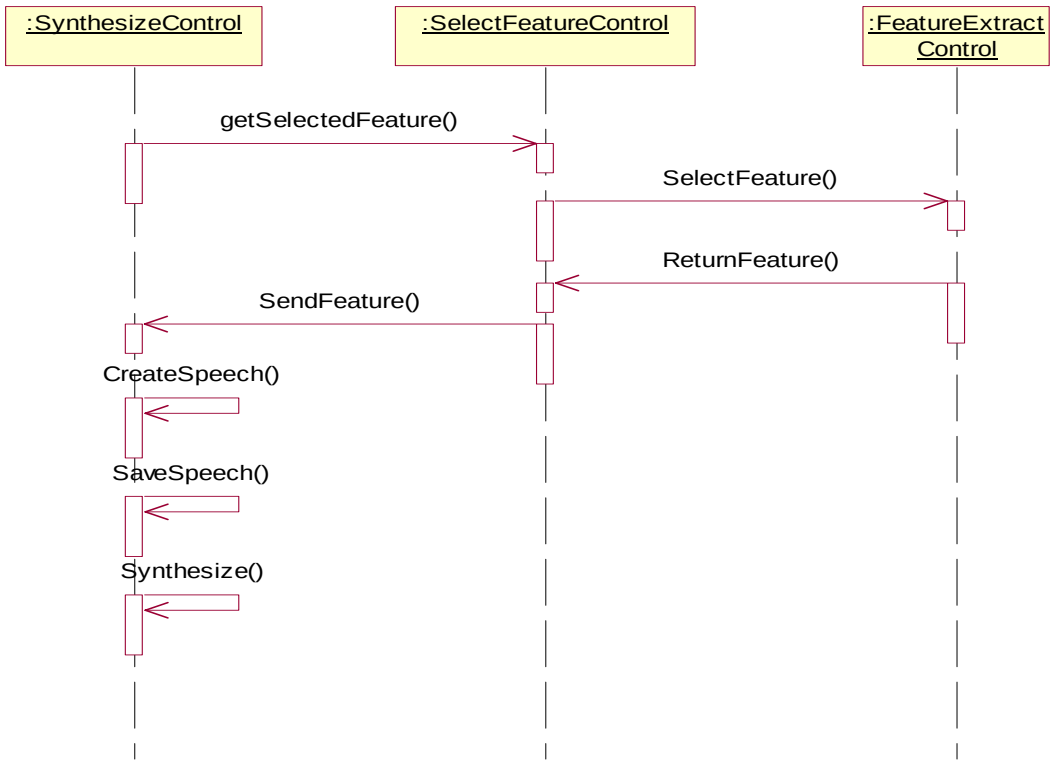

Figure 4.2: Sequence Diagram of synthesize use case

As shown in Figure 4.2, the *"Synthesize"* use case is initiating the *"SelectFeatureControl"* object of *"SelectFeature"* use cases. The *"synthesizeControl"* object receives the extracted feature data from "SelectFeatureControl" object that gets from the "FeatureExtractControl" object. After that, the *"SynthesizeControl"* object will be responsible to create, save, and synthesize speech.

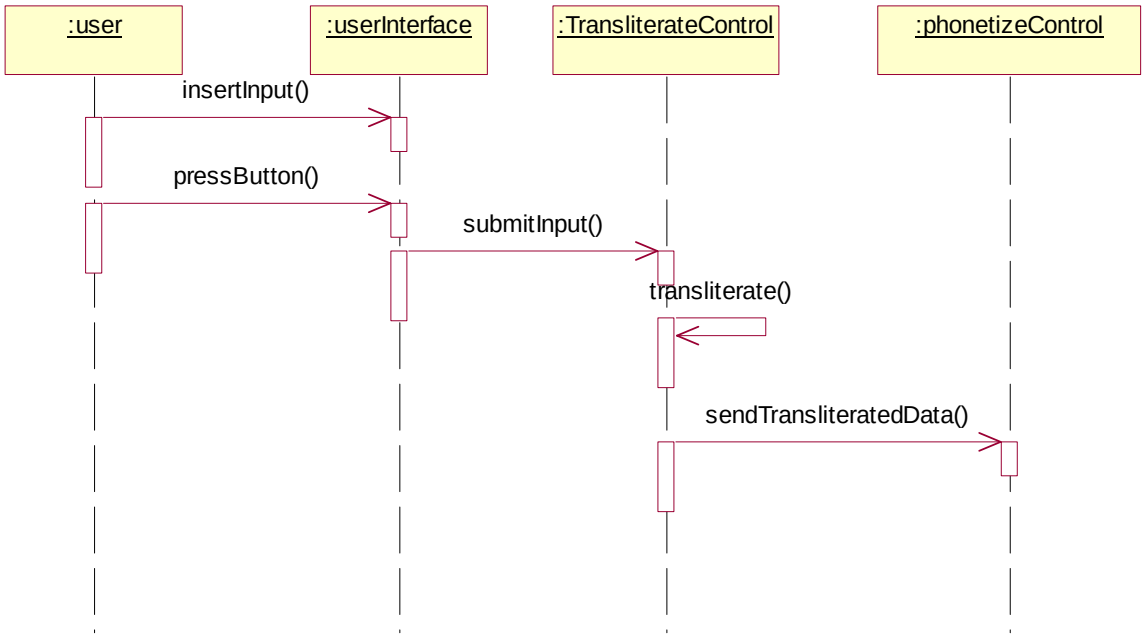

Figure 4.3: Sequence Diagram of Transliterate use case

"Transliterate" use case is initiated when the user inserts a text to the system. After the userInterface is activated by the user, the data will be submitted to *"transliterateControl"* object for processing.

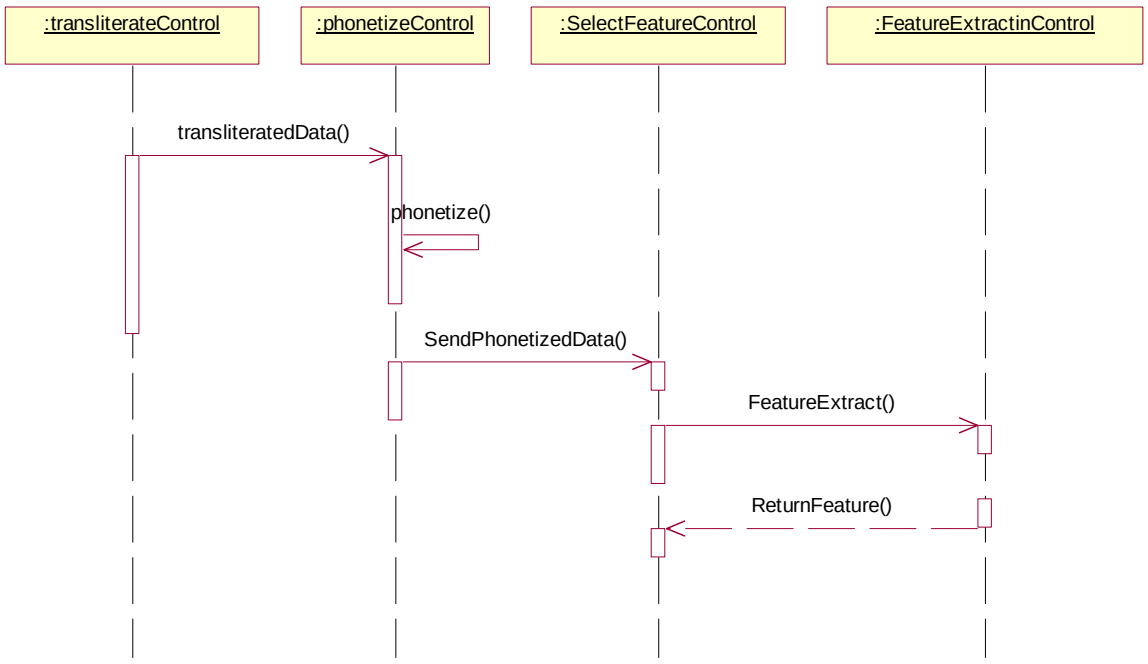

Figure 4.4: Sequence Diagram of phonetize use case

"*transliterateControl*" object initiates the "phonetize" use case. The "phonetizeControl" object initiates the "SelectFeature" use case and the "selectFeatureControl" object manages the feature selection process. Finally, the "FeatureExtractionControl" object returns the selected feature.

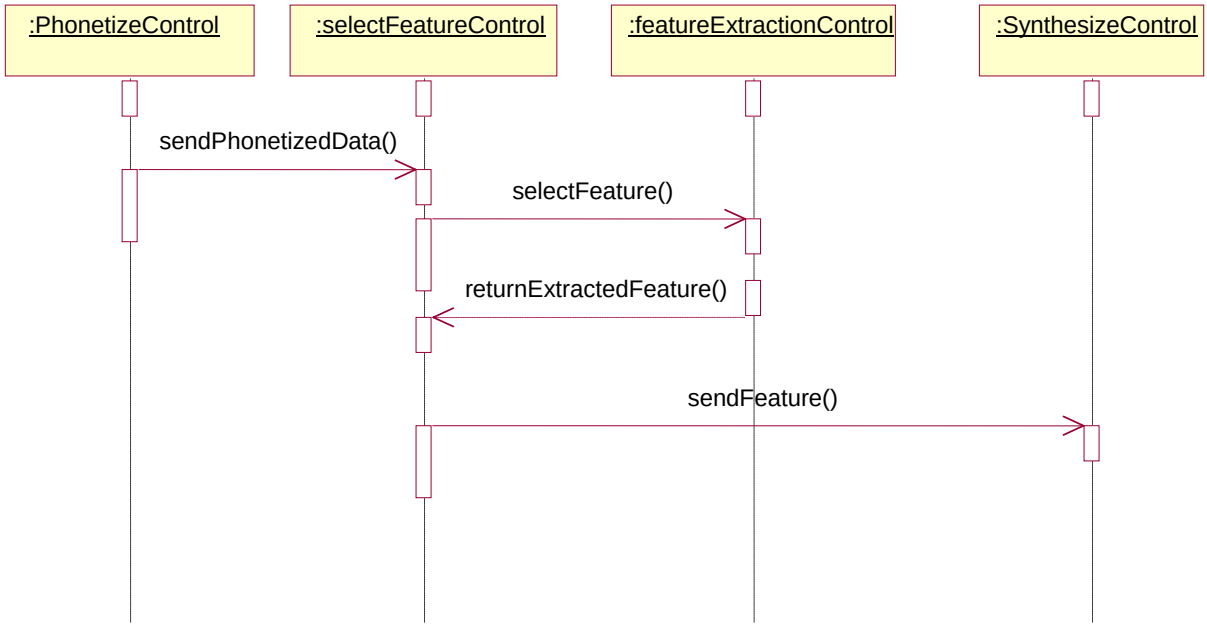

Figure 4.5: Sequence Diagram of SelectFeature use case

The “PhonetizeControl” object initiates the “selectFeature” use case and the “SelectFeatureControl” object initiates the “FeatureExtractControl” object and it manages the feature extraction process. Finally, the “FeatureExtractionControl” object returns to “selectFeatureControl” object.

4.3.3 Activity Diagram

Figure 4.6 shows the activity diagram of the system that can describe the set of operations executed and the order of execution of these operations.

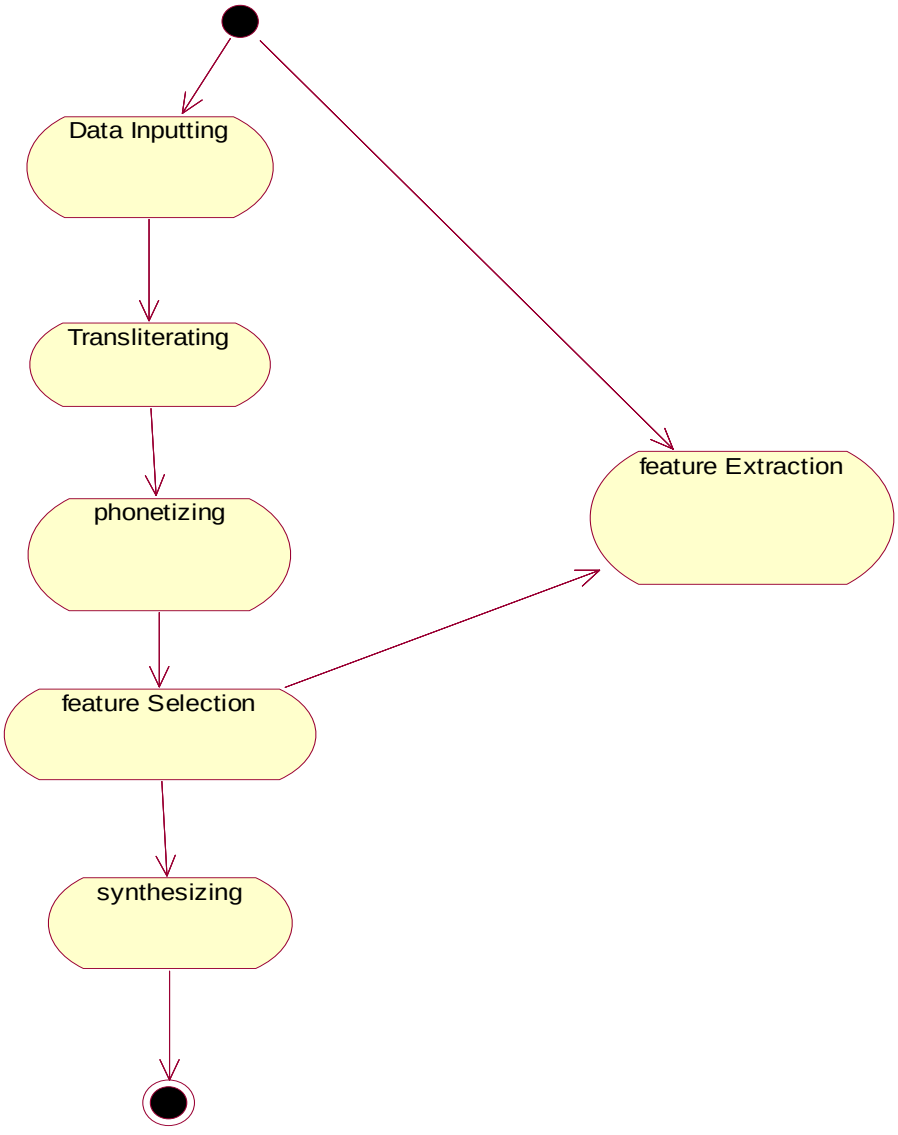

Figure 4.6: Activity diagram of the system

4.4 Subsystem Decomposition

Subsystem decompositions will help to reduce the complexity of the system. The subsystems can be considered as packages holding related classes/objects. The system is decomposed into four subsystems: Language Processor, Feature selection, inventory Data subsystems and synthesizer subsystem. Figure 4.7 shows the identified subsystems and their dependency of the system.

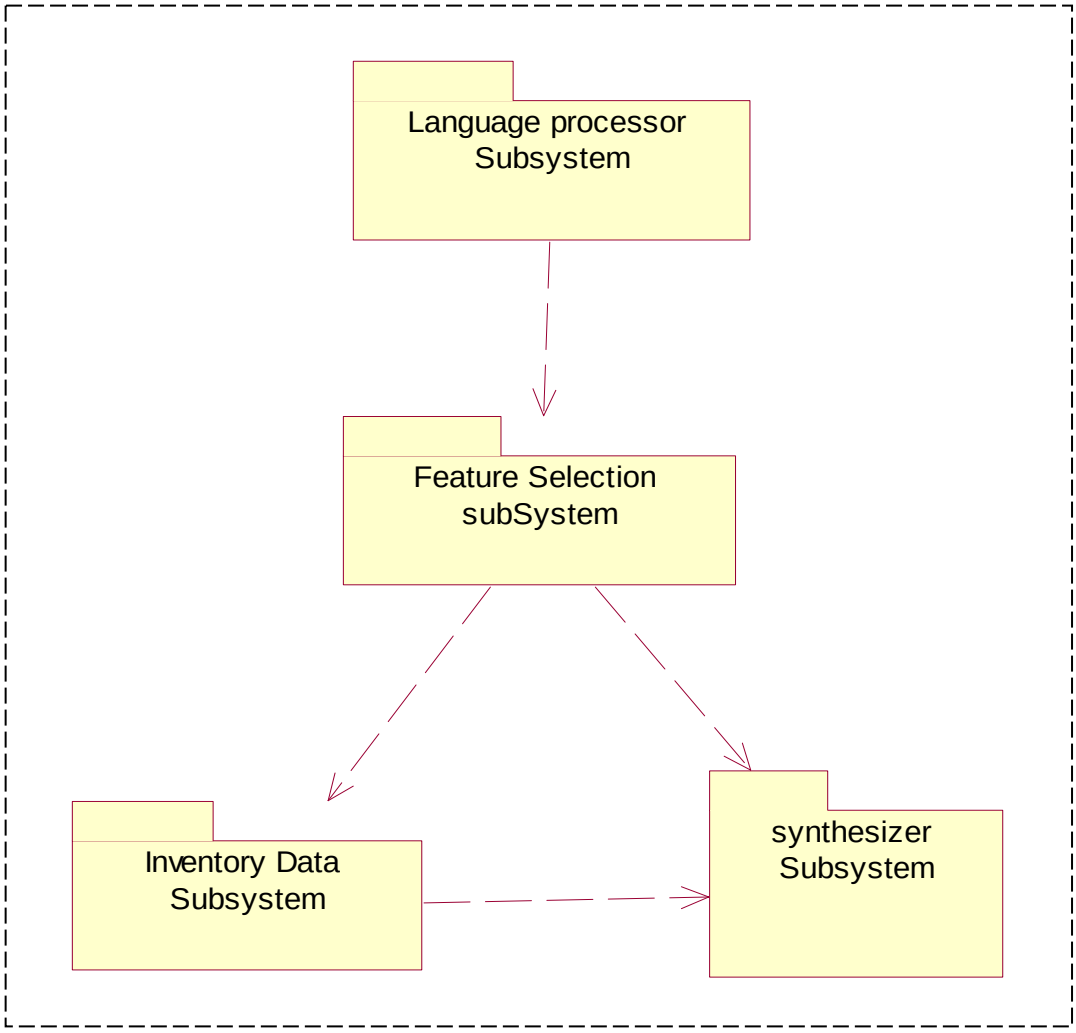

Figure 4.7: subsystems and their dependency of the system

CHAPTER FIVE

5 SYSTEM DESIGN AND DEVELOPMENT ENVIRONMENT

We have identified the requirements of the system and produced the requirement analysis models in chapter four. Based on this analysis, the design goals and architecture of the system is presented in the following section.

5.1 Design Goals

Design goals are used to identify the expected qualities of the system. Most of the design goals of the system are inferred from non-functional requirements. These are discussed below.

Complexity: complexity refers to the time complexity and space complexity of the desired system. The system should need a very small amount of time/space bounds since it should be applied to all types of devices that have even limited CPU and memory resource.

Expandable: the system should be designed to accommodate other additional new functionality without too much effort.

Usability: the system should be designed in such a way that users should find convenient to interact with it.

5.2 Architecture of the Speech Synthesizer

The general architecture of the system is shown on figure 5.1. The architecture shows the three main components of the system. These are the natural language processing component, feature

extraction component and the digital signal preprocessing component. Each of the components will be discussed briefly in the following section.

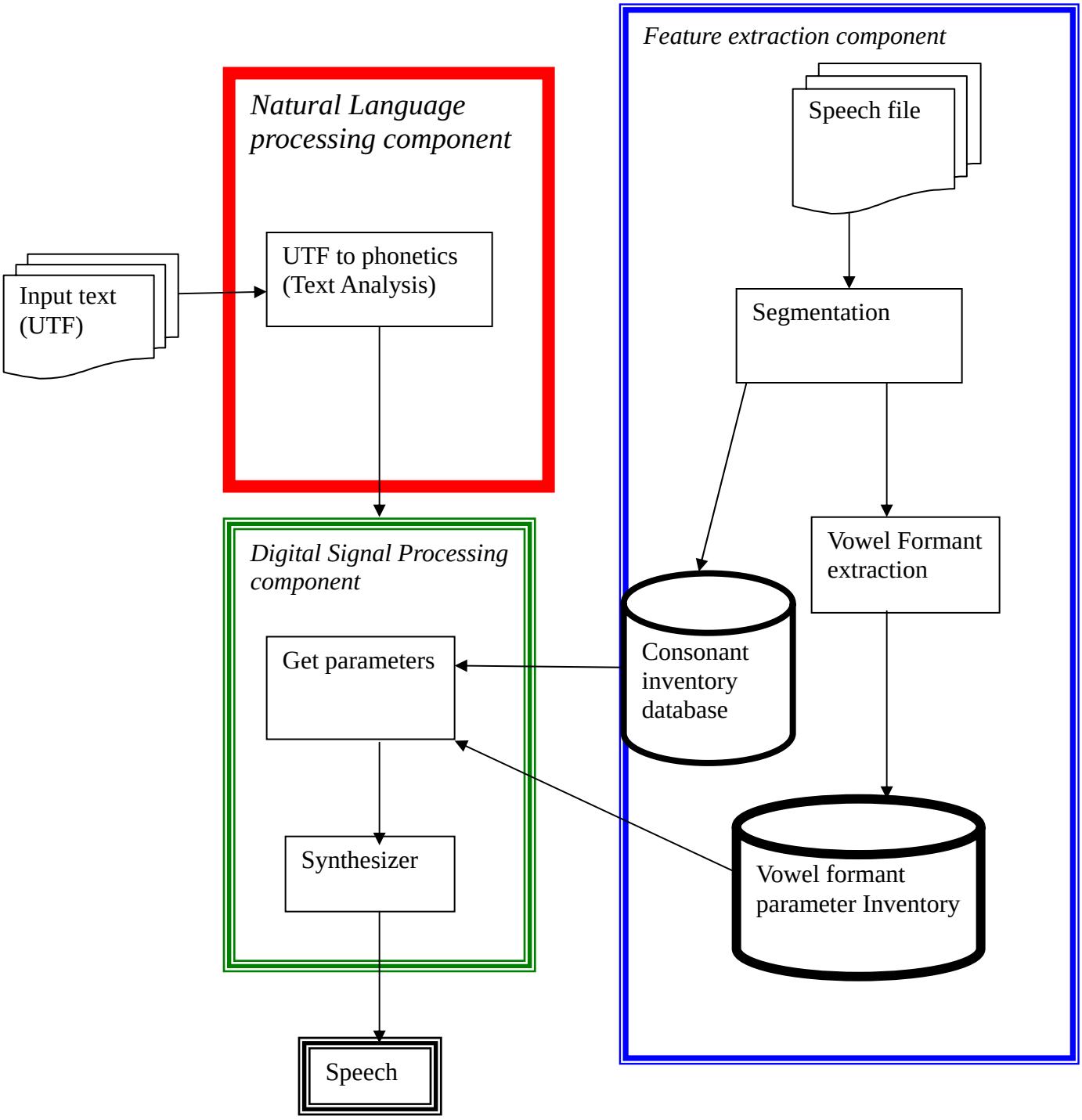

Figure 5.1: The architecture of Amharic speech synthesizer

5.2.1 Natural Language Processing Component

The natural language processing component is responsible for converting the input text into an internal linguistic description. The conversion of the linguistic description in orthographic form to phonemes is called phonetic analysis. In general, this component performs grapheme-to-phoneme conversion.

5.2.2 Digital Signal Processing Component

The digital signal processing component is responsible to generate speech waveforms by getting the appropriate input parameter from the feature extraction component. The detail is discussed in the next chapter.

5.2.3 Feature Extraction Component

5.2.3.1 Speech Data Preparation

For the preparation of an inventory data that contains vowel phoneme formants and consonant phoneme speech for Amharic alphabets, it is necessary to identify all possible consonant and vowel phoneme of the Amharic language. Amharic has 32 consonant and 7 vowels. This results in 39 possible phonemes.

The 39 phonemes are collected by taking all the Amharic CV (consonant-vowel) and V (vowel) type syllables. These are a total of (32 * 7) =224 units. These units are prepared for recording. While recording the speech, there are parameters that are required to be controlled for each wave file. These are the sampling frequency, the duration of the recorded segment, the number of bits used, and the format of the recorded speech. For this project, sampling rate of 8000 Hz and quantitative level of 16 are taken. The recorded file is saved in RIFF format.

5.2.3.2 Speech Data Segmentation

At the time of recording, each recording had silence to both end and the syllable at the middle. Hence, segmentation was performed to extract the required information about the desired unit. Segmentation of speech signal requires analyzing the speech and analysis tool.

In order to start the speech analysis process, speech files are recorded and prepared as stated above. A speech analysis program, wavesurfer, is used to analyse the speech. There are numerous ways to do the analysis. For segmentation, one of the most common ways is to label the wave spectrogram by hand, i.e., just by looking the speech waveform, the spectrogram and listening selected segment. This method had been used in this project.

Even if hand segmentation is usually closer to the correct value, it is labor intensive and tedious. Consequently, a high percentage of hand-correction was made, and then the inventory data was built. The segment labels were kept for speech unit analysis to do feature extraction

Even if the samples were recorded in area with low background noise, sometimes the electromagnetic noise generated by the computer got added to the recorded signal and it had a lot of impact on the feature extraction process.

5.2.3.3 Parameter Extraction

For this project, the most important features are the formant frequencies (F1, F2, and F3), their bandwidth (B1, B2, and B3), and pitch. There are also other important features though they are not considered in this project because of different constraints.

Feature extraction is the fundamental process in speech synthesis. Extracting these frequencies and bandwidth from continuously changing speech signal are very challenging. We tried to consider the wave to be stationary within a small time frame (10ms). When analysis is performed

on a “segment-by-segment" basis, useful information about the segment is obtained. Since our ear cannot respond to very fast change of speech data content, we normally cut the speech data into frames before analysis.

Wavesurfer, speech analysis tool, performs several acoustic analysis for the natural words, to extract a number of parameter values for each time frame of 10 ms, and to store the values in a parameter file, which was used as input to the formant synthesizer. First, formants (F1, F2, and F3) and F0 parameters were extracted. Second, the bandwidth of each formant frequency will be calculated.

We have seen how to extract the pole location of $H_i^{-1}(z)$ that can be used to extract F_i and B_i. The corresponding formant frequency and its bandwidth can be computed from a given pole location using eq.2.7 and.2.8 respectively given the pole of the inverse transform function.

5.2.3.4 Speech waveform

Speech consists of vibrations produced in the vocal tract. The vibrations themselves can be represented by speech waveforms. It is not possible to read the phonemes in a waveform, but if we analyze the waveform into its frequency components, we can obtain a spectrogram which can be deciphered [16]. When we look the speech waveform in figure 5.2 one can read from left to right as the silence on the left side of the signal, low energy for consonant e, high energy for vowel , low energy for consonant Ø , high energy for vowel ›< , low energy for consonant " and silence on the right side.

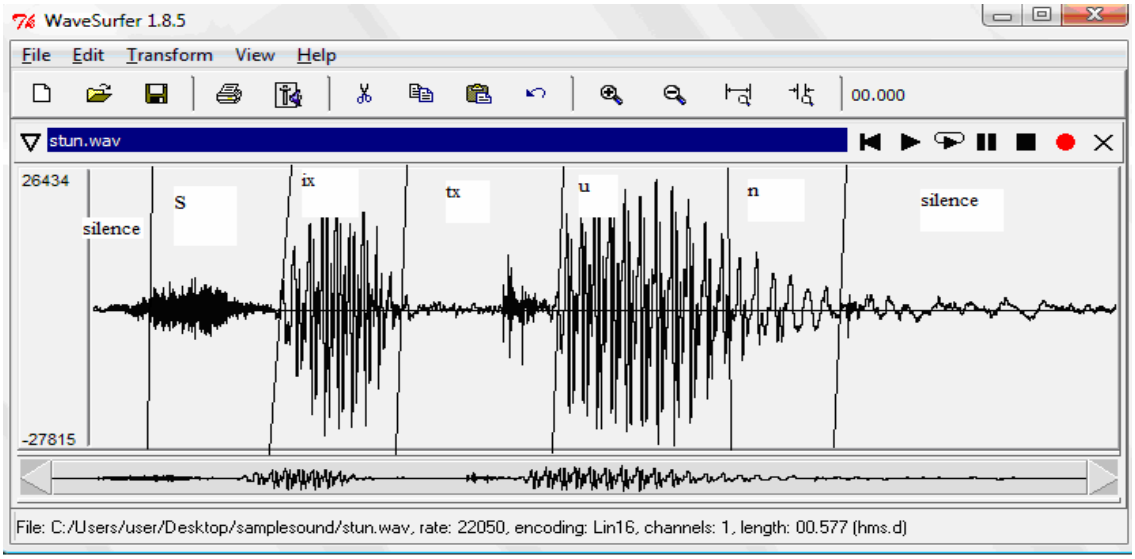

Figure 5.2: speech waveform of the word sixtxun/eÖ<"/

The characteristics of a speech signal varies with time since it is a sequence of different phonemes (consonants and vowel) with different frequency characteristics combined with pauses and periods of silence. To extract these characteristics, we need to chop the signal into segments which are more stationary and have some predictable behavior across time and frequency. These segments however should overlap to avoid the effect of discontinuity. To extract the formant trajectory of the signal, we need first to chop the signal into these overlapping segments and pre-process it [16].

5.2.3.5 Spectrogram

The spectrogram provides a picture of the energy in a signal as a function of time and frequency. It is normally displayed as a two-dimensional image, where the x-axis corresponds to time, the y-axis corresponds to frequency, and the intensity of the grey scale values corresponds to energy. By inspecting the spectrogram, important speech features can be observed, identified, and analyzed [16].

Each thin vertical slice of the spectrogram shows the spectrum during a short period of time, using darkness to stand for amplitude. Darker areas show those frequencies which have high amplitude.

Spectrum diagrams are useful for seeing the state of a complex wave during a very short period of time. But in speech, sounds are constantly changing. Spectrograms are a convenient way to diagram the changes in spectrum over time.

In Figure 5.3, Spectrogram of the word *sixtxun/eÖ<"/*, the vertical axis represents frequencies up to 8000 Hz, the horizontal axis shows positive time toward the right, and the intensity of the gray scale represent the most important acoustic peaks for a given time frame, with dark representing the highest energies, then decreases as the intensity moves toward white.

An experienced spectrogram reader has no trouble identifying the word "sixtxun" from the visually salient patterns in the image below.

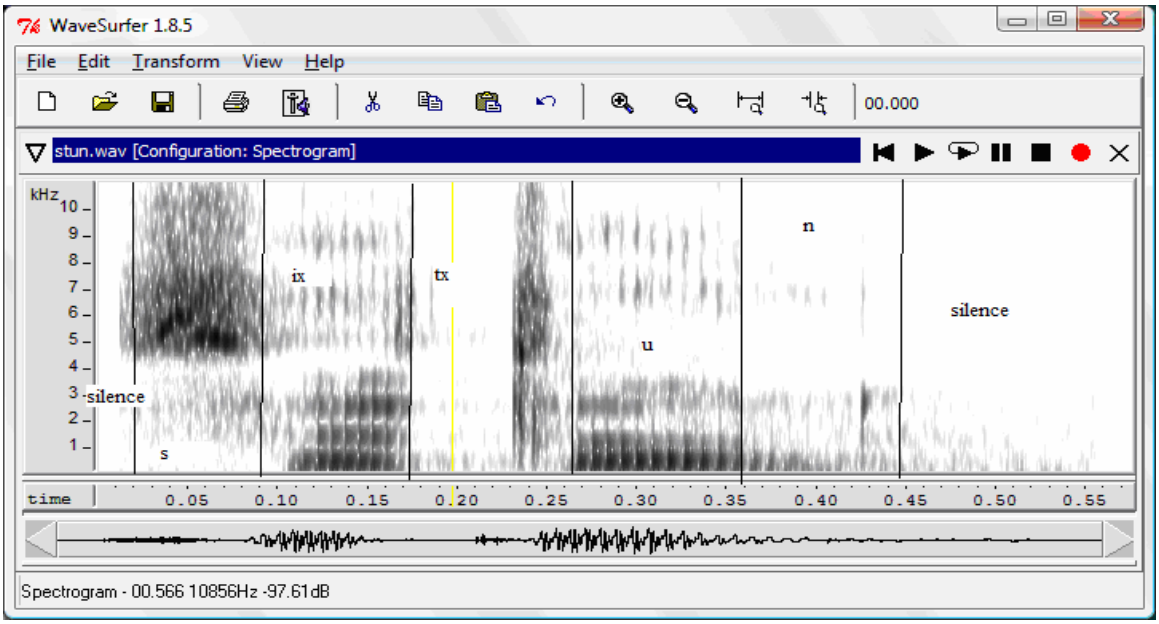

Figure 5.3: Spectrogram of the word "sixtxun/eÖ<"/."

5.2.3.6 Formants

Formant features can be interpreted as adaptive non-uniform samples of the signal spectrum that are located in the resonance frequencies of the vocal tract and normally happen to have higher signal-to-noise ratio than the other parts. The number and position of these frequencies along the frequency axis might differ depending on the phonemes and the position of the window along the phoneme (i.e. beginning or ending part of a phoneme) [16].

For the word sixtxun/ eÖ<"//, we can see the LP (Linear Predictive) coefficient analysis by taking a 10ms window near the starting position of the word. In addition, we see the formant frequencies (F1, F2, and F3) and formant bandwidth as well. To change the size of the window (frame size), it needs only adjusting the duration in the control window.

To extract the required parameters, first the speech units are divided into short frames (10 ms) of samples. Then the speech signal is analyzed through a "time window" to obtain a spectral representation of the unit. Parameterization of successive frames adequately models the speech if the frames are short compared to vocal tract motion. The first three formant frequencies are adequate to re-synthesize [13].

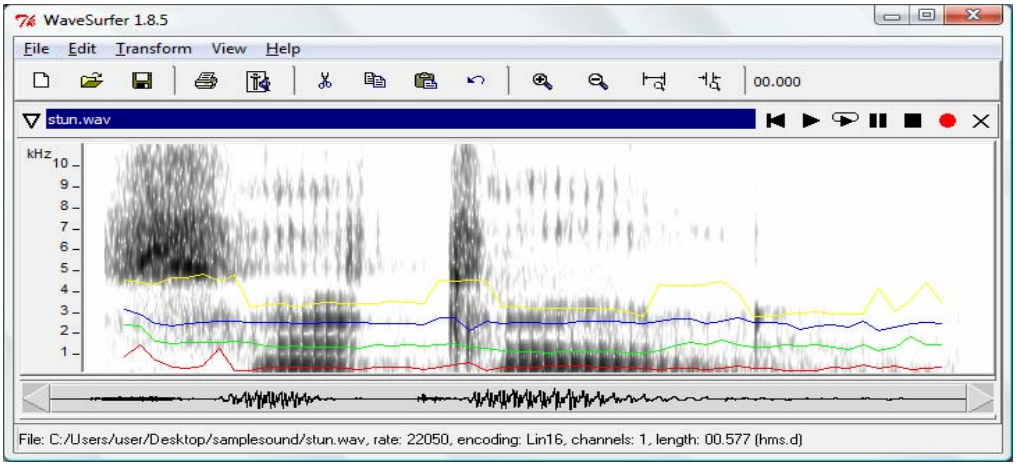

Figure 5.4: formant frequencies for the speech file sixtxun/ eÖ<"/

In figure 5.4, we can observe that F1 is at the bottom (with red color), F2 in the second from the bottom with green), F3 in the third from the bottom with blue) and F4 is at the top (with yellow) of the speech waveform. The LPC (Linear Predictive coefficients) parameters are dependent on the sampling frequency taken. In this project, the sampling rate 8000Hz and LPC order of 14 had been used.

In Figure 5.5, we can visualize the formant frequencies, spectrogram and speech waveform of the word sixtxun/ eÖ<"// from the bottom to the top.

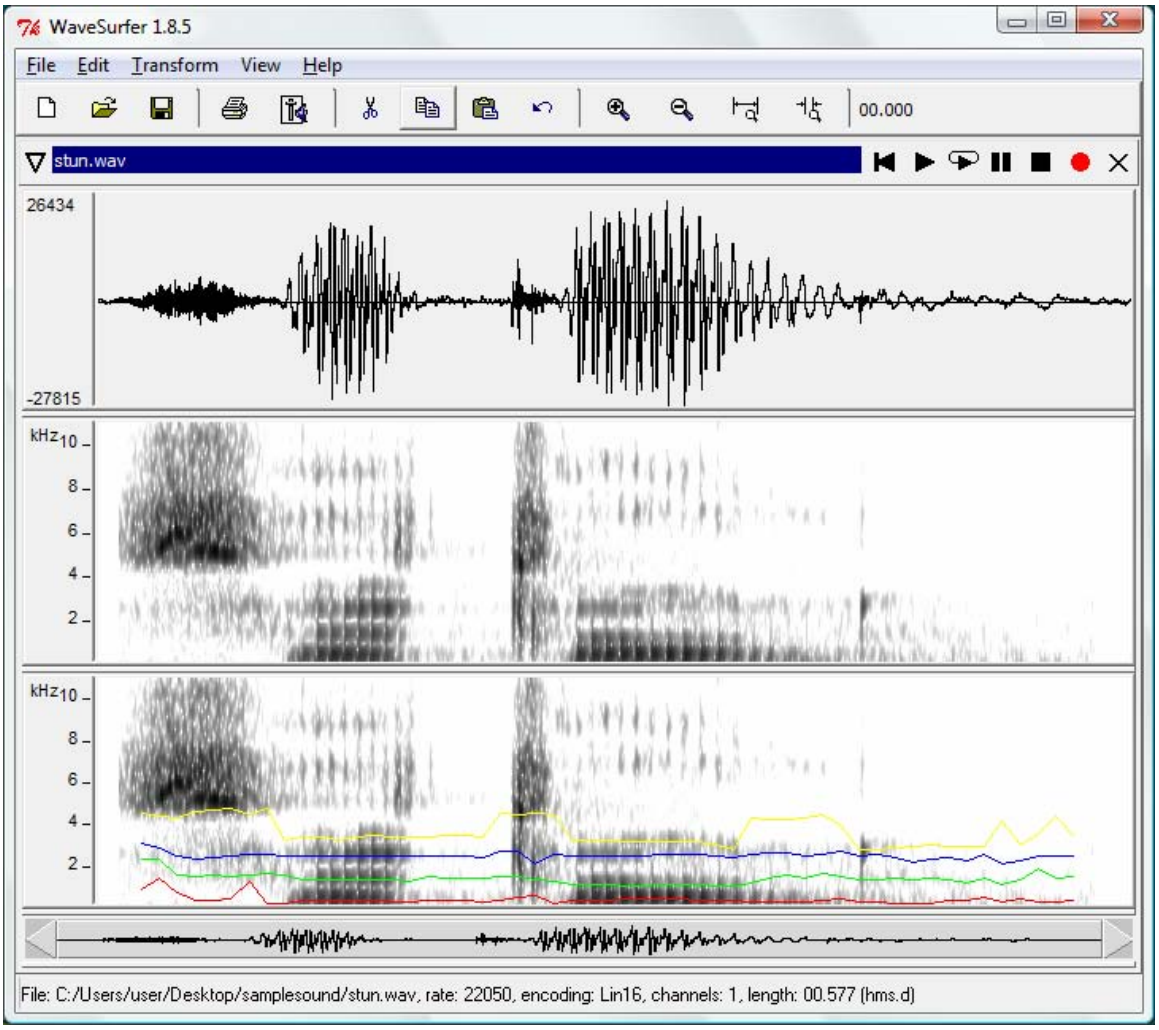

Figure 5.5: formant frequencies, spectrogram and speech waveform of the word sixtxun/ eÖ<"/

The formant data values extracted from formant contour with in 10ms interval from the speech signal, sixtxun/ eÖ<"//, using wavesurfer as shown in Fig 5.6.

For the word sixtxun/ eÖ<"//, the LP (Linear Predictive) coefficient analysis can be seen by taking a 10ms window near the starting position of the word. In addition, the formant frequencies (F1, F2, F3, and F4) and formant bandwidth as well can be observed. To change the size of the window (frame size), it needs only adjusting the duration in the control window.

To extract the required parameters, first the speech units are divided into short frames (10 ms) of samples. Then the speech signal is analyzed through a "time window" to obtain a spectral representation of the unit. Figure 5.6 shows the formant data values of the speech waveform sixtxun/ eÖ<"/.

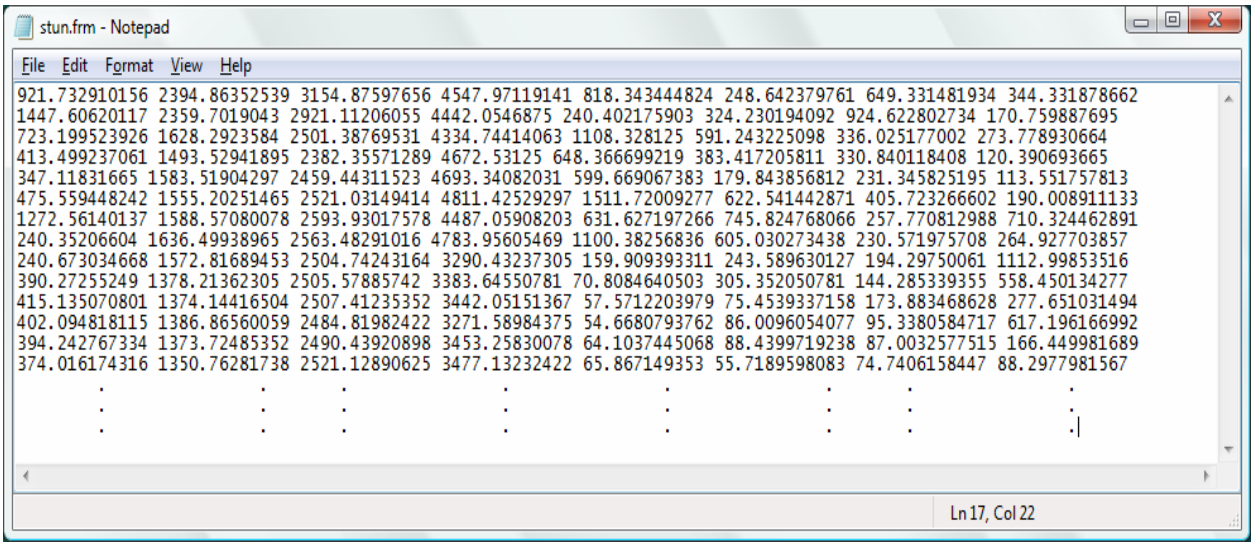

Figure 5.6: formant data values for speech waveform sixtxun/ eÖ<"/

5.2.3.7 Parameter Adjustment

After parameterization of several features like formants, pitch, bandwidth, etc, adjustments was needed at least to the major elements that had significant effect on the output.

Formants and bandwidth: Speech analysis using Wavesurfer occasionally failed to extract the correct formant values. However, this measure led to distortion of the phoneme qualities as well as in the transitions between phonemes. Eventually, attempts to adjust formant values were temporarily abandoned, until a better technique was found. Therefore, attempts had been made to adjust the formant bandwidths, although it often displayed a typical value.

Pitch: Pitch or fundamental frequency is the rate at which the vocal folds in the human speech production system vibrate, that is the opening and closing of the glottis. Voiced sounds like / ›/ cause the vocal folds to vibrate, however, the unvoiced sounds like /e/ does not vibrate the folds. For the harmonic coding of speech, correct pitch estimation is very important. The quality of the coded speech is severely degraded at the wrong pitch marks. Pitch adjustment is important since the pitch contributes to the perceived prosody most [13].

CHAPTER SIX

6 IMPLEMENTATION OF AMHARIC SPEECH SYNTHESIZER

This chapter provides the implementation details of Amharic speech synthesizer. The tools used in developing the system, the developed system and the challenges are discussed in the following section.

6.1 Tools used

Different tools were used to build the system. For the acoustic analysis, the speech analysis software, wavesurfer and colea were used. For the development of the speech synthesizer that comprises of the natural language processing (NLP) module and digital signal processing (DSP) module, matlab7.0 program were used.

Wavesurfer is an audio editor widely used for studies of acoustic phonetics. It is a fairly powerful program for interactive display of sound pressure waveforms, spectral sections, spectrograms, pitch tracks and transcriptions. It can read and write a number of transcription file formats.

Colea, a matlab software tool for speech analysis, used for spectrogram displays, displays time-aligned phonetic, manual segmentation of speech waveforms, formant analysis, Pitch analysis and the like.

Matlab is high-performance language for technical computing that integrates computation, visualization, and programming where problems and solutions are expressed in mathematical notation. Typical uses include Math and computation, algorithm development; signal processing, modeling, simulation, prototyping, and the like.

6.2 Transliteration and Phonetic Description of Input Text

In order to develop the Amharic speech synthesizer, synthesis units consisting of CV clusters along with consonants (C) and vowels (V) were used. Prior to this a transliterate routine perform transliteration as described in 3.3. The Phonetize routine shown in figure 5.1 reads a text in UTF-8 format and converts this text into the phonetic description. The phonetic description of the input is phoneme-based (consonant and Vowel). The text to be spoken out must be expressed in terms of these sound units: CV (Consonant-Vowel) a consonant followed by a vowel or VC (Vowel-Consonant) vowel followed by a consonant. This speech synthesizer has a phonetic-to-speech engine which is capable of generating speech from a suitable phonetic description.

6.3 The Structure of Inventory Data

First, a speech file was recorded for all syllables of Amharic and segmentation process was conducted, i.e., all possible phonemes (consonants and vowels) were identified. The next was automatic extraction of acoustic parameters from the segmented voiced speech file which was then used to generate synthetic versions of the same voiced speech. Consequently, the unvoiced speech segments were stored in appropriate storage device. The unvoiced speech was kept in the storage by appending in co-existing voiced signal as context information to improve the quality of the synthetic speech.

6.4 Speech Synthesizer

As discussed above, language processor component is responsible for generating the phonemic units of the input text. After identifying and generating the phonetic notation, these units were given to the digital signal processor so that it would generate the corresponding waveform to the user.

6.4.1 Voiced Speech Synthesis Technique

Voiced sound is one in which the vocal cords vibrate periodically. In order to synthesis these sounds, formant parameters (F1, F2, and F3) and the F0 fundamental frequency that were extracted from the segmented waveform using Wavesurfer in the feature extraction process were used. Formants are peaks in the frequency spectrum of a sound caused by acoustic resonance of sounds produced by the vocal tract vibration.

Therefore, the synthesizer takes the formants and corresponding bandwidths values from the formant inventory and generates the voiced sound and keeps the speech waveform in appropriate place for joining the different synthesised units.

6.4.2 Unvoiced Speech Synthesis Technique

In the production of unvoiced/voiceless sounds, the vocal cords are left open. These unvoiced sounds are then taken during the segmentation process. Since the speech waveform was taken from all Amharic syllables at the time of data preparation the appropriate unvoiced sound segment was taken from the database. For example when you read the speech waveform of the syllable so/f/ as shown in figure 6.1, from left to right , first it has silence , then the unvoiced sound e and the voiced sound * and at last silence would be found. Therefore the unvoiced sound segment will be taken and stored in the respective directory.

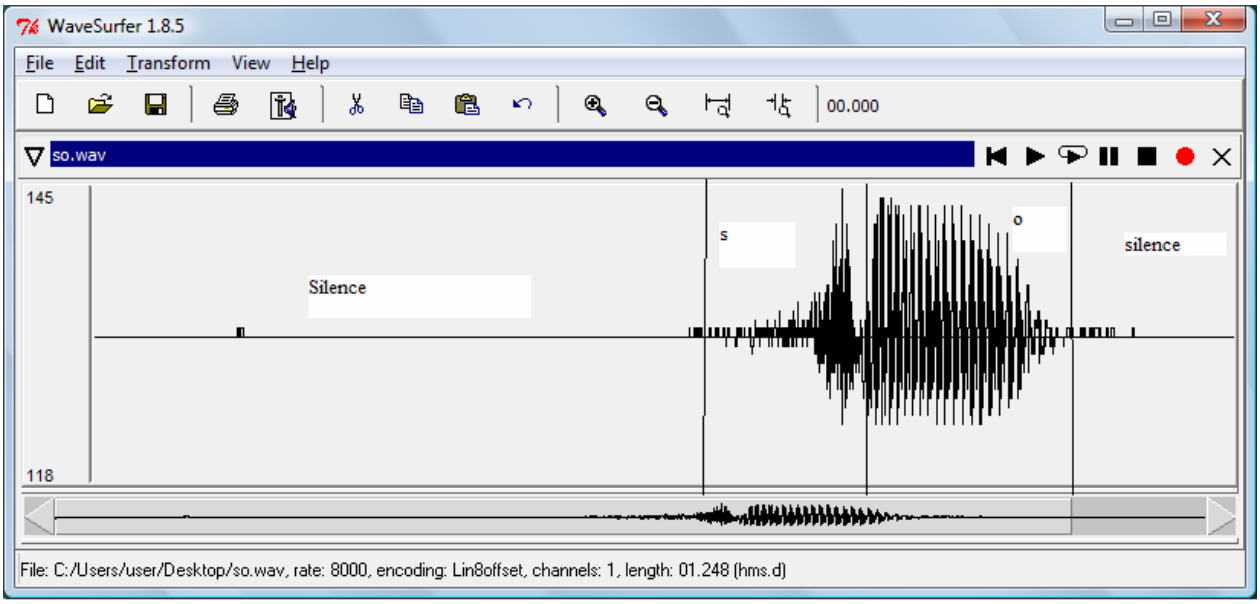

Figure 6.1: speech wave form of the syllable so/f/

6.4.3 Concatenation of Voiced and Unvoiced Units

In order to get a word level speech waveform, concatenation of the voiced and unvoiced sounds synthesized above in their proper order was needed. Concatenation of units was made in the following way. The final speech file was prepared and a short silence was written on it. Then for every voiced and unvoiced sound, their order of the word transcription, its synthesized signal was written on it and the file was saved as a wave file. Finally, it was played with the system transducer.

6.5 Interface of the System

The system was developed using matlab 7.0 including the user interface as shown in Figure 6.2. The interface provides text editor where the user can enter input text. Once the user enters the text and clicks on the synthesize button then the system would synthesize speech corresponding to the input text.

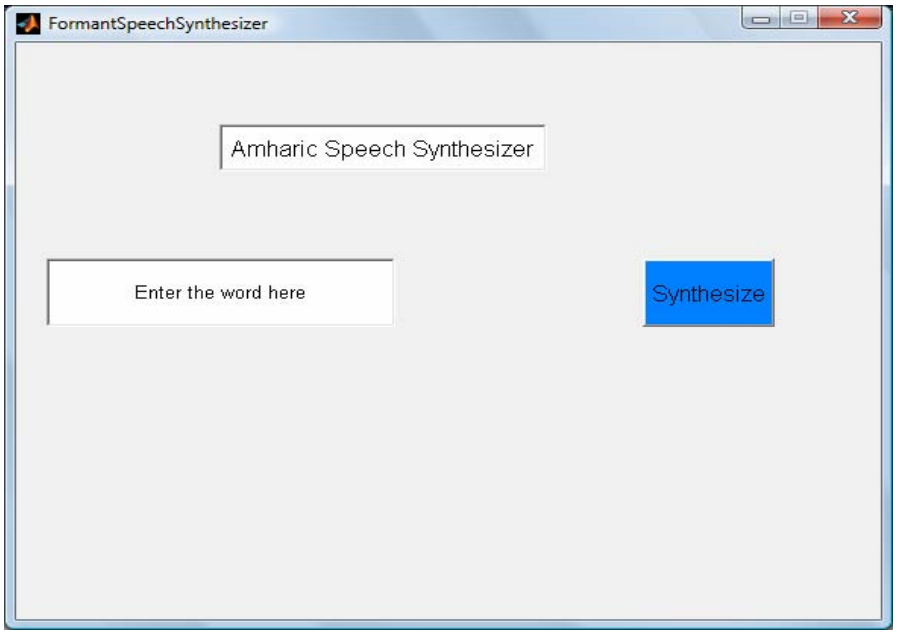

Figure 6.2: Interface of speech synthesizer

Finally, the corresponding speech generated by the synthesizer for the input text will be written and saved as a wave file. For example, if you synthesize the word "eÖ<"/s ix tx u n/", you can get the signal and spectrogram of the synthesized speech as shown below in figure 6.3 and figure 6.4 respectively.

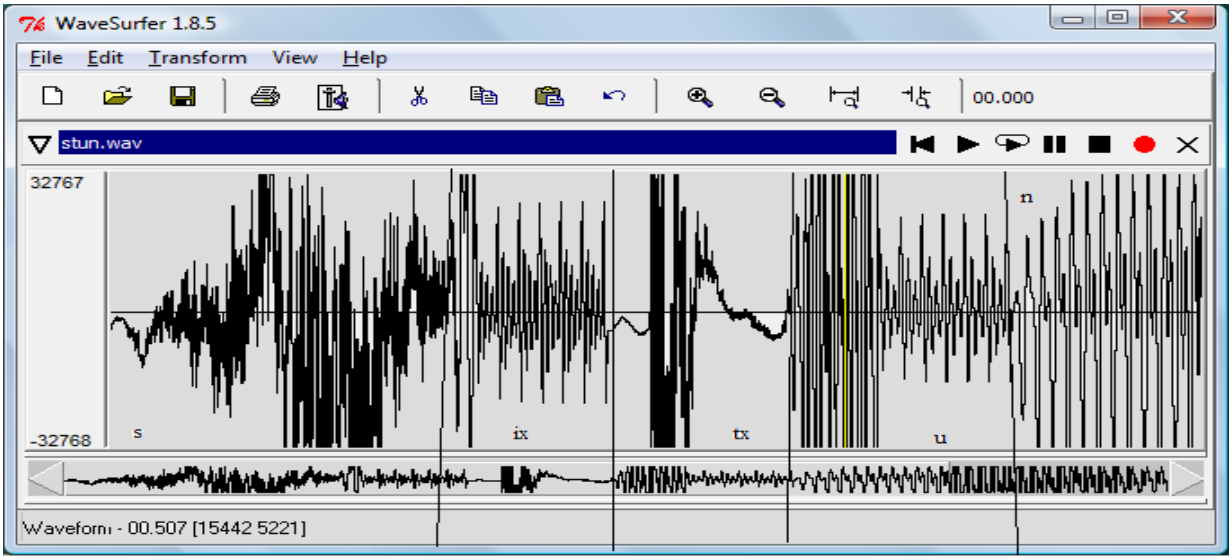

Figure 6.3: The signal generated for the word “s ix tx u n /eÖ<”/”

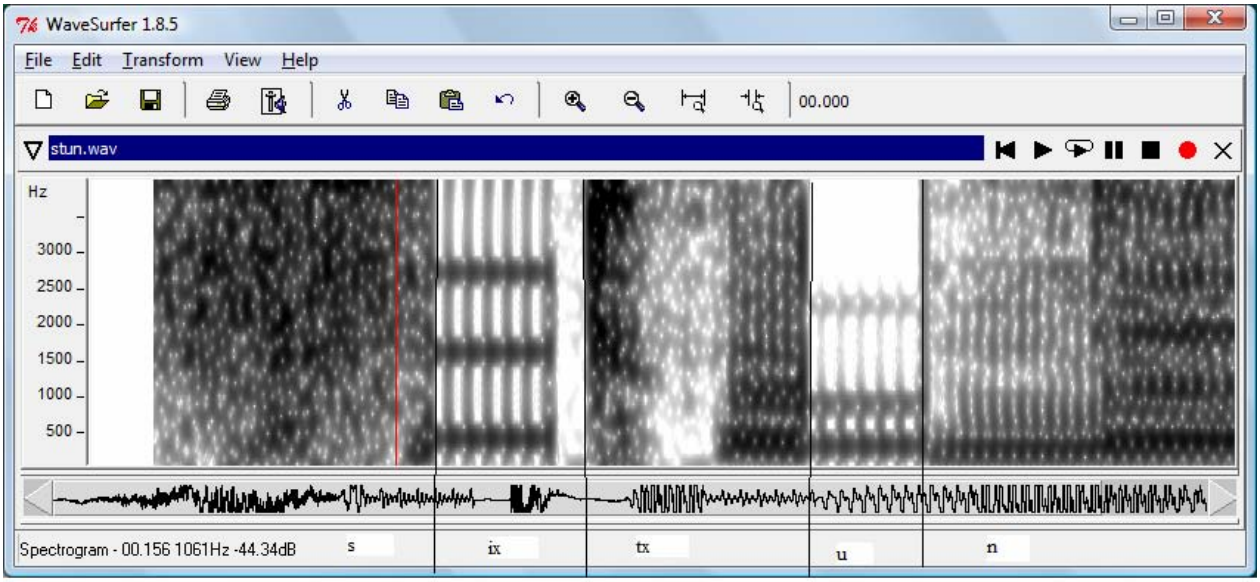

Figure 6.4: The spectrogram of the word “s ix tx u n / eÖ<”/”

In general, the main modules of the system and their flow are as follows.

1. The language processor module is used to accept the input text, transliterate and identify the phonemes of the text.
2. After getting the phonemes, the synthesizer reads the first phoneme of each text and sends the phoneme to select feature of the phonemic units form the data and synthesize them accordingly. It will perform repeatedly until it reaches to the last phoneme of the text.
3. After the corresponding file is stored for each phoneme, those basic sound units of the word were combined and played out.

6.6 Challenges

One of the challenges of this project was, whenever a different speech analysis tool such as colea was used, different types of formants, pitch, spectrogram, power spectral magnitude etc. were obtained. Even though there was a problem of getting good speech analysis tool, extraction of features was done by using wavesurfer.

The feature extraction process had slightly affected by the electromagnetic noise generated by the computer as well as the surrounding noise that gets added.

CHAPTER SEVEN

7 CONCLUSION AND FUTURE WORKS

7.1 Conclusion

Formant synthesis is one of the most popular methods of generating speech, which some times called synthesis by rule. It uses a set of rules to modify the pitch, formants and its bandwidth and so on. Even if speech synthesis has a history of many years, it could not still be employed to meet the demands of human beings in different areas as it requires high memory and computational power.

In this project work, a synthesizer was developed which would generate a speech for a given input text. The technique models the human speech production system in the form of source and filter, in which the source is completely independent from the filter. The source is identified by the air flow through the vocal and the filter represents the resonance of the vocal tract, which are also called the formant that changes from time to time. The resonance is due to the constriction of the vocal tract while generating different sounds.

Parameters like formants, its bandwidth, pitch, etc were extracted from collected speech for voiced sounds. The unvoiced sounds with co-existing context were segmented from all Amharic syllables and stored in appropriate place. The voiced sounds were generated by taking parameters from the inventory data. Finally, in order to synthesize the speech for a given word, the system should concatenate the voiced sounds and unvoiced sounds. The system would provide flexibility of a speech with low memory and data requirements.

7.2 Future works

Developing speech synthesis system needs reasonably good development environment. In order to improve the quality of the system developed in this project and extend it, the following should be considered.

- The speech analysis phase needs very serious attention. For example, it needs calm condition in order to get correct feature of the speech file. So, if one can record the speech in a sound proof room and extract with a very high quality speech analysis tool, this would improve the quality of the output. Because these were the major challenges of this project, i.e., absence of good formant extractor and interference of the background noises.
- Using better feature extraction tool helps to get representative features (formants, bandwidth, pitch, etc) of speech and definitely it would improve the quality of the output.
- The scope of this work was to synthesize at word level; but it could be extended to synthesize any size sentence/document.
- This project considered only Amharic language syllables. But it could be extended to all Ethiopian languages.
- This project assumed all words were normalized into appropriate form but normalization could be considered as a major issue to handle abbreviations, acronyms, numbers, symbols, etc.
- The co-articulation effect and the syllable weight are research issues.

REFERENCES

[1]. James L. F. (1965). “Speech Analysis: Synthesis and Perception”. Springer: Berlin.

[2]. Jilei T., etal (2006). “Modular Design for Mandarin Text-To-Speech Synthesis”. TC-STAR Workshop on Speech-to-Speech Translation, Barcelona, Spain.

[3]. Azhar A. S., Abdul W. A. and Lachhman D. (2000). “Bi-Lingual Text to Speech Synthesis System for Urdu and Sindhi”. Institute of IT, University of Sindh, Jamshoro, Pakistan.

[4]. Allen, J., Hunnicut S. and Klatt, D. (1987). “From Text to Speech: the MITalk System”. Cambridge University Press, England.

[5]. Nadew Tademe (2008). “Formant-based speech synthesis for Amharic vowels”. MSc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia.

[6]. Tanja S. and Katrin K. (2006). “Multilingual Speech Processing”. Elsevier Inc, San Diego, USA.

[7].Thierry Dutoit(1996). “An Introduction to Text-to-Speech Synthesis”. Kluwer Academic Publishers, Dordrecht.

[8]. Thierry Dutoit (1996). “High-quality text-to-speech synthesis: an overview”. Journal of Electrical & Electronics Engineering, Australia: Special Issue on Speech Recognition and Synthesis, vol. 17, pp 25-37.

[9]. M.J. LIBERMAN (1992). "Text analysis and word pronunciation in text-to-speech synthesis", in Advances in Speech Signal Processing, S. Furuy, M.M. Sondhi eds., Dekker, New York, pp.791-831.

[10].Pardeep Gera (2006). Text-To-Speech Synthesis for Punjabi Language. ME thesis in Software Engineering, Thapar Institute of Engineering and Technology, Patiala.

[11].Solomon Teferra Abate, Wolfgang Menzel and Bairu Tafila(2005). “An Amharic Speech Corpus for Large Vocabulary Continuous Speech Recognition”. Fachbereich Informatik, Universit¨ at Hamburg.

[12].Solomon Teferra Abate and Wolfgang Menzel(2007). “Syllable-Based Speech Recognition for Amharic”. Proceedings of the 5th Workshop on Important Unresolved Matters, pages 33–40,Prague, Czech Republic, June 2007.

[13]. Hasim Sak (2004). “A Corpus-Based Concatenative Speech Synthesis System for Turkish”. Masters Thesis, Computer Engineering and Information Science, Bilkent University.

[14]. Bekrie Ayele(1997). “Ethiopic: An African Writing System; its History and Principles”. RSP, Canada.

[15]. Philip Loizou (1999). “COLEA: A Matlab Software Tool for Speech Analysis”.

Retrieved on 10/05/2008, from http://www.utdallas.edu/~loizou/speech/colea.htm.

[16].http://books.huihoo.org/introduction-to-digital-filters-with-audio-applications/What_Filter.html. Retrieved on 04/05/2008.

[17]. Sebsibe H/Mariam , S P Kishore, Alan W Black, Rohit Kumar, and Rajeev Sangal (2005). “Unit Selection Voice for Amharic using Festivox”. 5th ISCA Speech Synthesis Workshop, Pittsburgh, page 103-107.

[18]. Hussien Seid and Bj¨orn Gamb¨ack (2005).“A Speaker Independent Continuous Speech Recognizer for Amharic”. INTERSPEECH, Lisbon, Portugal.

[19]. Habamu Taye (2006). “Diphone based text-to-speech synthesis system for Amharic”. MSc Project, Faculty of Informatics, Addis Ababa University, Ethiopia.

[20]. Laine Berhane (1998). “Text-to-Speech Synthesis of the Amharic Language”. MSc Thesis, Faculty of Technology, Addis Ababa University, Ethiopia.

[21]. Sami Lemmetty(1999). “Review of Speech Synthesis Technology”. Master’s Thesis, Helsinki University of Technology.

[22]. Huang, X., Acero, A., Hon, H. (2001). “Spoken Language Processing. Prentice Hall”. Upper SaddleRiver, New Jersey.

[23]. Chanwoo Kim, Kwang-deok Seo, and Wonyong Sung (2006). “A Robust Formant Extraction Algorithm Combining Spectral Peak Picking and Root Polishing”. Hindawi Publishing Corporation, EURASIP Journal on Applied Signal Processing Volume, Pages 1–16.

Declaration

I, the undersigned, declare that this project is my original work and has not been presented for a degree in any other university, and that all source of materials used for the project have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____

Place and date of submission: Addis Ababa, June 2008.