Chapter 4

Naturalness in Synthesised speech: the Voice Conversion way

4.1 Introduction

Voice Conversion or VC, which is the technique of converting speech from a source speaker to speech of a target speaker, has been used in a number of applications such as conversion of whispered to normal speech, conversion of speaking styles, conversion of emotion, accent, improving speech intelligibility, transformation of speaker identity, speech-to-speech translation etc. to name a few. The aim of VC is to transform the characteristics of a speech signal uttered by a source speaker in such a manner that the transformed speech sounds as if it is spoken by the target speaker. Such a conversion transforms not only the organic properties of speech such as voice quality but also linguistic cues such as regional accents and requires transformation of both spectral and prosodic features. In this work we explore to what extent VC techniques can help in generating dialectal speech from existing Text-to-Speech (TTS) systems. In other words, we explore to what extent VC can be used for incorporating dialectal features into speech synthesised from a TTS built for the standard variety of the Assamese language. A TTS built for the standard variety of a language will have no explicit linguistic or prosodic knowledge pertaining to a target dialect. Therefore it is obvious that dialectal speech generated from it will lack naturalness with respect to the dialect concerned. Our aim is to use VC techniques to incorporate some amount of naturalness into this speech. Three different techniques, i.e., Vector Quantization (VQ) with mapping codebooks, Gaussian Mixture Models (GMMs) and Artificial Neural Networks (ANNs), have been used to develop mapping functions. The mapping function maps from speech synthesised by a TTS for the standard form of the language to parallel speech recorded from a speaker in the target dialectal variant of the language. Our aim is not to compare existing VC methods or improve the quality of a VC system, but to explore the prospects of VC in the generation of dialectal speech or the prospects of VC in incorporating naturalness to synthesised dialectal speech. Therefore we carry out the conversions using VC mapping functions and then select the best conversions for resynthesis. Mel Cepstral Coefficients (MCEPs) are used to represent the spectral envelope, while pitch, intensity and duration values, collectively referred to as PID, represent the prosody of speech.

The rest of the chapter is organised as follows. Section 4.2 presents a brief introduction to VC and its applications, Section 4.3presents the motivation for this work, Section 4.4 presents a brief review of related literature and Section 4.5 describes the building of the four systems A, B, C and D. Experimental results are presented in Section 4.6 and finally in Section 4.7 conclusions and plans for future work are presented.

4.2 Voice Conversion

Voice Conversion is a special type of the Voice Transformation technique whose goal is to modify a speech signal uttered by a source speaker to sound as if it was uttered by a target speaker, while keeping the linguistic contents unchanged [26]. VC modifies speaker-dependent characteristics of the speech signal, such as spectral and prosodic, in order to modify the perceived speaker identity while keeping the speaker independent information i.e., the linguistic contents, same. An overview of a typical VC system is presented in Figure 4.1 [96]. In the training phase, the VC system is presented with a set of utterances recorded from the source and target speakers. The speech analysis and mapping feature computation steps encode the speech waveform signal into a representation that allows modification of speech properties. Source and target speakers speech segments are aligned (with respect to time) such that segments with similar phonetic content are associated with each other. The mapping or conversion function is then trained on these aligned features. The conversion phase consists of two steps. First mapping features from a new source speaker utterance are computed, and second, the features are converted using the trained conversion function. The speech features are computed from the converted features which are then used to synthesise the converted utterance waveform.

VC systems are categorized as using parallel or non parallel corpus, text dependent or independent, language independent or cross-language. The most common approach to VC uses recordings of a set of parallel sentences from both source and target speakers. However, the source and target speakers are likely to have different length recordings, and have dissimilar phoneme durations within the utterance as well. Therefore, a time-alignment approach must be used to address the temporal differences. Manual or automatic phoneme transcriptions can be utilized for time alignment. Most often, a dynamic time warping (DTW) algorithm is used to compute the best time alignment between each utterance pair or within each phoneme pair. The final result of this step is a pair of source and target feature sequences of equal length. More complicated approaches are required for non-parallel alignment. An important factor for VC categorization is the amount of data used for training the system. It is seen that conversion functions that memorize better are more effective for larger training data while those that generalize better are more preferable for smaller training data. The three most effective speech features with respect to VC are the average speech spectrum, formant frequencies and the average pitch level [96]. Most VC systems therefore aim to modify the speech features related to the short time spectral envelopes and the pitch value. Today VC systems are gaining popularity because of its varied applications. VC systems however pose a threat to speaker verification systems [154].

4.3 Motivation

Building a TTS for the standard variety of a language is a much simpler task than building a TTS for a dialect, the main reason being the ready availability of speech data of the standard variety. The quality of dialectal speech generated by a TTS built for the standard variety of the language, is poor in terms of naturalness. This is due to differences in pronunciation rules, syllabification rules, phoneme and syllable inventory and prosodic factors between the two varieties of the language. Therefore building a module for post processing this synthesised speech lacking naturalness, would help in achieving more natural sounding speech with respect to the concerned dialect.

There are two basic advantages of using VC in our work. Firstly, a small

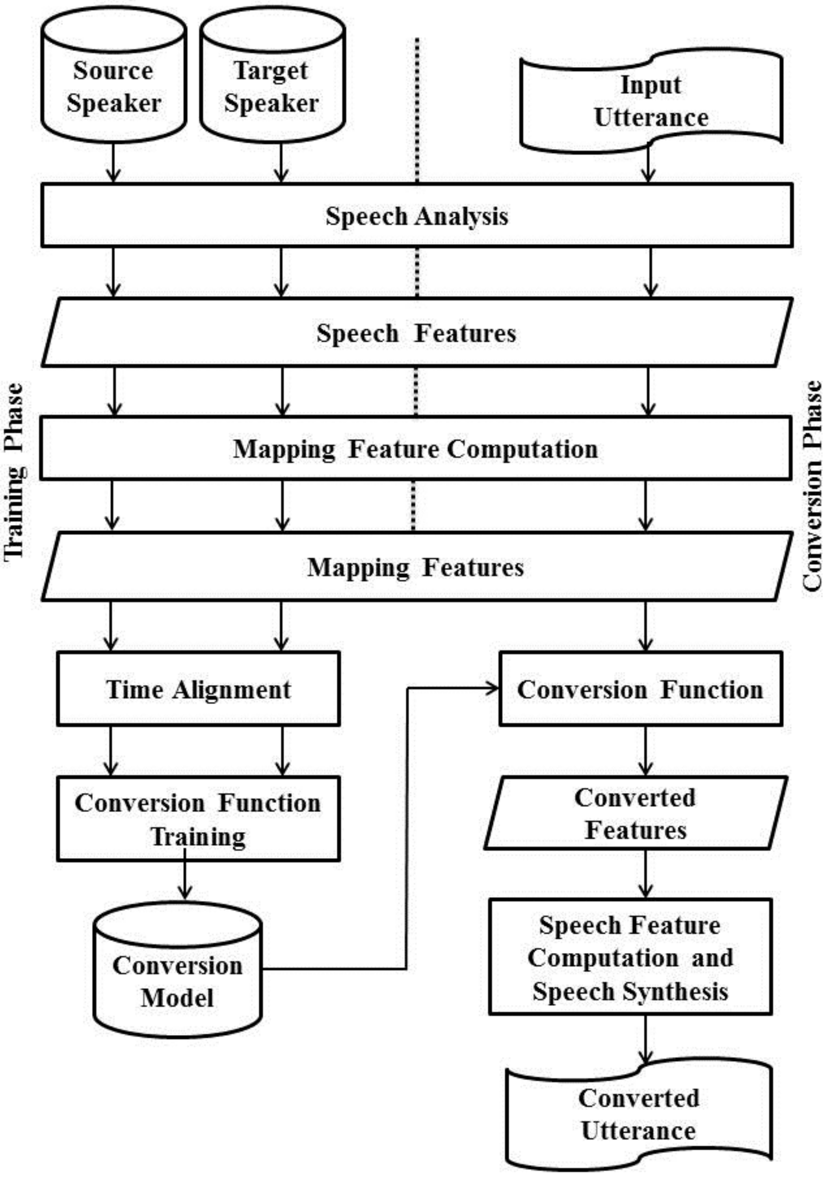


Figure 4.1. A typical Voice Conversion System. (courtesy:[96])

amount of target data is known to be sufficient to train the VC system. This would be particularly helpful since dialect data is scarce and limited. Secondly, VC can be performed with limited annotation of the training data since the classes of spectral features represented by the parametric components are determined purely on the basis of acoustic measures. This feature of a VC system is also desirable for our work since we do not have an automatic transcription tool for Assamese and manual transcription is both time taking and expensive. Our system as shown in Figure 4.2 includes two modules in addition to the TTS module: the preprocessing module which is a text-to-text translator for translating the text in the standard variety of a language to text in the target dialect. The output of this module, i.e., given the

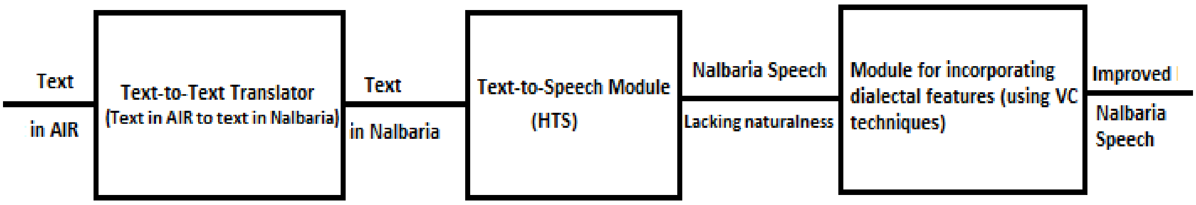


Figure 4.2. Block Diagram Representation of Proposed System

transcription of an utterance in the standard variety, the equivalent transcription in the dialect, will be fed to TTS module. The utterance synthesised by the TTS will resemble the speech of a non-dialect speaker and will lack naturalness with respect to the dialect concerned. This will be passed on to the post processing module for incorporating naturalness into the synthesised speech. This module will use VC techniques for converting spectral and prosodic features from the source (speech in standard variety of Assamese) to the target (speech in a dialectal variety of Assamese). By transforming the overall spectral characteristics, VC also realises the corresponding voice quality (VoQ) changes which is implicitly in the spectral conversion function [143], thereby contributing to improved naturalness. The final output speech is expected to be closer to speech in the target dialect. The current work focuses on the TTS module and the post processing module.

As previously mentioned in Chapter 3, Section 3.1, our work considers the All India Radio (AIR) variety of Assamese as the standard form and the Nalbaria variety (NAL) as its dialectal variant. The Nalbaria variety has been chosen for our study because it greatly differs from the standard form in terms of accent, vocabulary, pronunciation as well as grammar. The Nalbaria variety has additional syllables, consonant clusters and diphthong sounds. What mainly distinguishes it from the standard variety is however the tone or rhythm which is referred to as prosody and the manner in which various speech sounds are pronounced.

4.4 Literature Review

In an attempt to create new voices from existing synthesis systems researchers are using various methodologies. State-of-the-art techniques related to such topics are reviewed in the following subsections.

4.4.1 Dialect Synthesis

The dialects of a single language may be defined as mutually intelligible forms of that language which differ in systematic ways from each other. Although a lot of work has been going on in speech synthesis, work on dialect synthesis is relatively few. Recent speech analysis techniques are opening up new avenues of research in the processing of oral dialectal data. Adaptive parametric Hidden Markov Model (HMM)-based speech synthesis [163] allows for the usage of a back-ground model or average model to improve synthetic voice quality with small amounts of adaptation data and is especially suitable for designing TTS systems for dialects. A number of works ([109], [151] and [9]) report on systems built using this approach. Representing a dialect primarily requires a specialized pronunciation dictionary which reflects deviations from the standard variety on relevant linguistic levels. Pucher et al. [106] developed methods to derive Viennese dialect dictionaries from a standard Austrian German dictionary using sets of transformational rules. In addition to adaptive methods, a number of works exist which attempt to create dialectal speech from standard speech by manipulating prosodic features such as F0, segmental durations, intensity, etc. Yoon [160], showed that transformation of both F0 contour and segmental durations had major effects on listener responses of synthesised Korean dialects. A major challenge in the synthesis of dialects is lack of a proper phoneset and sufficient training data. Pucher et al. [107] describes HMM-based machine learning methods and supervised optimization for the definition of the phoneset of an unknown dialect.

4.4.2 Accent Conversion

While a dialect may be described as a variation of a given language spoken in a particular place or by a particular group of people, an accent may be considered to originate not only from regional variations but also from variations in the socio-economic status of the speakers, their ethnicity or their age. It may also be associated with a language which is not the first language of the speakers. Accent describes the way a speaker produces the sounds of a language while dialect describes a person's accent in addition to the phonology, vocabulary and grammar associated with the dialect. Therefore the problem of dialect conversion may be reduced to the problem of accent conversion provided the rest of the issues are taken care of. Zheng [166] identifies accent influential acoustic features of two English dialects and investigated accent conversion via formant modification and pitch contour manipulation. Zetterholm [165], states that to imitate a speaker's voice and speech behavior, one has to be aware of not only the group identity markers such as regional or social dialect, but also the personal markers in speech such as pronunciation or articulation. Chao-angthong et al. [21] developed the Northern Thai Dialect (NTD) TTS system which translates text input in the Central Thai Dialect (CTD) and synthesises speech in NDT. They modified two components in the TTS, the Grapheme-to-Phoneme (G2P) and Speech models used.

4.4.3 Voice Conversion

A number of mapping techniques are used for the VC task to learn the associations between the spectral mapping features. Vector quantization (VQ) can be used to reduce the number of source-target pairs in an optimized way. This approach creates M codevectors, where 'M' is the number of centroids, based on hard clustering using VQ on source and target features separately. At conversion time, the closest centroid vector of the source codebook is found and the corresponding target codebook is selected. The VQ approach is compact and covers the acoustic space appropriately since a clustering approach is used to determine the codebook. However, this simple approach still has the disadvantage of generating discontinuous feature sequences. Valbret et al. [147] proposed to use linear multivariate regression (LMR) for each codevector. In this approach, the linear transformation is calculated based on a hard clustering of the source speaker space. One of the simplest mapping functions is a look-up table that has source features as entry keys and target features as entry values. For an incoming feature, the function looks up to find the most similar key based on a distance criterion. In other words, it looks for the nearest neighbor of the incoming source feature and select its corresponding entry value. This category of approaches is called exemplar-based VC [155]. The most popular VC approach in the literature is Gaussian mixture model (GMM) based conversion. A GMM can be trained to model the density of source features only [135] or the joint density of both source and target features [66]. Nonlinear methods using Artificial Neural Networks (ANNs) have been applied to VC to capture the nonlinear relationships between source and target [31]. Laskar et.al [81] present a comparative analysis of ANNs and GMMs for the design of a VC system using LSFs. Spectral features like MFCCs and Linear Prediction Coefficients (LPC) [79], Line Spectrum Pairs (LSP) and Line Spectral Frequency (LSF) [115] have been used in various works of VC. VC can be used to transform the overall spectral characteristics for realising corresponding voice quality changes implicitly in the spectral conversion function. Voice quality is known to display significant variation across different speaking styles, or across different dialects. Three VC methods, weighted codebook mapping, weighted frame mapping and joint source-target GMM, used for transforming voice quality of neutral speech to emotional speech have been compared by Turk and Marc [143]. Likewise VC has been used to convert one form of speech to another, like adult to child speech [151] and whispered to normal speech [78].

4.4.4 Prosody transformation and Dialects

Regional dialects are known to display differences at the prosodic level also. In fact, the prosodic aspect may be one of the most outstanding aspects of regional dialects. Srikanth et al. [134] proposes a framework for converting both spectral and prosodic features whereby phoneme duration is modified using a Gaussian normalised transformation before mapping spectral characteristics of source speaker to those of the target speaker using ANNs. Their results confirm that incorporating durational modification has a significant improvement over a VC system using only spectral features. Most VC systems use a linear mean variance method to transform the pitch range of the source speaker to that of the target speaker. They overlook the local variations that affect the speaking style. Popular methods for pitch conversion are GMMs, codebook method [60] and non-linear pitch modification using ANN [17]. In addition to pitch and duration, controlling the intensities of speech segments also bring naturalness to synthesised speech. Reddy and Rao [118] use syllable specific features capable of capturing intensity variation patterns to train feed forward neural networks. Chiang [25] presents a cross-dialect adaptation framework for constructing prosodic models for Chinese dialect TTS systems. He adapts dialect prosodic models from an existing Mandarin speaking rate-dependent hierarchical prosodic model.

Although a lot of work has been reported for recognition/identification of dialects, very little work is present for dialect synthesis or bringing naturalness to synthesised dialectal speech. Likewise VC has been used for conversion of speaking styles, accent conversion, emotion conversion etc. However, the scope of using such techniques for incorporating dialectal features into synthesised speech has not been explored yet as illustrated in Figure 4.3. The current work is an attempt to apply VC techniques to synthesised dialectal speech to make it sound more natural. At

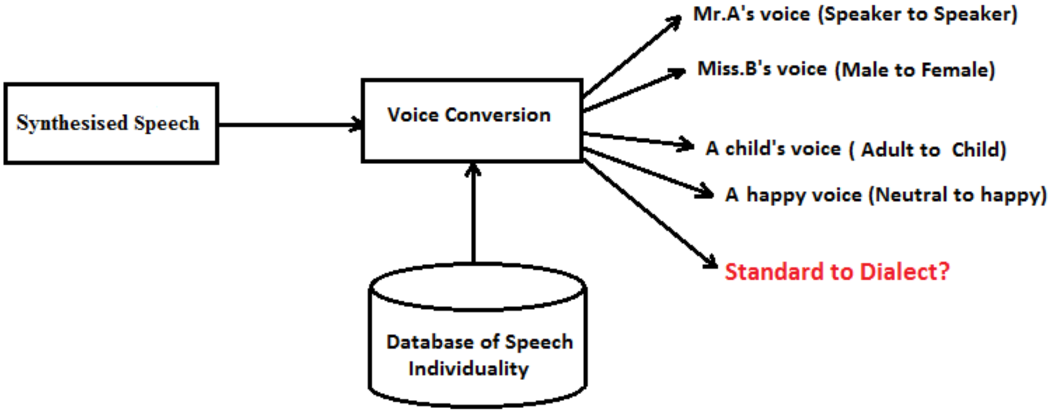

Synthesised Speech
Voice Conversion
Database of Speech Individuality
Mr.A's voice (Speaker to Speaker)
Miss.B's voice (Male to Female)
A child's voice (Adult to Child)
A happy voice (Neutral to happy)
Standard to Dialect?

Figure 4.3. VC for Dialectal Speech?

the same time, experiments are carried out to manipulate prosodic features such as vowel duration, pitch and energy contours of the synthesised speech to bring it closer to the target dialect.

4.5 Methodology

In this work dialectal speech is produced in four different ways by the four systems A, B, C and D as presented in Figure 4.4. System A is a TTS system for the standard variety of Assamese where the text input is transcriptions of dialectal speech samples. System B uses VC techniques to modify the output of System A. Systems C and D carry out manual manipulations of prosodic features on the outputs of Systems A and B respectively. We then compare the outputs of these four systems to analyse the scope of using these systems in generating dialectal speech.

4.5.1 System A: The standard HTS

The input to the VC system is synthesised speech lacking naturalness with respect to the target dialect. The first step therefore is to generate this synthetic speech from a TTS for the standard variety of Assamese. Initially an existing syllable based unit selection TTS built by IITG[1] using *Festival* is used to synthesise transcribed Nalbaria utterances. But because of differences in the syllable inventories in the two varieties, the TTS failed to produce quality output. For example, for the word

[1] `http://www.iitg.ernet.in/cseweb/tts/tts/Assamese/festival/iitg_ass_sameer/`

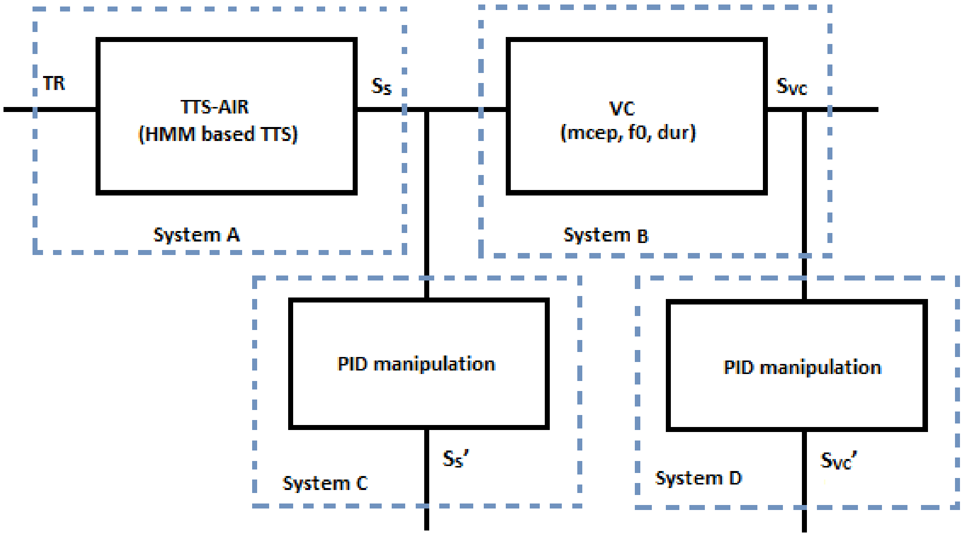


Figure 4.4. Block Diagram Representation of Systems A, B, C and D

'boirhaan' meaning 'rain' in Nalbaria, the syllables are 'boir' and 'haan' do not exist in the syllable inventory of the AIR variety of Assamese. Therefore the TTS system breaks them into the corresponding phonemes and uses the respective models to generate output speech, resulting in poor output quality. We attempted to generate the Nalbaria utterances by a HMM based TTS system (IITG-HTS) developed at IITG[2]. In order to normalise the effects of the vocal tract system and also to nullify the effects of speaker-dependent features, we finally decided to use training data for both systems A and B from a single speaker SPK, who is fluent in both AIR and Nalbaria. This required building an entirely new TTS system trained with speech data from speaker SPK. Therefore an HTS system which simultaneously models spectrum, excitation, and segment duration using context-dependent HMMs, is developed for our proposed system using training data in the AIR variety from SPK.

HMM based speech synthesis is a statistical parametric model that extracts speech parameters from the speech corpus, trains the system and produces sound equivalent to the input text. Advantages of this method include its ability to synthesise speech with various speaker characteristics/speaking styles and low memory requirements as it does not require recording of large databases. Adaptation to new speakers and speaking styles is simple as it involves modification of HMM parameters using relevant techniques. To develop such an HMM based system the tool

[2] `http://www.iitg.ernet.in/cseweb/tts/tts/Assamese/HTS/IITG_Assamese_MaleHTS/`

used is the HTS toolkit.

A typical HMM-based speech synthesis system consists of two parts, a training part and a synthesis part, as is shown in Figure 2.6 of Chapter 2. The training part extracts both spectrum and excitation parameters from a speech database and models context dependent HMMs taking into account phonetic, linguistic and prosodic contexts. In the synthesis part, text to be synthesised is converted to a sequence of context-dependent labels and an utterance HMM is constructed by concatenating context-dependent HMMs according to this label sequence. State duration probability functions are used to determine the state durations of the utterance HMM. A speech parameter algorithm is then used to generate the sequence of spectral and excitation parameters that maximize their output probabilities. Finally, a speech waveform is generated directly from the spectral and excitation parameters using a speech synthesis filter.

4.5.1.1 Speech Database for the HTS system

The speech corpus for our HTS (TU-TTS) is built using approximately 45 minutes of speech recorded from SPK. The text prompts (TP-TTS) are prepared with care to include most of the frequently occurring words and also all phones of the AIR variety, with a minimum frequency of 10. The text prompts are prepared mainly by collecting text from short stories and essays in Assamese. The text prompts are then recorded from the SPK at a sampling rate of 48kHz using a Zoom H4Next recorder in a sound proof room and saved in the '.wav' format. During recording care is taken to prevent DC wandering or baseline wandering, by using the unidirectional Zoom H4Next recorder with a pop filter mounted in front of the microphone. A pop filter serves to reduce or eliminate popping sounds caused by the mechanical impact of fast-moving air on the microphone during speech recording. The popping sounds lead to wandering of the DC-bias or base axis of the signal. This data is broken up into wave files of 6-8s resulting in around 400 wave files which are preprocessed to remove unwanted pauses, noise, pronunciation errors and clipping. The Assamese phoneset used, is prepared at IITG and consists of twenty-four consonants and eight vowels. Phone level transcriptions are carried out using the HTK label editor. Phone boundaries are further manually corrected by a person well versed in Nalbaria and then cross checked by another.

4.5.1.2 Building the HTS

The HMM-based Speech Synthesis System toolkit[3] is developed by the HTS working group and others, and is implemented as a modified version of the Hidden Markov Model Toolkit (HTK). HTS is designed to be patched to HTK and is released under a free license although it requires the user to obey the license of HTK to which it is patched. HTK[4] is a portable toolkit for building and manipulating HMMs. It is mainly used for research in speech recognition although it is also widely used in other applications such as research into speech synthesis, character recognition and DNA sequencing. To build an HMM voice, utterance files consisting of textual features and duration of each unit in the text to be synthesised, is required. These utterances need to be built in Festival. Festvox is required to generate these utterances in Festival. So together with $HTS-2.2$, $HTK-3.2.1$ and $HDecode-3.4.1$, $festvox-2.5.3$, $festival-2.1$, $SPTK-3.5$, $speech\_tools-2.1$, $ActiveTcl8.4.19.6$ and other support software tools are also installed to complete the set up for the TTS experimentation. All the software are freely downloadable from their respective websites.

Some of the basic components required for building the HTS are listed below :

1. Text data in the language: This refers to the text prompts that are used for recording speech data to be used for training the TTS. The text data (first 100 utterances) used for building the HTS is stored in the file 'txt.done.data' and is included as Appendix A.

2. Speech data corresponding to the text: This refers to the speech wav files ('.wav') corresponding to each of the utterances in the text data file.

3. Time-aligned phonetic transcription: The phonetic transcription together with the phoneme boundaries and duration of each phoneme in a speech file is stored in a corresponding label file ('.lab'). For a sample speech file 'TR_AIR_1.wav', the text data is " ek raxjaar dujoni raani aasil" and the corresponding label file is 'TR_AIR_1.lab' and is included as Appendix B.

[3] http://hts.sp.nitech.ac.jp/

[4] http://htk.eng.cam.ac.uk/

4. A phoneset for the language: The phoneset or phoneme set, i.e., the list of phones, for developing the HTS for Assamese consists of a total of 32 phones, i.e., 8 vowels and 24 consonants as can be seen in Figure 4.5. The phone features are defined in a separate file with features such as whether it is a vowel (v) or consonant (c), vowel height (high, mid, low), vowel length (short, long, diphthong, schwa), vowel frontness (front, mid, back), consonant type (stop, fricative, affricate, nasal, liquid), etc.

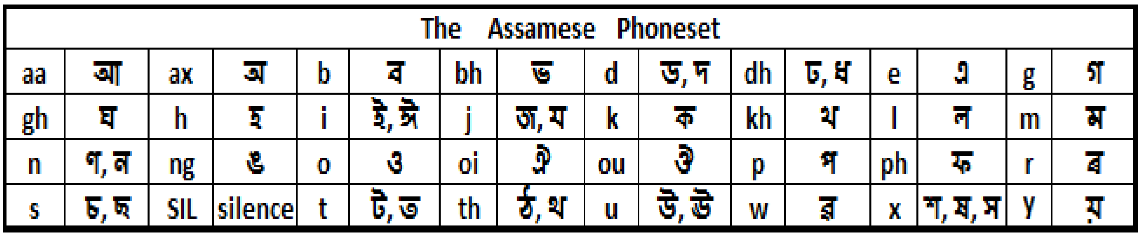
5. A set of letter to sound rules (L2S), Lexicon and Post-lexical rules: A lexicon is a long list of words together with their pronunciations. If a word is not found in the lexicon, L2S rules are used as a backup to to get its pronunciation. Post lexical rules are a general set of rules which can modify the segment relation after the basic pronunciations have been found. These rules operate on the phoneme string which is output by the L2S module in an attempt to rewrite the string to account for contextual effects when the word is spoken in context and not in isolation.

6. A set of syllabification rules: These are a set of rules used to syllabify the words in the utterances. A good set of syllabification rules is highly important in the development of a syllable-based TTS system like the one we have developed for Assamese.

7. Context specific features to be added to each of the context dependent phones. For each of the context dependent phone, context specific features such as phoneme identity before the previous phoneme, previous, current and next phoneme identity, whether previous, current and next syllable is stressed or not, number of syllables in the previous, current and next word, etc. can be found in the context-dependent label format for HMM-based speech synthesis[5].

8. A question set to be used during the training phase of context dependent phone models. A question set is a primary requirement for tree-based clustering in an HMM-based speech synthesis system. Relevant linguistic and

[5] http://www.cs.columbia.edu/~ecooper/tts/lab_format.pdf



The Assamese Phoneset

aa	আ	ax	অ	b	ব	bh	ভ	d	ড,দ	dh	ঢ,ধ	e	এ	g	গ
gh	ঘ	h	হ	i	ই,ঈ	j	জ,য	k	ক	kh	খ	l	ল	m	ম
n	ণ,ন	ng	ঙ	o	ও	oi	ঐ	ou	ঔ	p	প	ph	ফ	r	ৰ
s	চ,ছ	SIL	silence	t	ট,ত	th	ঠ,থ	u	উ,ঊ	w	ৱ	x	শ,ষ,স	y	য়

Figure 4.5. Phoneset for Assamese

phonetic classifications that influence the acoustic realisation of a phone are included in this question set.

The phoneset, letter to sound rules, question set, syllabifier etc. for the Assamese language, has been acquired from the Speech Processing Group at IITG. The basic files required for HTS building has been developed by Carnegie Melon University (CMU). We have made slight modifications to suit our needs.

Spectral parameters include cepstral coefficients and their dynamic features while excitation features include fundamental frequency (F0) and its dynamic features. 105 dimensional mel generalized cepstral coefficients (MGC), 3 dimensional log-F0 and composite features are extracted from raw files and are used to model context dependent phoneme HMMs using HTK. The training of phone HMMs using pitch and Mel cepstrum simultaneously, is enabled in a unified framework by using multi-space probability distribution HMMs and multi-dimensional Gaussian distributions. This simultaneous modelling of pitch and spectrum results in the set of context dependent HMMs. Decision based clustering based on the Minimum Distance Length (MDL) criterion is applied in isolation to MGCs, log-F0 and state durations of context dependent phoneme HMMs. In addition to tying contextual factors, the MDL clustering technique also generates spectrum and excitation parameters of newly observed vectors. The EM algorithm is further used to re-estimate clustered context dependent phoneme sequences.

In the testing part, text to be synthesised is input, analysed and transformed to a sequence of context dependent labels, using the *'dumpfeats'* function of Festival. Using this label sequence, trained context dependent HMMs are cascaded to generate the sentence HMM. State durations of the label sequence are determined by the state duration distributions. The speech parameter generation algorithm uses labels from the text analyser in Festival and context dependent HMMs to generate spectrum and excitation parameters which are further used by the Mel

Log Spectrum Approximation (MLSA) filter to produce the corresponding speech waveform. It is observed that the output generated by this HTS with transcribed Nalbaria utterances as input, is of good quality, but lacks naturalness with respect to the target dialect. System A is also able to synthesise correctly the set of transcribed Nalbaria text corresponding to the set of test utterances provided as input. The IITG-HTS has some limitations which have been overcome by the TU-TTS. For example, the word 'maxi' is generated as 'mi' by IITG-HTS probably because it assumes /ax/ to be the inherent /ax/ with the consonant /m/ instead of being the primary vowel in the diphthong /axi/. Similarly IITG-HTS generates the word 'jaxaakaali' as 'jaakaali', 'naxkaxu' as 'naxku' for the same reason.

4.5.2 System B: The VC System

The VC system is implemented in two basic modules, the training and testing modules. These modules consist of four basic steps: acoustic modelling, alignment of features, development of a mapping function, and finally synthesis using converted features. During the acoustic modelling step, short-term spectral properties of the speech signal are captured into a low-dimensional feature vector, for both source and target speech signals. This is carried out using the speech processing tool SPTK 4.5.2.3. During the alignment step, source and target utterances are time aligned, typically in an automatic fashion by using Dynamic Time Warping (DTW)[6] or Hidden Markov Models (HMMs) [104]. During the mapping step, a mapping function is learnt using techniques such as vector quantization(VQ) [1], neural networks [97], and GMMs [142]. Both warping for alignment and building of mapping functions is implemented in MATLAB. The final step of synthesis using converted features is again carried out with SPTK. The raw files generated with SPTK are imported into Audacity, an audio software, and saved as wav files.

4.5.2.1 Selection of features for conversion

According to the source-filter theory of speech production, the speech signal is the result of a convolution between the source of excitation and the impulse response of a filter. The filter represents the acoustic effects of the vocal tract, which depends not only on the shape and size of the vocal tract but also on the positions of the

[6] `http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/`

articulators like lips, jaws and tongue, corresponding to the uttered sounds. At the segmental level Dialect specific information is observed in the form of unique sequences of the shapes of the vocal tract for producing the sound units. The shape of the vocal tract is characterised by the spectral envelope which can be represented by Mel Frequency Cepstral Coefficients (MCEPs). At the prosodic level, dialect specific knowledge is embedded in the duration patterns of syllable sequences and dynamics of pitch and energy contours. The pitch contour is often considered to be an important signal of linguistic stress. Therefore MCEPs that take human perception sensitivity with respect to frequencies into consideration are selected to represent the filter parameters while fundamental frequency (F0) estimates are selected as source parameters like most VC systems do. Furthermore, a prior analysis carried out to study durational differences between the two varieties of Assamese under consideration, show significant difference in mean vowel duration in the two varieties indicating that in addition to MCEPs and F0, vowel duration can also be considered for transformation.

4.5.2.2 Choosing a model for the mapping function

Although the basic models used for developing mapping functions for carrying out feature transformation in the VC framework are based on GMMs [2], ANNs [31], Dynamic Frequency Warping(DFW) [42] and Mapping Codebooks [1], a number of improvements on the basic models have been proposed by researchers for the conversion function such as the use of Radial Basis Function Neural Networks [98], Weighted Codebook Mapping [150], Codebook Mapping with fuzzy VQ and difference vectors [128], Eigen Voice GMM [141] and Trajectory-based conversion using GMMs with Global Variance [59]. However review of the state-of-art techniques for developing mapping functions have revealed that the most widely used approaches are based on GMMs. Likewise it is also seen that ANN based techniques are also becoming increasingly popular. One of the earliest techniques is the VQ method using codebooks which is also popular for its simplicity. Therefore these two models, i.e., GMM and ANN, together with VQ, have been selected for developing our mapping functions.

4.5.2.3 The Speech Processing Toolkit (SPTK)

The Speech Signal Processing Toolkit[7] is a suite of speech signal processing tools for UNIX environments, e.g., LPC analysis, Partial Correlation (PARCOR) analysis, LSP analysis, PARCOR synthesis filter, LSP synthesis filter, VQ techniques, and other extended versions of them. For our work, we have used SPTK for extracting cepstral coefficients as well as pitch values from our source and target training data, and also for generating speech using the converted features. The *'mcep'* function extracts mel-cepstral coefficients from data which is windowed with fixed length frames, the *'pitch'* command extracts the pitch values using the RAPT or SWIPE algorithm, the *'excite'* command generates an excitation sequence from the pitch period information in the '.pitch' file which is then passed through a synthesis filter to generate the utterance. When the pitch period is nonzero (i.e., voiced) the excitation sequence consists of a pulse train at that pitch; when it is zero (i.e., unvoiced) the excitation signal consists of Gaussian noise. Finally the 'mlsadf' command of SPTK derives a MLSA digital filter from mel-cepstral coefficients and is used to filter an excitation sequence and synthesise speech data.

4.5.2.4 Speech Database for the VC system

Most VC systems require a parallel database containing the same set of utterances recorded from the source and target speaker. In our case we are trying to apply VC techniques to synthetic dialectal data. The building of the speech corpus starts with designing TP-VC, a set of text prompts in Nalbaria consisting of approximately 5-8 words each, carefully selected to contain all the phonemes of the Assamese language. We choose to use a set of 50 text prompts for our VC system considering the fact that a GMM based VC system requires about 30-50 parallel utterances [31]. Moreover our aim is to find out whether VC can be used to incorporate naturalness into the synthesized dialectal speech. This does not require us to concentrate much on the efficiency of the VC system output which could be affected by the size of the training data. Our target set 'T' is the set of utterances recorded from speaker SPK in Nalbaria using TP-VC as text prompts, with a sampling frequency of 16kHz and resolution of 16 bits. We manually transcribe the Nalbaria utterances to form the set of transcriptions TR. The set of utterances, generated from TTS-AIR using TR as input with Nalbaria vocabulary and grammar, but AIR phonetics

[7] `http://sp-tk.sourceforge.net/`

and prosodic rules, is our source set, ‘S’. In effect, we have a hypothetical person speaking Nalbaria without the knowledge of the phonetics and prosodic rules of Nalbaria and this hypothetical speaker is our source speaker. MCEPs of the order of 21 and pitch values are extracted from both ‘T’ and ‘S’ using a shell script in SPTK 4.5.2.3 after aligning each pair of utterances using DTW. MCEPS are extracted using a Hanning window, with a frame size of 25ms and a frame period of 5ms resulting in a [22532 × 21] dimensional feature matrix for each of source and target MCEPs. The energy coefficient is not used in the conversion and therefore a [22532 × 20] dimensional feature matrix for both source and target MCEPs is used as training data while developing mapping functions for the spectral features using (i) VQ with codebooks (ii) GMMs and (iii) ANNs. Pitch values are extracted using the RAPT algorithm, with an overlap of 5ms with upper and lower limits of F0 defined. Segmentation of ‘T’ and ‘S’, to mark vowel/diphthong boundaries, is carried out using HTK. The boundaries are manually corrected and segments are annotated with the PRAAT tool[8]. Vowel/diphthong duration values are then extracted using a PRAAT script. Since the aim of this work is not to develop a VC system but to explore the scope of using VC techniques on synthesised dialectal speech to make it more natural, we simply select 10 random utterances from the training data as test utterances. This is S_s which will be generated by system A and used as input by the VC system, i.e., system B. The output of System B is denoted by S_{vc}.
In short:

TP-VC: Set of 50 text prompts in Nalbaria.

SPK: A speaker who speaks both the varieties of Assamese (Nalbaria and AIR) fluently.

T: Set of utterances in Nalbaria variety (target) recorded from speaker SPK using TP-VC.

TR: Set of phonetic transcriptions of T

TU-TTS: A standard Assamese TTS trained with standard Assamese speech data from speaker SPK

[8] `www.fon.hum.uva.nl/praat/`

S: Set of utterances in AIR variety (source) generated using TR as input, from TU-TTS

S_s: Set of test utterances generated by System A.

S_{vc}: Corresponding set of utterances (output) generated by System B.

4.5.2.5 Transformation of Spectral features using VQ with Mapping Codebooks

In this method, the conversion of spectral features from the voice generated by the TTS (source) to those of the voice of the speaker speaking Nalbaria (target), is reduced to the problem of finding a correspondence between the codebooks developed for source and target speech [1]. A codebook is generated for the MCEPs in source data and another for MCEPs in target data. The vector correspondences between source and target speakers are accumulated as histograms. Using the histogram for each codevector of the source as a weighting function, a mapping codebook from source to target is defined as a linear combination of the target speaker's vectors. During conversion, for the test vector, the closest centroid vector in the source vector is found and then mapped to the corresponding codevector in the target codebook. The mapping function so developed is tested with different number of centroids 'm', i.e., m=32, 64, 128, 256.

4.5.2.6 Transformation of Spectral features using GMM

The VC algorithm based on GMMs was proposed by Stylianou et al. [135]. Such a transformation, aims to fit a GMM model to the augmented source and target feature vectors. During the training phase, the GMM is adopted to model the distribution of the paired feature sequence z_t, which represents the joint feature vector of source speech vector x_t and target speech vector y_t at frame t. The joint probability density is given as follows:

$$P(z_t|\lambda^{(z)}) = \sum_{m=1}^{M} w_m N(z_t; \mu_m^{(z)}, \Sigma_m^{(z)}) \tag{4.1}$$

M is the total number of mixture components and w_m is the weight of the m^{th} mixture component. $\lambda^{(z)}$ represents the GMM parameter set consisting of weights, means, covariance matrices for individual mixture components. Mean vector $\mu_m^{(z)}$ and covariance matrix $\Sigma_m^{(z)}$ of the m^{th} Gaussian component $N(z_t; \mu_m^{(z)}, \Sigma_m^{(z)})$, can be expressed in terms of mean source and target vectors $\mu_m^{(x)}$, $\mu_m^{(y)}$, covariance matrices of source and target feature vectors $\Sigma_m^{(xx)}$, $\Sigma_m^{(yy)}$, cross-covariance matrices of source and target feature vectors $\Sigma_m^{(xy)}$, $\Sigma_m^{(yx)}$ all corresponding to the m^{th} Gaussian component, in the following manner:

$$\mu_m^{(z)} = \begin{vmatrix} \mu_m^{(x)} \\ \mu_m^{(y)} \end{vmatrix}, \Sigma_m^{(z)} = \begin{vmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{vmatrix}, \tag{4.2}$$

The Expectation Maximization (EM) algorithm is used to train the GMM with the joint source and target vectors aligned with DTW to yield highly robust parameters. The conditional probability density of y_t given x_t can be represented as a GMM as follows:

$$P(y_t|x_t, \lambda^{(z)}) = \sum_{m=1}^{M} P(m|x_t, \lambda^{(z)}) P(y_t|x_t, m, \lambda^{(z)}) \tag{4.3}$$

where

$$P(m|x_t, \lambda^{(z)}) = \frac{w_m N(x_t; \mu_m^{(x)}, \Sigma_m^{(xx)})}{\sum_{n=1}^{M} w_n N(x_t; \mu_n^{(x)}, \Sigma_n^{(xx)})} \tag{4.4}$$

$$P(y_t|x_t, m, \lambda^{(z)}) = N(y_t; E_{m,t}^{(y)}, D_m^{(y)}) \tag{4.5}$$

Mean vector $E_{m,t}^{(y)}$ and covariance matrix $D_m^{(y)}$ of m^{th} conditional probability distribution is:

$$E_{m,t}^{(y)} = \mu_m^{(y)} + \Sigma_m^{(yx)} \Sigma_m^{(xx)^{-1}} (x_t - \mu_m^{(x)}) \tag{4.6}$$

$$D_m^{(y)} = \Sigma_m^{(yy)} - \Sigma_m^{(yx)} \Sigma_m^{(xx)^{-1}} \Sigma_m^{(xy)} \tag{4.7}$$

The converted target feature vector $\hat{y}_t$ is given by

$$\hat{y}_t = E[y_t|x_t] = \sum_{m=1}^{M} P(m|x_t, \lambda^{(z)}) E_{m,t}^{(y)} \tag{4.8}$$

The mapping function, so learnt, is used to convert MCEPs of a test utterance and is evaluated with different numbers of Gaussian mixtures ranging from 16 to 128 (M=16, 32, 40, 64, 72, 128).

4.5.2.7 Transformation of Spectral features using ANN

Multi-layer feed forward neural networks for mapping the MCEPs (20th order, excluding the energy coefficient), are used to capture the non-linear relations between acoustic features. These neural net models consist of interconnected nodes, each node representing the model of an artificial neuron. The interconnection between two such nodes has a weight associated with it. ANNs consist of multiple layers, each performing a mapping (usually non-linear) of the type y = f(Wx + b) where 'f' is called the activation function that can be implemented either as a sigmoid, a tangent hyperbolic, rectified linear units, or as a linear function. 'W' and 'b' are the weight and bias for that particular layer. The input and output size are usually fixed depending on the application while the size of the middle layer and the activation functions are chosen depending on the experiment and data distributions [96].

A feed forward neural network can be designed to perform the task of pattern mapping. Acoustic features of the source speaker is given as input to the network, while that of the target speaker is given as output, during the training process. These two data sets are used by the network to learn and capture a non-linear mapping function based on the minimum mean square error. In an attempt to minimize the mean squared error between the desired and the actual output values, the weights of the neural network are adjusted using Generalized back propagation learning [9]. Optimization parameters in training include the selection of initial weights, the architectures of ANNs, learning rate, momentum and number of iterations. On completion of the training process, we get a weight matrix that represents the mapping function between the acoustic features of the given source and target speakers. This weight matrix can be used to predict acoustic features of the target speaker from acoustic features of the source speaker. Different network structures are possible by varying the number of hidden layers and the number of nodes in each of the hidden layer.

Various network architectures with different parameters are experimented in MATLAB, before finally settling for the 5 layer network 20L-60N-60N-60N-20L shown in Figure 4.6, yielding the best results. The first and last layers are the input-output layers, with linear units (L), having the same dimension as that of input-output acoustic features (20 in our case, excluding the zeroth coefficient in the 21st order mceps). The second layer (first-hidden layer), third layer (second-hidden

[9] `https://en.wikipedia.org/wiki/Backpropagation`

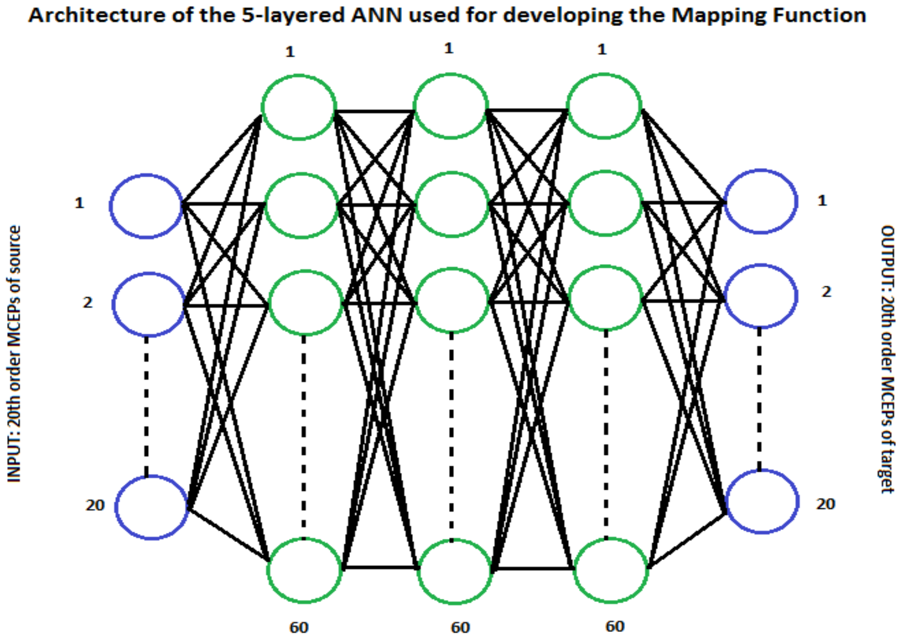

Architecture of the 5-layered ANN used for developing the Mapping Function
INPUT: 20th order MCEPs of source
OUTPUT: 20th order MCEPs of target
1
2
20
1
1
1
60
60
60
1
2
20

Figure 4.6. 5-layer ANN architecture with 20 input and 20 output nodes

layer) and fourth layer (third-hidden layer) have non-linear nodes (N), which help in capturing the non-linear relationship that may exist between the input-output features. The neural network is then trained with the number of epochs set to 200.

4.5.2.8 Transformation of Pitch and Duration

Glitches in the TTS generated F0 contours are observed which resulted in incorrect values of source mean and variance. The F0 contours of ‘S’ are therefore corrected manually by comparing them with corresponding target F0 contours of ‘T’ before calculating the mean and variance values from source and target F0 data. The F0 contour of a test utterance is then transformed to match that of the target speaker using the commonly used mean-variance formula 4.9.

$$f0_{conv} = \mu_{tgt} + \sigma_{tgt}/\sigma_{src}(f0_{src} - \mu_{src}) \tag{4.9}$$

where μ_{src} and σ_{src} are the mean and variance of the fundamental frequencies for the source speaker, μ_{tgt} and σ_{tgt} are the mean and variance of the fundamental frequencies for the target speaker, $f0_{src}$ is the speaker pitch of a test utterance and $f0_{conv}$ is the converted pitch for the target speaker.

F0 conversion is also carried out by a GMM based mapping function. A GMM is trained using the EM algorithm with the joint source and target F0 feature vectors aligned with DTW in a similar way as already described in section 4.5.2.6 for MCEP conversion.

In order to carry out the transformation of segmental duration from source to target, the first step is to calculate a scaling factor for normalisation of segmental durations. Durations of test utterances are modified to match that of the target by using a scaling factor K. K is the ratio of the average duration of phonemes in the source and target training data. The durations of the test utterances are normalised by multiplying the number of phonemes in a source utterance by the scaling factor K. In the second step, the durations of vowels and diphthongs in the normalised test utterances are further transformed to match that of the corresponding target using the mean-variance formula 4.9 with $f0_{conv}$ and $f0_{src}$ replaced by dur_{conv} and dur_{src} respectively. Thus this process of segmental transformation is a two-fold process which is seen to perform better than simply transforming the durations using either the scaling factor or the mean-variance method.

4.5.3 Systems C and D: PID Manipulation

Prosodic features or suprasegmental features, play an important role in bringing naturalness to speech and variation of such features have been observed across dialects in a number of studies. The mapping functions we have used in System B, converts the spectral features, ie, MCEPs, F0 global range and vowel/diphthong durations from source to target. In order to bring about the local F0 variations and also the variations in intensity and segmental durations in a more accurate manner, we have used the PRAAT tool for PID (P: pitch, I: intensity, D: duration) manipulation. However these manipulations can be carried out for a test utterance only when the target utterance is known since the manipulations are to be carried out to match the source PID values to the target PID values. These manipulations are therefore for experimental purposes only where the aim is to study the effect of PID manipulation on the outputs of Systems A and B.

The mean intensity of the test utterance is adjusted to match the mean intensity of the target utterance. Using the PRAAT tool the mean intensity of the target utterance is measured. The intensity of the test utterance is then scaled to match that of the target. The durations of the vowels and diphthongs in the test utterance are then modified to match their counterparts in the target utterance by adding duration points into the duration manipulation object and then modifying the duration points accordingly to lengthen or shorten the segment. PRAAT allows replacement of the F0 contour of an utterance by another F0 contour. Therefore in the final step, the test F0 contour is replaced with the target F0 contour. Manipulation of prosodic features on the outputs of System A is carried out in System C which results in output $S_{s'}$. Similar manipulations are carried out on the outputs of System B in System D resulting in output $S_{vc'}$ as can be seen in Figure 4.2.

4.6 Results and Evaluation

Output speech from the four systems, i.e., A, B, C and D, are evaluated and compared.

4.6.1 Results of spectral conversion

MCEP conversion from source to target is carried out with mapping functions developed using VQ codebooks, joint probability GMMs as well as neural networks. Since the GMM based mapping function gave better results in terms of Mel Cepstral Distortion (MCD), it is chosen to convert the MCEPs of the test utterances. Figure 4.7 shows the MCEPs of a test utterance plotted against the MCEPs of the corresponding target while Figure 4.8 shows the converted MCEPs plotted against the MCEPs of the corresponding target utterance. MCD is an objective error measure known to have correlation with subjective test results. The smaller the value, the better the conversion. It is essentially a weighted Euclidean distance defined as:

$$MCD = (10/ln10) * \sqrt{2 * \sum_i (mc_i^t - mc_i^e)^2}$$

where mc_i^t and mc_i^e denote target and estimated MCEPs respectively.

MCD values are calculated between MCEPs of test and target utterances (mcd_tt) and between corresponding converted MCEPs and MCEPs of target utterance (mcd_mt). The conversions are carried out via (i) VQ with mapping codebooks having 'm' centroids, (ii) the joint-density GMM method with different number of mixture components 'M' and (iii) by the ANN method with varying number of hidden layers 'h' and number of neurons 'n'.

Table 4.1, Table 4.2 and Table 4.3 presents a comparison of MCD scores for five of the test utterances (Test 1 to Test 5) using (i) VQ with codebooks (with m=32, 64, 128,256 centroids), (ii) GMM (with M=40, 64, 72, 128) and (iii) ANN (with h=2, n=40; h=2, n=60; h=3, n=60) mapping functions respectively. It can be seen that the GMM method with M=128, outperforms both the VQ method as well as the ANN method for the three different architectures used in our work. The best results for the test utterances with the three methods are compared in Figure: 4.9.

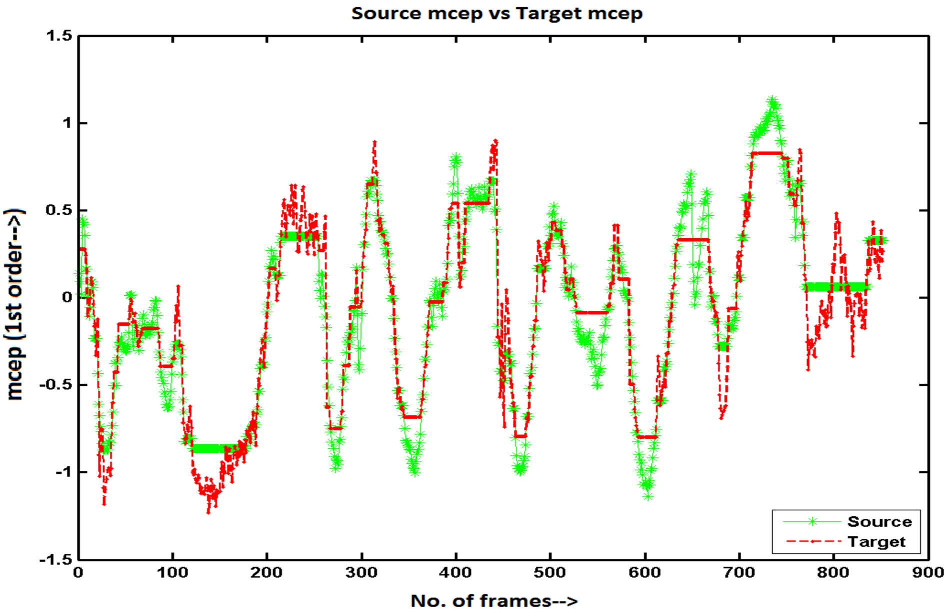


Figure 4.7. Source vs Target MCEPs of a Test Utterance

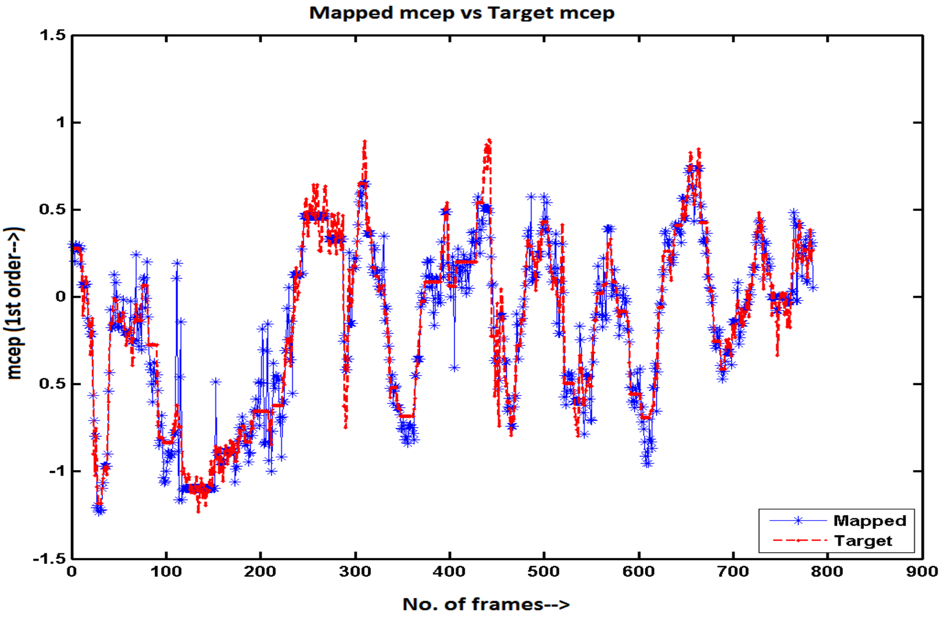


Figure 4.8. Converted vs Target MCEPs of a Test Utterance

Table 4.1 Comparison of MCD values before and after conversion of MCEPs via the VQ method

Utterance	mcd_tt	mcd_mt	mcd_mt	mcd_mt	mcd_mt
		(m=32)	(m=64)	(m=128)	(m=256)
Test 1	11.7	12.1	11.7	11.4	10.5
Test 2	11.8	12.2	11.4	11.3	10.9
Test 3	16.5	15.4	15.1	14.6	14.3
Test 4	18.4	17.3	16.6	15.9	15.2
Test 5	11.8	11.2	11.0	10.4	9.4

Table 4.2 Comparison of MCD values before and after conversion of MCEPs via the GMM method

Utterance	mcd_tt	mcd_mt	mcd_mt	mcd_mt	mcd_mt
		(M=40)	(M=64)	(M=72)	(M=128)
Test 1	11.7	8.5	8.6	8.1	7.3
Test 2	11.8	8.5	7.7	7.4	6.0
Test 3	16.5	11.9	11.5	12.0	9.9
Test 4	18.4	13.2	12.2	12.7	10.8
Test 5	11.8	8.3	7.8	7.9	7.0

Table 4.3 Comparison of MCD values before and after conversion via the ANN method

Utterance	mcd_tt	mcd_mt	mcd_mt	mcd_mt
		(h=2,n=40)	(h=2,n=60)	(h=3,n=60)
Test 1	11.7	8.9	8.5	7.6
Test 2	11.8	9.5	7.4	7.2
Test 3	16.5	12.3	11.5	10.0
Test 4	18.4	13.4	12.3	11.5
Test 5	11.8	9.0	7.9	7.6

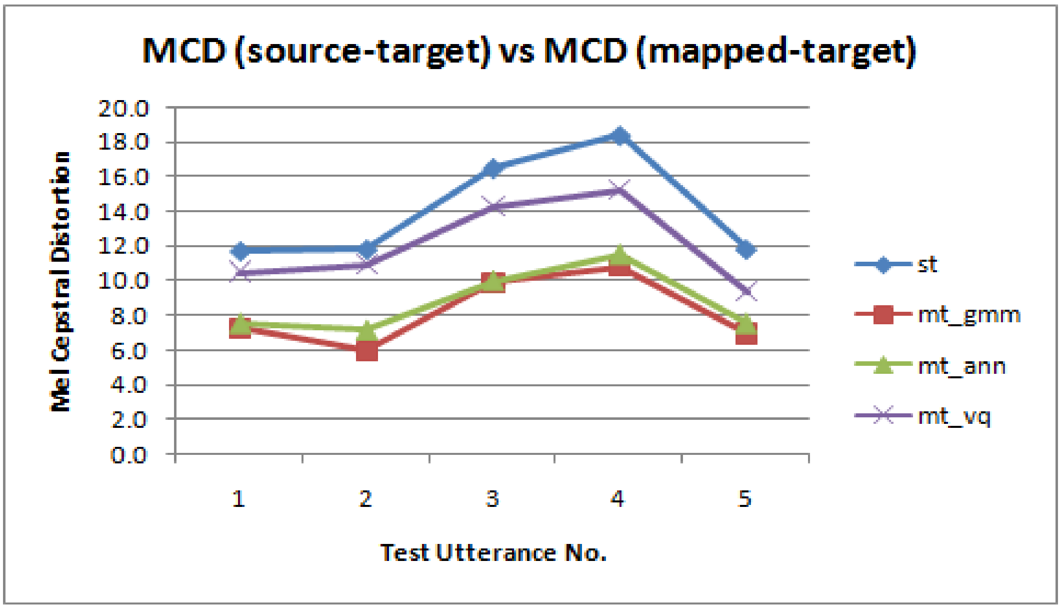


Figure 4.9. Comparison of MCDs with the 3 methods

4.6.2 Results of F0 conversion

The prediction accuracies of both the linear mean-variance mapping function and the GMM based mapping function used for predicting F0 values are evaluated using the objective measures of root mean square error (RMSE). RMSE is calculated with the following equation:

$$RMSE = \sqrt{(\sum_{n=1}^{N}(f0_n^t - f0_n^c)^2)/N}$$

where $f0^t$ and $f0^c$ are the target f0 and converted f0 respectively for each voiced frame and N is the total number of frames per utterance.

The conversion using the linear mean-variance method showed almost no improvement. While the second conversion using the GMM based mapping function with 64 number of mixture components, resulted in the lowering of RMSE values, by converting the global range of F0 and to some extent the local variations as well. Results of F0 conversion, for a test utterance is shown in Figure 4.10 and Figure 4.11.

RMSE values between test and target f0 (rmse_tt), between corresponding converted and target f0 using mean-variance method (rmse_mt_lmv) and between corresponding converted and target f0 using the GMM method (rmse_mt_gmm) for

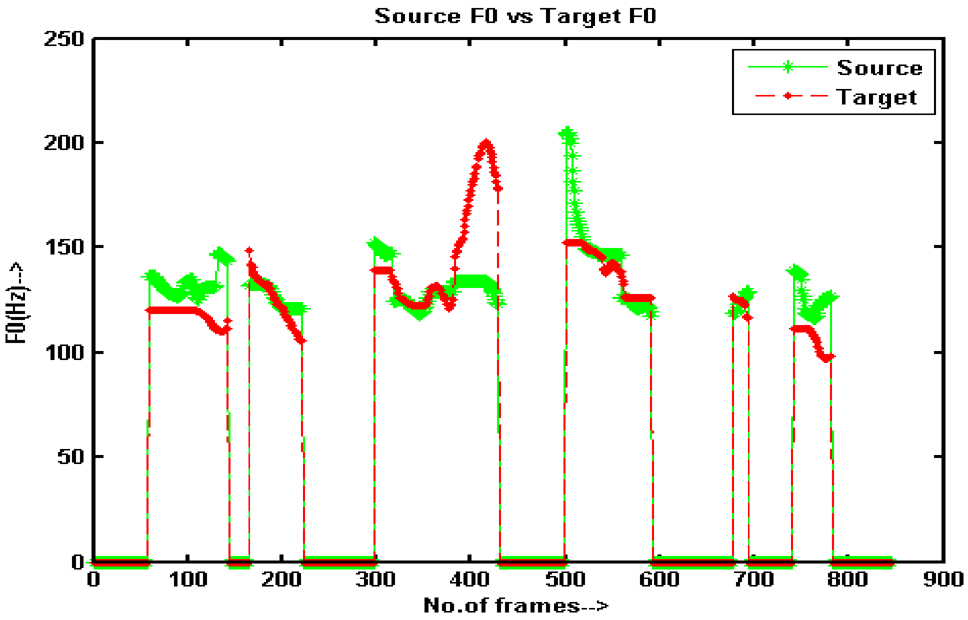


Figure 4.10. Source F0 contour vs Target F0 contour

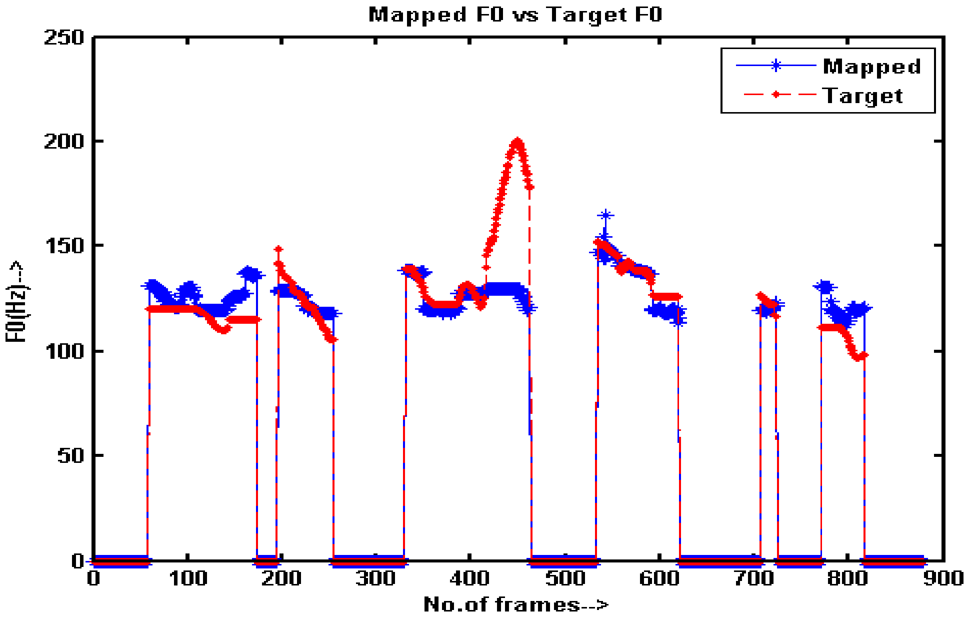


Figure 4.11. Mapped F0 contour vs Target F0 contour

five of the test utterances (Test 1 to Test 5) are presented in Table 4.4.

4.6.3 Results of duration conversion

Mean and standard deviation values of the duration of vowels/diphthongs in both source and target training data are represented in Table 4.5. Duration of vow-

Table 4.4 Comparison of RMSE(F0) values

Utterance	rmse_tt	rmse_mt_lmv	rmse_mt_gmm
Test 1	19.3	19.3	13.0
Test 2	15.1	15.4	13.8
Test 3	29.8	29.5	27.3
Test 4	17.3	16.6	14.3
Test 5	26.0	25.0	24.0

els/diphthongs in target data, i.e., in Nalbaria, is seen to deviate more from the mean. The bar diagram in Figure 4.12 shows the results of the conversion of vowel/diphthong durations in the test utterance ("tor bhuk nalgiliu khaba lagbo"). Durations of vowels/diphthongs in order of occurrence in the test utterance, before and after the conversion are plotted. It can be seen that for most of the segments, the converted vowel/diphthong duration is closer to that of the target.

Table 4.5 Mean and Standard Deviation(StdDev) of duration of vowels in Source and Target data

Vowel	Mean	StdDev	Mean	StdDev
(IPA)	(source)	(source)	(target)	(target)
/ɔ/	0.07	0.022	0.07	0.030
/a/	0.08	0.023	0.07	0.030
/i/	0.06	0.020	0.09	0.050
/u/	0.06	0.022	0.07	0.040
/e/	0.06	0.020	0.07	0.030
/oi/	0.12	0.038	0.08	0.037
/o/	0.07	0.034	0.07	0.020
/ou/	0.10	0.052	0.12	0.042

4.6.4 Subjective evaluation using MOS

Since the GMM method is seen to outperform the ANN method in the conversion of MCEPs from source to target, it is used convert the source MCEPs to match that of the target. The converted MCEPs and F0 values of the ten test utterances, are used to resynthesise the test utterances in SPTK. The resynthesised raw files

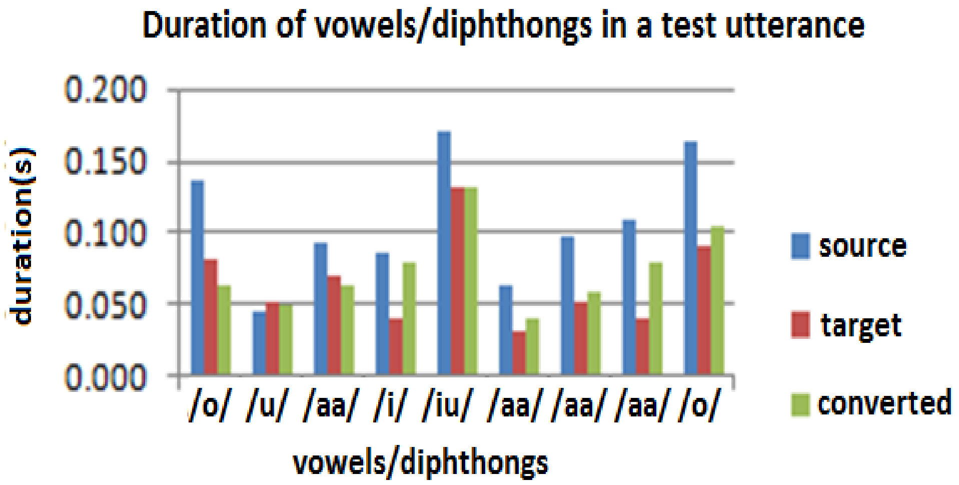


Figure 4.12. Transforming vowel/diphthong durations of a Test Utterance

are imported by the Audacity software [10] whereby they are saved as wav files. A Mean Opinion Score (MOS) test is carried out on the synthesised utterances with and without prosodic modification. The MOS is calculated as the arithmetic mean over single ratings (R) performed by human subjects for a given stimulus in a subjective quality evaluation test[11] as shown in Equation 4.10, where 'N' is the number of evaluators, (N=8). A total of 10 sets of utterances are given to each of the 8 evaluators, 5 male and 3 female, who are well versed in the Nalbaria variety of Assamese. Each set consists of 5 utterances, (1) the HTS generated utterance (System A), (2) the HTS generated utterance with PID modification (System C), (3) the HTS generated utterance after using VC (System B), (4) the HTS generated utterance after using VC with PID modification (System D) and finally (5) the target utterance spoken in Nalbaria. Therefore there are a total of 50 utterances which are to be given scores by the 8 evaluators. Finally the average MOS for each of the four systems, i.e., Systems A, B, C and D, is calculated using Equation 4.11, where 'M' is the total number of sets of utterances (M=10).

$$MOS = \sum_{n=1}^{N} R_n/N \tag{4.10}$$

$$MOS_{avg} = \sum_{m=1}^{M} MOS_m/M \tag{4.11}$$

Since the conversion and prosody modification are carried out on TTS gener-

[10] https://www.audacityteam.org/

[11] https://en.wikipedia.org/wiki/Mean_opinion_score

Table 4.6 MOS Results

Sl.No.	System	Utterance Type	Score(avg)
1	System A	HTS	2.3
2	System C	HTS + PID	2.9
3	System B	HTS + VC	3.2
4	System D	HTS + VC + PID	4.0

ated utterances, the final utterances are slightly noisy. However our aim is to find out how close the converted and manipulated utterances are to Nalbaria in terms of naturalness, and therefore the evaluators are asked to listen to each of the 5 utterances in each set, compare the first four utterances in each set to the target utterance and give a score using a 5-point scale in terms of naturalness, based on the question ‘Which utterance is closest to the target Nalbaria utterance?’ or ‘Which utterance is most likely to be spoken by a Nalbaria speaking person?’. Results of the MOS test are presented in Table 4.6 while Figure 4.13 provides the details of the test.

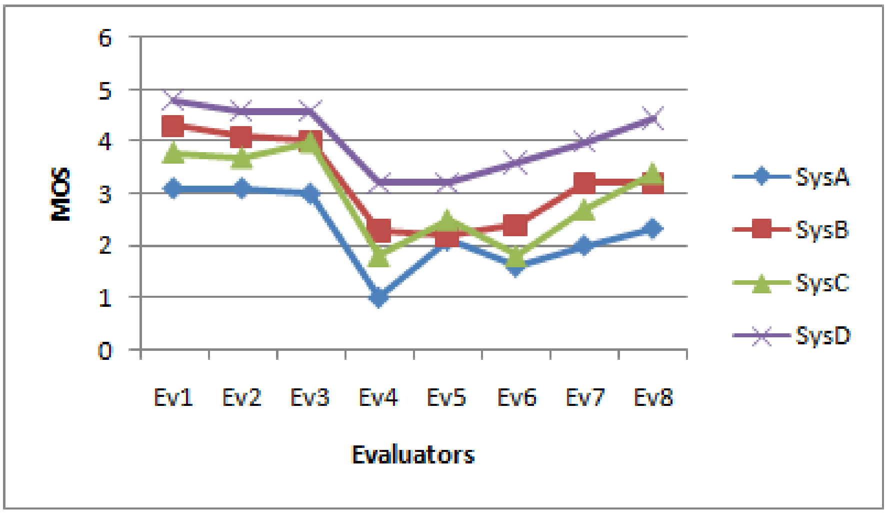


Figure 4.13. Details of the MOS test

An analysis of variance (ANOVA) test is also carried out on the MOS results to find out whether the effect of the different systems on the MOS scores is significant and at the same time to find the consistency of the scores given by the different evaluators. For this the MOS scores are first normalised and inserted into a table with columns representing the different evaluators and rows representing the different systems, i.e., A, B, C and D. A two-way ANOVA test is then carried out on this data. The results show that the between group (i.e., between rows in the above mentioned table) P-value is equal to 1.9x10^{-8} with F(=57.3) much greater than F_{crit}(3.29). This indicates that the type of systems used for generating the test utterances have a significant effect on the quality of the generated speech. The within group (i.e., between columns in the above mentioned table) P-value is equal to 0.376 (greater than 0.01) which indicates that the MOS scores assigned to the outputs of the different systems by the different evaluators is consistent and do not show any significant differences.

4.7 Summary

The GMM based mapping function outperformed the VQ based mapping function as well as the ANN based mapping function while converting MCEPs from source to target. However listening tests performed on the resynthesised test utterances using converted MCEPs using the three different mapping functions, and F0, indicate that there is a degradation in quality in the test utterances using MCEPs converted by the VQ based mapping function; this is also indicated by the MCD values. MCD values indicate a better conversion of MCEPs with the GMM based mapping function. However there is no perceptual difference between the resynthesised test utterances using MCEPs converted by either the GMM based or ANN based mapping functions. The GMM based method of F0 conversion converted the global F0 range while the local variations of target contours could be achieved only to a very minimal extent. This does not have much effect on a normal VC system. However, our aim is to convert speech from one dialect to another and the prosody of a dialect is reflected in the local variations of the F0 contour. Therefore inaccurate mapping of the local variations resulted in lack of naturalness in the converted utterances. Duration conversion using the mean-variance method after scaling with factor K, produced good results although it is expected that inclusion of contextual information such as position of vowel/diphthong segments in the word/phrase and stress, would lead to better results.

Results of the subjective test presented in Table 5, show that farthest from the target utterances are the ones generated by TU-TTS, meaning that a TTS for the standard variety of a language is not the best option for generating dialectal speech. Closest to the target utterances are the ones where prosodic manipulations are carried out manually on the outputs of system B indicating that VC techniques may be used for Dialect Conversion provided the prosodic features such as pitch, intensity and duration (PID) are also converted appropriately. System B using VC, produces better results than system A and C (prosodic manipulations on the outputs of A). This is an interesting result since it implies that in addition to prosodic features, spectral features also carry paralinguistic information and play a crucial role in bringing the HTS (for the standard variety) synthesised utterance closer to the chosen dialect. Another point to note here is that training data for both systems A and B are from the same speaker SPK. Therefore the fact that System B produces better results than System A in terms of naturalness, implies that the conversion function used in System B converts speaker independent features such as linguistic features pertaining to the dialectal variants as well.

In addition to the experiments carried out in this chapter, an attempt has been made to implement the three VC mapping function discussed in this chapter into a tool. In order to use VC to convert one type of speech to another, researchers have to carry out a number of steps such as speech feature extraction, feature alignment, development of mapping functions, conversion of features, evaluating the efficiency of the conversion using various measures, graphical representation of results etc. Based on the results of conversion the researcher decides whether or not to use VC for his work, or which mapping function to use. However this requires in-depth knowledge of each and every step and each and every mapping technique as well as intensive coding which is not only tedious but also highly time consuming. In an effort to ease the researcher of such tasks we have built a MATLAB GUI based tool which has been named as VoiCon. VoiCon carries out the various steps required for the VC process and at the same time also allows the researcher to analyze his results both objectively and graphically.

The present work mainly concentrates on using VC to convert speech synthesised by a TTS of the standard variety to target speech in the Nalbaria variety of Assamese. Although the findings are positive, local variations in prosodic features are necessary in order to bring the converted speech closer to the target dialect. One difference that has been observed in the speech of these two varieties of Assamese is that some of the vowel/diphthong sounds are different. This has also been proved by the results of the feature analysis in Section 3.3 whereby the formant spaces for the vowels and dynamic formant trajectories of diphthongs are seen to be different in the two varieties. The VC approach described in this chapter is also capable of shifting formants. However in order to convert the formant contours of vowels/diphthongs exclusively, since the vowel sounds in the two varieties of Assamese under study are seen to vary considerably, it may be beneficial to concentrate on the transformation of vowel/diphthong formants. So the next phase of work, elaborated in Chapter 5, is directed towards transforming the vowel/diphthong formant space from one variety to another, in order to make the vowels/diphthongs of one variety sound more like their counterparts in the other variety.