TÉCNICO LISBOA

Exploration of Audio Feedback for L2 English Prosody Training

Pedro Miguel Sonso Sousa

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Isabel Maria Martins Trancoso
Dr. Xavier Anguera

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Isabel Maria Martins Trancoso
Member of the Committee: Dr. Paula López Otero

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I would like to thank my supervisors, Prof. Isabel Trancoso and Dr. Xavier Anguera, for their support throughout this thesis. A special thanks in particular to Prof. Isabel Trancoso, for sparking the interest in the this specific subject during Speech Processing classes and for the continuous help and guidance throughout the whole process, from the choice the subject of the thesis to the delivery of the final version. I also extend my thanks to everyone at ELSA Corp., who provided me with every tool and information I required, and to Ivan Carapinha, who shared his experience with the VC algorithm.

I would also like to thank all my friends that supported me throughout all the years in University. It would be much harder to complete the degree without your help, positivity and friendship, all the study sessions and celebrations. You made me feel at home, both in IST and in my time in Erasmus.

To my girlfriend, thank you for giving up the time that would be spent on us, for listening to my outbursts and for cheering me up at every step of the way, for being understanding, caring, and giving me all the needed support.

Lastly, I thank my family for always being there, and specially to my parents for going above and beyond to provide everything I could ever ask for. Thank you for your effort, for your guidance, and for giving me the tools to grow and educate myself. To my mum, I wish I could thank you once again, for everything.

Abstract

The increase in the amount of English language learners has made using mobile apps or websites to learn English a viable, easily accessible and widely used option by many. This work explores two different approaches to tackle prosody training in the context of Computer Assisted Language Learning.

Both methods are applied to an exercise from a language learning app, where the learner is given a sentence and a recording of a native speaker uttering this sentence. The learner then tries to read this sentence and replicate the prosodic targets from the native speaker utterance. The app records and processes the utterance, returning a feedback on how close the learner is to the target in terms of duration and pitch. The task will be complementing or replacing the utterance from the native speaker with an utterance in the voice of the learner.

The first approach consists on manipulating the user's speech. It will take the learner's attempt and correct the pitch and duration markers through speech analysis with a vocoder-based system and a time alignment algorithm. The second approach uses a Voice Conversion method to convert the native speaker's utterances to the voice of the leaner. By removing the voice difference, it is expected that the learning process will be more efficient.

Both approaches are implemented and preliminary results are provided. A subjective evaluation performed by a listening panel of 40 subjects is presented and a method for objective evaluation is proposed. The results reveal that the Voice Conversion approach seems the best choice for future development, given the VC algorithm is tailored for this specific task.

Keywords

Computer Assisted Language Learning (CALL), Prosody Training, Voice Conversion (VC), L2 Learning, Dynamic Time Warping

Resumo

O aumento do número de estudantes de Inglês fez com que o uso de aplicações móveis e websites para aprendizagem dessa língua se torne uma opção viável, acessível e largamente utilizada por muitos. Esta tese explora duas soluções diferentes para o treino da prosódia com recurso a computadores.

Ambos os métodos são desenvolvidos tendo em conta um exercício específico de uma aplicação para aprender Inglês, em que o estudante tem acesso a uma frase e a uma gravação de um falante nativo de Inglês a ler essa frase. O estudante ouve a gravação e lê a frase, tentando replicar os contornos prosódicos da gravação que ouviu. A aplicação avalia a proximidade do discurso do estudante relativamente à gravação do falante nativo, tendo em conta marcadores de duração e frequência fundamental.

A primeira abordagem consiste em manipular o discurso do utilizador. Usando a tentativa anterior do mesmo exercício, o algoritmo corrige os marcadores de duração e de frequência fundamental utilizando um sistema com base em tecnologia Vocoder e um algoritmo de alinhamento temporal. A segunda abordagem utiliza Conversão de Voz para converter a gravação do falante nativo para a voz do estudante. Ao remover as diferenças entre a voz do estudante e das gravações de referência, é expectável que processo de aprendizagem seja mais eficiente.

Ambas as abordagens são implementadas em código, permitindo a obtenção de resultados que serão avaliados subjetivamente com recurso a um painel de 40 juízes. Um método de avaliação objetiva também será apresentado. Os resultados favorecem a abordagem com recurso a Conversão de Voz, que tem maior margem para melhorias.

Palavras Chave

Computer Assisted Language Learning (CALL), Treino de Prosódia, Conversão de Voz, Aprendizagem de Segunda Língua, Dynamic Time Warping

Contents

List of Figures

List of Tables

Acronyms

CALL	Computer Assisted Language Learning
CAPT	Computer Aided Pronunciation Training
VT	Voice Transformation
VC	Voice Conversion
L2	Second Language
AI	Artificial Intelligence
PSOLA	Pitch Synchronous Overlap and Add
F0	Fundamental Frequency
DTW	Dynamic Time Warping
MRI	Magnetic Resonance Imaging
MOS	Mean Opinion Score
SNR	Signal-to-Noise Ratio
LPC	Linear Predictive Coding

1

Introduction

Contents

1.1 Motivation

English can be seen as the lingua franca of the world - it is a global language that strongly dominates all areas of international communication. Depending on the criteria that defines an English speaker, the more conservative numbers show 1.132 billion total speakers, from which 753.3 million (around 2 thirds) speak it as a second language (Second Language (L2)) [9]. David Crystal goes further, giving the astounding number of over 2000 million total speakers in 2008 [10], from which over 1600 million are L2 speakers, and updating it to over 2300 million [11] 10 year later, a number which keeps increasing everyday.

It is clear that English is a very widely spoken language and that the majority of its speakers are non native. According to the Merriam-Webster dictionary, a native speaker is a person who learned to speak the language of the place where he or she was born as a child, rather than learning it as a foreign language. Another approach is taken by Penycook [12], who states that a native speaker is a ”person with a complete and possible innate competence in the language”. This approach does not limit a language nativeness to the place of birth, but instead connects it with the natural ability of using this language, independently of how the subject came to this knowledge.

Today, the ability to produce highly intelligible English speech is a powerful tool, and a vital one due to globalization. Because of this necessity, and to accommodate the raising number of learners, learning English has become immensely diversified, taking the form of in-person classes, private tutorship, usage of language learning books, online courses, computer games and even mobile apps. These last three are examples of what is called Computer Assisted Language Learning (CALL) systems [13] and its importance in language learning is undeniable. On the last decade the role of mobile apps in specific has increased and some language schools even use them as part of their program. It presents a cheaper, simpler and more versatile way to learn because it can be used anywhere at any time and the experience may be tailored to an individual user.

When perfecting a language, it is important to practice not only grammar and morphology but also other speech related features. As speech is the simplest form of communication, mastering a language includes the capacity of producing clear and intelligible speech. And in order to do that, it is important to master both the individual phonetic segments (vowels and consonants) as well as the properties of syllables and larger units of speech, in other words, the supra-segmental aspects of speech, that are known as prosody. This connection between intelligibility and prosody is highly researched [14] [15] [16] [17], so part of the focus on CALL systems should be prosody training. For example, a study by Laures and Weismer (1999) [18] revealed that synthesizing speech with flattened intonation patterns significantly lowers the intelligibility scores as compared to sentences uttered with naturally varying contours. Other researchers [19] have related some aspects of deaf speech like difficulty in producing stress, appropriate pausing, and intonation with the reduction of intelligibility. Even though this connection is still

not completely defined and quantified, its existence is recognized. To achieve proficiency, in addition to the competencies stated above, the student should master prosodic features, which include intonation, rhythm, word/sentence stress and speech rate/chunking and only then fluent communication will be achieved.

Prosody training through CALL systems can be done both with visual and audio feedback. A system that would evaluate pitch and duration of a student's utterance could give tips on where to place the stress, where to elevate the pitch and which phones to elongate or shorten. But repetition is a key factor on language learning [20], so listening to an utterance with the correct stress, for example from a native professional speaker, would provide the student with a reference to follow. This is an approach taken by ELSA Speak, a mobile app for Android and iOS that helps users improve their English accent. One type of exercise that it offers is exactly providing the users with a written word or sentence and an audio of a native speaker reading it, which the user reads back to the phone. The user's utterance is scored and some improvements are suggested with a graphic feedback, if applicable.

But what if instead of a native speaker, the student would listen to himself/herself uttering that same sentence with the correct stress? This would potentially eliminate any distracting factor related with the difference between the student and the native speaker's voices, and improve the student's focus on the real aspects that he/she needs to improve. This is the suggestion of [21], in a study made on native Italian speakers that were learning German as a second language, but the same concept may apply to learning English.

1.2 Objectives

The main objective of this thesis is to explore two methods that could possibly optimize prosody training using an app, available anywhere, at any time. The proposed methods will be developed around a specific type of exercise from ELSA Speak app. This exercise consists of presenting a student with a written sentence in English. The user may choose to press a button and listen to a recording of a native speaker, ELSA's speech artist, uttering the sentence. The student then needs to repeat the sentence which is sent to the app's servers, processed and a feedback is returned. There are several indicators in this feedback, but for the purpose of this work we will be focusing on the feedback relative to the intonation, more specifically markers like pitch and duration.

The objective of these methods is to provide the user with a reference audio with a voice close to the user's own voice. Two different approaches will be tested. The exercise targets users that already master the segmental aspects of the L2 language.

The first approach will consist of manipulating the audio of the user's attempt on this exercise, correcting the pitch and duration of the phones uttered, and playing it back to the user. It will still be possible

to listen to the reference speaker recording, but the option of listening to his/her own utterance with corrected prosody will be added. This will be made using a Vocoder-based system and it is intended to be used without any pre-training and to be available to the user since the first attempt of the first exercise. It is the same concept as [21], [22] and [23], but with a different implementation designed for real-time usage without any human intervention. It is supposed to be almost instant and with very light computational requirements.

The second approach is intended to replace the native-speaker's utterance by the same sentence with the target prosodic contours, but in the user's own voice. It requires gathering data (audio files) from previous exercises from the user to fine-tune a pre-trained Voice Conversion model. This model is then used to generate the reference when the user loads the exercise, in his/her own voice, without requiring any attempt on that exercise (which is needed in the first approach). The recording done by the native speaker will be used as source, and it is expected that his/her prosodic features will be kept. It makes use of state of the art Voice Conversion technology and requires relatively high computational power and long training times.

1.3 Outline

This thesis is organized into 5 Chapters. The first chapter gives a brief introduction to what motivated this work and what objectives are expected to be accomplished. The second chapter briefly reviews background concepts necessary for this thesis. The third chapter runs through the first approach presented on this work, using a Vocoder-based speech synthesis system and a time alignment algorithm. The fourth chapter explores a second approach, using a Voice Conversion method. The fifth chapter provides conclusions and discusses possible future work.

Chapter 2 is divided into three main sections. The first section starts by providing some background on learning English as a second language, its challenges and how technology improved its process. It also shows why it is important to practice prosody and presents ELSA Speak, the app that served as the foundation for the solutions presented in this thesis. The second section explores the evolution of Speech Synthesis and presents today's solutions to produce artificial speech. The third section introduces the Voice Conversion task, explains the main components and gives a general overview of different techniques and methods to evaluate its performance.

Chapter 3 reveals the thoughts behind the first approach. It mentions all the components used to build the algorithm and discusses their choice and how they were tested. It finishes with the evaluation of the final algorithm, using subjective tests and proposing a method for objective tests.

Chapter 4 presents an alternative method, using a Voice Conversion technique. It explores the requirements of the task, the choice of the algorithm and its architecture. Then, the different databases

used to train and test the algorithm are described, as well as the results that they provided. Similarly to chapter 3, it finishes with the evaluation of the proposed model, but only with subjective testing.

Chapter 5 contains the conclusions and proposes future improvements on both algorithms, but with special focus on the Voice Conversion solution.

2

Related Work

Contents

2.1 Secondary Language Learning

2.1.1 Computer-Assisted Language Learning

In the past, the evolution of pedagogical methods relied heavily on classroom experimentation, which is an environment filled with uncontrollable variables. With the introduction of Computer-assisted Language Learning (CALLs) systems, it became easier to control the environment of experiments, allowing control over the exact presentation and content of materials and over user participation [24]. CALLs was defined as "the search for and study of applications of the computer in language teaching and learning" [25], where the term computer may be broadened into containing any application of Information and Communication Technology. The development of CALLs can be split into 3 phases [26], each relating to a certain level of technology and pedagogical level.

- **Behaviouristic or Structural CALLs** - Conceived in the 1950s and implemented from the 1960s to the 1980s. The main role of the computer was to deliver instructional materials to the learners, essentially repetitive language drills, vocabulary, grammar and translation tests.
- **Communicative CALLs** - Used on the 1980s and 1990s, this approach contained interactivity both learner-computer and learner-learner. It was used for activities that involved communication, such as conversations, written tasks and critical thinking.
- **Integrative CALLs** - From late 1990s onward, after the appearance of the World Wide Web. It integrates listening, speaking, reading and writing into language learning, and adds a social component. Teaching advanced from a structure-based manner into task-based approaches, introducing authentic discourse and developing real social interactions, possible through a network.

CALLs is not to be seen as a substitute for language teachers, but as a complementary way that gives easy access to a wide variety of learning materials from any device connected to the internet. The access to these materials tends to be faster, easier and cheaper than taking in-person language courses, or even having a language tutor. Duolingo, Memrise or Babel are good examples of learning apps that fit into this description.

But CALLs systems kept getting more complex and their applications began to diversify, having different tools that focus on specific aspects of language training, one example being pronunciation. Computer Aided Pronunciation Training (CAPT) aims at detecting and diagnosing mispronunciations in the speech of learners, and then help the learners to correct them [27]. The first CAPT system dates back to 1972 and was developed by Kalikow and Swets [28]. It was a system that used visual feedback for teaching English pronunciation to Spanish students, focusing mainly on vowels. But only after 2000 there was a real progress in CAPT systems. Nowadays, there are sophisticated tools that make use of Artificial Intelligence (AI) to detect and classify mispronunciations, and give highly specific feedback on

how to correct these mistakes. Two examples of such tools are ELSA Speak, which will be described later on, and Rosetta Stone.

2.1.2 The Golden voice

During the last three decades, many studies have suggested that L2 students would benefit from having a tutor with a very similar voice to theirs (see for instance [29] [30] [24]). The reason behind this is that by stripping away all the unnecessary information, such as the differences between the student's and the tutor's voice, the student would be able to perceive more easily the differences between his/her own accented utterances and the ideal accent-free correspondents. This is specially useful in CAPT systems, where the student would be able to detect and correct the mispronunciations in his/her own utterances more easily.

A study about Lexical Stress Training with L2 German speakers [21] tested this idea. The study consisted of having a group of 12 L2 German students that had been learning the language for several years, read a series of texts in German, which were then evaluated. A native German speaker was then asked to record these same texts, which were stored and analyzed in terms of pitch-contour, local speech rate, and intensity. Half of the students was randomly selected and their recordings were manually manipulated and resynthesized to match the prosodic characteristics of the native speaker. One week later, the students were called back to listen to recordings of the same text and try to correct their mistakes. The previously selected group listened to their manipulated utterances and the remaining students listened to the native speaker recordings. The results were again evaluated and compared with the initial results, which revealed a bigger improvement in the group that listened to their own corrected utterances than on the group that listened to the native speaker's utterances.

The study concluded that utterances in the learner's own voice are a more effective form of feedback for stress pronunciation training than pre-recorded reference utterances spoken by a native speaker. The students were able to focus more efficiently on the speaker independent features, and the improvements of their intelligibility were more evident. This process was done by manually correcting the utterances, which is not applicable to a large scale system. But it opens the discussion for what would be the result if this system would be applied in real-time, on an automated way and with instant feedback.

2.1.3 Intelligibility, pronunciation and prosody

Speech intelligibility can be defined as how clearly a person speaks so that his or her speech is comprehensible to a listener [31]. Having a good pronunciation is associated with having a high intelligibility, but it is important to account for all factors that affect pronunciation. Often pronunciation training in language schools is focused on improving the segmental features of speech, which may be defined as

”any discrete unit that can be identified, either physically or auditorily, in the stream of speech” [32], such as consonants and vowels, which occur in a distinct temporal order. But mastering pronunciation also requires the dominion over the suprasegmental features of speech, or in other words, prosody. These features extend over more than one segment and take the form of lexical stress, pitch, rhythm and intonation [1]. The process of increasing a speaker's intelligibility may be hindered if prosody is not practiced and mastered. The connection between prosody and intelligibility has been made and studied in several occasions [35]. Deaf speech and neurological disorders have been linked with the lack of control of prosody, which include difficulty in producing stress, appropriate pausing, and intonation, and leads to a significant decrease in intelligibility.

It is then necessary to include prosody training into language learning and CAPT systems. And knowing that these systems benefit from providing feedback to the speakers in their own voice, as presented on 2.1.2, applying it to prosody training seems only logical. One study concludes that prosodic manipulation is beneficial in pronunciation training, and suggests that accent conversion can be a successful form of implicit feedback in CAPT [36]. This way it is possible to set a personal target of how the student aspires to sound like, with his/her own voice, making the target ”closer” to the student. It may even be beneficial to add some progression to this learning process [37], by having a reference that is adaptable to the listener as opposed to a fixed normative one. This gradual (also mentioned as ”floating” [36]) reference would be a mid point between the student's latest attempt and the target utterance from a native speaker (with a tendency to the latter), but in the voice of the student. Each time the student would correct some prosody mistakes, a new temporary reference would be set closer to the one with ideal prosodic features.

2.1.4 ELSA Speak

ELSA (English Language Speech Assistant) is a startup founded in 2015 by Stanford alum Ms. Vu Van, and Dr. Xavier Anguera, an expert in speech recognition. ELSA's app, which is called ELSA Speak, is a good example of a CAPT system that aims at helping its users to perfect their pronunciation using deep learning AI. Its exercises consist of asking the users to read a word or sentence, which is recorded by the device and sent to its servers to be analysed. It returns real-time feedback on their pronunciation mistakes, with over 95% accuracy, and gives specific suggestions on how to improve. The user may also chose to listen to this word or sentence uttered by a native English speaker. All its targets are defined according to the Western American English accent. ELSA Speak is available for iOS and Android and is used by more than 10 million users in over 100 countries.

[1] Mastering word coarticulation can also be a difficult task for L2 speakers (see [33] [34])

2.2 Speech Synthesis

Speech synthesis is defined as the artificial production of human speech. The first attempts at reproducing the human speech with a machine dates back to 18th century, when Wolfgang von Kempelen created his Speaking Machine. A few years later, on mid 19th century, Joseph Faber built his own talking machine, named Euphonia, inspired by von Kempelen's work. These were early forms of a speech synthesizer, a system implemented either in software or hardware capable of producing speech.

2.2.1 The Vocoder

On the 1930s the first Vocoder system was introduced by Bell Laboratories, invented by Homer Dudley. A vocoder, word originated from Voice Encoder, operates on the principle of deriving voice codes to recreate the speech which it analyses [38]. It is composed by a analyser and a synthesizer. The analyzer receives the speech waveform as input and outputs eleven speech-defining elements, which include pitch and 10 separate bands of the spectrum. The synthesizer receives these 11 signals and outputs a reconstructed speech wave, similar in pitch and spectrum to the original. Between the analyzer and the synthesizer, the decomposed signal is simpler. When applied to telephone systems, this allows for a reduction in the range required for the transmission of intelligible speech and provides privacy, because the signals don't carry speech, but only a representation of it.

On 1939, Bell Laboratories presented the Voder, also developed by Dudley. The Voder stands from Voice Operating Demonstrator, and it uses the knowledge gathered with the Vocoder system applied to speech synthesis. While the Vocoder is used to break down the acustic components of Human Speech, the Voder preforms the inverse process, like the receiver mentioned above, generating artificial sounds through commands given by a highly trained operator. The system aimed at emulating the human vocal tract and works on the principle of formants, which are produced by resonances in the vocal tract.The overview of the Voder system can be seen on figure 2.1.

The operator used a wrist bar to chose between two base sounds: a buzz tone - responsible for the generation of vowels and nasals and whose pitch was controlled by a foot pedal - or a hissing sound - responsible for unvoiced fricatives. These sounds were then passed through a set of 10 band pass filters that the operated controlled with a keyboard, combined, and played through a speaker. There were also extra keys for plosives and fricatives. This was a very complicated machine to operate and required an operator to press physical buttons in order to produce the sounds, but the concept served as basis for more advanced systems.

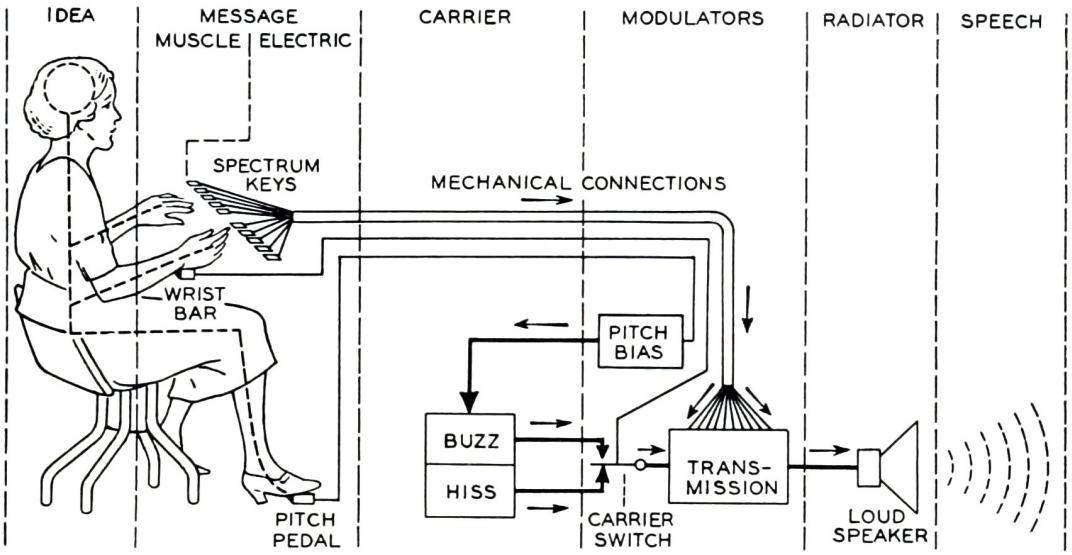


Figure 2.1: Schematic circuit of the Voder. Taken from [1]

2.2.2 Vocoder using Source-Filter model

The speech signal may be represented by a source-filter model, where the source is the vocal folds (or vocal cords) and the filter is the vocal tract (seem fig 2.2). The vocal folds produce the periodic sound, characterized by its intensity and frequency, which result in the perceived loudness and pitch. The vocal tract provokes resonances which generate the formants. The tongue, lips and throat originate hisses and pops, which are responsible for the production of the plosives and fricatives. By decomposing the speech into this two separate components, they can be modeled independently.

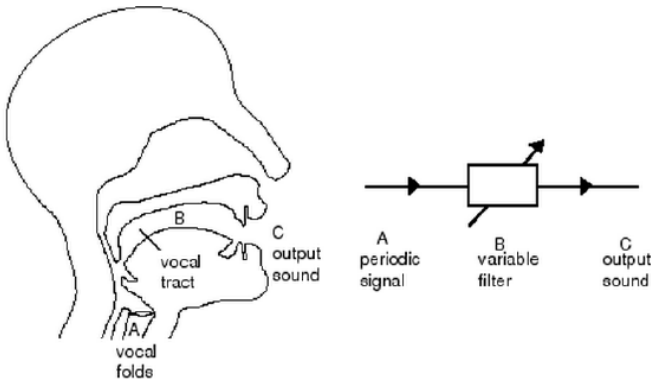


Figure 2.2: Representation of the vocal folds and vocal tract as a source-filter model. Taken from [2]

On the basis of this simplification, it is possible to produce a system that substitutes the human anatomical features responsible for speech production by components that can be electronically modelled and controlled, thus producing synthesized speech. The lungs are replaced by a DC source, the

vocal folds by an impulse generator for the voiced sounds and the vocal tract by a filter. A noise generator will produce the unvoiced sounds, assuming the role of the tongue and lips. On the case of human speech, voice and unvoiced sounds are mixed in different proportions and amplitudes throughout the speech, which could be represented by two potentiometers mixing both sources. To simplify the model, these two potentiometers are replaced by a switch that selects either the voiced or the unvoiced excitation.

To build a vocoder using a source filter model, it would required four main components:

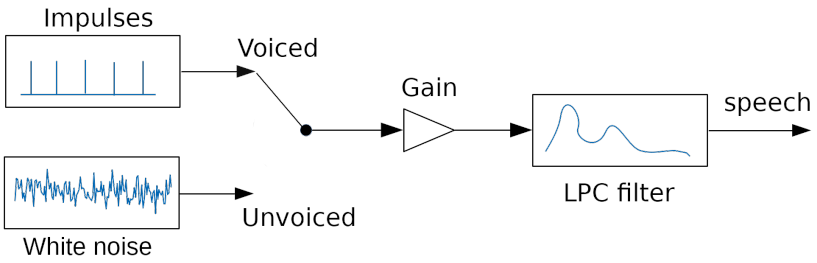
- Pitch Analysis - Detection of the fundamental frequency
- Voiced or unvoiced decision - A voiced sound happens when the vocal folds vibrate and produce a pitched sound in resynthesis, as with the production of vowels. An unvoiced sound does not have a defined periodicity and is associated with plosives and fricatives.
- Spectral Envelope analysis - It requires a filter to estimate the formants, which would replace the set of band pass filters seen in the early Vocoder technology.
- Amplitude Analysis - To determine Gain (or amplification factor) of the model

In this model, the filter will be responsible for simulating all the resonances from the vocal tract. So in order to produce such vocoder system, it is first necessary to find the filter coefficients, which may be done using LPC. This process provides the Linear Prediction Coefficients, which are not directly interpretable. But these may be converted to pairs of Line Spectral Frequencies (often called Line Spectral Pairs), which will resemble to the formants.

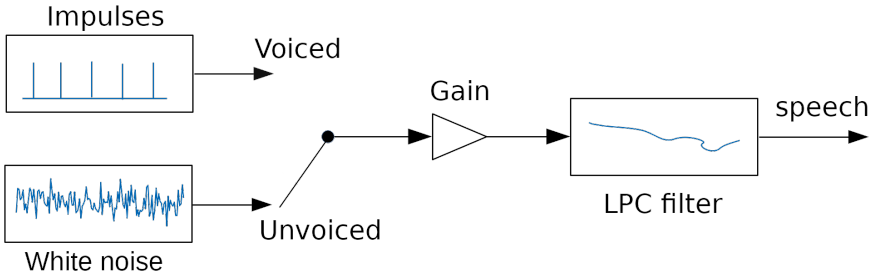
It is important to note that this is all done under the assumption that the waveform is completely separable into a source and a filter, which may be an oversimplification. Also, in order to predict the Linear Prediction Coefficients, the source is assumed to be very simple, with a flat spectrum, which means that the filter takes responsibility for all the spectral shape of the output sound. Nevertheless, this method still produces speech with some level of intelligibility.

The LPC Vocoder was improved extensively over the years in obtain a more natural generated speech. This was mainly done by introducing changes to the source, or in other words, by changing the excitation of the filter from an impulse train to more complex signals. A few systems may be mentioned:

- Residual-Excited Linear Prediction (RELP), uses the Residual as the excitation signal instead of an impulse train. The residual is obtained a process called inverse filtering, which consists of subtracting the predicted spectral envelope from the speech signal, and obtaining the remaining signal, which is the residual.
- Code-Exctited Linear Prediction (CELP), uses a fixed codebook as the exitation signal



(a) LPC Vocoder generating voiced speech



(b) LPC Vocoder generating unvoiced speech

Figure 2.3: Basic model of an LPC Vocoder, using the 4 main components: Pitch Analysis to determine the characteristics of the impulse train; Voiced/Unvoiced Decision to select the source; Amplitude Analysis to determine the Gain; Spectral Analysis to model the LCP Filter. Adapted from [3]

- Mixed-Excitation Linear Prediction (MELP), adds 5 features to the standard LPC, including mixed excitation, aperiodic pulses, adaptive spectral enhancement, pulse dispersion, and Fourier magnitude modeling.

2.2.3 Modern Speech Synthesis Systems

Today, the systems that decompose and re-synthesize speech are not just used as a way of compressing the speech signals and transmitting them in a faster, cheaper and more robust way. The fact that the acoustic features of speech can be translated into data opens the door to the manipulation and treatment of this same data, which can later be synthesized into speech that is different from the original. This is the backbone of the modern Text-to-Speech (TTS) systems that are used everyday on our mobile devices, phone answering machines, home assistants, and the list goes on.

2.2.3.A Concatenative Speech Synthesis

It is important to mention that the approach taken when developing the LPC Vocoder is not the only one possible. Instead of factorizing speech into components that are connected to the way the human body produces speech, Concatenative Synthesis crops the speech signal into very small temporal segments and re-synthesises speech by splicing these small units according to a complex set of rules. Signal processing techniques are also applied in order to smoothen the concatenation boundaries and

to superimpose the prosodic targets when using smaller segments. This achieves very good results, provided that the datasets used contain thousands of recorded sentences, the quality of the recordings is high, and that these recordings are made by a trained professional speaker. The speech database can have speech units of different sizes such as phones, diphones, syllables, words or sentences. If the system uses larger units, the synthesized speech will sound more natural, but it will require much larger datasets. On the other hand, a system that uses smaller units like diphones, requires a smaller dataset but demands more digital signal processing to introduce the desired prosody on the synthesized speech [39], using techinques such as Pitch Synchronous Overlap and Add (PSOLA) [40] and its variations. These systems are developed specifically for a set of voices and require a lot of engineering to perfect, which is a high effort and high cost task.

2.2.3.B Articulatory Speech Synthesis

While synthesis by concatenation generates speech by joining parts of pre-recorded speech, an alternative approach generates speech from scratch. In order to do this, it requires either physically modeling the human tract, or modeling the speech signal itself. An example of the former would be the model developed by Chiba and Kajiyama on 1941 [41], which was able to produce vowels. To do this, it used a series of tubes designed after the different shapes that the vocal tract assumed to produce these same vowels, but it wasn't able to produce consonants.

In the past, the progress of progress in articulatory synthesis has been hindered due to the lack of adequate data on vocal tract shaping [42]. With the usage of X-ray imaging and later Magnetic Resonance Imaging (MRI) technology it was possible to expand the knowledge on the vocal tract movements that produce human speech, and it was possible to model these movements with more detail [43]. In order to produce speech, an articulatory speech synthesis system would need to integrate elaborate and realistic models of the vocal tract [44] [45] [46], the vocal folds [47] [48], aero-acustics [49] [50] [51] and articulatory control [52] [53].

2.2.3.C Parametric Speech Synthesis

Even though articulatory speech synthesis as seen extensive research [43] and it is still being actively improved [54], the high complexity of these systems leads to worse results than alternative approaches that model the speech signal. These models focus only on how human-like the output sounds, without making deeper claims that the model is a true model of the human speech production.

Nevertheless, some inspiration is still drawn from how human produces speech. As previously stated, human speech is produced as the result of individual parts of the human body, such as the tongue, the lips, the vocal folds and the shape of the vocal tract, combined together, which implies that speech is composed by a number of processes running concurrently. To model this, the components need to be

separated to some degree and controlled separately. The current models of these parametric vocoders encode speech into the three following components, according to a given time frame:

- **Spectral Envelope** - Contains the formants originated by the shape of the vocal tract . Perceived as the overall timbre.
- **Fundamental Frequency** - Rate of vibration of the vocal foals. Perceived as the pitch.
- **Non periodic Energy** - Associated with the fricatives.

Usually, these components are expressed in numbers, and vectors of numbers, taken at fixed intervals of time, for example at each 5ms. Once it is in this form, the data can be manipulated for purposes such as Voice Transformation (VT) or analysed for detection of illnesses or emotional states. These coded representation of the waveform can also be used together with the text they represent to train TTS systems, without needing to actually store the original waveforms. Later, when the model is trained, it can receive text as input and output the predicted coded representations of the waveform. These representations can then be used by a waveform generator to output artificially generated speech.

2.2.3.D Neural Vocoders

While the decomposition of speech into these parameters mentioned above are still the state of the art in the feature extraction, there are new methods to perform waveform synthesis that provide more natural speech. Parametric vocoders tend to approximate the human mechanism to produce speech under certain simplified assumptions and the audio generated by them tends to be buzzy or robotic. To overcome these limitations, a new generation of vocoders was introduced.

Since the introduction of WaveNET [55], neural vocoders have gradually became the most common vocoding method to generate waveform audio, achieving increased audio quality of generated speech. These systems are data driven and they do not assume any mathematical model, which appears to be a solution to some inherent problems of the parametric vocoders. One downside is that these models require big amounts of data and take very long to be trained. This is a hindrance, specially because it is still difficult to make a universal system that is able to generate any voice, even though some attempts already achieve good results, like WaveGlow [56] and Universal Vocoder [57], and fine-tuning the algorithm every time the speaker changes is very time-consuming. Also, the inferences take long to run, making them not suitable for real-time applications.

On the training phase, these models receive two kinds of input files: waveforms and some intermediate representations of speech extracted from these waveforms. These representations may be, for example, the 3 components of speech (Spectral Envelope, F0 and non periodic energy) extracted by a parametric vocoder. It then proceeds to establish a connection between these representations and the waveforms they refer to. On the synthesis phase, the model receives the speech representations

as input and outputs the predicted waveform. Even though it is a recent technology, it is being widely researched and there are several algorithms already developed that provide very high quality audio [58].

2.3 Voice Conversion overview

Voice Transformation is the term used to designate the various modifications one may apply to human speech. It aims at modifying one or more aspects of the speech signal while retaining its linguistic information. VT approaches can be applied to perform tasks such as changing the emotion conveyed through speech, improving the speech intelligibility of speech or changing whisper/murmur into speech, without modifying speaker identity. Voice Morphing and Voice Conversion (VC) are two tasks derived from Voice Transformation that also retain linguistic content but change the identity of the speakers. The first blends the voices of two speakers creating a third virtually new speaker. The latter is the study that deals with the conversion of the perceived speaker identity.

2.3.1 Objective of the Voice Conversion task

The objective of Voice Conversion is to modify a portion of speech from one speaker, the source speaker, so that it sounds as if it was uttered by a second speaker, referred to as the target speaker. In other words, VC modifies speaker-dependent characteristics of the speech signal, such as formants, fundamental frequency, intonation, intensity and duration, in order to modify the perceived speaker identity, while keeping the speaker-independent information (textual contents) unchanged. There are several applications for this kind of task, including safety related usage (to disguise a speaker's identity), voice reconstruction, entertainment and even translation or accent reduction. Evaluating this task constitutes a challenge by itself. It is very difficult to classify whether or not a VC task was performed successfully since the identity of the speaker and the perception of the quality are highly subjective. Nevertheless, VC today is a significant aspect of AI and benefits greatly from the increase computational power now available.

To further understand what is involved in the VC task, it is necessary to first understand what constitutes the speakers identity and how to decompose and analyse all the information included in speech. Speech conveys a information through different means, that according to John Laver can be categorized into the following three distinctions [59]:

- **Linguistic behaviour** - Coded, informative and communicative. Uses the dual-level code of spoken language made up of the phonological (audio) and grammatical (text) units, which can be descibed as follows:

  - **Grammatical unit** - Connected with morphology and syntax, reflected in sentence structure, lexical choice, and idiolect;
  - **Phonological unit: Segmental factors** - related to short term features, such as spectrum and formants.
  - **Phonological unit: Supra-segmental factors** - Related with the prosodic characteristics of a speech signal, including intonation, tone, stress, and rhythm.
- **Paralinguistic behaviour** - Coded, informative and communicative. It is non-linguistic and non-verbal and communicates the speaker's current affective, attitudinal or emotional state.
- **Extralinguistic behaviour** - Non-coded, non-informative, but communicative. It is what remains from the speech signal that doesn't fit into the Linguistic and Paralinguistic behaviours. It retains information about the identity of the speaker, particularly with respect to habitual factors such as the speaker's voice quality, the overall pitch and loudness.

With the Grammatical unit (textual content) fixed, an effective Voice Conversion task is expected to convert not only the Extralinguistic and Paralinguistic behaviour, but also both segmental and supra-segmental factors of the Linguistic behaviour, which are relevant factors concerting speaker individuality.

2.3.2 Pipeline of a typical Voice Conversion system

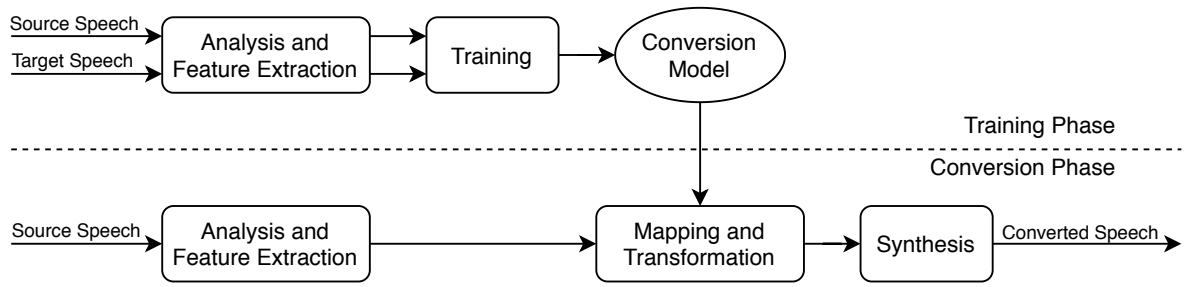


Figure 2.4: Overview of the flow of a typical Voice Conversion system, divided intro the training and conversion phases

There is a wide range of algorithms designed to perform a VC task, which have different requirements and different purposes. But there are a set of steps that are common to the majority of these algorithms and produce a simplified pipeline that can be seen on the figure 2.4. These components are:

- **Analysis and Feature Extraction** - Estimation of the parameters that represent the acoustic features of speech, as mentioned in the previous chapter. This is applied to the audio files containing both the source's and the target's speech.

- **Training** - Receives the features extracted from both the source and target speakers and attempts to represent the relation between similar features of both. Outputs a conversion model, or a mapping function, with the perceived correspondences between each set of features.
- **Mapping and Transformation** - Performs a process similar to the one on the training phase to map the features of the source speaker into the representations from the conversion model, and performs the transformation of these features using the mapping function. Outputs the converted features.
- **Synthesis** - Receives the converted features and attempts to reconstruct the waveform audio containing the text originally uttered by the source speaker but with the voice of the target speaker.

2.3.3 Parallel vs Non-Parallel

The VC frameworks can be split into 2 broad groups according to the nature of the data they use to train the models, having on one side the algorithms that use parallel speech corpora and on the other side those that use non-parallel speech corpora.

On a parallel speech corpus, all the speakers record the exact same sentences. These corpora are more difficult to construct and usually restrict the applications of the algorithm. During training, the conversion model captures the correlation between source and target speaker and examines their acoustic correspondences or dissimilarities.

A non-parallel training corpus is built from different utterances uttered by different speakers. This means that Voice Conversion models that require this kind of corpus may be employed where it is impossible to record the same utterances for all speakers, making them more versatile. Usually, these algorithms are not fundamentally different, and instead include additional steps to adapt to the non-parallel dataset. There are two general approaches to this problem. The first one consists of factorizing the linguistic and speaker related representations carried by acoustic features and transforming only the speaker related representations, which are then concatenated with the original linguistic features from the source to generate the output. The second approach attempts to train an Auto Encoder model in speech reconstruction and enforce speaker independence in the latent representations. Such latent features are then concatenated with speaker dependent representations in a similar way to what the first approach does with the speaker independent representations.

2.4 Evaluation Methods

Evaluating a Speech Synthesis system is a complex task on its own. Although there are different methods to evaluate the performance of these systems, there is no universal guidelines on how to proceed.

Nevertheless, these evaluations are necessary to compare new systems with their alternatives and to guide future development. The evaluation o speech synthesis systems is usually made according to 2 main aspects:

- **Speech Quality** - Describes the quality of the generated speech in terms of naturalness and audible artifacts.
- **Speech Intelligibility** - Measures the intelligibility of the generated speech.

When referring to the VC task in particular, a third aspect needs to be added:

- **Speaker Similarity** - Quantifies how close the voice of the converted speech sounds to the voice of the target speaker.

Some other minor aspects may be evaluated, specially when the tests are performed on specific components of the systems. This evaluation may be done through objective or subjective methods. The most common objective evaluations methods include the Mel-cepstral distortion (MCD) [60] for evaluating the spectrum, and Pearson Correlation Coefficient (PCC) [61] and Root Mean Squared Error (RMSE) [62] to evaluate prosody. But objective evaluations do not necessarily correspond to human judgments. So, in most studies, subjective evaluations are preferred. They consist of asking human listeners, referred to as subjects from here on, to assess the perceived performance of the Speech Synthesis algorithms by presenting them with utterances generated through these algorithms and in some cases, other references. Some of the most used methods are mentioned below.

2.4.1 MOS

The mean opinion score (MOS) is the most common test used to rate speech synthesis systems, allowing the ranking of different algorithms. The evaluation is done according to a 5-point Likert scale grading, divided as follows: 5=excellent, 4=good, 3=fair, 2=poor, 1=bad. The mean is then calculated and presented, usually associated with its 95% confidence interval.

2.4.2 CMOS

The Comparative MOS (CMOS) is a comparative test where the subjects are asked to chose how one speech sample compares to a baseline, for example in terms of audio quality or speech intelligibility. The subject listens to the two speech samples and rates usually on a scale from 1 to 5, according to the instructed metric, if the former (new method) is better or worse than the latter (baseline). The scale may divided as follows: 5=definitely better, 4=better, 3=same, 2=worse, 1=definitely worse. The average is then computed, also including the 95% confidence interval. It may be presented as is or inside the

interval [-2,2], where 0 means both techniques are similar, positive means that the technique is better than the baseline, and negative means that its worse.

2.4.3 AB test

On the AB test, the subjects are presented with 2 speech samples. After listening to both samples, they chose which one is preferred according to the instructed metric. The neutral option may be given to avoid a forced random choice in case there is no clear preference. The order effect is counterbalanced by using both A–B and B–A pairs.

2.4.4 ABX test

Similar to the AB test, but with an added reference. The subjects are asked to listen to the sample A, sample B and the sample X. Usually, sample A and sample B are generated by two alternative systems and sample X represents a target for the system. The subjects need to chose whether A or B is more similar to the sample X, according to the instructed metric. The neutral option may be also given for the same reason as the AB test. The order effect is counterbalanced by using both A–B and B–A pairs.

2.4.5 Thurstone's Paired Comparison

The Thurstone's Paired Comparison [63], or Pairwise Comparison consists of asking the subjects to choose the stimulus, or item, in each pair that has the greater magnitude on the choice dimension they were instructed to use. Each subject is presented with the same full set of choices and is not offered the indifference choice, meaning it is a binary choice test. All combinations excluding the same stimulus pairs are presented to the subject and the order effect is counterbalanced by using both A–B and B–A pairs. The result of all the choices represents a preference score for each item, resulting of the number of times the subject preferred one item to other items. When the subject prefers the stimulus 1 over the stimulus 2, it is denoted by $p_{1>2}$, and when the choice is the opposite we have $p_{2>1}$. This value is between 0 and 1, meaning that 0.50 would be the intermediate point where the choices of each of the stimulus over the other are in equal number for both stimulus.

The result of this test is calculated through the law of comparative judgement:

$$S_1 - S_2 = x_{12} \cdot \sqrt{{\sigma_1}^2 + {\sigma_2}^2 - 2r\sigma_1\sigma_2}$$

Where:

$S_1 - S_2$ is the psychological scale value of the two compared stimuli. It is positive in case the stimulus R_1 is preferred over the stimulus R_2 and negative if the opposite happens.

x_{12} - the sigma value corresponding to the proportion of judgments $p_{1>2}$. When $p_{1>2}$ is greater than .50 the numerical value of x_{12} is positive, and it is negative otherwise.

σ_1 - discriminal dispersion (same as standard deviation of its distribution) of stimulus R_1

σ_2 - discriminal dispersion (same as standard deviation of its distribution) of stimulus R_2

r - correlation between the discriminal deviations of R_1 and R_2 in the same judgement.

2.4.6 MUSHRA - Multiple Stimuli with Hidden Reference and Anchor

The MUSHRA methodology [64] (MUltiple Stimuli with Hidden Reference and Anchor) of perceptual testing was initially developed to evaluate audio codecs, but it has been found to be very powerful at detecting differences between speech synthesis systems [65] [66] [67].

This methodology consists of allowing the subjects to switch at will between the different stimuli from the different systems being tested and a reference utterance labeled as such. The recommendation [64] specifies a hidden reference and low and mid-range anchors should be added. These anchors will serve to calibrate the scale so that minor artifacts are not unduly penalized and typically result of a 7 kHz and a 3.5 kHz low-pass version of the reference, when evaluating audio coding. In the case of speech synthesis systems evaluation, these anchors are difficult to attain, and they are usually dropped so only the hidden reference is used, mixed with the actual stimuli from the systems in test.

The subjects evaluating the audio are asked to rate the stimuli on a scale from 0 to 100, and all the different stimuli should contain exactly the same utterance. In case the subject classifies the hidden reference with a score lower than 90% on more than 15% of the test items, the evaluation should not be considered. The utterances should be shorter than 10s each test shouldn't contain more than 12 (including references and anchors), to avoid the subject's fatigue.

3

Pitch Transplant

Contents

3.1 Introduction

The objective of this thesis is to propose an algorithm that presents the user with a goal utterance in his/her own voice. The user can then use it as reference to perform a language learning exercise, as specified in section 1.2.

The first approach that was adopted consists of taking the user's audio containing the exercise utterance and manipulating it through signal processing to produce the target. This allows the introduction of the gradual reference [36] that will change at each iteration, as alternative to a voice conversion algorithm which will produce a single static target, no matter how the user utters the sentence.

This approach is named Pitch Transplant because of the way the output is generated. In broad terms, it is built by fusing the user specific features with the scaled and aligned pitch contour of the audio from the reference. It is aligned through a DTW algorithm with specific restrictions so that the produced utterance is close to the target, but doesn't contain significant distortion.

The algorithm was designed to perform the pitch transplant end-to-end. It receives two audio files as input, one from the user and another from the reference, and outputs a single audio file containing the re-synthesized audio of the user with the corrections on pitch and duration contours.

The input audio files need to obey to the following restrictions:

- Format: wav
- Sampling rate: 16KHz
- Bitrate: 16 bits/sample
- Channels: 1 (mono)

The Speech-Server will return the alignment at the phone level of both the reference utterance and the user's utterance. This exercise is expected to only be available to users that already have a good level of English, in terms of phonetics, and consists of uttering a sentence that is shown in the screen. To guarantee that the phone sequence from the users' utterance is exactly as expected for each specific sentence, the audio and the text are force-aligned using technology form ELSA.

The outline of the Pitch Transplant algorithm is presented on image 3.1. First, both utterances are normalized in terms of volume and the audio and text are aligned, which provides the information of which frames correspond to the beginning and end of each phone. Then, a feature extraction process provides the F0 contour, the spectral envelope and the aperiodicity for each utterance. The utterances are both aligned on the phone level with a DTW algorithm, using the features previously extracted. The F0 contour from the reference is scaled to match the user's F0 and aligned, and it is used together with the user's aligned spectral envelope and aperiodicity to produce the transplanted audio.

The code of the Pitch Transplant algorithm was all written in Python. The original implementation of the WORLD algorithm is written in C++, so a Python Wrapper was used [68].

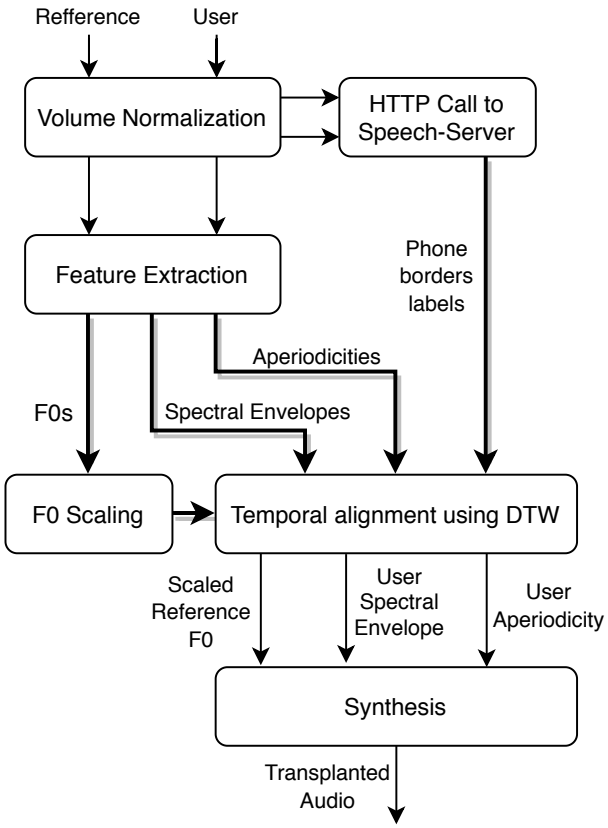


Figure 3.1: Outline of the Pitch Transplant Algorithm. The thicker lines represent data from both the reference's and the user's utterance.

The first section of this chapter introduced the Pitch Transplant Algotihm. The second section will present an overview of WORLD [4], DTW and the F0 scaling function, which are the minimum modules required to establish a working baseline. Third section shows the datasets used to test the Pitch Transplant algorithm, together with their purpose and how they were constructed. The forth section describes all the work put into fine-tuning the parameters of the WORLD methods, tweaking and refining the DTW algorithm, and adding extra methods to improve the overall result of the Pitch Transplant. It also includes some comments to the implemented methods and presents some other methods that were tested but did not improve the results and were consequentially abandoned. The last section presents the results of the final version of the Pitch Transplant, which was evaluated by 40 subjects through an online survey.

3.2 Algorithm

3.2.1 WORLD

WORLD is a vocoder-based high-quality speech synthesis system [4] that allows the easy manipulation of speech and meets the requirements of high sound quality and real-time processing. In this work, the latest version available of WORLD was used, having received its last major update in 2018. This same version was used as part of the baseline model of 2020's Voice Conversion Challenge [69]. Its results are better than any other similar vocoder-based system to this date [70], it does not require any training, and its processing time is very reduced, making it compatible with the requirements for the Pitch Transplant.

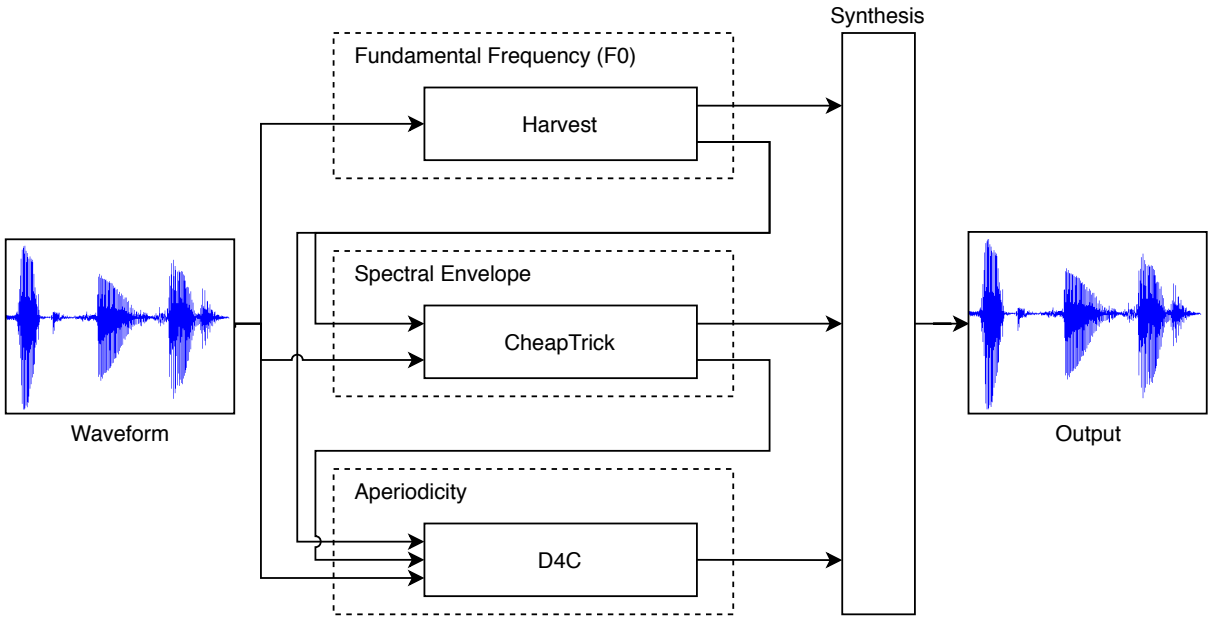


Figure 3.2: Overview of WORLD, including the methods for estimation of the 3 speech components, Fundamental Frequency, Spectral Envelope and Aperiodicity. Adapted from [4].

WORLD can be divided into 2 main stages: feature extraction and synthesis. Feature extraction in WORLD is carried out by three modules that run in sequence and extract three speech parameters. The synthesis stage performs the inverse process, joining the 3 speech components to produce a waveform signal containing speech.

3.2.1.A Harvest - Fundamental Frequency Estimator

The first module used in WORLD is called *Harvest* and outputs the F0 contour of the input file. It consists of two steps: estimation of F0 candidates and generation of an F0 contour on the basis of these candidates [5].

The module starts by passing the input waveform through a series of band-pass filters with different center frequencies to obtain basic F0 candidates. These basic F0 candidates are then scored and refined, resulting in a number of F0 candidates estimated for each frame. Because this is a frame-by-frame process, there are frames where it is not possible to estimate a F0 candidate due to high temporally local noise. In order to overcome this problem, a smoothing algorithm that uses the F0 values from neighbouring frames is applied.

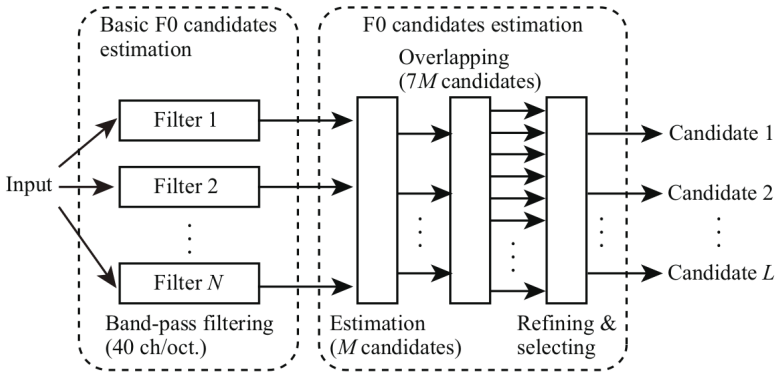


Figure 3.3: Outline of the first step of Harvest. Taken from [5]

The second step of this algorithm is responsible for selecting the best F0 candidates. Since voiced speech is a quasi periodic signal, it does not change abruptly within a short interval, so all the F0 candidates that represent rapid changes are removed. Short voiced sections with duration below a given threshold are also removed and classified as unvoiced, as they most likely represent noise with a short period continuous F0. Then, each voiced section is expanded by verifying if F0 candidates from adjacent unvoiced sections are within an interval of $\pm$18% (value tuned by the authors), in which case they are selected and expanded as voiced sections. After this expansion another run removes the short voiced sections and selects the F0 with higher average reliability score where there are overlapped F0 contours.

One of the aims of the Harvest algorithm is to ensure that no voiced sections are marked as unvoiced, so in the last step all unvoiced sections smaller than 9ms are classified as voiced and their F0 contour is estimated by interpolation. The final result is smoothed by a zero-lag Butterworth filter and all the F0s of unvoiced sections that were padded in the boundaries are reset to 0, resulting in the output of this algorithm.

In the code, the Harvest algorithm is implemented in a function that receives as argument the waveform signal, the sampling frequency, and 3 parameters:

- $f0\_ceil$ - Maximum value of F0 in Hz (*default = 800.0Hz*)
- $f0\_floor$ - Minimum value of F0 in Hz (*default = 71.0 Hz*)

– $frame\_period$ - Period between consecutive frames in ms (*default = 5.0 ms*)

On the baseline version of the algorithm, the maximum and minimum values of F0 were kept default and the frame period was set to 10ms, to match the default interval of text and audio alignment software. These values were later fine-tuned using results from the users' utterances, which will be described on section 3.4.1.A.

The function returns an array containing the estimated F0 contour and another array of the same dimension containing the temporal positions of each frame where the F0 was estimated. This array will be useful for the remaining feature extraction that will be carried out by WORLD.

3.2.1.B CheapTrick - Spectral Envelope Estimator

The second module used in WORLD is called CheapTrick. Its goal is to output a temporally stable spectral envelope, given the F0 contour (calculated with Harvest) and the audio file. This is the speech component that weights more on the DTW's alignment, although it is stacked with the aperiodicity component for added precision.

The algorithm consists of three steps: F0-adaptive windowing, smoothing of the power spectrum, and a liftering process for smoothing and spectral recovery. Cheaptrick begins by sweeping the signal with a Hanning window with the length of three times the fundamental period and with a power at $\omega_0 Hz$ 30dB lower than that of the main lobe (0Hz) [71]. In the second step, the power spectrum obtained is smoothed, by way of a simple filtering with a rectangular window, to ensure that it has no 0s. This is necessary because the third step uses a logarithmic power spectrum and the presence of 0s would lead to the presence of $-\infty$ and cause a fatal error.

The third step consists of preforming a liftering in the quefrency domain. It is carried out by a $sinc$ function and it allows the removal of the time-varying component that may be present in all frequencies due to discretization. Simultaneously to this process, the spectral recovery is carried out on the basis of consistent sampling theory [71].

The authors tested the method extensively since its development and compared it with alternative methods that also estimate the spectral envelope. The results of MUSHRA subjective tests reveal that the quality of re-synthesized audio obtained with Cheaptrick is very similar to TANDEM-STRAIGHT [72] [73] and STAR [74] methods, all scoring above 90 points. But Cheaptrick outperforms the alternatives in terms of robustness. Tests were made with Thurstone's Paired Comparison using audios generated with original value of F0, 0.75F0 and 1.25F0, resulting in a total number of 108 stimulus pairs. Ten subjects were instructed to select the stimulus with the highest sound quality and the results revealed that Cheaptrick was 10% better than TANDEM-STRAIGHT and 35% better than STAR. It is also more robust to additive noise and F0 error. On an objective test that consisted of using a periodic impulse train with a standard F0 and adding white noise or F0 error between -20% to 20%, Cheaptrick managed

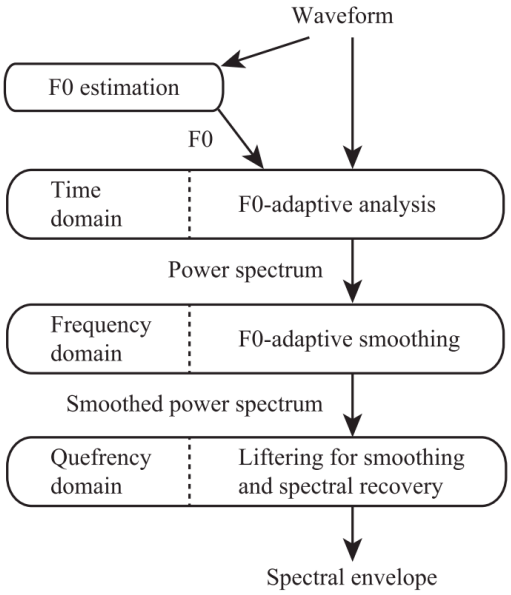


Figure 3.4: Overview of Cheaptrick Method. Taken from [6]

to keep the time varying component and the estimation error lower than both the other two methods. It is also the fastest of the tested algorithms, which adds to its desirability.

The function that implements CheapTrick in the code receives as input the waveform signal, the sampling frequency, the F0 contour, the temporal array returned by Harvest, and 3 optional parameters:

- $q1$ - Spectral recovery parameter. This value was tuned by the authors of WORLD and it is advised to keep it in the default value of -0.15
- $f0\_floor$ - Lower F0 limit in Hz.
- ft_size - FFT size to be used. If this parameter is kept at default value - 'None' - the FFT size is computed automatically as a function of the given input sample rate and F0 floor.

This function outputs the 513-dimensional spectral features which makes it difficult to interpret, as opposed to the output of the Harvest algorithm. This means that its output can only be tested by re-synthesizing the audio and verifying it against the original audio.

The audio produced with cheaptrick has a very decent audio quality, accounting for the speed with which the whole algorithm runs. With good recording conditions (High Signal-to-Noise Ratio (SNR)), the synthesized audios have very similar quality to the original ones, and artifacts were rarely added. The performance with real user's audios will be further discussed in section 3.4.

3.2.1.C D4C LoveTrain (Definitive Decomposition Derived Dirt Cheap) - Band-Aperiodicity estimator

This is the last module to run as part of the feature extraction from WORLD. Its purpose is to extract the aperiodicity of the speech signal which is defined as the power ratio between the speech signal and the aperiodic component of the signal [75]. To achieve this, D4C uses a group-delay-base parameter, which forms a sine wave of F0 Hz from arbitrary periodic signals with a fundamental period of T0. The power ratio between this sine wave and the other frequency components will correspond to the aperiodicity and the band aperiodicity can be obtained by limiting the frequency band used for this calculation.

The latest version of D4C, called D4C LoveTrain, which is used in this work, contains an added last step. D4C occasionally gives a low value in the lower frequency band, and as a result, the periodic component is perceived as the noise. This step identifies the voiced/unvoiced segments and gives the aperiodicity value of 1.0 in the whole aperiodicity band. This way, the whole component of the spectral envelope comes from the aperiodic component, even if the frame contains an F0, avoiding the degradation of the resulting waveform.

As with the previous 2 modules, the authors tested the algorithm extensively and made comparisons with the alternative aperiodicity algorithms including: STRAIGHT and TANDEM-STRAIGHT. Subjective evaluation included a MUSHRA evaluation that was done in 40 audios uttered by two male and two female voice artists, using the STRAIGHT F0 and spectral envelope estimation and changing only the aperiodicity estimator. Sixteen subjects rated the audios synthesised with D4C with the highest sound quality. Also objective testing to analyse the impact of AM modulation of vocal cord vibration, additive noise and F0 estimation error, similar to the tests made for Cheaptrick, revealed that D4C was more robust and generated a closer estimate than the alternatives.

D4C provides a way to remove any segment that was misclassified as voiced during the F0 contour estimation. This is done by going through the frames and verifying if the aperiodicty for a given frame is above 0,5. If is is, this frame will be assigned a zero F0 value, like the other unvoiced frames, correcting the Voiced/Unvoiced classification. This way it is possible to get the best possible F0 value estimation from Harvest, complementing it with the Voiced/Unvoiced classification from the D4C to produce an optimum result.

D4C is implemented in a function that receives as arguments the waveform signal, the F0 contour, the temporal array returned from Harvest, the sampling frequency and 3 parameters:

- $q1$ - Spectral recovery parameter, also tuned by the authors and left with the default value of -0.15.
- $threshold$ - Threshold for aperiodicity-based voiced/unvoiced decision, in range 0 to 1 which changes as following:

    1. $threshold = 0$: voiced frames will be kept voiced, similar to older versions of the algorithm

2. $0 < threshold < 1$: some voiced frames can be considered unvoiced by setting their aperiodicity to 1 (thus synthesizing them with white noise)
3. $threshold = 1$: all frames considered unvoiced

– fft_size - FFT size to be used, similar to Cheaptrick.

The default values were kept unchanged on the baseline version of the Pitch Transplant algorithm. The module returns the 513-dimensional aperiodicity features, which again are difficult to interpret, making the analysis of the re-synthesized speech the only way to evaluate the performance of this module. As with cheaptrick, it produced very good results when audio files with high SNR were used. Lower SNR audio files still produce an acceptable result, but with a noticeable deterioration as will be discussed in section 3.4.

3.2.1.D Synthesis Algorithm

The second stage of WORLD consists of synthesizing an audio file from the 3 components obtained in the first stage: F0 contour, spectral envelope and band aperiodicity. This algorithm calculates the vocal cord vibration in the basis of the convolution of the minimum phase response and the extracted excitation signal. The computational cost of this operation is lower than the synthesis algorithm of STRAIGHT and TANDEM-STRAIGHT and the output waveform is the most similar to the input waveform, according to the authors [4].

In this case, as opposed to the previous algorithms, there aren't any parameters to be changed, and the function only receives the 3 components containing the extracted features and the sampling frequency and outputs the waveform signal.

3.2.2 Dynamic Time Warping - DTW

3.2.2.A Introduction

The F0 transplant that is described throughout Chapter 3 consists of extracting the F0 contour from a reference utterance, which is considered as ideal, and forcing it into the audio uttered by the user trying to replicate the same sentence. A problem with this is that different acoustic renditions of the same speech utterance are seldom realized at the same speed, originating high fluctuations between each of the user's attempts and between those and the reference audio. This becomes a bigger issue when running the Pitch Transplant in long sentences, where silences between words are important and add prosodic value. So, the F0 can not be directly and naively replaced without making the proper adjustments. A non-linear wrapping mechanism is required which will shrink or extend each segment accordingly, for the F0 contour to have the closest possible match to the recipient waveform.

To preform this alignment, the DTW algorithm was chosen. DTW is a well-known algorithm widely used in many areas. It was introduced in 1958 [76] and applied to speech processing for the first time in the 70s [77]. Since then, several time alignment algorithms were proposed, including much more complex architectures, but DTW is still hard to beat to this day. [78]. Even though it may be out-preformed in terms of accuracy in specific applications, it is usually at the cost of a very high computational power requirements [78] and the improvement is not very significant. Due to the restrictions of this problem, where the alignment needs to be done in very few seconds, DTW seemed like the best choice.

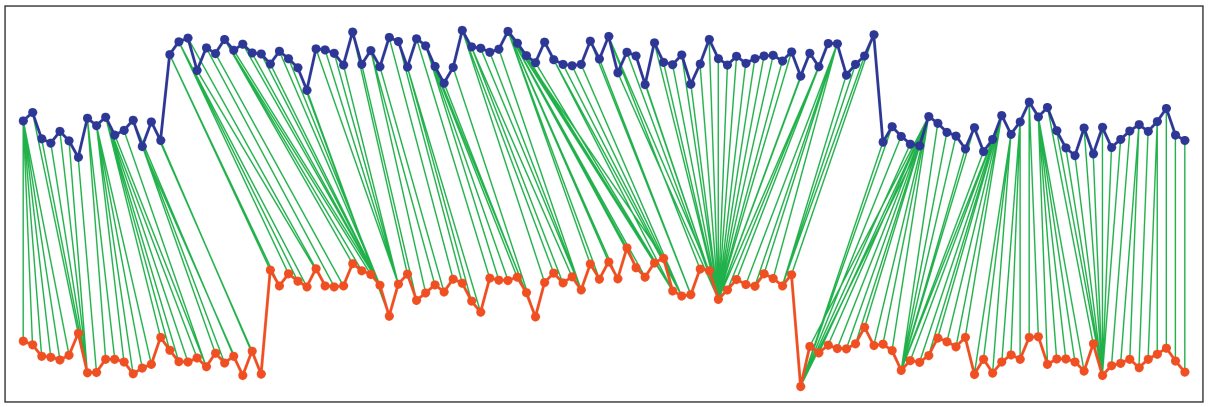


Figure 3.5: Example of the alignment made by DTW algorithm to two one-dimensional arrays. Taken from [7]

3.2.2.B The algorithm

Take the reference audio and one user's attempt to replicate the same utterance as two different speech patterns X and Y, respectively [79, Chapter 4.7]. These speech patterns are represented by the features extracted beforehand by WORLD, resulting in an array $(x_0, x_1, ..., x_i, ..., x_{T_X})$ and $(y_0, y_1, ..., y_i, ..., y_{T_Y})$, each element, x_i and y_i, corresponding to a frame of 5ms and containing a vector of short-time acoustic features. The durations T_x and T_y are most likely not identical.

DTW will start by calculating a matrix containing the distances between each element of X and Y, or in other words, the similarity between a the features corresponding to a frame x_i and a frame y_i. Then, the optimum alignment will correspond to the path, ϕ, in this cost matrix that minimizes the cumulative distances and obeys to a set of restrictions. These restrictions minimize distortion, reduce computational power and ensure the usability of the results. These are:

- **Endpoint Constaints** specify that the alignment must start in the first frame pair (x_0, y_0) and finish in the last (x_{T_X}, y_{T_Y}).
- **Monotonicity Conditions** do not allow for the warping path to have a negative slope, following the restrictions: $\phi_x(i+1) \geq \phi_x(i)$ and $\phi_y(i+1) \geq \phi_y(i)$.

- **Global Path Constraints** restrict the region in the matrix where the the distances are calculated and consequentially, the optimal path is searched.
- **Local Path Constraints** specify the allowed jumps between each 2 adjacent elements on the path. It is recommended [79] that the selection of the local continuity constraints should be based on heuristics and observations that result from an experimental process.

The restrictions implemented on the DTW algorithm used in Pitch Transplant will be explained in detail in section 3.4.

The DTW returns a variable $wrap\_path$, containing two arrays $wrap\_path_X$ and $wrap\_path_Y$, each of these arrays containing indexes corresponding to one of the utterances, reference or user. The correspondence between the i^{th} frame from the speech patterns X and Y can be obtained by:

$$X(wrap\_path_X(i)) \Leftrightarrow Y(wrap\_path_Y(i))$$

3.2.3 F0 Scaling

After both utterances are aligned with the DTW and the path is obtained, it is time to synthesize the resulting audio with the F0 from the reference and the remaining two speech components from the user. But the F0 contour can not be directly used without scaling it to match the user's average fundamental frequency. This is particularly relevant because there is only one reference, ELSA's speech artist, who is female and has a high average F0 of 228Hz, and the user's are both male and female, with a wide range of F0 average values. The average F0 value was calculated with 2000 audios containing sentences longer than 3 words, using WORLD's Harvest method.

The formula used to scale the F0 [80] is the following:

$$f0_{final} = \frac{\sigma_{user\_s}}{\sigma_{ref}}(f0_{aligned} - \overline{f0_{ref}}) + \overline{f0_{user}}$$

Where:

σ_{user_s} - Scaled standard deviation of user's F0

σ_{ref} - Standard deviation of reference's F0

$f0_{aligned}$ - F0 contour from the reference aligned with DTW. It is similar to the reference's F0 but with some dropped or repeated frames.

$\overline{f0_{ref}}$ - Mean value of the reference's F0

$\overline{f0_{user}}$ - Mean value of the user's F0

It is important to note that originally the formula contained the fundamental frequency in the log domain as well as a mean and variance adaptation of it. This happened because the features used by the authors were extracted with the Ahocoder [81] which gives the outputs in the logarithmic scale.

WORLD on the other hand, provides the F0 on a linear scale, so an adapted version of the formula was used.

Another difference from this to the original formula is the usage of a scaled user's standard deviation. At first, and as an attempt to conserve the user's characteristics as much as possible, the user's standard deviation was kept untouched. This worked well in the cases where pitch prominence existed, but it was put in the wrong phonemes. An issue arose where no or too low prominence was given in the sentence and the pitch contour was flat overall. With no tool to increase this variance, the output also failed to hit the prominence targets. The opposite was also prone to happen, with to high variance, the result would be an audio containing unnatural speech.

To solve this issue, the user's standard deviation was scaled according to the following formula:

$$\sigma_{user\_s} = \frac{\sigma_{ref} * \overline{f0_{user}}}{\overline{f0_{ref}}}$$

3.3 Datasets

The algorithm described in this thesis was developed in an incremental way. Every change in the code was followed by a round of testing to determine if the change improved or deteriorated the results.

In every round of testing, several audio files were used to verify the results. These audio files were selected to include a fair representation of audio files from real users that the algorithm will be processing, namely in terms of gender, diversity of L1 and diversity of recording conditions (including noise levels and overall amplitude). To have a consistent input within all the test rounds, three pilot datasets were created with selected audio files from real users of ELSA's application.

The first two datasets, DS1 and DS2, were built to aid in the development of the algorithm. This means that they needed to have enough diversity in terms of noise levels, audio quality, mean F0 and duration of the phonemes and silences, but no effort was made to include variability in terms of linguistic contents. In fact, the number of variable factors of the audio files in the testing should be kept to a minimum to extract meaningful results, so these datasets have only one exercise containing one sentence each. They were used to test different parameters in WORLD, its robustness to different recording conditions, the different restrictions to the DTW algorithm and to preform overall debugging.

The last dataset, DS3, had the purpose of testing the final full algorithm and its results were used to perform minor tweaks and optimizations. It was only used after all the parameters from WORLD and DTW were already fixed and the alignment was being done at the phoneme level. This required more dynamic sentences with higher prosodic variability, containing for example, questions and exclamations, as well as long silences in the middle of the utterance. This means that the utterances would be longer and the Speech-Server would return more markers of pitch and duration to work with and evaluate.

The first exercise contains 3 sentences, the fifth exercise contains 1 and the remaining three exercises contain 2 sentences each. There are 10 audio files from different users from each of the 5 exercises.

The datasets are described in detail in the appendix A. These tables contain information about gender and the L1 of the speaker, as well as the duration of the audio and a measure of the speaker nativeness.

The sentences that correspond to each dataset are the following:

- **DS1**: ”This is the best vest I’ve ever worn.”
- **DS2**: ”Our last product launch was smooth sailing.”
- **DS3**
  - **1**: ”Certainly. Would it be for this evening? What time would you like?”
  - **2**: ”Welcome to Tasty! What name is the reservation under?”
  - **3**: ”May I recommend the baked barramundi? It’s exquisite!”
  - **4**: ”Great. Thanks for making that change! How about adding more filters into the search section?”
  - **5**: ”I am thinking about it, do you have any advice for me?”

3.4 Model Fine-Tune and Comments

On section 3.2.3, a baseline version of the Pitch Transplant algorithm was described. Using the methods of WORLD, the DTW algorithm and F0 scaling, it was possible to generate transplanted utterances, and from this point forward, additional methods could be implemented and its results evaluated. In this section, these methods are presented, together with tweaks to the baseline version and other attempts that did not improve the overall results and were abandoned.

3.4.1 WORLD

3.4.1.A Harvest Parameters

The exact upper and lower limit of the F0 used on the final version of the Pitch Transplant algorithm were determined by taking into account values from bibliography [82] and values suggested by the authors of Harvest. These were validated using results from tests with real users’ utterances. The validation was made with the dataset DS1 and is described below.

An audio file was sent to WORLD, where it was decomposed in the three speech components and synthesized back to a waveform signal. All the parameters on all the components of WORLD were kept constant, except for the $f0\_ceil$ and $f0\_floor$ which were initialized at the values similar to the values suggested by the authors and changed at each iteration. The result was analyzed and compared with

the F0 generated by Wavesurfer. The objective was to avoid situations where the F0 of the utterance could not be accommodated by the estimate, either capped on the lower or upper limit, or there was F0 detection on non-speech segments, which is associated with periodic noise.

The starting point of the upper limit was set at 800Hz and it was lowered in steps of 25Hz. The lower limit set at 70Hz and it was increased and lowered in steps of 5Hz. In both cases, when the next iteration had a negative impact on the estimated F0, in comparison with the output of Wavesurfer, the value was rolled back 2 steps (50Hz in case of upper limit and 10Hz in case of lower limit). Mainly on the lower frequencies, there was a significant gap between male and female speakers, which originated recognition of F0 in noise segments on the latter. But since gender information is not available, the chosen limits needed to be suitable for both genders. The algorithm is not designed to accommodate children speakers because there are no children speakers on the datasets available for testing. This means that the upper limit of the F0 does not need to be as high as the suggested by the authors of Harvest. This parameter can easily be changed to accommodate children speakers in the future, if required. The limits used on the final version of the algorithm are presented below.

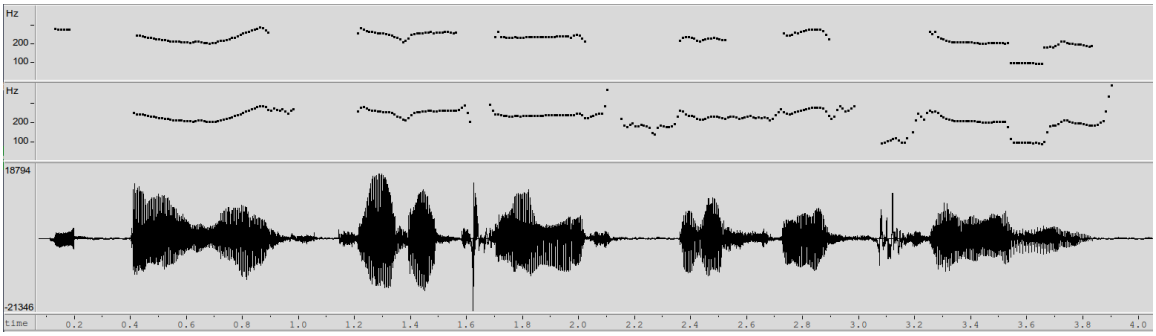
- $f0\_ceil$: 450Hz
- $f0\_floor$: 60Hz

Once these parameters were set, the frame period was changed. Initially it was set to 10ms. This way the frame indexes returned from the phone alignment could be used directly without any manipulation, making it much easier to construct the aligned F0 and preform the transplant. But testing revealed that the frame period of 5ms produced more accurate results and outputted a smoother F0 contour, without having a significant effect on the running time of the algorithm. The tests were similar to the tests made for the F0 limits, and consisted of having a cycle that would run the algorithm with constant parameters, except for the frame period, which assumed the values of 10ms or 5ms.

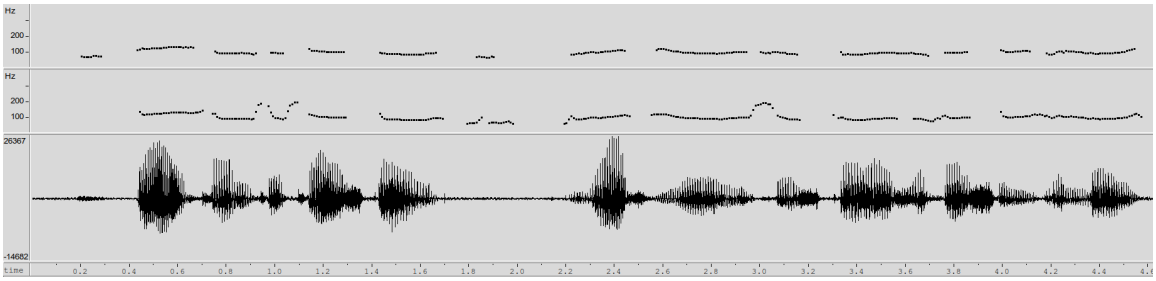
On figure 3.6 there are two examples of the F0 estimation using using Harvest, as well as the F0 contour estimated with wavesurfer. The estimate made with Harvest is slightly worse because, due to its nature, many unvoiced segments were classified as voiced, which raises some problems when using it as a standalone F0 contour estimation algorithm. This is corrected when Harvest is used together with WORLD's D4C algorithm, which produces the information of whether a segment is voiced or unvoiced.

3.4.1.B F0 Smoothing

The last step of the Harvest module is smoothing the result with a zero-lag Butterworth filter, resetting the padded sections to 0. When the audio signal has low noise, this results in a smooth F0 contour. But Pitch Transplant will be applied to noisy audio files, which will produce some undesired fluctuations on the estimated F0 contour. There was an attempt attenuate these fluctuations with a simple moving average filter, followed by the removal of the padded sections to 0. But although this filter produced a



(a) F0 estimation of audio file ds2-id-f-1



(b) F0 estimation of audio file ds3-2-vi-m-1

Figure 3.6: Each figure contains F0 estimation by Wavesurfer on top, F0 estimation by Harvest in the middle, and the Waveform in the bottom.

smoother F0 contour, it created more artifacts in the synthesized audio.

In a second attempt, a moving average box filter by convolution was implemented, also followed by removal of padded sections. Different placements for the filter were attempted, before the spectral envelope estimation and after the full feature extraction. The only situation where the filter did not reduce the quality of the output audio files was when it was applied only to the F0 estimation of the reference utterance, after the full feature extraction. But even in this situation, there were no evident benefits of this smoothing in the final result and tuning the size of the box was impossible because every audio seemed to have slight benefits and drawbacks with very different box sizes (from 3 to 21 frames). Without any recognizable pattern and no significant overall benefit, the F0 smoothing was set aside and it is not implemented in the last version of the algorithm.

3.4.1.C CheapTrick parameters

The only parameter that is changed in Cheaptrick is the fft_size which was set to 1024, to match the $f0\_floor$ of 60Hz that was used in Harvest. The reason the fft_size was calculated instead of just letting cheaptrick make the calculation from the $f0\_floor$ is because the same value will be used to obtain the aperiodicity. The remaining parameters were left as default.

3.4.1.D D4C Parameters

The value of $q1$ was kept default as recommended, the fft_size used was 1024, as in Cheaptrick, so the only value that was changed was the $threshold$. To determine the value that provided optimum results, tests were made with the DS1 dataset. These tests consisted of running the feature extraction and synthesis keeping all conditions constant except for the $threshold$, which was swept from 0 to 1 in increments of 0.05, and comparing the produced audio files. The value of 1 was immediately excluded, as clearly classifying every frame as unvoiced did not produce any listenable result. The best result was produced with the $threshold$ of 0.85, which is the suggested by the author. Some of the input audio files contain noise because of varying recording conditions. This may cause some unvoiced segments to be classified as voiced due to presence of periodicity in the background noise. So a way to prevent this would be by raising slightly the $threshold$, thus classifying these segments as unvoiced and obtaining better results. But the tests revealed that higher than 0.85, the audio quality of the files with no or low noise is greatly reduced due to the opposite effect, classifying voiced segment as unvoiced, while not increasing significantly the results of more noisy files. So the value was finally kept at 0.85.

3.4.1.E Comments on WORLD

WORLD is not yet capable of producing speech that is as natural as the input, but the quality it achieves in such small running time is remarkable. It was chosen because it decomposes an audio in 3 workable components and re-synthesizes it in under a third of the duration of the audio in a device with average computational power [4].

The biggest plus in the choice of WORLD is that it may be used from the moment the user makes the first attempt on the first exercise, getting its feedback in just a few seconds. It does not require the collection and maintenance of a database and does not require a long pre-training process. Also, and as mentioned in the introduction, one of the main objectives of this work was to present users with a gradual reference when doing language exercises. This reference needs to be updated or generated again at every attempt on the exercise. With WORLD there is the possibility of generating a new reference at each attempt.

There are of course some limitations to this algorithm. WORLD is highly sensitive to the quality of the audio that it receives as input. In this work, the system was tested both DS1 and DS2 datasets, containing audio files recorded by real ELSA users. It was noticeable that the performance of WORLD was severely degraded when the audio files were recorded by the user in highly noisy environments or with very low overall volume.

For the cases where the recording had very low overall volume, the estimators, and specially Harvest, had trouble identifying the target features. This was fixed by normalizing the volume of the files received as input. But raising the level of the audio also brings more noise, which leads to a higher number of

artifacts in the synthesized audio and, consequentially lower quality overall.

3.4.2 DTW

3.4.2.A Distance Function

DTW calculates the distances between elements to obtain the optimum alignment path, so it is important to determine which distance function performs better for the provided data. Before running the DTW algorithm, a call was made ELSA's Speech-Server to determine in which frames each phone starts and ends, both from the user and the reference utterances. Then the DTW algorithm was applied to align each pair of phones individually. In the earlier stages, a python implementation of the DTW algorithm [83] that used the *cdist* function from *scipy* python library was employed. This made it possible to test the results of the DTW using all the distance functions from this library, including *braycurtis*, *euclidean*, *sqeuclidean*, *cityblock*, *minkowski*, and others. This test consisted of having a cycle that ran the Pitch Transplant with the different distances, and saving each result (audio and alignment path) in individual files. Euclidean distance was then chosen because it produced the best results together with *braycurtis* and *chebyshev*, and it had the simplest implementation.

The distance function computes the similarity between each possible pair of frames, one from the user's utterance and another from the reference's utterance, in order to build the cost matrix. Each frame is represented by a vector containing the features extracted by WORLD. On the baseline version, only the spectral envelope was used to calculate the similarities between frames. But tests with the DS1 and DS2 datasets revealed that the alignment of the unvoiced consonants was more accurate when the aperiodicity was included.

3.4.2.B Global Path Constraints

The DTW algorithm performs the alignment on each pair of corresponding phones, one from the user's utterance and another from the reference's utterance. Since the phone is the same, it is expected that the alignment path will be close to the diagonal of the cost matrix. In order to reduce the number of distances calculated, a global path constraint was introduced. The choice landed on a simple Sakoe-Chiba band [77] with the width of either 15 or 31 samples, which is chosen according to the difference between the length of the segments for each phone. In case this difference is smaller than 10 samples, the smallest window is used and the largest window is chosen otherwise. For the rare cases where there is a difference between segments larger than 31 samples, the window size is changed to 1.1 times the difference between segments.

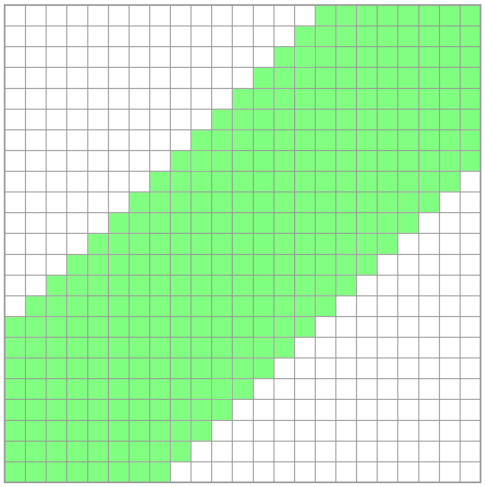


Figure 3.7: Example of a Sakoe-Chiba band with window size of 8 (for simplicity). The green area represents the points in the matrix where the distances are calculated and the warping path may be located.

3.4.2.C Local Path Constraints

These constraints were obtained after experimental observations by running the algorithm with the datasets DS1 and DS2. The resulting utterances were analyzed together with the visualization of the resulting path and the labels returned by the Speech-Server. The allowed steps are represented in Figure 3.8

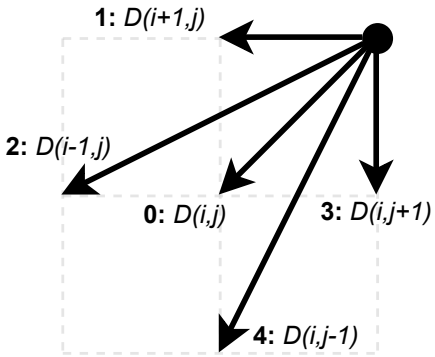


Figure 3.8: Local continuity constraints. The five step patterns selected.

- **0**: $D(i, j)$
- **1**: $D(i + 1, j)$ (max of 3 successive steps)
- **2**: $D(i - 1, j)$ (max of 1 successive step)
- **3**: $D(i, j + 1)$ (max of 3 successive steps)
- **4**: $D(i, j - 1)$ (max of 1 successive step)

These restrictions were chosen taking into account that the objective is to obtain an intermediate point between the reference and the user utterance, and not an utterance that follows the exact F0 contour of the reference. If DTW is left unrestricted, it may introduce too much distortion on the synthesized utterance, by removing too many frames in some phones and repeating too many in other phones. If it is too restricted, the F0contour from the reference w ill not be identifiable in the resulting audio and the purpose of the algorithm will be lost.

From the above possibilities, the steps 2 and 4 allow for the jump to happen between two non adjacent frames of either the reference or the user's utterance. Although this results in non-continuity of the alignment, because some frames from either the user or the reference may be lost, for the purpose of this work this is entirely acceptable. As the objective is obtaining this mid-point between both utterances, some loss may even be necessary in order to reduce the length of a specific phone, and the resulting sound should not be noticeably degraded due to the small size of the frame interval in the feature extraction process.

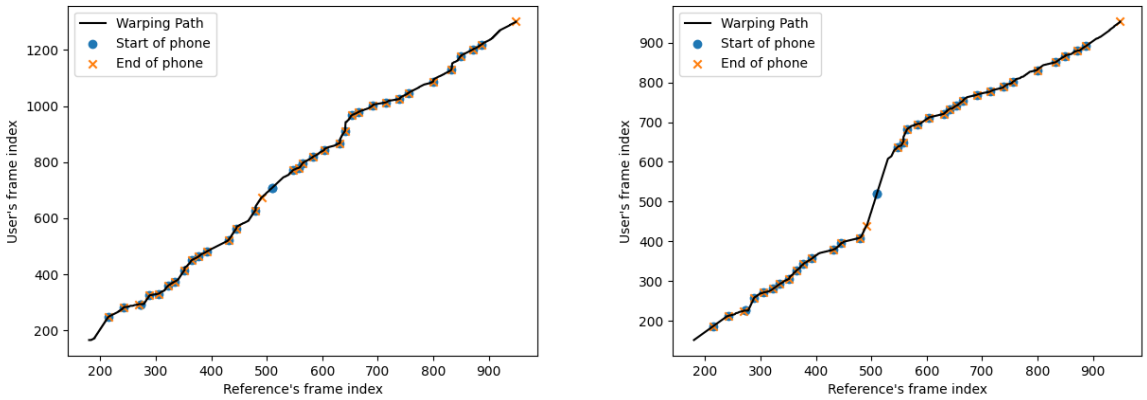
There are some situations where, due to the possible jumps, the alignment will be shorter than both the user's or the reference's corresponding segment. This tends to happen in the cases where the phones are either very short or very long, which makes less than 30% of the phones in tests made with the DS1 dataset. For these cases, there is an extra cycle added which runs again the last step of the DTW without the possibility of preforming at any time the steps described in 2 and 4. This last component is the trace-back algorithm, which runs through the calculated distance matrix from the end to the beginning and obtains the optimum path, respecting all the restrictions mentioned above. It is important to note that in the tests made, from all 3 datasets, DTW never returned a path longer than both inputs, so there was no need to prepare the code for this event, as there is also no way of testing it.

Figure 3.9 shows the warping path that resulted from aligning 2 utterances from two different users, one male and one female. It also contains markings for the beginning and end of each phone. The resulting F0 contours from the utterances synthesized after these alignments are presented on figure 3.10. The F0 contour from the user's utterance is presented on the top, the one from the reference's utterance on the bottom and in the middle is the F0 contour from the synthesized transplanted utterance.

3.4.3 SNR verification

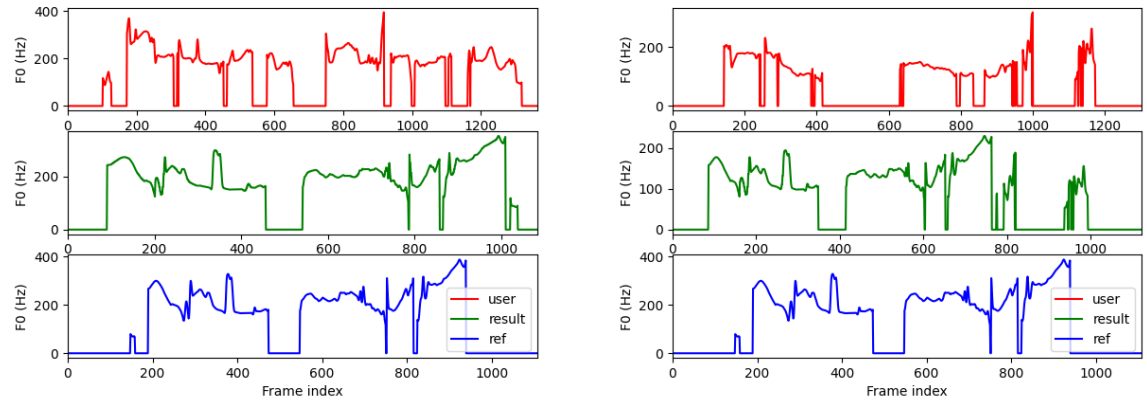
Pitch Transplant was designed as a proof of concept for a real application and was tested with real audio files recorded by users that use ELSA's app on their smartphones. Even though it is advised to use the app in quiet environments for the best possible experience, it is not always the case. The level of noise varies and it is possible to identify sources like loud dialogues in the background, continuous noise similar to fans, high levels of echo or wind and others.

To guarantee that the output of the Pitch Transplant does not contain a high level of noise, and due



(a) ds3-5-vi-f-1 **(b)** ds3-5-vi-m-2

Figure 3.9: Warping path of two utterances. Utterance (a) was more similar to the reference than utterance (b), which is noticeable by the less diagonal path.



(a) ds3-5-vi-f-1 **(b)** ds3-5-vi-m-2

Figure 3.10: F0 contour of the user, reference and the converted reference (result of Pitch Transplant algorithm) for two utterances from two different users

to the high sensitivity of WORLD to it, a verification of the noise level of the audio was introduced. The chosen method is named WADA SNR [84] and it estimates the SNR based on Waveform Amplitude Distribution Analysis. It assumes that the amplitude distribution of clean speech can be approximated by the Gamma distribution with a shaping parameter of 0.4, and that an additive noise signal is Gaussian. It has a python implementation [85] which made it simple to test and integrate into the Pitch Transplant algorithm.

The threshold value chosen for this noise verification is SNR = 50, which means that utterances with a SNR lower than 50 will not be processed. This value was obtained through testing with the DS1 dataset. The tests consisted of cycling through all the utterances, computing the SNR value and using WORLD to extract the features and synthesize an output utterance. In the end, both the original utterances, the synthesized utterances, the SNR value and the average SNR of all utterances were evaluated together. The threshold set at 50 results in the exclusion of 12 out of the 30 audio files from DS1, which is perfectly acceptable knowing that there was no criteria in the choice of the files in terms of noise. The audio files above the threshold produce a better quality output, containing less artifacts both with and without the F0 manipulation through DTW.

An alternative to this method would be applying a noise filter on the whole audio. The fact that the files are recorded with different microphones (depending on the mobile device used), in very different environments makes it difficult to apply a universal method for noise reduction, although there are already some possibilities using Deep Neural Networks [86], as for example KrispNet DNN developed by NVidia.

3.4.4 Feature normalization

The DTW algorithm receives as input a stack made of both the spectral envelope and the aperiodicity features, that result from CheapTrick and D4C methods, respectively. But while the aperiodicity values range between 0.001 and 1, the range of the values of the spectral envelope differs greatly in several orders of magnitude, from 10^3 to 10^{-18}. To test the impact of the aperiodicity in the alignment, a normalization of both the spectral envelope and the aperiodicity was introduced. It is important to note that this normalization was used only to perform the alignment. The original non-normalized features were kept untouched to perform the synthesis once the time alignment was determined.

Two different normalization techniques were tested. The test consisted of running the Pitch Transplant algorithm keeping all parameters constant except for the pre-processing of the input of the DTW. The results with non-normalized features were compared with the results from both normalization methods by listening to the results and looking at the outcome of the evaluation method proposed with this work. The tests were done with the DS3 dataset.

The first technique is Z-Normalization [87], also called Standardization, and it consists of a normal-

ization to zero mean and unit of energy. This normalization is achieved with the following formula:

$$x' = \frac{x - \mu}{\sigma}$$

Here, x represents the raw features extracted from WORLD and x' the normalized features. First the mean of the time-series is subtracted and the result is divided by the standard deviation. This is a commonly suggested normalization done as a pre-processing step for data mining problems where DTW is commonly employed. The objective is to remove the amplitude variation between time-series, so that the DTW algorithm can focus solely on on the structural similarities. But when applied to the Pitch Transplant, this normalization led worsened results, possibly because it does not produce normalized data with the exact same scale. The features extracted with WORLD are highly sensitive, and this method introduced more noise into the data and made the alignment task harder.

The second method tested was the min-max Normalization, also known only as normalization of feature scaling. The mathematical formulation is the following:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where x_{max} and x_{min} are the maximum and minimum values of the dataset. In this case, the spectral envelope and the aperiodic features were normalized separately and in this way the values were all kept within 0 and 1, without maximizing its variance. An extra multiplier of 0.5 was added to the aperiodicity with the intent of lowering the impact of these features in the final alignment. Due to the presence of noise, the aperiodicity may influence greatly the warp path and worsen the results. The tests performed with this normalization produced better results and reduced at least one error marker in each processed audio (with the proposed evaluation method), in comparison to the non normalized data.

3.5 Evaluation

There is no established protocol on how to evaluate any speech synthesis system and different metrics may be taken into account, like intelligibility, naturalness and even the speed with which the algorithm performs the designated task. The pitch Transplant algorithm presented in this work makes use of WORLD, a Vocoder based system. A typical subjective evaluation done on Vocoder systems when performing copy synthesis is the MUSHRA listening test, already presented here and used by the authors of WORLD. But for the purpose of this task, which involves mainly F0 manipulation, this would not be appropriate as the audio samples in analysis are not intended to be similar.

Both subjective and objective evaluation is proposed in this work. On the subjective analysis side, Mean Opinion Score (MOS), AB and ABX tests will be performed. On the objective side, a method will

be proposed that will use the markers returned by ELSA's Speech-Server for both pitch and duration of each word to determine the approximation of the synthesized audio to the reference.

The evaluation methods proposed above take into account mainly the quality of the audio produced by the algorithm and the improvement on the fluency, nativeness and naturalness of the speech. Pitch Transplant was design to provide a floating point reference to the users learning English, and the best way to test it would be including it in the exercise and evaluating the improvement of the students. Although a script was prepared as a standalone tool for recording the audio, preforming the Pitch Transplant, and playing back the result to the user, even giving a feedback on the improvements similarly to the ELSA App, it was impossible to make tests at a large scale to obtain meaningful results. Both restrictions in terms of proximity to the users due to the COVID-19 pandemic and the inability to run this test for several weeks and follow the users' improvements on the long term made this test impossible to achieve at this time.

Pitch Transplant is performed exclusively using CPU and all tests were done in a standard laptop (Intel Core i5-8250U CPU MAX 3.4 GHz, and 8 GB RAM).

3.5.1 Subjective Testing

The goal of this evaluation is to determine if the algorithm improves the audio in terms of nativeness, fluency and naturalness, while keeping the audio quality at a similar level. For this purpose, 4 different audio files were chosen from datasets DS2 and DS3, two from female speakers and two from male speakers, containing each a different sentence. Then, 8 samples were produced, 4 were the result of the Pitch Transplant algorithm and the remaining 4 were copy synthesized (without any manipulation) using WORLD. This was made to remove the impact that WORLD may have in the audio quality of the result, due to the presence of noise. Also, the four reference audios from ELSA's speech artist were added to the tests, two of which were also copy synthesized with WORLD, leaving the remaining unchanged. All the audio files were in wav format, single channel, with a sampling rate of 16KHz at 16bits/sample.

The tests were condensed into an online survey for easy distribution, using the sogosurvey platform. Each test was performed on the 4 sets of 3 samples: the original audio (synthesized with WORLD), the transplanted audio (which results from applying the pitch transplant to the original audio file) and the reference audio (either unchanged or synthesized with WORLD). A total of 40 subjects answered the tests for each of the 4 sentences.

3.5.1.A AB test

The pair of samples on this test are the original and the transplanted audio. Subjects were asked to listen to both samples and chose which sounds more native, fluent and natural. The neutral option was

also given and the sample order was mixed. The results may be found below.

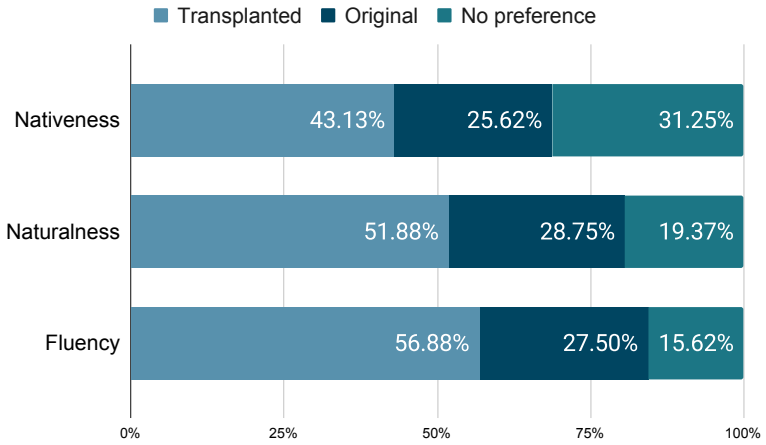


Figure 3.11: AB tests results

In terms of nativeness, there is no strong tendency towards the transplanted or the original utterance, and more than 30% of the subjects chose the option of no preference. As nativeness takes into account not only prosodic features, but also pronunciation, it is likely that the presence of pronunciation mistakes on the utterances led to inconclusive results. This algorithm is intended to be presented to users that already have a very good pronunciation, which was not the case on the available user audio files, and pronunciation correction is not contemplated in its design.

On naturalness and fluency, it is possible to see a tendency towards the transplanted algorithm. The indifference option was chosen by less subjects, and choice of the original utterance remained similar in all three tests. The majority of the subjects classified the Transplanted utterances as better in terms of Naturalness and Fluency than the original utterance.

3.5.1.B ABX test

The same pair of samples from the AB test were used and a third audio sample (X) containing the reference audio from ELSA's speech artist was added. The subjects were asked to choose from the first pair of samples (A and B) which one was more similar to the reference audio file in terms of intonation, tone, rhythm, and stress. The results may be found in figure 3.12.

The majority of the subjects classified the transplanted utterance as the closest to the utterance from ELSA's speech artist in terms of prosody. This is in line with the results of the AB testing, where this utterance was also chosen by the majority as better in terms of fluency.

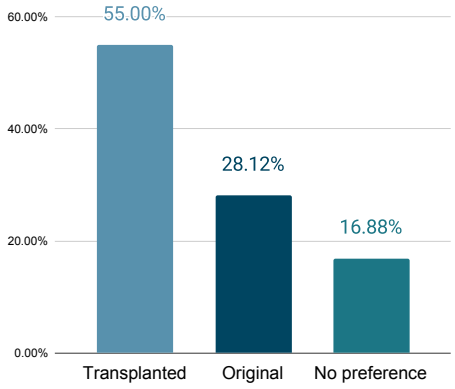


Figure 3.12: ABX tests results

3.5.1.C MOS

The subjects were asked to rate the audio quality of each of the three samples in a 5-point Likert scale score (1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent). On the user's side, all the samples were either copy synthesized with WORLD, or the result of the Pitch Transplant algorithm. To keep the test centered on the quality of the algorithm, no original samples, without being passed through WORLD, were included. On the reference side, half of the samples were original and half were copy synthesized with WORLD. The objective is to verify how much WORLD reduces the quality of the samples. The results are given in the table below, together with the 95% confidence interval.

Table 3.1: Mean Opinion Score together with the 95% confidence interval

Audio Sample	MOS
User Transplanted Audio	2.83 ± 0.14
User Audio synthesized with WORLD	2.89 ± 0.14
Reference Audio synthesized with WORLD	4.39 ± 0.16
Reference Audio (original)	4.72 ± 0.14

It is immediately noticeable that there is a significant gap between the MOS of the user's audio files and the one from the reference audio files. Comparing only the files that were copy synthesized with WORLD, this difference of score is 1.5, having on the lower side the user's audio with the score of 2.89, revealing a very poor quality with reduced usability, and on the higher side the reference audio, with an MOS of 4.39. Even though the results for the users' audio files may be influenced by the their poor English speaking abilities, this still indicates that these audio files have a much lower quality to begin with. The presence of high levels of noise and distortion on the voice render the audio files on ELSA's datasets almost unusable for any possible speech synthesis purpose. This was an issue raised since the beginning of this work, and these results seem to confirm it. This will be further discussed in the following chapter about Voice Conversion, where the similar conclusions may be drawn.

As to the audio quality reduction caused by WORLD, it seems that even though it causes a slight decrease, looking at the original and copy synthesized reference audio files, this decrease is not significant enough to make the audio files unusable in this context. It confirms the tests made by WORLD authors, placing it as one of the best speech synthesizers suitable for real-time applications.

In terms of the effect of the Pitch Transplant algorithm on the audio quality, results indicate that it is almost non existent. The users' transplanted audio scores just 0.06 below the users' copy synthesized audio with WORLD on this MOS test, which is too small to be significant, and sits within a difference that the majority of the subjects probably cannot notice [88]. Performing the Pitch Transplant in such a way that the quality of the audio would not be significantly dropped was one of the objectives when developing this algorithm, that seems to be fulfilled.

3.5.2 Objective Testing

When performing a Prominence exercise on ELSA app, a prominence marker is calculated for each word in the sentence. These markers include the analysis of the duration, pitch of each word and other metrics, and return either ”normal” or ”error” whether the submitted recording is close enough to the reference or not. In case a pause is made by the user between two words that should not have a silence between, an extra marker of silence error is added. This algorithm is intended to approximate the user's speech utterances to the reference, creating a floating point reference that will improve as the user improves. So it is expected that at each iteration a percentage of these markers will go from ”error” to ”normal” until the user manages to replicate almost entirely the prosody of the reference. From hereby on, the ”normal” markers will be referred to as correct markers, for simplicity.

The proposed objective evaluation method makes use of these markers to compute a percentage of how many markers were correct before and after the pitch transplant takes place. A request is made to the Speech-Server to get the total and the correct number of markers (both pitch and duration) for each utterance. The percentage of correct markers from a given utterance can calculated with the following formula:

$$avg\_marker\_score(\%) = \frac{\sum correct\_markers}{\sum total\_markers} \times 100$$

Where, for each utterance:

$$total\_markers = correct\_markers + error\_markers + silence\_markers$$

This is made for all the utterances of each dataset, resulting in the average marker_score per dataset from DS3 that can be seen in 3.2. The results shown on the column ”Before Pitch Transplant” were produced using the original utterances from the users and on the column ”After Pitch Transplant” are

Table 3.2: Results of Marker Score Test

Dataset	Before Pitch Transplant %	After Pitch Transplant %
DS3-1	79.23 ± 5.59	95.0 ± 2.35
DS3-2	87.01 ± 5.15	91.67 ± 6.00
DS3-3	86.12 ± 6.72	93.75 ± 3.65
DS3-4	88.89 ± 4.50	94.0 ± 2.19
DS3-5	88.63 ± 3.96	97.08 ± 2.01
Average	85.98	94.30

the results produced with the transplanted audio files. In the end, an average is computed to indicate an overall improvement of the algorithm.

For every sentence, the audio synthesized with the pitch transplant results in a higher score, with an improvement between 4% and 11%. As it is an algorithm designed to provide a gradual reference, it is not expected that all errors are corrected, so the observed improvement is as expected, not reaching a score of 100% in any case. The algorithm presents an improvement of 8.32%, resulting from the correction of the 59% of the error markers, on average.

3.5.3 Yes/No question

At the end of the survey, the subjects were asked if they would be comfortable listening to their own manipulated (corrected) voice as a reference in a language learning context. It was a yes or no question, but the indifference option was also given. The answers are shown below, and indicate that the majority would be comfortable with it. This means that if high audio quality is achieved, an algorithm like Pitch Transplant could be added to a CAPT system.

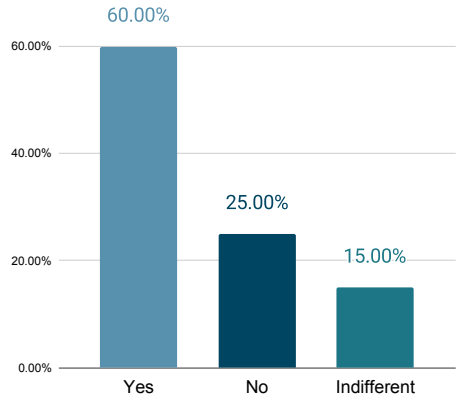


Figure 3.13: Responses to the question ”Would you be comfortable if, in a language learning context, you would listen to your own manipulated (corrected) voice as a reference?”

4

Voice Conversion Approach

Contents

4.1 Introduction

This chapter explores an alternative approach to tackle the problem of this thesis. Instead of manipulating a recording from the user, the idea is to produce the user's goal utterance in his/her own voice, by way of VC. This technique has been an increasingly hot topic, specially with the introduction of Deep Learning in the last decade, and new models are proposed every year that provide better results and higher adaptability to the restraints of different problems. Therefore, in this context, the terms reference and user will be replaced by source and target, respectively.

On a typical Voice Conversion application, the linguistic content is fixed and it is expected that the remaining speech features are converted from the source to the target, resulting in an utterance that sounds as close as possible to the target speaker's natural speech. But in this work, for the purpose of prosody training in a language learning exercise, the prosodic patterns from the source should be kept. The objective is to generate an audio file with the voice of the user, but with all the communicative information, namely the linguistic and paralinguistic activity (as defined by John Laver [59]), from the source. Today's Voice Conversion models can achieve high quality speech synthesis, which means that this method would present the user with a high quality reference utterance in his/her own voice, correct in both segmental and supra-segmental aspects.

The main objective of this chapter is to generate discussion and to explore the potential of Voice Conversion applied to language learning. Due to time limitations, after the development of the Pitch Transplant algorithm presented in the previous chapter, no extensive research on all the existing VC alternative methods was made. Also, the chosen algorithm was used with minimum tweaking, and the time was spent mainly on the search and preparation of a dataset with high prosodic variance, that could be used for the pre-train of a VC model as well as verifying if the user's data from the ELSA app was usable for this purpose. We did not intend to propose a finalized algorithm and a solution to the problem.

This chapter starts by introducing the Voice Conversion approach. In the second section, a brief overview of the chosen VC algorithm is made, including the requirements and restrictions of the task, the necessary preprocessing, the basic architecture of the algorithm and the chosen waveform generator method. The third section contains information about all the datasets used for both pre-training and fine-tuning of the model. The fourth section goes through all the tests performed with these datasets and discusses of the results. The fifth section presents the evaluation of the results, which was done by 40 subjects through an online survey.

4.2 Algorithm

4.2.1 Requirements and restrictions

In order to choose a VC algorithm that would fit the requirements of the task at hand, it is first necessary to determine exactly where it would be used and what are its requirements. As with Pitch Transplant, the objective is to generate a goal utterance in the user's voice to be used as reference in a language learning exercise. But this time, instead of taking an utterance from the user and modifying it, the result will be produced by using a pre-trained VC algorithm. This means that ELSA's reference speech artist (currently a female voice) will be the source and the task will be converting her audio files to the voice of the user, which will be the target.

For this to take place, the VC algorithm needs to fit the following constraints:

- **Small amount of target data** - The option of listening to the reference utterance in his/her own voice should be available to the user early on, so the algorithm should require a small amount of the user's audio files.
- **Generate audio files fast** - Even though the training of the algorithm can be made offline, the generation of the audio files should be done fast. Ideally, it should happen during the loading of one exercise, or when an entire module is downloaded. Generating all the converted audio files beforehand and storing them could also be an option, but it is not the current goal.
- **Keeping prosodic features** - The algorithm should allow for the maintenance of the prosodic patterns of the source speaker.

Apart from these constraints, it is necessary to use an algorithm that is not too complex to implement. Since it is the second method tested on this thesis, and because the first method was developed from scratch, the choice of the Voice Conversion algorithm was conditioned on being able to obtain results in a short amount of time. Because of this, the choice landed on a Non-Parallel Sequence-to-Sequence Voice Conversion Algorithm with Disentangled Linguistic and Speaker Representations, presented in [8]. This method was being used as a basis for the MsC Thesis of Ivan Carapinha, also a member of the research group and under the supervision of Prof. Isabel Trancoso, who provided valuable input regarding data pre-processing, bug fixing and tips related to the learning rate decay. His initial tests indicated that the algorithm fitted into the first two of the above constraints. The inference process was fast, taking under 5 seconds per utterance (excluding the waveform generation) and the fine-tune process could converge and achieve decent results with under 200 audio files of the target speaker. As for the third constraint, related with the prosodic features, it will be determined by testing.

The algorithm is shipped with an implementation of a Griffin-Lim Vocoder [89] as the waveform generator. This is a simple and computationally cheap algorithm that uses the predicted spectrograms to

generate the audio waveforms. Per Ivan's suggestion, and to obtain improved results, a pre-trained model of the Universal Vocoder [57] will be used instead to generate higher quality waveforms. This requires an extra stage on the pre-processing which will be explained below.

ELSA's speaking exercises consist of the users reading a sentence which is unique for each exercise. Because of this, it is possible to generate a dataset structured in a parallel way, where each user is chosen on the condition of having completed a specific set of exercises and only these exercises are chosen, which would result in a perfectly balanced parallel dataset. So, even though it may seem counter intuitive that the VC method chosen is prepared for non-parallel data, in reality this allows for the usage of different datasets on the pre-train. This is particularly useful because the data collected from the users has, in general, very low sound quality. The audio files are recorded with hundreds of different mobile devices in noisy environments, and it is not clear if this will allow convergence of the model, so an alternative dataset may be required. Furthermore, the results presented by the authors [8] place this model's performance very close to the state-of-the-art parallel seq2seq VC method, so there is almost no trade-off in this choice.

4.2.2 Preprocessing

The code for the Non-parallel Seq2seq Voice Conversion algorithm is provided in the authors github repository [90]. The algorithm was used mostly as is, with very few changes. Nevertheless, since this is non-released code, still under development, it contained legacy code and several bugs that needed to be fixed. Apart from this, the data reader needed to be customized for the datasets that were used, and there was no code written for the generation of the list that divides the files into training, validation and test sets. This code was written with Ivan's help, since he had already went through this process for his own work, and adapted for the different datasets used.

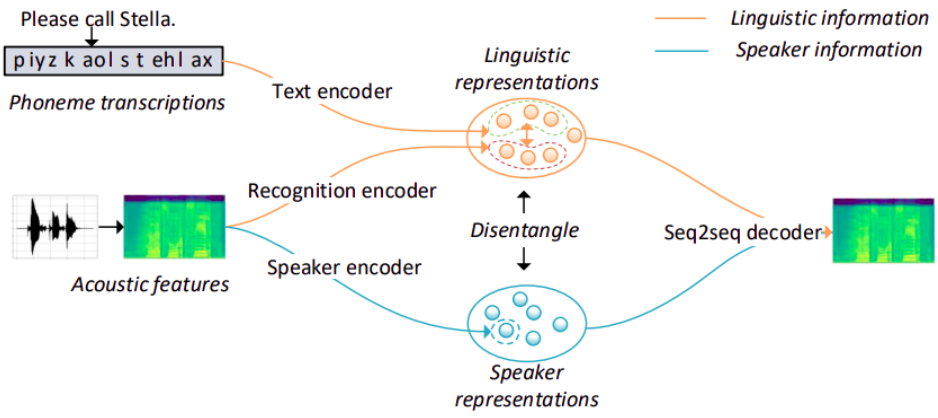
The author provided a feature extraction script, extract_features.py, that extracts the linear spectrograms and mel-spectrograms from the audio files using the librosa [91] python library and the phone sequences from the text files using the phonemizer [92] python library. Because the waveform generator used in this work requires only mel-spectrograms, the linear spectrograms used on Griffin-Lim vocoder are not kept, and the remaining code was adapted to remove the necessity of these features. This was made according to suggestion by the author on an issue in the same repository and does not change in any way the behaviour of the algorithm. One significant change was introduced on the feature extraction script, to improve the quality of the audio file generated with the Universal Vocoder. This change was the addition of a peak amplitude normalization and a digital filter to add pre-emphasis to the utterances. It was added so that the pre-trained model could be used, which preforms this pre-emphasis to the input files used on training.

One script, preprocess.py, was created to walk through the dataset directory and generate the files

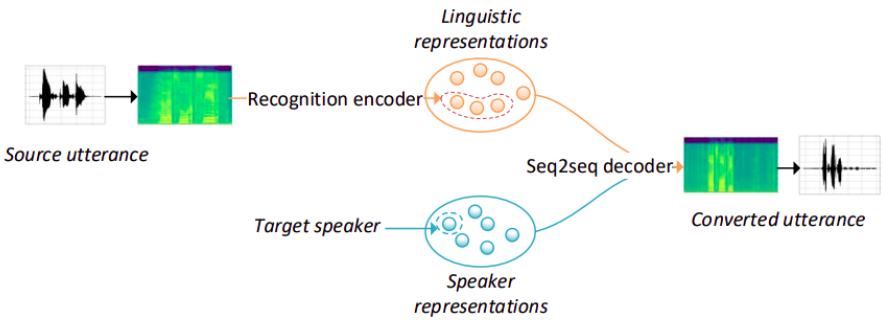
containing the train, validation and test lists. These lists contain the paths to the files containing the mel-spectograms that were generated with the feature extraction script, and the algorithm is prepared to also obtain the paths to the files containing the phone sequence through them. To prevent out of memory errors, all the utterances longer than 7.5 seconds were not added to these lists, as per recommendation of the author. This script was changed to preform the preprocessing of the LibriTTS dataset. Since this is a less balanced dataset, two functions were added. One removes from the lists all utterances that contain less than 4 words, and the other removes the speakers with less than 50 audio files after this filtering. This was done to avoid bias on the training process of the algorithm when processing smaller utterances.

One bash script was written to call both the feature extraction script (feature_extraction.py) and the list generation script (preprocess.py) and fully prepare the necessary data for the training to start. This data was all saved in one folder with the name of the dataset and all the paths in the code were adapted to this directory layout.

4.2.3 Architecture



(a) Training Phase



(b) Conversion Phase

Figure 4.1: Overview of both phases of the Voice Conversion model. Taken from [8]

A brief overview of the architecture of this system, which is built under the framework of encoder-decoder neural networks, will be made in this section. The system performs sequence-to-sequence (seq2seq) voice conversion using non-parallel training data. The process can be broadly divided into two distinct phases, training and conversion. Since the model was used with minimal tweaking, with a black-box approach, the description of the algorithm will not be done in detail.

The training phase is responsible for the estimation of the model's parameters and it is done in two stages, the pre-training stage, which uses a multi-speaker dataset, and the fine-tuning stage performed on a specific pair of speakers. The conversion phase receives the acoustic features of the source audio file and converts them to the target using the parameters estimated in the training phase. The converted audio file can then be synthesized by a waveform generator.

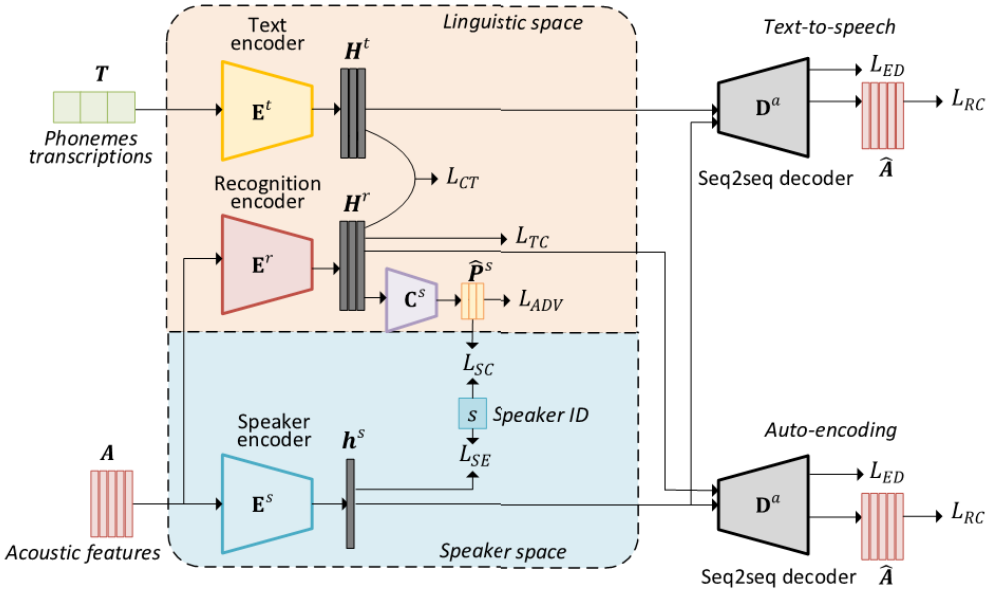


Figure 4.2: Structure of the Non-parallel Seq2seq Voice Conversion algorithm. Taken from [8]

The model is composed by the five following components:

- **Text Encoder** E^t - Transforms the text inputs T into linguistic embeddings H^t.
- **Recognition Encoder** E^τ - Receives the acoustic feature sequence A and predicts the phoneme sequence T, aligning the acoustic and phoneme sequences automatically. Since one phoneme usually corresponds to tens of acoustic frames, the encoding is a compression process. Its output H^τ has the same length as the phoneme sequence T regardless of the speaking rate of speakers, and resides in the same linguistic space as H^t, containing only linguistic information.
- **Speaker Encoder** E^S - Embeds the acoustic feature sequence A into a speaker embedding vector h^S which can discriminate speaker identities and should contain only speaker-related information.

It is only employed at pre-training, whereas at fine-tuning stage a trainable speaker embedding is introduced for each speaker, initialized by h^S.

- **Auxiliary classifier** C^S - Employed to predict the speaker identity from the linguistic representation H^τ of the audio input. It is introduced for adversarial training in order to eliminate remaining speaker information within linguistic representation H^τ. Each element of its output $\hat{P^S}$ is the predicted probability distribution among speakers.
- **Seq2seq decoder** D^a - Recovers acoustic features from the combination of speaker embeddings h^S and linguistic embeddings, H^τ or H^t, either of each fed into the decoder at each training step. It can be viewed as a decompressing process, in which the linguistic contents are transformed back in acoustic features $\hat{A}$, conditioned on the speaker identity information. The structure is similar to the Tacotron model [93] for speech analysis.

As previously stated, the algorithm was used mostly without changes, apart from the removed legacy code. The majority of the work was devoted into the preprocessing scripts, to create the datasets, extract acoustic features, make the lists for each of the train, validation and test sets. On the algorithm, only one significant change was made. A learning rate decay was added to the pre-train script, which was already included in the fine-tune script. In the initial tests, the model kept collapsing, and the results were only noise and silence, so this portion of code was copied from the fine-tune scripts. After this change, the algorithm managed to train successfully.

After the training completed, the predicted mel-spectograms were generated by running the inference script, inference.py, with the files in the test set. These were then used to generate the wavefiles that could be heard and analysed.

4.2.4 Waveform Generation

The Universal Vocoder [57] is a WaveRNN-based neural vocoder developed to overcome the over-fitting that other neural vocoders are prone to. It can be used with speakers unseen in training, making it ideal to test in a black-box approach. The authors provide a pre-trained model, making audio generation an easy and seamless process. The pre-train is made with audio files sampled at 16KHz, which is the same sampling frequency used on the VC algorithm.

Even though the generation is not as fast as the Griffin-Lim Vocoder, the higher quality of the generated audio files is a priority when the goal is to use these audio files in a language learning exercise. In a productized version of this algorithm, the waveform generation would have to be done with a much faster waveform generator, otherwise it could not be used for the intended purpose. On average, for the last test done, described in section 4.4.8, 66 generated files containing 265 seconds of speech took 2543 seconds to generate, which gives on average of almost 10 seconds of running time to generate 1

second of audio. It is important to notice that it was possible to optimize the generation algorithm, but the improvement would only be on the loading of the pre-trained checkpoint, and would change less than 3 seconds per utterance. Due to the time constraints, it was not possible to investigate and test other waveform generators that would fit better into the requirements of the problem.

4.3 Datasets

In this section, all the datasets relevant to the Voice Conversion chapter will be presented, including their characteristics and a brief subjective comment related to their prosodic relevance for the task at hand. This prosodic relevance is mostly related to the variance of the prosodic features in the utterances that compose the dataset. For example, a dataset composed of only declarative sentences, recorded in a monotonic tone will have less prosodic relevance than a dataset composed of dialogues, questions and some level of emotional speech. This is important to evaluate because ELSA's speech artist records the reference audio files for the exercises with exaggerated stress, pauses and pitch variation for learning purposes, and the converted speech should keep these characteristics.

Table 4.1: Table containing the summary of the main characteristics of each dataset. The prosodic value is a very subjective assessment based on the textual content and listening of a small subset of the audio files. "env" stands for the environment where the recording was made.

Dataset	Sampling Freq. & Bit Depth	Length of audio min	Prosodic value	Comments
VCTK	48KHz/16bit	1640	Low	Studio recording, clean speech
ARCTIC-rms	16KHz/16bit	66	Medium	Studio recording, clean speech
ARCTIC-slt	16KHz/16bit	57	Medium	Studio recording, clean speech
ELSA-REF$^{(*)}$	16KHz/16bit	183	High	Silent env, clean speech
LibriTTS	24KHz/16bit	12372	Medium	Studio recording, clean speech
ELSA-USR1$^{(*)}$	16KHz/16bit	2	High	Noisy env, low intelligibility speech
ELSA-USR2$^{(*)}$	16KHz/16bit	1	High	Silent env, clean speech
ELSA-USR3$^{(*)}$	16KHz/16bit	10	High	Noisy env, low intelligibility speech
L2-ARCTIC-NCC	44.1KHz/16bit	70	Medium	Silent env, clean speech
L2-ARCTIC-HQTV	44.1KHz/16bit	69	Medium	Silent env, clean speech

$(*)$ Non public datasets constructed or adapted for the purpose of this work using protected data from ELSA Corp.

4.3.1 VCTK

The CSTR VCTK corpus [94] is a public multi-speaker dataset composed by audio files of 109 native English speakers with different accents. Each speaker recorded around 400 audios of clean speech, composed by a set of sentences common to every speaker and another set of unique sentences, taken from a newspaper. The dataset contains both the audio files and the respective transcriptions for every

speaker except for speaker 315, whose transcriptions are not present. The audio files have a bit depth of 16bits and a sampling frequency of 48KHz.

Because sentences are taken mostly from newspapers, they are in general declarative, without high prosodic variability. No background or static noise is noticeable.

4.3.2 ARCTIC

The CMU ARCTIC database [95] is a dataset designed to support speech synthesis systems. It contains approximately 1200 phonetically balanced utterances recorded in a studio per speaker. The dataset includes wavefiles, transcriptions and several other files required to support a Festival Speech Synthesis System. Out of the 4 speakers on the original version, only ”rms” (male) and ”slt” (female) are mentioned in this work, both US English native speakers and with around 51 minutes of audio per speaker. The corpus is released as free software. The audio files have a bit depth of 16bits and a sampling frequency of 16KHz.

The sentences are taken from out-of-copyright books of English Language that are part of the Gutenberg Project. Because the dataset contains some dialog and more expressive reading, the prosodic relevance of this dataset was considered higher than the VCTK database, for the context of this work.

4.3.3 ELSA-REF

In order to train an algorithm that would be able to convert the voice from ELSA’s speech artist to the voice of the user, a dataset containing these audio files needs to be created. This dataset was named ELSA-REF because it is composed exclusively by the audio references from ELSA’s exercises, recorded by a female speaker.

A Python script was developed to generate this database. The script would verify a TSV file (tab separated values) containing the sentences and the URL location of these audio files in mp3 format. In case the sentence has 4 words or more, the file is downloaded and converted to wav format, 16KHz of sampling rate, 16 bits per sample and single channel. The files are numbered incrementally and a file of the same name but with txt extension is stored in the same folder, containing the sentence.

This resulted in a corpus of 2204 files, containing 183 minutes of clean speech. The recording conditions change slightly in some audio files, which may contain some echo. Nevertheless, the quality of the audio files is still high and close to studio quality. The sentences of 3 words or less were removed because the existence of many short sentences in the dataset could introduce bias in the Voice Conversion algorithm when used in training.

4.3.4 LibriTTS

LibriTTS [96] is a speech corpus designed for text-to-speech use. It is derived from audiobooks that are part of the LibriVox project and similar to the LibriSpeech [97] dataset, but with three main differences that make it more desirable to use in the task proposed in this chapter:

1. The audio files correspond to shorter segments, which means that more utterances will fit into the restrictions of the chosen VC algorithm.
2. The utterances with a significant background noise are excluded, using a WADA SNR estimator to filter out noisy utterances, similarly to the Pitch Transplant algorithm.
3. The text is not normalized into uppercase and contains punctuation, which are useful features to learn prosodic characteristics such intonational groups and the length of pauses.

Although this non-paralel dataset is composed by 2456 speakers, which is far more than VCTK, and it is gender balanced, not all speakers were used. The authors split the speakers into groups of ”clean” and ”other”. The first group contains the speakers whose utterances have lower word error rates (WERs) and a higher SNR, and the remaining are included in the second group. Having in mind the requirements of the VC models, the speakers chosen for the pre-processing fit into the following restrictions: they need to belong to one of the ”clean” groups, which means their audio files will have low WERs and high SNR; they have at least 50 utterances with sentences at least 4 words long, the latter being a restriction also made on the ELSA-REF dataset. Including speakers with less than 50 utterances could result in a bias in the training phase. The resulting dataset contains 12372 minutes of single channel audio from 983 speakers, with a 24kHz sampling rate at 16 bits.

Because it is taken from audiobooks and includes a more emotional reading, the prosodic relevance of this dataset seems to be higher than VCTK. It is read by native speakers and freely available to download and use.

4.3.5 ELSA-USR

Several tests need to be preformed in order to determine in which conditions this VC model achieves a successful voice conversion. To perform these tests, it is necessary to gather different datasets containing user's data with different characteristics. In all the cases, I requested the help of ELSA's Speech Team, who have access to the audio files, and have some already pre-prepared datasets I could use. All the audio files are wav files sampled at 16KHz, with 16 bits per sample. The chosen datasets are the following:

- **ELSA_USR1** - 13 Audio files from ELSA's assessment test, with a score of 37% of nativeness. The speaker is male and his L1 is Vietnamese. The audios have noticeable background noise and the

pronunciation is poor. The audio files needed to be broken into smaller files, due to restrictions of the algorithm, resulting in 23 audio files with a total of 130 seconds of speech.

- **ELSA_USR2** - 13 Audio files from ELSA's assessment test, with a score of 97% of nativeness. The speaker is female, her L1 is American English. The audio files have very low noise and the pronunciation is excellent. The total duration of this subset is 67 seconds.
- **ELSA_USR3** - 170 Audio files from ELSA's exercises. The speaker is male, his L1 is Vietnamese and the pronunciation is poor. The audio files have very different recording conditions. There are three factors that are noticeable in this user's audio files, which can be heard across the majority of ELSA's users: the noise levels vary from barely noticeable to very high, including some audio files where the wind noise is higher than the user's own voice; some audio files contain highly distorted voice, most likely from speaking too close to the headphones microphone; some of the audio files seem to be spoken by a different user (also male). These factors and the problems they lead to will be commented on section 4.4, together with the results from this training.

4.3.6 L2-ARCTIC

L2-ARCTIC [98] is a speech corpus of non-native English intended for research in voice conversion, accent conversion, and mispronunciation detection. It was recorded in a quiet environment using 1132 sentences in the CMU ARCTIC prompts. From this dataset, two speakers were chosen: "NCC" - a female Mandarin native speaker; and "HQTV" - a male native Vietnamese speaker. The choice of these users was made after listening to a few samples and detecting that both speakers had poor English pronunciation. The files from the first speaker amount to 70 minutes and ones of the second speaker to 69 minutes of clean audio. The L2-ARTIC dataset is released as free software. The audio files are wav files with a bit depth of 16 bits and a sampling frequency of 44.1KHz.

Similarly to CMU ARCTIC, it is expected that this dataset's prosodic relevance is higher than VCTK. It is similar to the user's audio files in terms of representation of the speaker because it was recorded with L2 English speakers, but with higher audio quality. Because of this, the Voice Conversion results are expected to be significantly improved in comparison with the results from the users.

4.4 Tests

This section will describe the tests made with the chosen algorithm for the Voice Conversion task. Both training and the conversion are highly computational intensive tasks done using a Graphics Processing Unit (GPU) which was not available locally. All the tests were performed in a remote AWS EC2 instance, made available by ELSA Corp. This instance had a NVidia Tesla K80 GPU, with 11Gb of memory and

CUDA version 11.0.

Table 4.2: Table containing the division of each dataset into train, validation and test sets

Mode	Dataset	Train	Validation	Test
Pre-Training	VCTK	40298	1949	1620
Fine-Tuning	VCTK + ELSA-REF	2183	166	145
Pre-Training	LibriTTS	73798	5594	4878
Fine-Tuning	ELSA-REF + ELSA-USR1	1389	299	396
Fine-Tuning	ELSA-REF + ELSA-USR2	1571	298	225
Fine-Tuning	ELSA-REF + ELSA-USR3	1959	151	131
Fine-Tuning	L2-ARCTIC-NCC	1899	146	126
Fine-Tuning	L2-ARCTIC-HQTV	1941	149	131

4.4.1 Pre-Training with VCTK

The VC algorithm chosen for this task has no pre-trained model. So in order to test its performance it is necessary to chose a dataset, train the model and generate converted audio files for evaluation. For their experiments, the authors used the VCTK corpus for pre-training and two speakers from the CMU ARCTIC, namely one female speaker, ”slt”, and one male speaker, ”rms”, to fine-tune the model. These same datasets are suggested for testing purposes on the github repository that contains the code, so this recommendation was followed in the initial tests.

The first test had the main objective of verifying that the code was stable and the model was training properly. First, some legacy code was removed from the files cloned from the repository and some minor bugs were fixed. It was also necessary to verify if the code written to perform data pre-processing produced the expected files, like the mel-spectra and the phoneme files. The utterances from the speakers were shuffled and split into train, test and validation sets. In order to prevent out-of-memory errors on the GPU, sentences longer than 7.5s were removed. According to the authors, excluding long utterances should not affect the convergence of the algorithm, as long as the size of these sets is not reduced significantly. The resulting sets are displayed in table 4.2

The algorithm trained for a total of 82h and stopped after 90 thousand iterations. The alignment graphs indicated that the algorithm converged successfully, even though it stopped training earlier than expected due to an error. This error was related to the lack of space in the ssd drive where the checkpoints were being stored and did not impact the training until that point. The algorithm trained for 15 thousand more iterations after some space was cleared, but there were no visible improvements in the model. Using the Universal vocoder to synthesize the audio files from the Mel-Spectrogram files, it was possible to listen to the final result of the pre-training.

The quality of the synthesized audio through the Universal Vocoder was compared with one audio file from the target speaker copy synthesized with this same Vocoder and the results were very similar. This was done because Universal Vocoder reduces the overall sound quality of the audio, so comparing

two audio files produced by the same vocoder makes the comparison fair and helps remove the effect from the waveform synthesizer on the results. The converted audio was highly intelligible and the voice of the target was easily recognizable, as expected.

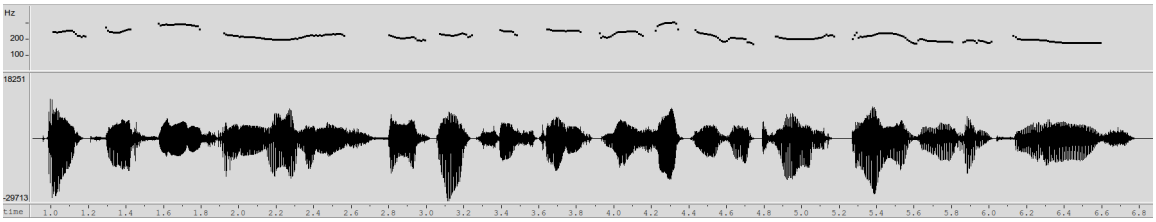
4.4.2 Fine-tuning with VCTK and ELSA-REF

The next step is fine-tuning the model for a pair of speakers. The objective of the VC task is to take the reference utterances from ELSA's voice artist and generate them with the voice of a user, so these utterances needed to be included in the speaker pair used in the fine-tuning process. The speaker p360 from VCTK dataset was chosen to pair with ELSA's speech artist, taking the place of the user. This speaker has around 22 minutes of speech across 424 files. The size of train, validation and tests set can be seen in 4.2.

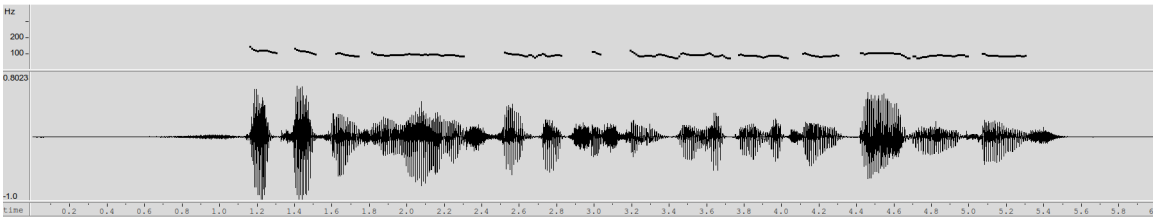
The fine-tuning ran for 15h, reached 50 epochs and stopped after a checkpoint on 18 thousand iterations. Using this model, 50 utterances from ELSA-REF were converted to the voice of p360 speaker. The pretrained model of the Universal Vocoder was used again to generate waveforms from the resulting mel-spectrograms. As expected, the audio quality was highly intelligible and natural, and the voice of the target speaker was very recognizable. However, none of the prosodic features from the source speaker were kept. The resulting audio file had all the prosodic traits from the target speaker, thus it was not usable for a context of prosody training.

To provide a graphical visualization of this issue, the pitch contour of both utterances, the source utterance from ELSA's speech artist and its converted counterpart to the voice of speaker p360 of VCTK corpus, are shown in image 4.3. The first pitch contour was estimated using wavesurfer, which could not detect the pitch of the second utterance. The pitch of the converted utterance was estimated using WORLD's F0 estimator, Harvest, which provided the result seen below. The converted utterance has a faster speaking rate, uttering the sentence in 4.5 seconds, while the source takes 6 seconds. It is also clear that the pitch contour of the converted utterance is almost flat, which contrasts with the source's high variance pitch.

With this test it was possible to understand that the model provided very good results in the Voice Conversion task. The fact that it also converted the prosodic features, which can be seen as a characteristic of the speech that is connected to the speaker, can be seen as a success on a simple VC task. But in this case, losing the prosodic features from the source is a set back that needs to be addressed. The VCTK dataset contains utterances mainly from a newspaper, which are predominantly declarative and plain, without significant pitch variations as a dialogue would have for example. One interesting thought is that the lack of prosodic variance of this dataset, with which the pre-training and fine-tuning was done, may be responsible for a bias on the algorithm towards generating equally neutral utterances. To test this, a next test was made with different datasets.



(a) Utterance 0838 from ELSA-REF



(b) Converted utterance 0838 to the voice of p360

Figure 4.3: F0 estimation and waveforms of the source utterance 0838 and the converted utterance to the voice of VCTK speaker p360. Both waveforms are in the same temporal scale.

4.4.3 Pre-training with LibriTTS

On this test, a new model was pre-trained with a different dataset. The previously pre-trained model, on section 4.4.1, was trained with the VCTK dataset, which does not introduce enough prosodic variation on the training phase of the model, and leads to neutral and plain converted utterances, with low prosodic variance. Also, the model wasn't pre-trained with speech from ELSA's speech artist, and since the Voice Conversion will always have this speaker as source, it makes sense to include it in the initial training, and not only on the fine-tuning.

The resulting dataset used to pre-train the model on this test is LibriTTS (as described in 3.3) augmented with ELSA-REF files. Similarly to the initial pre-training, all the sentences longer than 7.5 seconds will be removed. The remaining utterances are shuffled and split into the three sets shown on table 4.2.

Due to the higher number of utterances on the dataset, the pre-training took longer, totaling 134h of clock time and reaching 117k iterations, in the 50th epoch. At this point, the training was stopped because the alignment graphs indicated that the model collapsed. Looking back to previous checkpoints, it seemed that between 90k and 98k was where the alignment graphs seemed better, so all the checkpoints in between (at each 500 iterations) were used to generate mel-spectrograms of the converted audio. These were then synthesized into waveforms using the pre-trained Universal Vocoder and the results were compared.

The best results were obtained for the checkpoint at 95k iterations, corresponding to 41 epochs and 105h of training. Again, at this point, the converted audio files were compared with the original audio files from LibriTTS synthesized through Universal Vocoder. The converted audio files were once again highly

intelligible and the voice of the target was easily recognizable. At this stage, it was still not clear whether the prosodic features from the source were kept or not, and fine-tuning the model for two speakers was necessary.

4.4.4 Fine-tuning with ELSA-REF and ELSA-USR1

This was the first test done with audio from real users, using one of the datasets created for this purpose. The ideal case would be having a user do the assessment test, which is usually the first set of audio files a user records when the app is installed, and have the Voice Conversion available right away, allowing to have a reference in his/her own voice immediately from the beginning. The assessment test is composed by 13 long sentences, and its purpose is to evaluate the user's English pronunciation at the start of the usage, recommending exercises to start working on his/her weaknesses first.

The audio files in the ELSA-USR1 dataset were partitioned into segments because they were too long to be used in this algorithm. The maximum length for this fine-tuning, to avoid out-of-memory errors, was set at 8 seconds, which result in 23 audio files instead of the initial 13. The pronunciation of the user is very poor and the recording quality is low, containing background noise and small distortions in the voice. All of these conditions summed up make this test an extreme case. Test and validation sets each contained 3 audio files from the user, and the remaining files were assigned to the train set. Their size can be seen in table 4.2.

The model quickly collapsed and it was not possible to get any useful result from it. The audio files that were generated contained only silence and/or noises. This result was expected since the number of audio files was very low and the audio quality was poor, but it was important to perform this test and incrementally improve the quality of the dataset to determine the limits of this algorithm.

4.4.5 Fine-tuning with ELSA-REF and ELSA-USR2

In this test, the model was fine-tuned for the speaker pair ELSA-REF and ELSA-USR2. This time, the audio files from the user (ELSA-USR2) did not require any cropping, because all were under the 8 second mark. The audio quality of this dataset contained minimal background noise and the voice was not degraded in any way. The speaker's pronunciation was excellent and there was only one repetition in the last sentence, so this dataset can be classified as clean speech. In terms of duration, it is shorter than for ELSA-USR1. Since the sentences are exactly the same, the shorter duration can be attributed to the higher speech rate that characterizes more fluent L2 speakers.

This time, the number of files of the test set was reduced, increasing the size of the training set. Both validation and test sets included 2 audio files of the user and the remaining ones were assigned to the train set, which explains the difference of 1 file on the validation set. Their sizes can be seen in table

4.2.

The model did not collapse, and it trained for 10 hours, reaching over 12k iterations after 50 epochs. The audio files were generated with the converted mel-spectograms using the pre-trained Universal Vocoder. The majority of the generated audio files had fluent and intelligible speech, but the perceived speaker identity didn't change. The utterances still sounded mostly like the source speaker, which was ELSA's speech artist, and contained only small portions (words or sometimes only individual phones) that sounded similar to the target speaker. This means that overall, the model failed to perform the voice conversion task, and produced utterances with high pitch fluctuations and with no usability on the context of prosody training. Also, a small portion of the files included included multiple repetitions of phones and/or entire segments, insertion of noise and utterances containing only part of the sentence.

This test seems to indicate that the dataset of utterances from the assessment test is not enough to generate a meaningful training and validation set. Even with high quality and good pronunciation, it was not possible to get a successful conversion. But this time, even though it achieved poor results, the model did not collapse during training and it was possible to generate intelligible utterances. This indicates that the quality of the audio may have a decisive impact on whether the algorithm will converge or not.

4.4.6 Fine-tuning with ELSA-REF and ELSA-USR3

One of the factors that led to poor results on the previous test was the small number of utterances from the user included on the training. To prevent this, a new dataset was brought in as the second speaker in the fine-tuning pair, ELSA-USR3.

This dataset is larger in size, amounting to 170 utterances from a real user and a little under 10 minutes of speech. Since these audio files were in a pre-made dataset given by ELSA's speech team, it was not possible to determine what was the overall nativeness score of the user at the time of recording. Nevertheless, by listening to the audio files it is possible to informally rate the speaker's pronunciation as average/poor. Also, the recording conditions change significantly between practice sessions and some of the audio files seem like they do not belong to the same speaker, even though they are recorded by the user. This is common among ELSA users, and it is one of the reasons why it is difficult to get a clean single speaker dataset to test the VC algorithm. Since the access to the full database containing user's audios is limited, it was not possible to create other datasets with verified audios that would remove these hindrances, but possible solutions for this will be discussed further in section 5.2.2. The size of the train, validation and test sets is shown in table 4.2.

The algorithm ran for 37h, reaching 17k iterations after 50 epochs. During the training, it was possible to observe that the alignment graphs had no changes throughout the whole training, but the training was not interrupted. After generating the audio files, it was clear that the model had trouble converging. The

audio files were very long, with over 20 seconds, and had only silence and noise similar to speech with the target's voice, but without uttering any word. This noise seemed like specific phones, mostly vowels, elongated and repeated without any meaningful order. It is now clear that the audio quality has a very high influence on the outcome, higher than expected initially, and may indicate that using the available audio files from real users to train the model will not give any usable result for speakers with these characteristics. Another division of the audio files was attempted, specifically with a bigger validation set and smaller train set, but the model collapsed and the results were worse. The generated audio files had only silence or buzzing and hissing noises.

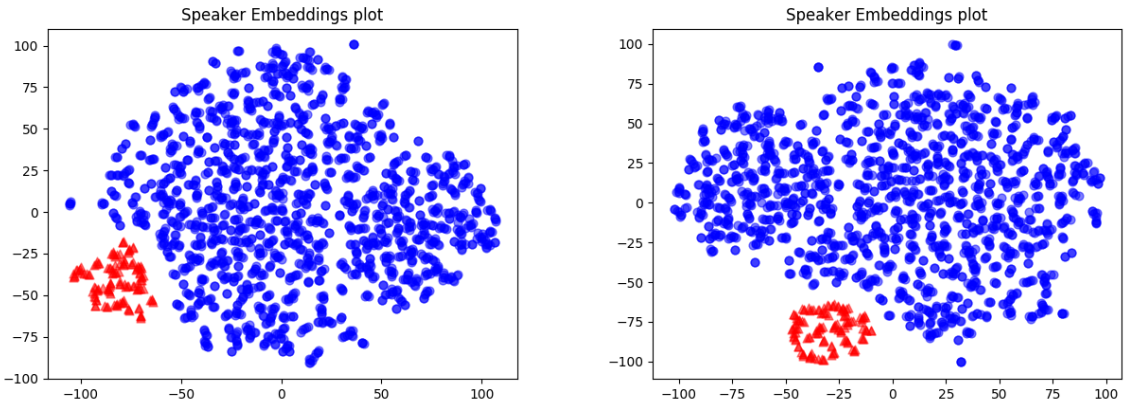
4.4.7 Fine-tuning with ELSA-REF and L2-ARCTIC-NCC

Due to limitations in time, resources and restrictions in the access to users' audio files, it was not possible to generate a dataset that would guarantee audio files from a single speaker, with low noise and no distortion. So in order to explore the potential of this VC method, an alternative dataset was used. The L2-ARTCIC was chosen, and two speakers were selected to take the user's place on the fine-tuning pairs. This dataset was publicly available and fitted the conditions stated above, while keeping enough similarity with the user's audios, more specifically, with L2 English speakers without excellent pronunciation.

The fist attempt was done with 100 audio files from L2-ARCTIC-NCC, resulting in 6 minutes of speech. Even though the dataset contains almost 4 minutes less than the previous attempt, the recordings were made in a studio environment, which should improve the overall results. The size of the sets used for training, validation and test is shown in table 4.2.

The algorithm reached the 50 epochs at 16400 iterations after running for slightly under 12h, with the alignment graphics indicating that it converged. The wavefiles were generated from the converted mel-spectograms using the Universal Vocoder, and in this test both conversions were made, from ELSA-REF to L2-ARCTIC-NCC and the opposite. All the utterances were complete, without long silences, repetitions or unfinished sentences. The speech was intelligible but it was not natural. The converted voice had some resemblance with the target speaker, but with an added creakiness that made it sound unnatural. To try to determine the origin of the issue, the speaker embedding was plotted for all the utterances used for the fine-tuning process. It is possible to see in figure 4.4 a separation between both speakers, but this separation may not be so evident in the context of all pre-training speakers. It seemed that the model simply did not have enough data to achieve total convergence and a good result. To avoid this issue, the number of utterances from the speaker that will take the user's place will be increased.

Although the conversion was not yet achieved successfully, this experiment allowed us to verify the behaviour of the method in the presence of target speakers with poor English speaking skills, which includes not only poor prosody but also mispronunciation errors, namely substitutions, deletions, and



(a) Blue - ELSA-REF, Red - L2-ARCTIC-NCC

(b) Blue - ELSA-REF, Red - L2-ARCTIC-GQTV

Figure 4.4: Plot of the speaker embeddings for the utterances used in fine-tuning. The plot (a) refers to test 4.4.7 and the plot (b) refers to 4.4.8. There is a separation between speakers in both embedding plots.

additions. Disfluencies such as repetitions, repairs, false starts and silent pauses are not present in the used dataset because the speaker was allowed to repeat the utterance in these cases. The audio files generated through this VC model contained barely any of these mispronunciation errors. It is not possible to conclude anything in this case because the limited number of utterances from the target speaker also harmed the naturalness of the converted voice, but further discussion will be held on the next test.

4.4.8 Fine-tuning with ELSA-REF and L2-ARCTIC-HQTV

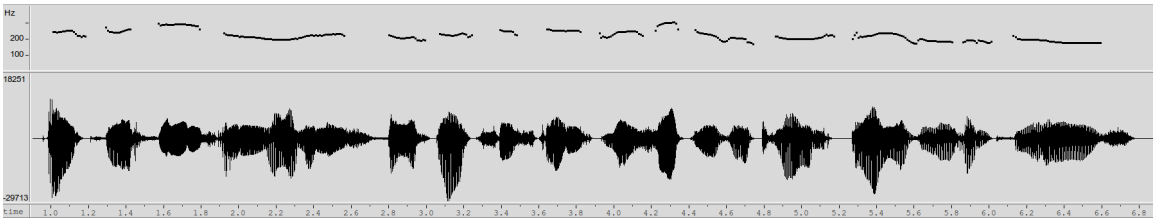
This test constitutes the second attempt using L2-ARCTIC datasets. This time, 150 audios resulting in 9 minutes of speech were used from L2-ARCTIC-HQTV to replace the user in the speaker pair for which the model was fine-tuned. The size of the train, validation and test sets is shown in table 4.2.

The training ran for 10 hours, reaching 16800 iterations. One outcome from this test is noticeable right away. Excluding the tests presented on 4.4.4 and 4.4.5, that were done with only 13 sentences, this was the fastest fine-tunning of the model. The number of files is similar to the test 4.4.6 but it took less than a third of the time to reach the 50 epochs. On the other hand, with 50 more utterances than the test done on 4.4.7, it took 2h less to finish, and the results are significantly better. This raises the question of how many files is the sweet spot between training time and quality conversions, and if it is possible to predict this number using audio files from users that have varying nativeness scores and recording conditions. The embedding of the utterances used in this fine-tuning were plotted and can be seen in figure 4.4, showing a clear separation between speakers.

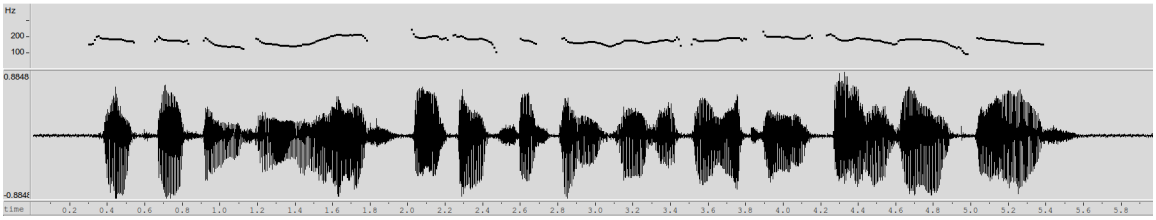
In terms of the Voice Conversion task, the results are the best between all the fine-tuning processes using the model pre-trained with LibriTTS, and comparable with the initial fine-tuning performed on

4.4.2. The converted audio files were generated with the Universal Vocoder from the predicted mel-spectograms and compared with copy synthesized audios from L2-ARCTIC-HQTV. The voice is very similar to the target and the speech is very natural, even though the fine-tuning was made with under 10 minute of speech from the source speaker.

Both pitch contours from the source utterance and the converted utterance are shown in figure 4.5. As in figure 4.3, the pitch contour of the converted was estimated with Harvest, since Wavesurfer was not able to provide any results. Even though the pitch contour is not an exact copy, which was not expected, it is possible to see the influence of the source's prosody in the converted utterance. The variance is much higher than in the previous test done on 4.4.2 and also higher than the utterances from the target speaker.



(a) Utterance 0838 from ELSA-REF

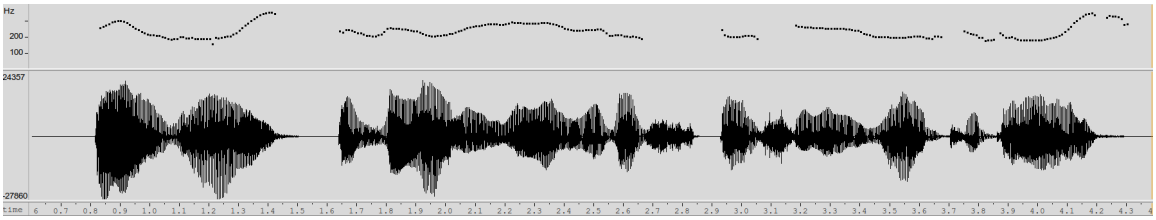


(b) Converted utterance 0838 to the voice of HQTV

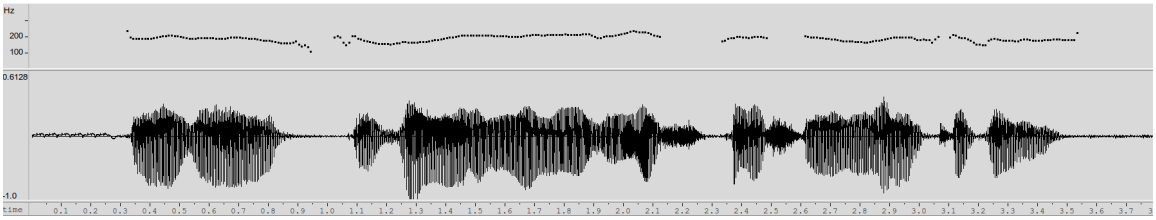
Figure 4.5: F0 estimation and waveforms of the source utterance 0838 and the converted utterance to the voice of L2-ARCTIC speaker HQTV. Both waveforms are in the same temporal scale

The pitch contour from another converted utterance can be seen in figure 4.6, together with the pitch contour from the same utterance converted with the model from test 4.4.2, for an easy comparison. On the top of figure 4.6, the pitch contour of the original audio sample containing the sentence uttered by ELSA's speech artist is presented. It is possible to see a high variation of the F0 value. In the middle, the converted utterance in the voice of the speaker HQTV, from L2-ARCTIC dataset, has more modest pitch variations, but similar temporal markers and speech rate. In the bottom, it is possible to see the waveform produced in test 4.4.2 of the utterance converted to the voice of the speaker p360 from VCTK. The speech rate is higher, the pitch is almost flat and the temporal alignments differ greatly from the source.

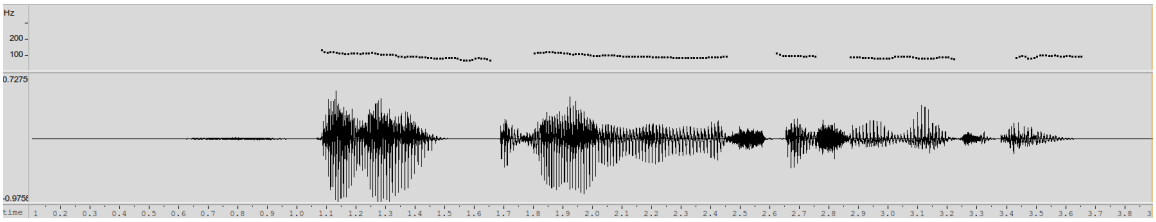
With this test, it becomes clearer that with these conditions, the algorithm mostly retains the pronunciation, and even the accent, of the source speaker. There are some phones that hint a slight mispronunciation from the target speaker, more specifically deletions, but this mostly appears to be cases



(a) Utterance 1970 from ELSA-REF



(b) Converted utterance 1970 to the voice of L2-Arctic speaker HQTV, from test 4.4.8



(c) Converted utterance 1970 to the voice of VCTK speaker p360, from test 4.4.2

Figure 4.6: F0 estimation and waveforms of three audio samples. All waveforms are in the same temporal scale.

where the phones are too short rather than not present. This is a very interesting application for English learning exercises, such as the ones found on ELSA Speak mobile app. It may allow the speaker to hear himself/herself with an almost native pronunciation, either as a reference for the exercise, or as a motivational feature.

With the results obtained in this test, it is possible to move on to the evaluation of the algorithm.

4.5 Evaluation

Similarly to what was done on the previous chapter, the results from the Voice Conversion task were evaluated. Despite the fact that this was an initial exploratory approach to the application of Voice Conversion to language learning, several audio files were produced with enough quality to make this evaluation. An objective evaluation is not possible, since the audios from the source speaker, L2-ARCTIC-HQTV, do not contain the same sentences as ELSA-REF. For this reason it is not possible to run them through the proposed objective evaluation in the previous chapter or any of the objective evaluations mention in chapter 2. This means that only subjective testing will be made, more precisely MOS and AB testing. The survey was responded by 40 different subjects.

4.5.1 Mean Opinion Score

Three scores are calculated and presented in the table below. These 3 scores correspond to three different questions. The first two questions were made after presenting the subjects with three audio samples each:

A - Audio sample from ELSA-REF, corresponding to the source speaker

B - Audio sample from L2-ARCTIC-HQTV, corresponding to the target speaker

C - Converted sample, resulting from the conversion of the source sample (A) to the voice of the target speaker produced during test 4.4.8

Since the datasets are not parallel, there are no utterances from the target speaker with the same sentence as the source speaker, which means that the source and converted samples have the same textual content and both are different from the target sample.

The subjects were then asked to respond to the following questions, rating from 1 to 5:

1. Would you say that sample C imitates the duration and intonation pattern of sample A?
2. On a scale of 1 to 5, would you say that sample C retains the voice of sample B?

The third question was made to the subjects after presenting them with 3 different converted utterances produced with the test 4.4.8:

3. How native do these samples sound when compared to an American English Native Speaker?

Table 4.3: Mean Opinion Score and 95% confidence interval of the responses to 3 evaluations of the converted audio according to 3 different metrics

Metric	MOS
Retention of duration and intonation patterns from source	3.31 ± 0.15
Voice similarity to target speaker	3.46 ± 0.16
Nativeness when comparing to American English Accent	3.45 ± 0.25

It is important to mention again that this experiment was made to explore the potential of similar methods, and not to provide a productized version. Nevertheless, the results will be discussed as for their applicability in a language learning context.

In terms of the retention of the duration and intonation patterns from the source, since this is a fixed reference (as opposed to the gradual reference presented on the Pitch Transplant algorithm), it has to capture perfectly the prosodic features from the source. Otherwise, it would be possible that the reference given to a student would not be correct, and that he/she would be learning these features in a wrong way. Due to the human tendency to avoid perfect ratings, a score between 4.2 and 4.5 would be considered enough to use as a reference. This would be necessarily accompanied by a similar rating in

terms of nativeness, due to the fact that the source is a native speaker. An interesting test on a similar study would be questioning the subjects about the nativeness of the utterances from the source speaker, the target speaker and the converted utterances. Then it would be easier to determine the improvement from the target utterances to the converted ones, in relation to the utterances from the source.

4.5.2 AB test

The subjects were presented with 3 pairs of audio samples. Each pair contained one sample taken from L2-ARCTIC-HQTV dataset and one converted sample produced in the test 4.4.8. They were instructed to listen to each pair and chose which utterance had better English pronunciation. The order of the samples was selected at random.

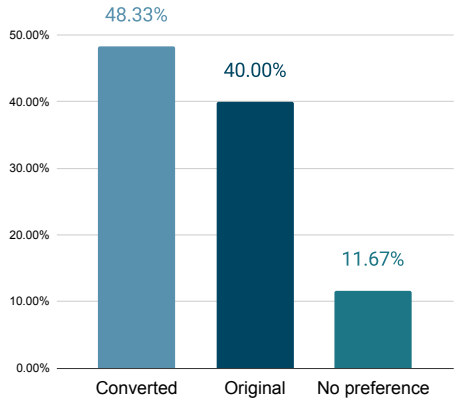


Figure 4.7: AB tests results for evaluating pronunciation

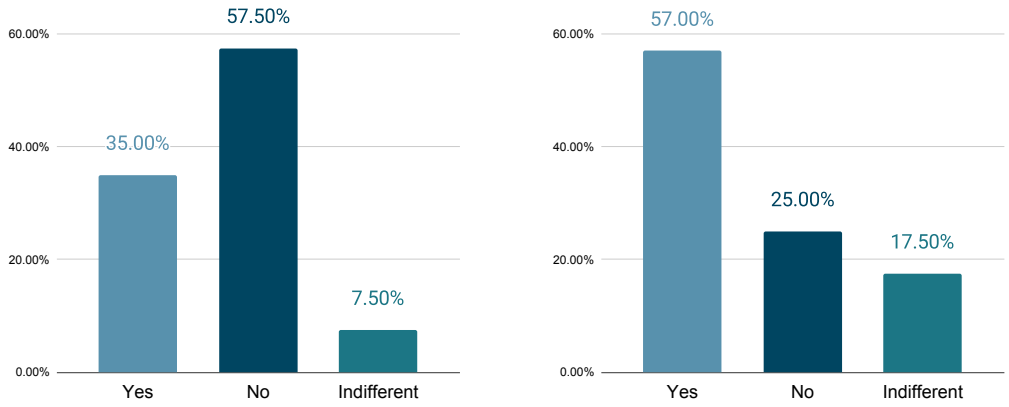
The results reveal a very slight tendency towards classifying the converted utterances as the ones with better pronunciation. Two different factors may have swayed the opinions to either side. On one hand, the audio quality of the converted samples is notably worse, which may have led less trained subjects to chose the audios from the target speaker. On the other hand, the difficulty to read the different sentences may have tilted the results the other way. The Flesh-Kincaid reading-ease score [99] was calculated for all utterances. The scores were between 80 and 100 in every sentence (5th and 6th grade level of difficulty) except for one sentence from the from L2-ARCTIC-HQTV dataset, which scored 42.4 (college level of difficulty). Because only 3 pairs were evaluated, this one sentence may have had a strong weight on the overall result.

4.5.3 Yes/No questions

Two extra questions were made on this survey. The first question was made in relation to the same audio files that were used in the 3rd MOS, which were converted utterances produced with the test 4.4.8. The

second question was general and did not involve any audio sample. Both are presented below, as well as the results of the answers. The indifference option was also given.

1. Would you consider that these samples have enough sound quality to be used as a reference in an English language learning exercise?
2. Would you be comfortable if, in a language learning context, you would listen to your voice saying a sentence you never said before?



(a) Answers to question 1.

(b) Answers to question 2.

Figure 4.8: Answers to the yes or no questions presented to the subjects on the online survey

Due to the experimental character of this Voice Conversion approach, it is not surprising that the majority of the subjects responded that the audio generated did not have the required quality to be presented as a reference on a language learning exercise. But it is important to notice that a smaller percentage disagrees, which may indicate that a huge improvement is not required to achieve this goal. As to the second question, only a quarter of the subjects would not be comfortable to listen to their own voices as reference on a language learning exercise, so designing a CAPT system as proposed in this work would likely be well received, at least as an added option.

5

Conclusion

Contents

5.1 Conclusion

The objective of this thesis was to apply two different techniques to prosody training. It was done in the context of the exercises available in the ELSA Speak app.

The first approach was taken through signal analysis and manipulation. The output audio was produced by manipulating the input audio file, which comes with its advantages and disadvantages. Since there is no pre-training or dataset required, it can be applied instantly to first time users and it generates results in real time. But the output is generated by manipulating the input audio, so it will maintain its pronunciation mistakes and audio quality. Also, there is a limitation to how much the audio can be manipulated without losing the speech naturalness, allowing only for small tweaks.

The second approach was made using a deep learning Voice Conversion algorithm, which is a more current technology that still has huge potential to improve. It was not intended as a final result, but as a preliminary exploration of the applicability of this technology in pronunciation training. Several tests were made with different datasets to understand the limitations of the chosen algorithm. They revealed that this model maintains some of the prosodic features from the source, but does not copy the exact speaking style. Also, it is a computationally heavy process and does not cope well with noisy recording environments. Nevertheless, its results were evaluated and they reveal that this is a capable alternative and should be developed further.

Even though the first approach presents results faster, it is fairly limited. The second method not only produces more natural speech, but once the model is trained, it may convert virtually any sentence without any further input from the user. Also, every year new VC algorithms are developed, and it is only a matter of time until an algorithm such as this can produce near human speech naturalness with very high quality audio.

On another note, the surveys that were handed out to evaluate both methods contained extra questions. The answers to these questions revealed that the majority of the responders were comfortable in having their voices manipulated and used in a language learning context. This encourages further the application of such systems in English learning apps such as ELSA Speak, or even as complementary work for English language courses.

5.2 Future Work

5.2.1 Pitch Transplant

In chapter 3, the Pitch transplant algorithm was proposed. All the tests performed with this algorithm, and the audio files it produced were made using audio files from real users. The only control over the audio quality was made with the SNR filter, which only applied the algorithm when the audio files had

what was considered workable noise. But it is obvious, from the datasets that were built, that the quality of these audios is very low. It is difficult to make any manipulation on an already degraded audio source, and reducing noise would definitely improve results. A method similar to the one implemented in the Audacity software, which takes a portion of audio containing only noise and attempts to remove it from the whole audio could provide a significant improvement on audio files with continuous noises. This could be applied by first recording a small sample, for example one second, before each practice session, and then using this sample to recognize the noise pattern and remove it on the user's utterances. Another option would be implementing a noise-detection method in the app which would only enable the Pitch Transplant feedback when the background noise would be under a given threshold.

In the Pitch Transplant method, DTW performs the temporal alignment between the user's and the reference's utterance. The allowed jumps between frames that the algorithm does were chosen in a conservative way, to try to avoid unnatural speech. But these restrictions influence greatly the behaviour of the algorithm, and it would be wise to adjust them with the help of a linguist, so that it would be guaranteed that the pitch and duration markers would be corrected.

Also, the algorithm's strength comes from the fact that it allows for a gradually evolving reference. So, in order to really evaluate it, it is necessary to use it in a language learning context. A script was developed that records the audio, sends it for processing and returns the score and the transplanted reference for another attempt. But this algorithm includes protected data from ELSA Speak and could not be sent for remote testing, meaning that the tests would need to be made in person, which was not possible due to the restrictions imposed by the COVID-19 pandemic. Also, in order to evaluate the effects of using such technique in language learning would require continuous testing for several months, and due to time limitations this evaluation could not made. But changing the implementation and allowing for remote testing could allow for this test to be performed, which would be an interesting work in the future.

5.2.2 Voice Conversion

The Voice Conversion approach presented on Chapter 4 is, as previously mentioned, a preliminary exploration of the application of this technology to prosody training. So, even though some results were obtained and an evaluation was made, these results could still be significantly improved.

In all tests where the model was fine-tuned with audio files from real users, it either collapsed or failed to converge, most likely due to the low quality of the users' audio files. But the environments where the users use the app are not constant, and it is possible that some of the audio files have higher audio quality and less noise than others. The use of an SNR estimator, such as [84], could be an alternative to select only the audio files with less noise and run the fine-tune with them.

In fact, the model never saw lower quality recordings during the pre-training, which was done with

a clean dataset of studio quality audio. It is possible that if the same dataset used in pre-training was augmented with users' lesser quality audios, it would produce better results after being fine-tuned. On an extreme case, would the model converge if the whole training dataset was made exclusively of user's audios? If it did, it would probably be easier to fine-tune the model and obtain better results. In order to do this it would first be necessary to guarantee the audios from each user contain recordings from one single speaker, which would be possible using a speaker recognition algorithm.

As mentioned before, one of the side-products that was seen in the results, is the fact that the model removes the majority of the pronunciation mistakes seen in the utterances of the target speaker. It would be interesting to perform further testing on the application of this model to pronunciation training focused on the segmental aspects. It could be applied to an earlier phase of language learning as a motivational goal for new learners.

After the work on this thesis was finalized, during the INTERSPEECH 2020 conference, new VC methods were proposed that could possibly produce better results than the chosen algorithm. One example is the algorithm proposed in [100]. The audio samples shared by the authors seem to keep the prosody features from the source. Another paper focuses its work in developing a technique to transfer the source speaking style in a non parallel voice conversion task [101]. Its performance is compared with two baseline models, one of which is the selected model in chapter 4. The evaluation showed that the model retains the source speaking style better than both baselines. Even though the code is not publicly available, it would be interesting to test the performance of such an algorithm in the context of prosody training.

Similarly to the future work suggested for Pitch Transplant, it would be interesting to productize this model and apply it in a real environment where the progress of the students could be monitored. The study would consist of having a group A doing prosody training with the reference sentences in their own converted voices, and group B practicing with the native speaker's audio. After a period of time, the progress of each group of students would be analyzed and compared.

Bibliography

[1] H. Dudley, “The carrier nature of speech,” vol. 19, no. 4, p. 509, Oct. 1940.

[2] G. Kouroupetroglou and G. Chrysochoidis, “Formant tuning in byzantine chanting,” 07 2014.

[3] J. Valin, “Lpcnet: Dsp-boosted neural speech synthesis,” 2018, online. Accessed November 29, 2020. [Online]. Available: https://jmvalin.ca/demo/lpcnet/

[4] M. Morise, F. Yokomori, and K. Ozawa, “World: A vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE Transactions on Information and Systems*, vol. E99D, no. 7, pp. 1877–1884, Jul. 2016.

[5] M. Morise, “Harvest: A high-performance fundamental frequency estimator from speech signals,” *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2017-August, pp. 2321–2325, Jan. 2017, 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017 ; Conference date: 20-08-2017 Through 24-08-2017.

[6] ——, “Error evaluation of an f0-adaptive spectral envelope estimator in robustness against the additive noise and f0 error,” *IEICE Transactions on Information and Systems*, vol. E98D, no. 7, pp. 1405–1408, Jul. 2015.

[7] Z. Zhang, R. Tavenard, A. Bailly, X. Tang, P. Tang, and T. Corpetti, “Dynamic Time Warping under limited warping path length,” *Information Sciences*, 2017.

[8] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, “Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, no. 1, pp. 540–552, 2020.

[9] D. M. Eberhard, G. Simons, and C. Fennig, “Ethnologue: Languages of the World. Twenty-second edition,” 2019.

[10] D. Crystal, “Two thousand million?” *English Today*, vol. 24, pp. 3 – 6, 03 2008.

[11] ——, *The Cambridge Encyclopedia of the English Language*, 3rd ed. Cambridge University Press, 2018.

[12] A. Pennycook, *The Cultural Politics of English as an International Language*, ser. Language in social life series. Longman, 1994. [Online]. Available: https://books.google.pt/books?id=5o9mAAAAMAAJ

[13] S. Scott, David and Beadle, "Improving the effectiveness of language learning: CLIL and computer assisted language learning," *Educatiom and Training*, 2014.

[14] N. S. McGarr and M. J. Osberger, "Pitch deviancy and intelligibility of deaf speech," *Journal of Communication Disorders*, vol. 11, no. 2-3, pp. 237–247, 1978.

[15] M. S. De Bodt, M. E. Hernández-Díaz Huici, and P. H. Van De Heyning, "Intelligibility as a linear combination of dimensions in dysarthric speech," *Journal of Communication Disorders*, vol. 35, no. 3, pp. 283–292, 2002.

[16] M. Klopfenstein, "Interaction between prosody and intelligibility," *International Journal of Speech-Language Pathology*, vol. 11, no. 4, pp. 326–331, 2009.

[17] D. R. McCloy, "Prosody, intelligibility and familiarity in speech perception," 2013. [Online]. Available: https://digital.lib.washington.edu:443/researchworks/handle/1773/23472

[18] J. S. Laures and G. Weismer, "The effects of a flattened fundamental frequency on intelligibility at the sentence level," *Journal of Speech, Language, and Hearing Research*, vol. 42, no. 5, pp. 1148–1156, 1999.

[19] A.-M. Öster, "The effects of prosodic and segmental deviations on intelligibility of deaf speech," *STL-QPSR*, 1990.

[20] L. Kimppa, T. Kujala, A. Leminen, M. Vainio, and Y. Shtyrov, "Rapid and automatic speech-specific learning mechanism in human neocortex," *NeuroImage*, 2015.

[21] M. P. Bissiri, H. R. Pfitzinger, and H. G. Tillmann, "Lexical stress training of German compounds for Italian speakers by means of resynthesis and emphasis," *11th Australasian International Conference on Speech Science & Technology*, pp. 24–29, 2006. [Online]. Available: http://www.assta.org/sst/2006/sst2006-25.pdf

[22] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Prosodic and segmental factors in foreign-accent conversion," 03 2012.

[23] S. Zhao, S. Koh, I. Soon, and K. Luke, "Feedback utterances for computer-adied language learning using accent reduction and voice conversion method," 05 2013.

[24] K. Probst, Y. Ke, and M. Eskenazi, “Enhancing foreign language tutors – In search of the golden speaker,” *Speech Communication*, vol. 37, no. 3, pp. 161–173, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639301000097

[25] M. Levy, J. Tainton, J. Lavis, C. W. Australia, A. E. C. A. N. C. W. Project, and A. E. C. A. N. C. W. P. Q. C. W. Committee, *Computer-Assisted Language Learning: Context and Conceptualization*, ser. Clarendon paperbacks. Clarendon Press, 1997. [Online]. Available: https://books.google.pt/books?id=RRGgrjteVjUC

[26] M. Warschauer, “Computer Assisted Language Learning: an Introduction,” *Multimedia language teaching*, 1996.

[27] C. Agarwal and P. Chakraborty, “A review of tools and techniques for computer aided pronunciation training (CAPT) in English,” *Education and Information Technologies*, 2019.

[28] D. Kalikow and J. Swets, “Experiments with computer-controlled displays in second-language learning,” *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 1, pp. 23–28, 1972.

[29] M. Jilka and G. Möhler, “Intonational foreign accent: speech technology and foreign language teaching,” *Proc. of the ESCA Workshop on Speech Technology in Language Learning*, no. L, pp. 115–118, 1998. [Online]. Available: http://ifla.uni-stuttgart.de/institut/mitarbeiter/jilka/papers/STiLLproc.pdf

[30] M. Tang, C. Wang, and S. Seneff, “Voice transformations: From speech synthesis to mammalian vocalizations,” *EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology*, pp. 353–356, 2001.

[31] M. Leddy and G. Gill, “Improving the communication of people with down syndrome,” 1999.

[32] D. Crystal, *A Dictionary of Linguistics and Phonetics*, ser. The Language Library. Wiley, 2003, pp. 408–409. [Online]. Available: https://books.google.pt/books?id=3JtAOHLtlHoC

[33] E. P. Altenberg, “The perception of word boundaries in a second language,” *Second Language Research*, vol. 21, no. 4, pp. 325–358, Oct. 2005. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00572082

[34] H. Baumotte and G. Dogil, “Coarticulation in non-native speakers of English: /əəlV/-sequences in no-proficient vs. proficient learners,” in *ExLing 2008: Proceedings of 2nd Tutorial and Research Workshop on Experimental Linguistics,*, 2019.

[35] M. Klopfenstein, “Interaction between prosody and intelligibility,” *International Journal of Speech-Language Pathology*, vol. 11, no. 4, pp. 326–331, 2009. [Online]. Available: https://doi.org/10.1080/17549500903003094

[36] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, “Foreign accent conversion in computer assisted pronunciation training,” *Speech Communication*, vol. 51, no. 10, pp. 920–932, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2008.11.004

[37] D. Kewley-Port and C. Watson, “Computer assisted speech training: practical considerations,” in *Applied speech technology*, A. K. Syrdal, R. W. Bennett, and S. L. Greenspan, Eds. CRC Press, 1995, pp. 565–582.

[38] “The vocoder,” *Nature*, 1940.

[39] R. A., “Concatenative speech synthesis: A review,” *International Journal of Computer Applications*, vol. 136, pp. 1–6, 02 2016.

[40] F. Charpentier and M. Stella, “Diphone synthesis using an overlap-add technique for speech waveforms concatenation,” in *ICASSP ’86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, 1986, pp. 2015–2018.

[41] T. Chiba and M. Kajiyama, *The Vowel: Its Nature and Structure*. Tokyo-Kaiseikan, 1941. [Online]. Available: https://books.google.pt/books?id=tpkKAAAAMAAJ

[42] C. Shadle, “The acoustics of fricative consonants,” 1985.

[43] B. J. Kröger and P. Birkholz, “Articulatory Synthesis of Speech and Singing: State of the Art and Suggestions for Future Research,” in *Multimodal Signals: Cognitive and Algorithmic Issues*, A. Esposito, A. Hussain, M. Marinaro, and R. Martone, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 306–319.

[44] ——, “Articulatory Synthesis of Speech and Singing: State of the Art and Suggestions for Future Research,” in *Multimodal Signals: Cognitive and Algorithmic Issues*, A. Esposito, A. Hussain, M. Marinaro, and R. Martone, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 306–319.

[45] S. Maeda, “Compensatory Articulation During Speech: Evidence from the Analysis and Synthesis of Vocal-Tract Shapes Using an Articulatory Model,” in *Speech Production and Speech Modelling*, 1990.

[46] J. Dang and K. Honda, “Construction and control of a physiological articulatory model,” *The Journal of the Acoustical Society of America*, 2004.

[47] “A theoretical model of the pressure field arising from asymmetric intraglottal flows applied to a two-mass model of the vocal folds,” *The Journal of the Acoustical Society of America*, 2011.

[48] K. Ishizaka and J. L. Flanagan, “Synthesis of Voiced Sounds From a Two-Mass Model of the Vocal Cords,” *Bell System Technical Journal*, 1972.

[49] P. Birkholz, D. Jackel, and B. J. Kroger, “Simulation of losses due to turbulence in the time-varying vocal system,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1218–1226, 2007.

[50] S. Maeda, “A digital simulation method of the vocal-tract system,” *Speech Communication*, 1982.

[51] B. Elie and Y. Laprie, “Extension of the single-matrix formulation of the vocal tract: Consideration of bilateral channels and connection of self-oscillating models of the vocal folds with a glottal chink,” *Speech Communication*, 2016.

[52] P. Birkholz, “Modeling Consonant-Vowel Coarticulation for Articulatory Speech Synthesis,” *PLoS ONE*, 2013.

[53] B. J. Kröger, “A gestural production model and its application to reduction in German.” *Phonetica*, 1993.

[54] R. Alexander, T. Sorensen, A. Toutios, and S. Narayanan, “A modular architecture for articulatory synthesis from gestural specification,” *The Journal of the Acoustical Society of America*, 2019.

[55] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” 2016.

[56] R. Prenger, R. Valle, and B. Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” 2018.

[57] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, “Towards achieving robust universal neural vocoding,” 2019.

[58] P. chun Hsu, C. hsuan Wang, A. T. Liu, and H. yi Lee, “Towards robust neural vocoding for speech generation: A survey,” 2020.

[59] J. Laver, *Principles of Phonetics*, ser. Cambridge Textbooks in Linguistics. Cambridge University Press, 1994.

[60] R. F. Kubichek, “Mel-Cepstral distance measure for objective speech quality assessment,” in *IEEE Pac Rim Conf Commun Comput Signal Process*, 1993.

[61] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson Correlation Coefficient,” 2009.

[62] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature,” *Geoscientific Model Development*, 2014.

[63] L. L. Thurstone, “A law of comparative judgment,” *Psychological Review*, 1927.

[64] ITU-R BS.1534-3, “Method for the subjective assessment of intermediate quality level of audio systems,” *International Telecommunication Union*, 2015.

[65] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, “Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2014.

[66] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, “From HMMS to DNNS: Where do the improvements come from?” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016.

[67] T. Merritt, B. Putrycz, A. Nadolski, T. Ye, D. Korzekwa, W. Dolecki, T. Drugman, V. Klimkov, A. Moinet, A. Breen, R. Kuklinski, N. Strom, and R. Barra-Chicote, “Comprehensive Evaluation of Statistical Speech Waveform Synthesis,” in *2018 IEEE Spoken Language Technology Workshop, SLT 2018 - Proceedings*, 2019.

[68] C.-C. Hsu, “Pyworld - a python wrapper of world vocoder,” https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder.

[69] P. L. Tobing, Y.-C. Wu, and T. Toda, “Baseline system of voice conversion challenge 2020 with cyclic variational autoencoder and parallel wavegan,” 2020.

[70] M. Morise and Y. Watanabe, “Sound quality comparison among high-quality vocoders by using re-synthesized speech,” *Acoustical Science and Technology*, vol. 39, no. 3, pp. 263–265, 2018.

[71] M. Morise, “Cheaptrick, a spectral envelope estimator for high-quality speech synthesis,” *Speech Communication*, vol. 67, pp. 1–7, Jan. 2015.

[72] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, “Tandem-straight: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, ser. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Sep. 2008, pp. 3933–3936, 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP ; Conference date: 31-03-2008 Through 04-04-2008.

[73] H. Kawahara and M. Morise, “Technical foundations of tandem-straight, a speech analysis, modification and synthesis framework,” *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 36, no. 5, pp. 713–727, Oct. 2011.

[74] M. Morise, “An attempt to develop a singing synthesizer by collaborative creation,” in *Proceedings of the Stockholm Music Acoustics Conference 2013 (SMAC2013)*, Nov. 2013, pp. 287–293.

[75] ——, “D4c, a band-aperiodicity estimator for high-quality speech synthesis,” *Speech Communication*, vol. 84, pp. 57–65, Nov. 2016.

[76] R. Bellman and R. Kalaba, “On adaptive control processes,” *IRE Transactions on Automatic Control*, 1958.

[77] H. Sakoe and S. Chiba, “Dynamic Programming Algorithm Optimization for Spoken Word Recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1978.

[78] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances,” *Data Mining and Knowledge Discovery*, 2017.

[79] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. USA: Prentice-Hall, Inc., 1993.

[80] C. Magariños, P. Lopez-Otero, L. Docio-Fernandez, E. Rodriguez-Banga, D. Erro, and C. Garcia-Mateo, “Reversible speaker de-identification using pre-trained transformation functions,” *Computer Speech and Language*, 2017.

[81] D. Erro, I. Sainz, E. Navas, and I. Hernaez, “Harmonics plus noise model based vocoder for statistical parametric speech synthesis,” *IEEE Journal on Selected Topics in Signal Processing*, 2014.

[82] H. Traunmüller and A. Eriksson, “The frequency range of the voice fundamental in the speech of male and female adults,” *Department of Linguistics, University of Stockholm*, 1994.

[83] P. Rouanet, “Dynamic time warping python module,” https://github.com/pierre-rouanet/dtw.

[84] C. Kim and R. M. Stern, “Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008.

[85] J. Meade, “Wada snr estimation of speech signals in python,” https://gist.github.com/johnmeade/d8d2c67b87cda95cd253f55c21387e75, 2020.

[86] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 1, p. 7–19, Jan. 2015. [Online]. Available: https://doi.org/10.1109/TASLP.2014.2364452

[87] D. Q. Goldin and P. C. Kanellakis, “On similarity queries for time-series data: Constraint specification and implementation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1995.

[88] S. Katsigiannis, J. Scovell, N. Ramzan, L. Janowski, P. Corriveau, M. A. Saad, and G. Van Wallendael, “Interpreting MOS scores, when can users see a difference? Understanding user experience differences for photo quality,” *Quality and User Experience*, 2018.

[89] D. Griffin and Jae Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[90] J.-X. Zhang, “Non-parallel seq2seq voice conversion,” https://github.com/jxzhanggg/nonparaSeq2seqVC_code/commit/4c03a6be3bc76207b7cf8222c985dc85c7018cde, 2020.

[91] B. McFee, V. Lostanlen, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, J. Mason, D. Ellis, E. Battenberg, S. Seyfarth, R. Yamamoto, K. Choi, viktorandreevichmorozov, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, D. Hereñú, F.-R. Stöter, P. Friesch, A. Weiss, M. Vollrath, and T. Kim, “librosa/librosa: 0.8.0,” Jul. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3955228

[92] M. Bernard, hadware, R. Riad, Isn0gud, and J. Benjumea, “bootphon/phonemizer: phonemizer-2.2,” Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3689438

[93] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, “Tacotron: Towards end-to-end speech synthesis,” 2017.

[94] C. Veaux, J. Yamagishi, and K. MacDonald, “CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit,” *The Centre for Speech Technology Research (CSTR)*, 2016.

[95] J. Kominek and A. Black, “The CMU Arctic speech databases,” *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[96] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, “Libritts: A corpus derived from librispeech for text-to-speech,” *CoRR*, vol. abs/1904.02882, 2019. [Online]. Available: http://arxiv.org/abs/1904.02882

[97] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[98] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, “L2-arctic: A non-native english speech corpus,” in *Proc. Interspeech*, 2018, p. 2783–2787. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1110

[99] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, “Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel,” Naval Technical Training Command Millington TN Research Branch, Tech. Rep., 1975.

[100] S. Ding, G. Zhao, and R. Gutierrez-Osuna, “Improving the speaker identity of non-parallel many-to-many voice conversion with adversarial speaker recognition,” in *INTERSPEECH*, 2020.

[101] S. Liu, Y. Cao, S. Kang, N. Hu, X. Liu, D. Su, D. Yu, and H. Meng, “Transferring source style in non-parallel voice conversion,” 2020.

A

User Datasets

This Appendix includes detailed information about the audio files in each dataset used for the development and test of the Pitch Transplant Algorithm. The audio files in these datasets contain protected data from Elsa Corp. and are therefore omitted in the public version of this text.