UNIVERSIDAD
DE GRANADA

MASTER's THESIS

TELECOMMUNICATION TECHNOLOGIES ENGINEERING

Speech Synthesis from Voice Biosignals Using Machine Learning Techniques

Author
Javier Lobato Martín

Supervisor
José Andrés González López

Superior Technical School of Computer and
Telecommunication Engineering

—

Granada, July, 2024

Síntesis de Voz a partir de Bioseñales de Habla usando Técnicas de Machine Learning

Javier Lobato Martín

Palabras clave: Síntesis de voz, Bioseñales, Unit Selection, PMA, WORLD, Redes Neuronales, Síntesis de voz directa, CCA

Resumen

Existen múltiples enfermedades y traumatismos que pueden causar la pérdida o afectación severa del habla, como ocurre en casos de ictus cerebral, la Esclerosis Lateral Amiotrófica (ELA) o laringectomía. Asimismo, muchas de estas diversas afecciones carecen de cura a día de hoy, haciendo que la pérdida de la capacidad del habla sea irreversible y, en muchos casos, un proceso progresivo e inevitable. Si bien es cierto que existen dispositivos en el mercado diseñados para ayudar a estas personas a paliar los problemas de comunicación que sufren, es habitual que este tipo de soluciones sean lentas y difíciles de usar, lo que impacta de manera significativa la calidad de vida de los afectados.

Este proyecto de investigación forma parte del esfuerzo colectivo enfocado en restaurar la capacidad de comunicación oral a estas personas mediante un punto de vista de aplicación de la tecnología y, en especial, las telecomunicaciones. De esta manera, en este Trabajo de Fin de Máster, se implementan una serie de algoritmos cuyo foco es la síntesis de voz a partir de bioseñales del habla, capturados a partir del movimiento de labios y lengua de los participantes, utilizando una técnica de captura conocida como Articulografía por Imanes Permanentes (PMA).

Para lograr el objetivo del estudio, se proponen múltiples algoritmos de síntesis de voz: Síntesis de Voz Directa, Unit-Selection con CCA, Regresión Lineal por medio de CCA y dos tipos distintos de redes neuronales: Deep Neural Network (DNN) y Gated Recurrent Unit (GRU). Los algoritmos toman la información de una base de datos que contiene bioseñales PMA y voz grabados de forma simultánea en individuos sanos. En ciertos algoritmos se usarán unos parámetros derivados de la voz en vez de la voz sin tratar: los MFCC's (coeficientes cepstrales en escala mel). En función del algoritmo, las señales utilizadas se tratarán en base a pequeñas porciones denominadas unidades. En el estudio se implementan diversos métodos estudiados en la literatura disponible, que se cree pueden dar un buen resultado para la casuística concreta o pueden suponer una mejora con respecto al método implementado en el trabajo previo del alumno.

Los resultados que arrojan los tres grupos de algoritmos que se han estudiado en este proyecto demuestran que es posible sintetizar voz inteligible

a partir de bioseñales PMA.

Los resultados del estudio se basan en métricas de evaluación objetivas y en escuchas subjetivas. Los resultados obtenidos en términos de distorsión cepstral (MCD) se encuentran entre 9.41 dB y 12.4 dB para todos los conjuntos de datos, mientras que para inteligibilidad (STOI) se encuentran entre 0.32 y 0.606. Las pruebas subjetivas confirman que los algoritmos desarrollados tienen un rendimiento superior en términos de inteligibilidad con respecto a los métodos base, superando en ocasiones al algoritmo original creado por el alumno. Los algoritmos son capaces de realizar síntesis de voz inteligible a partir de bioseñales PMA, tanto para bases de datos de dígitos como para bases de datos de oraciones completas.

Speech Synthesis from Voice Biosignals using Machine Learning Techniques

Javier Lobato Martín

Keywords: Speech Synthesis, Biosignals, Unit Selection, PMA, WORLD, Neural Networks, Direct Speech Synthesis, CCA

Abstract

Many diseases and traumas can cause the complete loss or severe impairment of speech, such as in cases of stroke, ALS or laryngectomy. In addition, many of these various conditions are, to this day, incurable, making the loss of speech irreversible and, in many cases, a progressive and inevitable process. While there are devices on the market designed to help these people alleviate the communication problems they suffer from, such solutions are often slow and difficult to use, which significantly impacts the quality of life of those affected.

This research project is part of the collective effort focused on restoring the verbal communication ability to affected people through the application of technology and, in particular, telecommunications. Thus, in this Master's Thesis, a series of algorithms are implemented whose focus is voice synthesis from speech biosignals, captured from the movement of the lips and tongue of the participants, using a capture technique known as Permanent Magnet Articulography (PMA).

To achieve the objective of the study, multiple speech synthesis algorithms are proposed: Direct Speech Synthesis, Unit-Selection with CCA, Linear Regression by means of CCA and two different types of neural networks: DNN and GRU. The algorithms take information from a database containing PMA and speech biosignals recorded simultaneously from healthy individuals. In certain algorithms, parameters derived from speech will be used instead of raw speech: the Mel Frequency Cepstral Coefficients (MFCC). Depending on the algorithm, the signals used will be processed by dividing them in small portions called units. Several methods studied in the available literature are implemented in the study, which were chosen on the belief that they will give a good result for the specific case or may be an improvement on the method implemented in the student's previous work.

The results of the three sets of algorithms studied in this project demonstrate that it is possible to synthesise intelligible speech from PMA biosignals.

The results of the study are based on objective evaluation metrics and subjective listening. The results obtained in terms of cepstral distortion

(MCD) are between 9.41 dB and 12.4 dB for all data sets, while for intelligibility (STOI) they are between 0.32 and 0.606. The subjective tests confirm that the developed algorithms have superior performance in terms of intelligibility with respect to the base methods, sometimes outperforming the original Unit-Selection algorithm created by the student. The algorithms are able to achieve intelligible speech synthesis from PMA biosignals, both for single digit and full sentence databases.

I, **Javier Lobato Martín**, student in the **Telecommunication Technologies Engineering Master's Degree**, authorize the placement of a copy of my Master's Thesis in the university library, so that it can be consulted by whoever wishes so.

Sgd: Javier Lobato Martín

Granada, July 2024.

Mr. **José Andrés González López**, Professor in the Signal Theory, Telematics and Communications department, University of Granada.

Informs:

That the present work, under the title ***Speech synthesis from voice biosignals using machine learning techniques*** was written under his supervision by **Javier Lobato Martín**, and authorize the defense of such work before the corresponding tribunal.

For the record, he issues and signs this report in Granada on July, 2024.

The supervisor:

José Andrés González López

Acknowledgments

To my parents, who have supported me unconditionally at every stage of my life and without whom I would not be able to achieve the goals I set myself.

To my grandparents, who, after a life of effort and sacrifice, can see their children and grandchildren succeed.

To my friends, who accompanied me on this adventure and are an essential part of everything I do.

To my supervisor, José Andrés, for his infinite patience, professionalism and great work.

Acronyms

AAC Augmentative and Alternative Communication. 18

ALS Amyotrophic Lateral Sclerosis. 7, 17

CCA Canonical Correlation Analysis. 5, 7, 18, 19, 61–63, 71, 74, 78, 86, 91, 92, 95

DCT Discrete Cosine Transform. 33

DFT Discrete Fourier Transform. 32

DNN Deep Neural Network. 5, 7, 18, 40, 67, 69, 72, 81, 82, 88–90, 93, 96

DSS Direct Speech Synthesis. 18, 33–36, 65, 66, 72, 73, 92–97

ECoG Electrocorticography. 28, 29

EEG ElectroEncefaloGraphy. 27–29, 91

EMA Electro Magnetic Articulography. 25, 26, 35, 40

EMG Electromyography. 26

EPG ElectroPalatoGraphy. 24

ERP Event Related Potential. 27

GRU Gated Recurrent Unit. 5, 7, 18, 43, 44, 67, 69, 70, 72, 82, 89, 93, 94, 96

LSTM Long Short-Term Memory. 43, 44

MCD Mel Cepstral Distortion. 6, 8, 44, 45, 71, 72, 75, 77, 79, 86, 93, 95, 96

MFCC Mel Frequency Cepstral Coefficients. 5, 7, 30, 32, 36

MSE Mean Square Error. 39

NN Neural Network. 38

PCA Principal Component Analysis. 61

PMA Permanent Magnet Articulography. 18, 26, 35, 37, 47, 49–51, 91, 93, 95

RNN Recurrent Neural Network. 41–43, 97

SDR Signal to Distortion Ratio. 46

sEMG Surface Electro-Myogram. 27, 35

SNR Signal to Noise Ratio. 46

SSI Silent Speech Interface. 18, 19, 21, 24, 28, 33, 34

STOI Short Term Objective Intelligibility. 45, 71, 72, 75, 79, 81, 93

WHO World Health Organization. 17

Chapter 1

Introduction

Language is one of the most important inherent capabilities of human beings. Human evolution cannot be explained without this phenomenon, which we use to transmit knowledge, experiences and feelings. Language is one of the phenomena that allowed us to prosper as a species.

Unfortunately, there are multiple conditions that can result in the impairment or total loss of the speech ability. The most common causes of these conditions are traumatic injuries (caused by any type of sudden impact), laryngectomies (partial or total remove of the larynx due to multiple affections), stroke (interruption of blood supply to a part of the brain) or Amyotrophic Lateral Sclerosis (ALS) (progressive neurodegenerative disease that affects brain cells and spinal chord).

This problem has deep repercussions in society. Numerous studies evidence that speech impairments affect a significant proportion of society. A study carried out by the European agency Eurostat [7] concluded that 0.4% of the European population has an impairment in speech.

It has been found in [65] that speech disorders lead to lower academic skills, contributing to higher unskilled working rates. Notable social consequences are also found: difficulties socializing, anxiety disorders, increased bullying and lower engagement with partners.

Another study carried out in 2011 by the World Health Organization (WHO) in 70 countries [25], concluded that 3.6% of the population suffers moderate or severe difficulties in participating in the community.

Language is of utmost importance for human development, thus its absence has significant consequences for the individual: Daily communication is greatly hindered, as well as medical assistance (because of the ineffectiveness of information exchange). This impairment can develop a sense of social isolation and even clinical depression. Consequences also affect economy and the labor market. According [7], 78% of European population with severe speech disabilities are excluded from the labor market, the figure being 27% for the population not facing such a condition.

Unfortunately, as of today, there is not a universal solution to reverse most of the conditions that cause speech impairment, hence the paramount importance of finding solutions that can restore this capacity to the extent possible. Devices known as Augmentative and Alternative Communication (AAC) [62] can help restore the communication ability to varying degrees. Solutions range from simple handwriting to text-to-speech conversion systems. These systems have their limitations and their use is not feasible in all cases.

1.1 SSI's

In recent years, there has been a growing interest in the field of Silent Speech Interface (SSI) [62], [19]. These type of systems enable oral communication without the need of utterance vocalization, by means of interpreting biosignals of varying natures [20]. In this context, we understand biosignal as any human-originated signal that has some level of correlation with the speech process and that is susceptible to be used to synthesize voice.

Biosignals that are used as a source of information originate from diverse parts of the human body that take part in the speech production process. Some examples include lip-movement sensing, vocal tract movement recording or procurement of speech-related neural activity from the brain 2.1.

SSI's count with numerous potential applications. For example, patients who have undergone a laryngectomy or elderly patients whose speech production demands significant effort but the remainder movement allows for the use of these kind of interfaces.

There is a great diversity of approaches that make use of the information provided by biosignals for speech or text synthesis, depending on the biosignal's nature, philosophy applied or algorithm implemented.

SSI's have the capability to generate natural sounding speech with an easy and intuitive interface, and therefore are a fast-growing area of study that can improve significantly the living standard of, potentially, millions of people.

1.2 Objectives

The goals of this work are divided into main and secondary ones. The main goal of this project is to achieve **intelligible speech synthesis from biosignals obtained via Permanent Magnet Articulography (PMA)** (mouth and lips movement capture). Multiple algorithms were created for this purpose in this investigation: Direct Speech Synthesis (DSS), Unit-Selection with CCA and Neural Networks using a DNN and GRU Neural Network. All details on the algorithms and the biosignals used will be specified in later sections. Specifically, the database used for the project's main

goal is composed of individual digits.

Secondary goals include:

- Speech synthesis for a database that includes complete sentences. These are phonetically balanced and have increased complexity.
- Creation of a base method for speech synthesis using CCA Linear Regression, in order for it to be compared with the student's previous work on the field and the new algorithms.
- Evaluation of the performance of all different architectures and synthesis approaches studied throughout the project.

1.3 Structure of the thesis

This work is structured in chapters, according to the following distribution:

- **State of the Art** 2: In depth study of SSI's and overview of most notable works on the field in recent years.
- **Proposed Methods** 3: Explanation of the specific use case for the work, as well as presentation of the designed algorithms for speech synthesis.
- **Obtained Results** 4: Evaluation of experimental results obtained for the proposed speech synthesis methods.
- **Conclusions and Future Lines of Action** 5: Final conclusions arrived after evaluating the obtained results. Proposal of future improvements on the methods and alternatives lines of research.
- **Annexes** A: Project timing A.1 and budget A.2.

Chapter 2

State of the art

This chapter serves as a more in-depth presentation of SSI's. A breakdown of the main components of a speech synthesis system will be presented, as well as a review of most relevant works in the field in recent years. In this manner, the first section 2.1 specifies a typical speech synthesis system that feeds on biosignals (SSI), which is the main objective to implement during this project. The rest of the sections explore each of the components, including physiological processes involved in speech production, techniques for obtaining biosignals, parameters extracted and the eventual decoding of speech from the biosignals.

2.1 Arrangement of an SSI system

This section will focus on the description of the main components of a silent speech interface. The following diagram 2.1 illustrates the main stages that an SSI comprises of: from record, to processing and eventually to output.

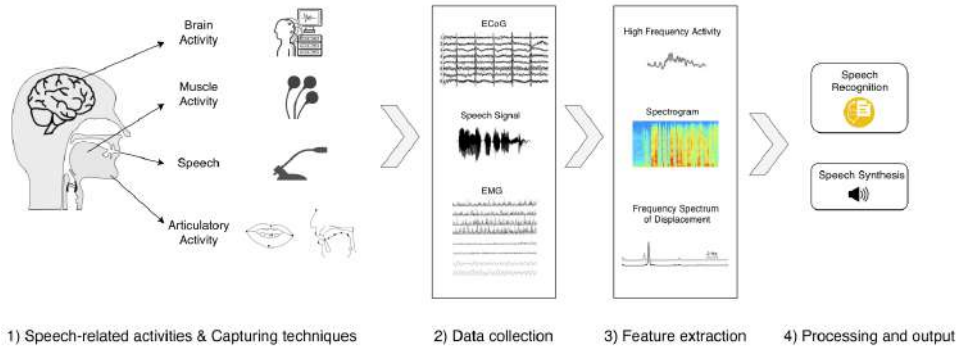

Figure 2.1: Communication system based on SSI. Source: [62]

As shown in diagram 2.1, speech synthesis process follows some steps that can be organised into four different procedures:

1. First step comprises of finding what speech-related phenomenon is going to be recorded.

2. Biosignals of interest are obtained using the chosen method.

3. Features are extracted from the biosignals using different kinds of signal-processing techniques.

4. Speech is decoded from the features obtained from the biosignals.

The following sections will follow the logic presented in 2.1. A review of the different techniques and important aspects will be performed according to the relevance, scope of application of bibliographical abundance.

2.2 Physiological processes involved in speech

This section summarizes the physiological phenomena that take place during speech production [5]. This processes are the physical source of biosignals that we will be using in our SSI system and represent the very first step when it comes to addressing the problem.

Following figure 2.2 illustrates the different parts of the human body that participate in the language production process. The conception of speech takes place in the human brain, source of the signals that activates the muscular response for vocal articulation.

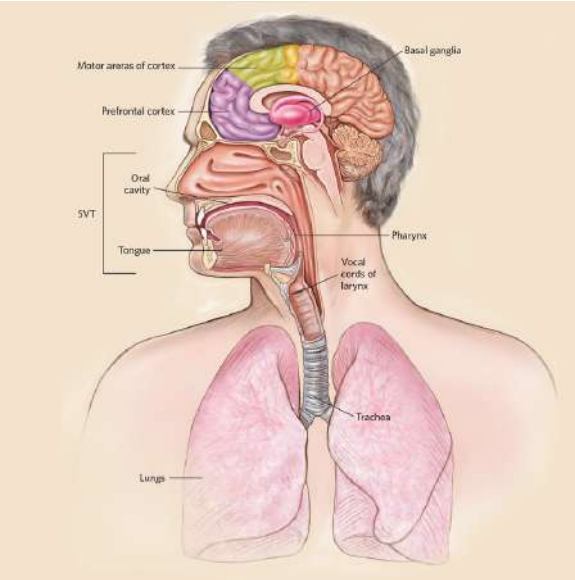

Figure 2.2: Physiognomy of the different body parts involved in speech production. Source: *thescientist.com*

2.2.1 Brain Activity

Brain-originated signals is the root of all verbal communication [14]. There is a clear and unbeatable advantage in using this activity as a source for biosignals: it serves as a solution for all the possible types of conditions: from diverse disorders that do not lead to total loss of voice (dysastria, apraxia, laryngectomy) to the most severe cases like aphasia (total loss of speech-ability). Many studies have linked the process of speech production to a specific region of the brain: the superior temporal gyrus [60]. Speech can be decoded by means of treating with signals from this area of the brain. However, significant problems arise with this technique. Brain mechanisms that originate speech are of great complexity and as of today are not fully understood. Additionally, these signals need to be obtained with great resolution to obtain acceptable results, which is another challenge in itself due to the invasiveness of the signal recording. These problems are a huge constraint for speech synthesis from brain signals.

2.2.2 Muscular activity

Verbal communication needs movement in the face muscles, mouth and tongue as well as in larynx. There is a high level of correlation between audible voice itself and these movements, so speech synthesis is possible by sampling and processing signals that represent muscular activity [66].

In the speech production process, once the message conceptualization

and motor activity planning has taken place in the brain, electrical impulses are transmitted by the motor neurons in the peripheral nervous system, which innervate the muscles involved in the speech production process. The electrical impulses coordinate the muscle contraction and relaxation, creating specific movements. It is this movement, combined with the variable airflow through the vocal tract, that generates speech. Thus, there are two distinct sources of information from which biosignals can be obtained: sampling of electrical impulses that generate muscle movement or sampling the muscle movement itself. Sampling of muscle activity is of great interest for speech synthesis, as it is much more specific than brain signals and whose conversion to audible voice has been proven feasible for years now. As a drawback, not every patient case is compatible with muscular activity sampling, given that in many cases the voice articulators are absent of movement or electrical activity from which signals can be sampled.

2.3 Biosignals Acquisition

In the field of SSI, speech biosignals are defined are physiological signals that are related to various aspects of the human speech production process. These signals may or may not be electrical. Signal acquisition is performed using specialized sensors for each kind of biosignal. This section will focus on the most relevant biosignal extraction methods for their use in SSIs.

2.3.1 Articular Movement

Speech production requires movement in the voice articulators: lips, tongue, palate and larynx. Biosignals are obtained by means of placing magnetic or imaging sensors in different areas of the vocal tract [11], [23]. These methods are not designed to capture glottal activity (whose functioning influences the pitch and intensity of the voice). Thus, the scope is normally people with speech disorders such as people who have undergone laryngectomy. Four methods may be highlighted:

ElectroPalatoGraphy (EPG)

EPG. An electrode array is situated in the palate to register the sequence of contacts between tongue and palate [8]. The contact pattern conveys information about phoneme pronunciation. Usability for the current case of study is limited because information is mainly related to phonetics. Also, tongue movement is needed. [3]

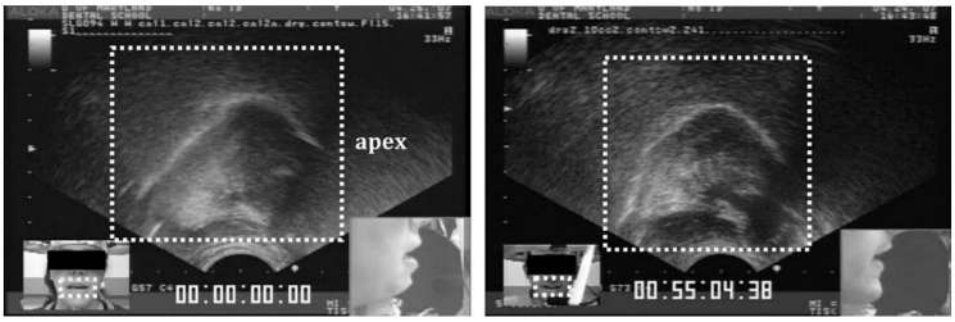

Figure 2.3: Vocal tract and mouth image for two separate patients. Source: [21]

Imaging techniques

Individual images and video is obtained from voice articulators as they move. This is a practical and simple practice. Different sensors may be used, such as radar, ultrasonic, optic, etc. In general, it need to be implemented accompanied by some other type of information to show a complete representation of speech process [21]. Figure 2.3 shows an example.

Electro Magnetic Articulography (EMA)

Consists in arranging a set of magnetic field sensors along voice articulators, wiring them to external processing units. Transmission coils are laid near the patient's head, generating an alternating magnetic field. The continuous sampling of this EM signal allows to track the spatial coordinates of the receptors and thus obtain voice-related parameters. The temporal resolution of this method is very high, but the glottal movement cannot be registered, difficulting the speech synthesis process. Additionally, the need of external machinery increases inconvenience and difficults its utilization [32].

Permanent Magnet Articulography (PMA)

It is a similar solution to EMA but it implements some clear advantages. Using this approach, numerous small-sized magnets are arranged in specific parts of voice articulators. The movement of the magnets generates variable electromagnetic fields, which are recorded by a compact set of magnetic sensors placed outside of the mouth. This sum captures the temporal evolution of the position of the various voice articulators. Depending on the configuration: lips, tongue and jaw. The main advantage of this method consists in that no wired routings to an external machine are needed. As a consequence, this technique is significantly more convenient for the user, also enabling portability. Sensors are easy to place and can be fixed only for

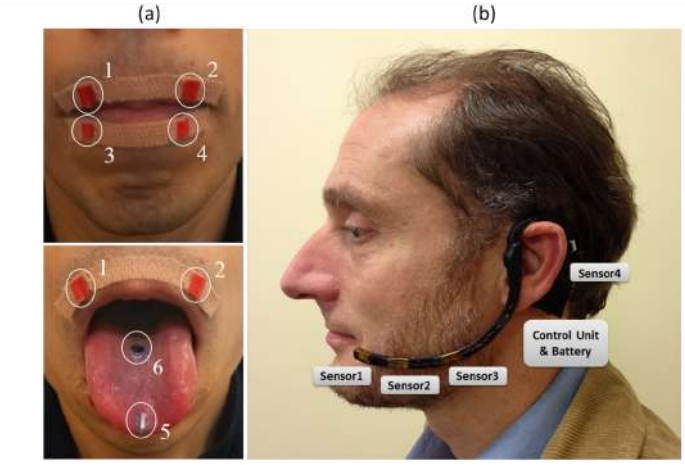

Figure 2.4: Magnet and sensor arrangement for PMA. Source: [47]

a given amount of time or permanently. There is one drawback: the signal used is a sum of the magnetic field generated by the movement of several magnets, so the relationship between signal information and precise position of the magnets is less explicit than in EMA [62]. Following figure 2.4 shows a typical configuration for obtaining biosignals using PMA

Left part of the image 2.4 (a) shows an example of magnet placement. In this case the chosen locations are lips and tongue. Right part of the image 2.4 (b) shows and example of sensor arrangement. As its visible, final system is compact and self-contained (does not require external wiring connections with any device) This method will be further documented in following sections, as it will be the method used to capture the biosignals used in the created algorithms.

2.3.2 Electromyography (EMG)

This method harvests the electrical potentials that activate facial muscles in the contraction phase. It can either be obtained using invasive or non-invasive techniques. The latter is considered to be the most preferable. However, the obtained signal (which is already complex, depending on the nervous system and the pysiological and anatomical properties of the muscle) has added noise originated from its path through skin and possible interference with other muscles. One clear advantage of EMG is that the electrical signal that is sampled appears some 60ms before the actual muscular contraction, allowing a big improvement in terms of latency and real-time systems. Its major downside is that results may vary notably between training sessions, given that is incredibly difficult to match the sensor placement from session to session. Following figure 2.5 shows an example of sensor arrangement.

Figure 2.5: Sensor arrangement for sEMG. Source: [36]

2.3.3 Brain Biosignals

There is a wide range of sensors design to register brain activity. For the scope of this work, the main difference for electrodynamic methods is between invasive and noninvasive methods. Hemodynamic methods also exists, although they are outside the focus of this project.

ElectroEncefaloGraphy (EEG)

It is one of the most popular methods for obtaining brain activity because it is non-invasive and has been in use for a long time. Electrodes are distributed along the scalp for the acquisition of electric signals. The result is a signal with a good temporal resolution but a poor spatial resolution. This is because what is obtained is a quite smoothed version of the firing pattern of the neurons in the area. Additionally, the skin and skull that the electrical pulse traverses has the effect of a low-pass filter. As a consequence, this method is only used to obtain broad patterns on neuron firing. A widely used example is the P300 potential. P300 is a specific biosignal obtained by means of EEG which is an example of what we refer to as Event Related Potential (ERP). This is a brain electrical response as a consequence of a stimulus, whether this be cognitive, motor or sensory. P300 is obtained by means of what is known as the 'Oddball Paradigm', where visual stimuli are repeated continously and mixed with others less frequent, which must be pointed by the patient and consequently triggers the P300 potential [56]. Figure 2.6 shows an example layout of sensor placement for the obtention of EEG biosignals.

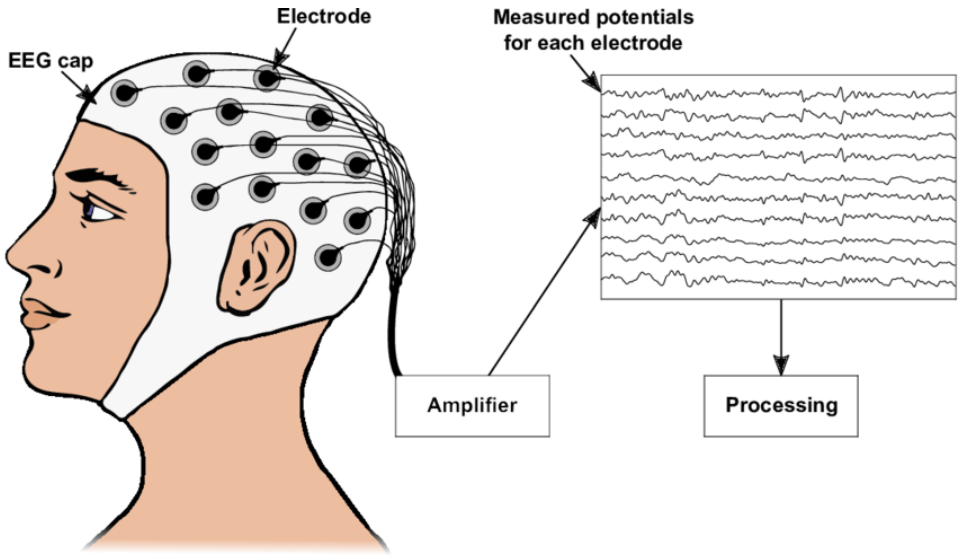

Figure 2.6: Sensor placement scheme for EEG biosignal recording. Source: [59]

Electrocorticography (ECoG)

It is an invasive method, by means of which an electrode array is directly laid over the brain cortex. This way, a very high temporal and spatial resolution is yielded, allowing also for very high portability. These features would allow for this method to be used for restoring speech after fitting the prostheses. Diverse studies have analyzed the capability of creating SSI models, with varying degrees of success [54], [55]. Following figure shows an example 2.7.

2.4 Feature extraction

This section will serve as summary of the main biosignal signal processing techniques used for representing the silent-speech biosignals described in the previous section, so they can be presented as sequences of feature vectors amenable for speech recognition and/or speech synthesis. As will be seen, this is a task very tailored to the specific characteristics of the biosignal under treatment.

2.4.1 EEG features

Alpha Waves

Electrical signals that take place in the brain are oscillations that can be of varying frequencies. In the case of Alpha Waves, their spectrum ranges from 8 to 12 Hz and originate from the synchronous and coherent combination of

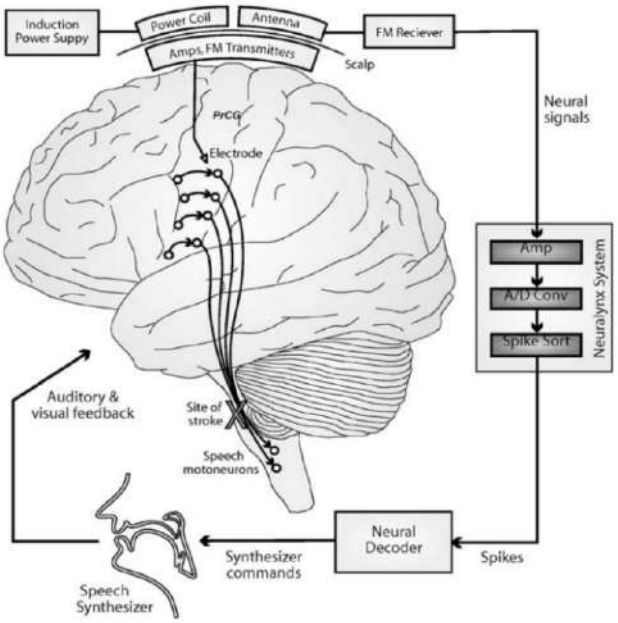

Figure 2.7: Sensor placement and speech production scheme for ECoG. Source: [18]

electrical activity in the thalamus' cells. They are used in the speech synthesis field because it is hypothesized that the waves are somewhat related with communication process [30].

Beta Waves

Similar to Alpha Waves, these comprise frequencies ranging from 12.5 to 30 Hz and are also obtained by EEG methods. Its use in speech synthesis originates in the link between Beta Waves and isotonic muscle contractions, as well as with active though process and concentration [15], [13].

2.4.2 Speech Features

Spectrogram

Used extensively in signal processing, specially for audio. It is a visual representation of the evolution of the spectrum along a given time. It is a three-dimensional signal: Time, frequency and spectral density. Applied to voice, allows to identify individual utterances, as well as spectral components of voice and its evolution. Figure 2.8 shows and example.

Spectrogram is an interesting representation in that it can allow us to clearly evaluate the similarity between original an synthesized signals, being

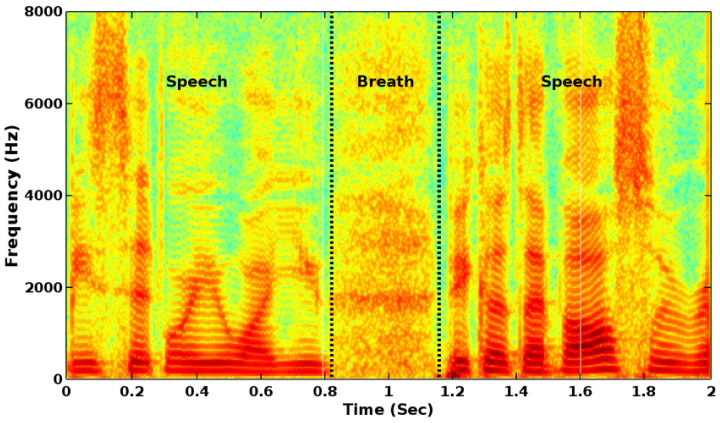

Figure 2.8: Typical speech spectrogram. Source: [46]

able to check the proximity of the spectral components of each instant at a glance.

WORLD Vocoder

A VoCoder is a signal processing technique used for analyzing, synthesizing, and manipulating voice waveforms. It works by breaking down the speech signal into its spectral components and applying these characteristics to a carrier signal, effectively encoding the voice information. In this work, we will use a well-known speech VoCoder known as WORLD [43].WORLD is a state-of-the art VoCoder designed for the synthesis of high-quality speech in real time. According a study carried out in [43], it is 10 times faster than conventional methods of speech synthesis. It is composed of 3 analysis algorithms and one synthesis algorithm.

Figure 2.9 shows that the 3 analysis algorithms obtain the fundamental frequency *F0*, the spectral envelope and the aperiodicity parameter. The synthesis algorithm uses these three parameters to get to the synthetic speech. For the parts of this project that use this VoCoder, the spectral envelope of the signal will be obtained (represented by the spectral coefficients MFCC), so the VoCoder analysis will only be focused in the retrieval of this envelope, as well as the speech synthesis itself.

Human voice is composed by an overlap of single-frequency waves. The lowest frequency of all is known as the fundamental frequency (F0) and it's used for speech characterization.

The aperiodicity parameter serves the purpose of stating the presence of non-periodic components in voice, which have diverse sources and contribute to speech quality.

The spectral envelope is a key parameter in speech synthesis. Algorithms used for its calculation are typically *Cepstrum* [44] and *LPC* [2]. The main

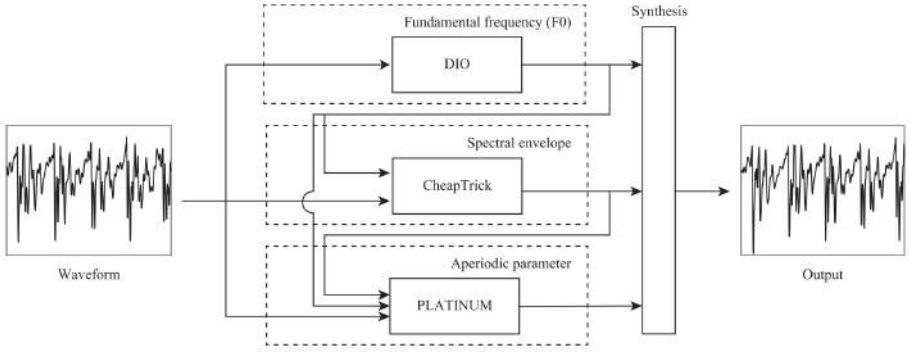

Figure 2.9: Outline for WORLD Vocoder for analysis and synthesis. Source: [43]

issue is that the output of the algorithm varies with timing, so it's key to eliminate the temporal variation as much as possible, if possible without losing quality on the estimation. The estimation of the spectral envelope uses an algorithm called CheapTrick [31] in various steps:

1. Power spectrum of the waveform is calculated, applying previously a Hanning window.
2. The power of the windowed waveform is stabilized in time by means of an integral.
3. Power spectrum is smoothed with a rectangular window.
4. Time-Variant component is eliminated using *liftering* (applying a window in the cepstral domain).

WORLD uses a synthesis algorithm that uses the minimum possible convolution products for obtaining speech. Following figure 2.10 shows an outline of the VoCoder.

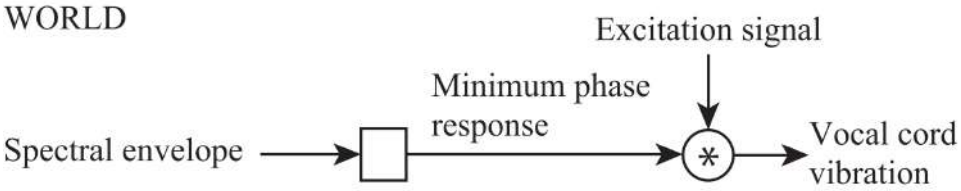

Figure 2.10: WORLD VoCoder Outline for speech synthesis. Source: [43]

The final result when synthesising voice is the the vibration of vocal cords in each instant. For that, the vibration is obtained as the convolution

between the minimum-phase response of the spectral envelope and an excitation signal. Fundamental Frequency F0 is used for establishing the start of each vocal cord vibration. WORLD VoCoder needs less convolution operations for speech synthesis than other options [43], so the computational cost of the whole process is reduced, allowing its use in real-time scenarios.

Cepstrum

Cepstrum is the result of applying the Inverse Fourier Transform (IFT) to the log-spectrum of a signal. It is used in speech analysis and conveys information on the rate of change in the spectrum bands of a signal.

MFCC

MFCC's are coefficients used to characterize speech in a compact format and are based in the human perception of hearing. Together they make up a representation of the power spectrum of a sound in a short period of time. These coefficients represent features of the vocal tract transfer function. They are widely used in speech recognition applications. For a portion of the project, MFCC's will represent the targets that will be predicted, so that they can be fed to the VoCoder. This will imply that instead of working with the audio signal itself, MFCC's are obtained from the audios using WORLD. For speech synthesis, the inverse process is performed. Calculation is carried out the following way:

- Signal is divided into short sections, typically from 20 to 40 ms, with a certain overlap so that continuity is not lost.
- Discrete Fourier Transform (DFT) is applied to each of these sections.

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N} \quad 1 \leq k \leq K \tag{2.1}$$

  Where $s_i(n)$ is the original signal divided into sections and $h(n)$ is the window applied to $s(n)$. For this calculation, a Hanning window is usually applied. K indicates the longitude of the DFT.

  Next, DFT is squared and its the absolute value obtained, attaining the power spectrum.

$$P_i(k) = |\frac{1}{N}S_i(k)|^2 \tag{2.2}$$

- A *Mel* scale Filter Bank is applied to the spectrum using overlapped triangular windows or Hanning windows. The reasoning behind using *Mel* scale comes from it being perceptual. That is, it is based in

the human hearing sensitivity at different frequencies. This scale is approximately linear up to 500 Hz, beyond which, increasingly wider frequency intervals are established for equally wide increments in the human perception of pitch. The next equation is used for conversion

$$M(f) = 1127 \ln(1 + f/700) \tag{2.3}$$

Whereas for the inverse procedure, following equation is used:

$$M^{-1}(m) = 700(e^{m/1127} - 1) \tag{2.4}$$

- Finally, logarithm is performed for the energies of every *Mel* scale frequency and Discrete Cosine Transform (DCT) is applied.

Hilbert Transform

Widely used in the field of signal processing and in mathematics, Hilbert Transform is a linear operator that transform a function using the following equation:

$$\hat{x} = x(t) \circledast \frac{1}{\pi t} \tag{2.5}$$

Where $\circledast$ indicates the convolution operation. Hilbert transform has a very simple frequency representation. It shifts the phase of the positive spectral components by $-90°$ and for the negative ones it shifts $+90°$, while the spectrum stays unaltered in magnitude. By means of this operation one can extract the complex envelope of a signal.

2.5 Speech decoding from biosignals

As shown in figure 2.1, the final objective of an SSI system is to decode the message that the user wants to transfer through interpretation of the recorded speech biosignals. In general, biosignals are not used directly, they are processed in order to have a more compact representation and to maximise the correlation with the speech process features, as commented on the previous section. The algorithm for calculating the voice sequence will depend on the type of biosignal that one is working with. As shown in 2.1, there are two main alternatives for decoding speech from biosignals: Text conversion (from biosignals to text) and direct speech synthesis (from biosignals to voice). For the scope of this project, one of the spotlights will be cast on direct voice conversion, more known as DSS. Under this premise, what is sought after is to perform a transformation $\boldsymbol{f} : x \rightarrow y$, where $\boldsymbol{x}$ represents the feature vector extracted from biosignals and $\boldsymbol{y}$ represents the

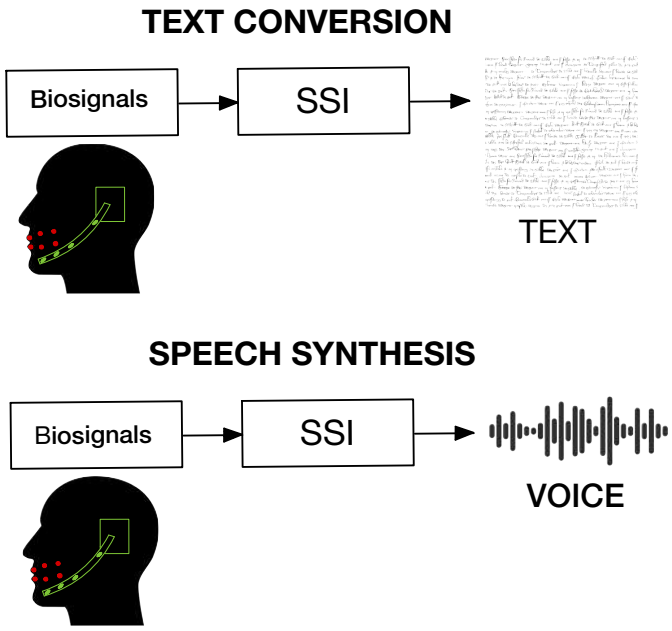

Figure 2.11: Approaches for speech decoding from biosignals

feature vector extracted from the voice acoustic signal. For each time unit, the following is computed:

$$y_t = f(x_t) + \epsilon_t \tag{2.6}$$

The challenges of DSS arise from the fact that this function is nonlinear and, additionally, the transformation performed is not univocal, that is, the same acoustic features might be mapped to multiple biosignal features. Also, most biosignal recording systems do not possess an infinite temporal or spatial resolution, so it is inevitable that part of this information is lost, thus affecting the quality of the speech decoding process.

Depending on whether the objective of the SSI is speech or text synthesis, a differentiation can be made between two alternative approaches. Figure 2.11 illustrates this concepts. The following sections briefly describe both approaches.

2.5.1 Text Conversion

The process of synthesizing text from biosignals has one advantage: it can be predicted more accurately, thanks to language and pronunciation models. However, they are not capable of identifying words that were not previously recognized during the training phase. The process of recording a number of

biosignals big enough in order to have a consistent model requires a significant sample mass. Text conversion systems are usually coupled with Text To Speech (TTS) [16], [53] systems to obtain an audible voice. As a consequence, all the paralinguistic context of the speech process (intonation, differences depending on the mood or personality of the user) is lost in text synthesis. These are disadvantages worth mentioning, although usually not critical. The major problem with these type of systems is the decoupling between the generation of the biosignals themselves and the audible feedback from the text being uttered. This implies that the system is not able to work in real time, which has negative consequences for its use in patients. According [6], in spoken communication, a delay that ranges from 100 to 300 ms causes hesitation by the speaker. When the delay surpasses 300 ms, users start to avoid speaking in order not to interrupt. As for audible feedback, in text speech process, the negative side effects start to appear at 50 ms of delay, considering it acceptable up to 100 ms [1]. These ranges of delay are not achievable by the current state of the art in text synthesis systems.

2.5.2 Speech Synthesis

Direct speech synthesis works with biosignals to produce speech in a direct manner. This technique allows for a much lower latency, enabling real time operations. For instance, speech synthesis systems have been proposed in the literature for sEMG [10], PMA [41] and EMA [26] biosignal modalities. Performance regarding delay opens the possibility for the audible feedback received by the user to be assimilated as its own voice. This allows for a better acoustic parameters modulation by the user, besides from improving acceptance among these type of devices [12].

Within the DSS technique, a distinction can be made into two different methodologies, depending on the approach chosen [62]:

Model Based Conversion

The mapping from sensory to acoustic features is divided into two sections [17]:

1. First, a physical model of the vocal tract is estimated from the biosignals.

2. Speech synthesis is then performed by means of simulating airflow through the vocal tract.

A clear disadvantage of this synthesis technique is that the model must be highly accurate to obtain acceptable results. Achieving precise models is a complex process and computationally demanding [40].

Data Based Conversion

The methodology of Data Based speech synthesis is, as of today, the most widely used. The mapping between sensory and acoustic features is modelled as a parametric function of the type: $\boldsymbol{f(x;\theta)}$, where θ are the function's parameters. A reasonable way of estimating the conversion shown in 2.6 is by making use of a dataset that is pairwise labeled (x, y), that is, a statistical approximation is used where the parameters of $f(x)$ are estimated in order to minimize a loss function. This is the approach most commonly used as of today in Machine Learning techniques and is the one adopted for this work by means of the various versions of Unit-Selection and Neural Networks. The process is carried out in two essential stages:

1. **Training phase:** In the model training, voice parameters are estimated using a data set with pairwise labeled *source* and *target* vectors $D = (x_1, y_1),....,(x_N, y_N)$. The dataset is obtained by means of capturing simultaneously voice and biosignals in an early enough stage that the patient still has intact or slightly impaired speech abilities. Voice parameters can either be used raw or through an acoustic parametrization, typically using MFCC's.

2. **Synthesis phase:** Once the θ parameters are estimated, a mapping function can be used to synthesize the patient's voice through predicting the acoustic features using only the biosignals. Voice can then be obtained directly by concatenating each segment or by using a VoCoder (such as WORLD) when MFCC's are used.

2.6 Machine Learning techniques for speech synthesis from biosignals

This section will focus in the review of the most relevant techniques found in different studies for Direct Speech Synthesis

2.6.1 Linear Methods

As stated previously, the requirement of real-time operations is key when it comes to determining the viability of implementing a DSS system in patients. One way to simplify and speed up the process is to establish a linear relation between the biosignals obtained from the sensors and the voice or parameters derived from it like MFCC's. Linear regression models search for the linear equation that best describes the link between the available signals and the computed voice parameters. The most popular fitting is least squares. When calculated, the equation of the line is as follows 2.7:

$$y = mx + b \tag{2.7}$$

Where, for our use case, x would be our PMA vector (source) and y our MFCC's vector (target). The line equation is calculated for **N** pairs of points (x, y), such that the squared error between the cluster of dots and the line is minimized. In the equation, x comprises the biosignal features and y the voice parameters. To obtain the expression, the square error is derived and equaled to zero, whose expression is shown below 2.8.

$$SSE = \sum (y - \hat{y})^2 \tag{2.8}$$

SSE refers to Sum of Squared Errors. The next equations show how each of the parameters of the line are calculated by means of least squares 2.9, 2.10.

$$m = \frac{N \sum (xy) - \sum x \sum y}{N \sum (x^2) - (\sum x)^2} \tag{2.9}$$

$$b = \frac{\sum y - m \sum x}{N} \tag{2.10}$$

For a linear regression method to work with an acceptable performance, it is necessary to assume that the relationship between variables is of a very low complexity. That is because, as stated previously, this relationship is usually nonlinear. The main advantage of this method is that its simplicity allow it to be implemented in real time, even in mobile devices.

As example of the application of this technique, in [24], a linear regression model is used to synthesize voice directly from PMA biosignals obtained from the lips and tongue a healthy participant. In particular, a linear regression model is used to estimate the first two speech formants (F1 and F2) from 9-dimensional PMA samples extracted with a sampling rate of $F_s = 100Hz$. The corresponding model has a very low computational complexity, but the results have a big room for improvement, as the correlation coefficients range between [0.48 , 0.72] for the linear adjustment. The feasibility of this approach is limited. The following figure 2.12 shows the prediction distribution for each speech formant.

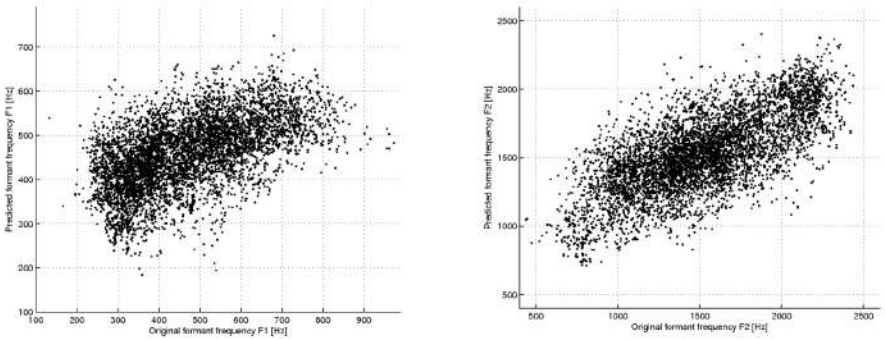

Figure 2.12: Distribution for predictions of each patient in the study. Source: [24]

2.6.2 Nonlinear Methods

As discussed in previous sections, the nature of the association between biosignal parameters and speech parameters has an underlying nonlinear behaviour. That is why methods that take into consideration the fact that this relationship is nonlinear have a significant importance in this field of study. Among the nonlinear techniques for speech synthesis, the one that has gotten the highest level of attention are the Neural Networks (NNs) [38], [39].

Neural networks are loosely bio-inspired, artificial intelligence algorithms. They find their origin in the process of trying to replicate the functioning of the human brain. In a brain, you can find millions of neurons which are interconnected via axons. They transmit and receive information among them using electrical pulses through their dendrites.

In an artificial neural network there are multiple processing nodes (called neurons) which are connected among them (just like brain neurons are). Neural Networks are intended to be able to recognize patterns for all kinds of inputs (vectors, matrices, images, sound, text, etc), their use is more indicated when large databases are available in order to train the net.

Neurons in neural networks are arranged in multiple layers. Each layer has a given number of neurons. The following figure 2.13 shows the basic elements of the most common type of neural network: feedforward.

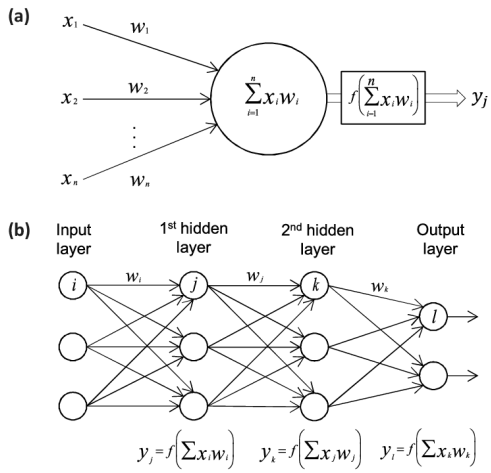

Figure 2.13: Functioning basics of a feedforward Neural Network. Source: [52]

In the upper part of figure 2.13 (a) we have a representation of a neuron. Each neurons receives a series of inputs, each one with a corresponding weight applied to them. Then all of the inputs are summed and the result is passed through a nonlinear activation function, that determines the learning rate and precision of the model itself, apart from normalizing the final neuron output. This nonlinear nature of the neuron is key, because it allows the neural network to recognize patterns or link variables which relationship may not be linear or complex. The output of each neuron has the following aspect:

$$y_j = f(\sum X_i W_i) \tag{2.11}$$

Where X_i indicates each of the inputs of the neuron and W_i the weight applied to each of them. In the learning phase, a starting value is given to each W_i. Multiple iterations are performed. In every iteration, the output is evaluated to check if it is correct or not, typically using an error metric (i.e: Mean Square Error (MSE) or *cross entropy loss*, among many). With this metric, backpropagation is performed, which consists in passing on the error metric in the opposite direction (in 2.13 would be from right to left), so that neurons can use it as feedback in order to adjust their weights for future iterations.

The lower part of figure 2.13 shows the general look of a neural network, in this case with 3 input neurons, 2 output neurons and 2 hidden layers.

Neural Networks were previously studied for their application in speech synthesis, where they were used to model the link between biosignals and speech features. This studies date form the 1990's. Its limitations at the time implied that the results were not as good as other methods, they were no

longer used and fell out of favour. Neural networks were recently revisited, with much better results, to the point where now the represent one of the best shots at synthesizing speech from biosignals. Some of the advancements that made possible this shift were:

- More capable computers mean that neural networks with more hidden layers can be implemented, increasing performance significantly.
- Thanks to cheaper and faster data storage, larger datasets are much more common. The bigger the training set, the better for a neural network.
- Advancements in all the other complementary systems for a speech synthesis: pre-processing of biosignals, filtering, vocoder's, etc.

Various studies have been carried out regarding the use of Neural Networks in speech synthesis. In [49] a biosignal-to-text conversion system is implemented using EMA biosignals. In [35] a DNN is used for direct speech synthesis, also using EMA biosignals. The use of DNNs improves the synthesis velocity compared to a standard gaussian distribution map.

Neural Networks are an extensive field of study and there is a wealth of different possible implementations. Following, the most important are reviewed [45].

Deep feedforward Neural Network (DNN)

The feedforward is the most common type of Neural Network: it is just a normal Network with a given amount of layers. There is not an exact number of layers from which a Neural Network is a Deep one, but the consensus is typically 4. Its implementation includes a Neural Network with a number of hidden layers that map the features of the biosignals with the features of the speech (or the speech itself). This linkage between parameters is performed, by definition, frame by frame, so there is no contextual information to be shared among frames. This information is valuable, as it is important to construct a coherent and smooth speech sequence. No temporal context is, as a consequence, the biggest drawback of feedforward Deep Neural Networks. On the advantage side, they are the easiest and most computationally efficient of all.

Figure 2.13 shows a feedforward Neural Network. A DNN uses the same scheme but includes a higher number of hidden layers.

The specific operation that is performed by each neuron is shown in 2.12 and 2.13:

$$h_t = H(W^{xh}x_t + b^h) \tag{2.12}$$

$$y_t = W^{hy} h_t + b^y \tag{2.13}$$

Where H is the activation function in the hidden layer. Some examples for this activation functions are Sigmoid, ReLu, Heaviside, etc. The equation for the ReLu activation function (it will be used for the proposed methods that use Neural Networks) and Sigmoid are shown in the following equations 2.14, 2.15:

$$f(x) = \max(0, x) \tag{2.14}$$

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.15}$$

W^{xh} and W^{hy} are weight matrices. b^h and b^y are vectors that induce a bias, which improves the convergence process in the training phase (optimization of weights such that the error reaches a minimum value). Speech features are predicted by the $W^{hy} h_t$ linear regression, using the information from all preceding layers

Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are a type of Neural Network architecture designed to work with sequential information. This is of a special interest to our scope because among its applications we can find natural language processing, speech recognition and synthesis, automatic translation, etc. All of the mentioned applications work with speech sequences. In a speech sequence long-term dependencies are a very important factor, as voice is very much context dependent: small portions of speech largely depend on past and future small portions of speech. It is, then a valuable source of information to be able to retain this context information. RNN's can maintain information from previous sequences that is later used to influence following inputs. In the same manner as in traditional feedforward Neural Networks, the system is composed of multiple layers of interconnected neurons, with the key difference that in this case, the connections are recurrent: the input of one layer receives information from the previous layer and also from its immediate past self. One key advantage of RNN's is that it works with inputs and/or outputs of variable length.

In recurrent Neural Networks, biosignal features are mapped to speech features, but with the added advantage that there is a recurrent hidden layer that includes contextual information. As stated before, this information is valuable and can increase the quality and smoothness of the synthesized speech.

The following figure 2.14 shows a basic description of the different layers in a RNN:

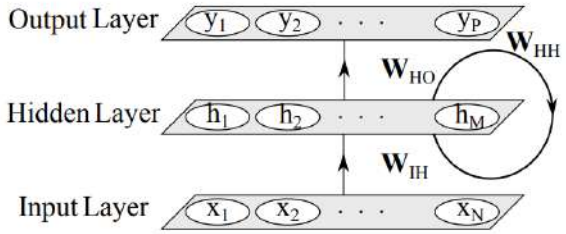

Figure 2.14: Functioning basics of a Recurrent Neural Network. Source: [51]

As it can be seen 2.14, in this case, there are Recurrent Hidden Layers between the input and output layer that implement the recurrence. Not necessarily all of the hidden layers need to be recurrent.

The recurrence in an original RNN is implemented by adding a hidden state h_t. The activation of this state at each instant depends on the activation at the previous instant. For a given sequence $x = (x_1, x_2, x_3, ..., x_T)$ the hidden state is updated according to 2.16

$$h_t = \begin{cases} 0, & \text{t=0} \\ \phi(h_{t-1}), & \text{otherwise} \end{cases} \tag{2.16}$$

ϕ is a nonlinear function (i.e: sigmoid function with an affine transformation). The output sequence $y = (y_1, y_2, y_3, ..., y_T)$ can also be of variable length.

The update of the hidden state h_t is as shown in 2.17

$$h_t = g(Wx_t + Uh_{t-1}) \tag{2.17}$$

Where g is a smooth function like a hyperbolic tangent or a logistic sigmoid. A probability distribution is passed over to the next element, stated the present one h_t. Sequences of variable length are supported thanks to an output symbol that indicates the end of the sequence. The probability in the sequence is as follows 2.18

$$p(x_1, \ldots, x_T) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_T \mid x_1, \ldots, x_{T-1}) \tag{2.18}$$

Where $p(x_T \mid x_1, \ldots, x_{T-1})$ is this output symbol that states the end of the sequence. Each conditional probability is modelled with the equation 2.19

$$p(x_t \mid x_1, \ldots, x_{t-1}) = g(h_t) \tag{2.19}$$

The implementation just shown is the one for a basic, unaltered RNN and it has one inconvenience: It is difficult for it to take into consideration

long term dependencies. The recurrent unit just outputs a weighted sum with a nonlinear function on top of if. The consequence is that gradients usually vanish (they gradually disappear as they are propagated back in time) or they explode (they grow exponentially). The former is the less common.

There are variations on the original RNN that do consider long term dependencies and solve the previous stated problem. The main difference is the implementation of a better activation function by using an affine transformation followed by a nonlinearity. This operations are introduced by a gating unit [29]. The most successful versions of this improvement are Long Short-Term Memory (LSTM) [4] and GRU [28].

- **Long Short-Term Memory:**

  LSTM Neural Networks are capable of retaining previous information for several time steps. This is implemented thanks to some key components. Each unit has a memory cell that stores information and three types of gates that moderate the flow of information:

  1. Forget gate: Decides what information is discarded before being stored in the memory cell. f in the figure 2.15
  2. Input gate: Decides what information is added to the memory cell. i in the figure 2.15.
  3. Controls the output of the memory cell and what portion of stored information is outputted. o in the figure 2.15.

  The following figure 2.15 shows a graphical depiction of a LSTM:

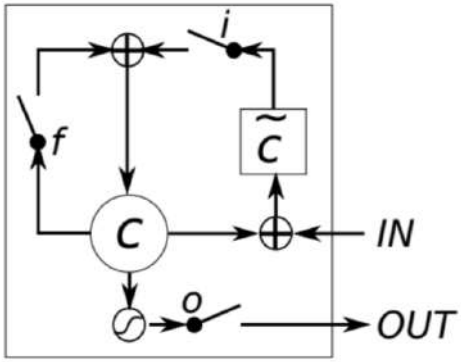

Figure 2.15: Functioning basics of a LSTM Recurrent Neural Network. Source: [29]

In 2.15, i, o and f are, respectively, the input, output and forget gates. c is the memory cell, while $\tilde{c}$ is the new memory cell content.

- **Recurrent Gated Unit:**

  GRU Neural Networks have a somewhat similar implementation to LSTM. The main difference is that in GRU, the input and forget gates are combined (candidate activation) into one single update gate, and the output gate is also dropped (instead, there is only the activation). Everything adds to create a more compact and efficient architecture.

  Two key gates make up a GRU:

  1. Reset gate: Determines how much of the information from the previous state should be forgotten. r in the figure 2.16
  2. Update gate: Controls how much of the previous information is retained and what portion of the new memory should be added. z in the figure 2.16

  The following figure 2.16 shows a graphical depiction of a GRU:

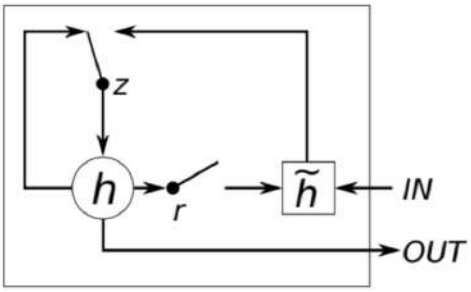

Figure 2.16: Processing blocks of a GRU Recurrent Neural Network. Source: [29]

In 2.16, r and z are the reset gate and update gate, whereas $\tilde{h}$ is the candidate activation and h is the activation itself.

2.7 Metrics

As a conclusion for this chapter, we present in the following the objective quality metrics that will be used in this work to assess speech quality. These serve as a way to evaluate the quality and naturalness of synthesized speech or to compare synthesized target results among them.

2.7.1 Mel Cepstral Distortion

Mel Cepstral Distortion (MCD) is a metric used in order to measure the dissimilarity between two MFCC sets. Its main scope of application is speech

synthesis and speech recognition. To compute the MCD, the original set of MFCC's are compared with the synthesized MFCC's set. The way to compare them is based on distance (the lower the distance, the better). The following equation 2.20 is used to calculate MCD:

$$MCD = \frac{10}{ln(10)} * \sqrt{2 * \sum_{d=1}^{D_a} (mc_d^t - mc_d^s)^2 (dB)} \tag{2.20}$$

Where mc_d^t comprises the original MFCC's set (that is, the target one), and mc_d^s comprises the synthesized set. The sum goes along all of the dimensions of the set. MCD is measured in decibel scale (dB)

2.7.2 Short Term Objective Intelligibility (STOI)

In the scope of speech synthesis and natural language processing, there is a crucial need to evaluate in some way the intelligibility of synthesized audios. Normally, this task would be performed manually, by having several people listen to a set of audios, compare them between original an synthesized and evaluate the quality of the result. These results need to involve a relatively large number of people and a decent dataset in order for them to be statistically significant. The task of evaluating audio quality is inherently subjective due to it being subject to a individual valuation. Moreover, this process is slow and costly, taking into consideration the considerable amount of time needed for the different people to listen to the audios and report their evaluation.

This inherently slow and inefficient nature of manually evaluating audios created the need for an automatic and objective tool. STOI is one of the many objective metrics that tries to objectively evaluate the degree of speech clarity and intelligibility in a noisy environment.

To compute the STOI metric, the linguistic and acoustic features of speech are obtained. The process of calculating STOI is carried out in parallel for two signals: original and modified (not synthesized, is the one where operations are performed). The following four step process is needed to compute [22]:

1. A time-frequency representation of both signals is obtained by segmenting both signals into frames, with a frameshift such that they are 50% overlapped, then a Hanning window is used for processing. A padding is concatenated to each frame (zeros are added) and the Fourier Transform is applied.

2. An analysis is performed for each one-third octave. An octave symbolizes a duplication or split-in-half in frequency. In total, 15 one-third octaves are studied.

3. The intelligibility metric for a time-frequency frame proceeds from a region of multiple consecutive frames. For said frame region, a normalization is needed to make the energy of the 'modified' region match with that of the 'original' region. With that operation, a metric known as Signal to Distortion Ratio (SDR), which is somewhat similar to Signal to Noise Ratio (SNR).

4. The final intelligibility metric is obtained as a linear correlation coefficient estimation between the original frames and the modified frames.

Chapter 3

Proposed methods

As commented in previous sections, the main goal of this project is to achieve synthesis of intelligible speech from PMA biosignals. For that purpose, after extensive literature investigation, multiple solutions are proposed in order to test the performance of each one, but all of them share a common main goal. This proposed methods start from the previous work carried out by the student, where a speech synthesis (Unit Selection) algorithm was created with the aim of synthesizing intelligible speech from PMA biosignals.

The aim of this chapter is to describe the proposed algorithms devised for synthesising audible speech from articulator movement data. The particularities of the use case in this project will also be described in depth.

As stated in previous chapters, the main goal of this work is to achieve intelligible speech from biosignals related with the voice production process. In my previous project, I aimed at synthesizing speech from PMA data by means of well-known, non-parametric algorithm known as Unit Selection in the speech synthesis literature. Linear Regression was also studied as a starting point and used as a baseline model. In the current work, several more advanced methods are developed and tested with the same objective. As some of the new approaches build on the basics of the Unit Selection algorithm, this method will have to be described thoroughly in order to get a proper understanding of the whole algorithm .

Regarding the database used for our work, the best candidate was a biosignal repository obtain by the PMA technique and used in various previous studies [41, 47]. This database contains simultaneous recordings of PMA signals and audible speech for healthy participants, so it is perfectly suited for creating a *data-based* model. All algorithms were coded using Python and its machine learning and signal processing libraries.

After reviewing the database used, the legacy unit selection algorithm for speech synthesis from PMA data will be described in detail. Next, three improvements over the base algorithm will be introduced and justified.

3.1 Datasets for Speech Synthesis

For this investigation, signal obtained in previous studies were used [41] [47]. Voice and PMA biosignals were sampled synchronously in healthy individuals. When it comes to the words/sentences uttered by the individuals, it is convenient to refer to previously designed databases for digit or sentences. The advantage is that they are already created and tested, and they usually come with additional files like a list of uttered digits or the frequency of appearance of each one. It is a plus to use databases that are phonetically balanced, in the sense that all phonemes appear in the database in similar proportions.

With reference to the database used for individuals from the first study [41], the *TiDigit* database was chosen. Each recording consists of a sequence of one to seven digits pronounced in English. The total vocabulary contains 11 words: the digits 'one' to 'nine', plus 'zero' and 'oh' (the latter, an alternative pronunciation for zero). In total, there are 21 phonemes, 11 vowels and 10 consonants.

For each individual, 308 recorded sentences are available. For this study, the available information is comprised of the digits uttered by two male speakers.

As for the database used in the second study [47], patients uttered sentences from a database created by the Carnegie Mellon University (CMU). This database, named *Arctic*, contains a range of phonetically rich sentences that allows the assessment of speech reconstruction over a wide phonetic range. There are 1132 sentences selected from English books. 470 and 510 sentences recorded from two healthy subjects are available in the same way as for the first study. Both were male.

In order to have a reference of the different biosignal/speech recordings that will be used, the following table 3.1 shows the features of each of the datasets used to synthesize speech.

Database	Speaker	Nº of files	Total duration (minutes)
TiDigit	LC	308	8
TiDigit	TP	308	8
Arctic	JG	470	26
Arctic	RM	510	28

Each one of the datasets contains a PMA biosignal and recorded speech, recorded in synchronous manner.

Each of the datasets contains a set of PMA biosignals and synchronously recorded voice audios. In each dataset file, specific digits or sentences are enunciated. To create the database, 90% of the files are used for training, while the remaining 10% is used for speech synthesis. The process is repeated

10 times, since the partition parameter *K-Fold* is $K = 10$ (more specification about this concept in 4).

3.2 Speech Synthesis by Unit Selection

Unit selection synthesis is a classical algorithm that has been widely popular for speech synthesis over the years. It operates on the principle of selecting and concatenating small portions of speech, known as 'units', to construct natural-sounding speech output. These units typically consist of small segments of recorded speech, such as single phonemes of around 100 ms or longer segments like syllables or even words. Unit Selection is an approach that links PMA *units* with voice *units*. Each of these units is small portion of the different files from the database. The files are obtained by synchronously recording biosignal and speech for a healthy individual. The process of linking PMA units with speech units is governed by the calculation of two main metrics: target cost and concatenation cost. This section will be focused in describing in depth the algorithm implemented for the last project carried out by the student. This method will be refered to as 'legacy Unit Selection' as it is the original implementation. It serves as the foundation from which all other methods implemented during this project are built upon.

The following figure 3.1 shows a block diagram of the legacy method:

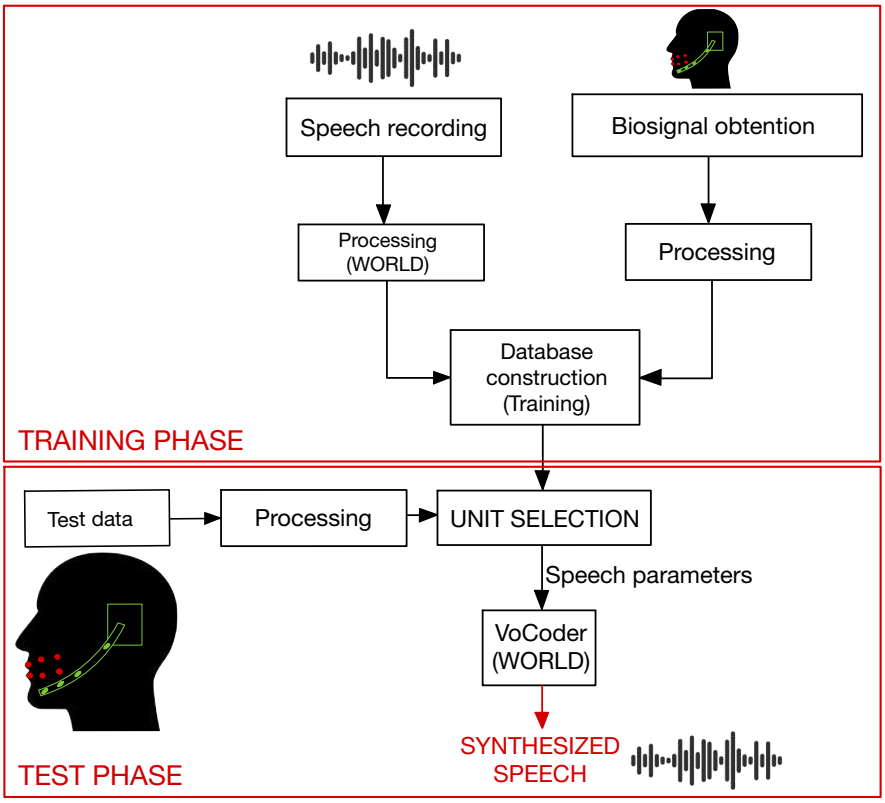

Figure 3.1: Block diagram for legacy Unit Selection

The very first step is to synchronously obtain the voice and PMA biosignal recordings. They are processed and modified in parallel according to the specific configuration of the algorithm to form the training database. this 'legacy Unit Selection' algorithm does not use voice units, it creates MFCC units from the voice recordings. These MFCC's are Mel-Frequency Cepstral Coefficients: Coefficients that collectively represent the short-term power spectrum of a sound. The Unit Selection algorithm uses the information available in the training database to try to predict the speech features (MFCC's) that link to the voice features. Lastly, the WORLD VoCoder will use these MFCC's parameters to synthesize audible speech. The following subsections will centre the attention on the details of each step.

3.2.1 Data collection

This subsection focuses in explaining the nature and the process of collecting the biosignals used, also the preprocessing work. As stated, PMA biosignals are used from past research [41], [47].

Acquisition and processing of PMA biosignals

The biosignal data set used was recorded in healthy individuals and was recorded in two different studies. The figure below 3.2 shows the device that was devised to obtain the PMA signals. This instrument was custom-made to fit this specific use case.

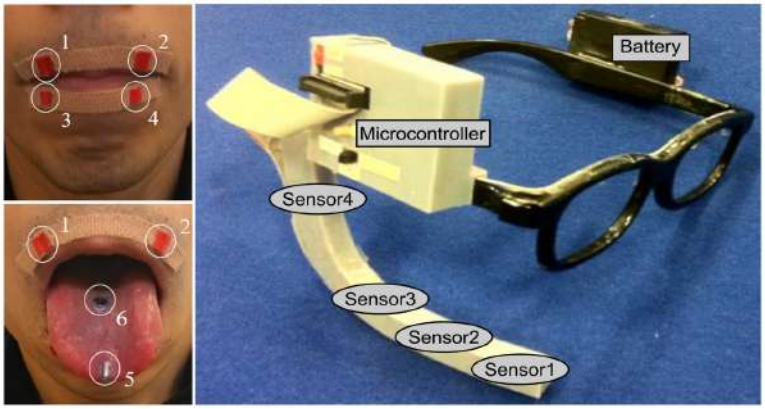

Figure 3.2: General overview of the PMA articular biosignal obtention system. Source: [47]

The left part of figure 3.2 shows the arrangement of the different sensors. For this study, two voice articulators were chosen: lips and tongue. 6 Neodymium-Iron-Boron (NdFeB) of 1 mm diameter were laid out in 6 different positions. The magnets are fixed using a special surgical adhesive that loses adherence within hours after application (permanent fixation of the magnets is possible for patients, special surgical operations are performed for that purpose).

To collect the magnetic fields emitted by the magnets, 4 sensors are disposed in the specified position in the right part of figure 3.2. The exact placement of the sensors depends on the physiognomy of the individual. Sensor 1 to 3 are arranged such that they optimize the reception of the magnetic field coming from speech articulators, while the fourth one is used for measuring the background magnetic field and compensate for the effect of earth's magnetic field and external EM noise. The PMA data is sampled at 100 Hz and are transmitted via a *Bluetooth* transceiver Additionally, a low-pass filter is applied at 50 Hz to avoid the possible electrical noise. All the patients used the same device for capturing the signals, the only custom part of the process is the actual placement of the sensors in each individual.

The recording session is synchronous, where audio and PMA biosignals (dimensionality is 9) are simultaneously obtained, using sampling rates of 16 kHz and 100 Hz, respectively.

To process the audio recordings, the processing will depend on the exact

method used. For legacy unit selection, untreated voice is not used, instead, the information consists on arrays of Mel-Frequency Cepstral Coefficients (MFCC's) obtained from the voice recordings (dimensionality is 25). For computing the MFCC's, WORLD VoCoder is used. The functioning basics of WORLD are detailed in 2.4.2.

To process the PMA signals, the only necessary modification is the subtraction of the earth's magnetic field, as explained previously. For that purpose, each individual is asked to make head movements in a certain way, all while maintaining voice articulators (lips and tongue) static. This way, the earth's magnetic field can be differentiated from the actual useful signal and subtracted.

It has to be noted that the biosignal obtained from PMA method is not a positional display of voice articulators (in the sense of explicitly showing where the articulators are), instead, it is a sum of the different sampled signals (a sum of magnetic fields).

The relationship between this sum of magnetic fields collected by the sensors and the cepstral coefficients of voice follows a nonlinear logic. Additionally, the PMA technique used in the studies does not allow to obtain information on additional speech parameters such as F_0 and *aperiodicity*. The consequence of this is that the synthesized audio would be *unvoiced* (that is, like a whisper). For simplicity, the decision was made to maintain this parameters from the original recorded audios, leaving as a possible future evolution the procurement of this parameters using other methods.

Processing of collected signals is identical both articles [41], [47]. The sampling rate also stays inaltered through both studies.

3.2.2 Speech Synthesis algorithm using biosignals

This section will be dedicated to specifying the Unit-Selection method created, including the creation of the units database used for synthesis. The algorithm is based on the idea that, for the training phase, the available information can be divided into multiple small portions, forming pairs of PMA-MFCC. There is, thus, a univocal relationship between these pairs of units. In the test phase (where synthesis happens) the only available information are PMA biosignals, which are again divided into small portions of the same size as the training database. The closest match for each unit in the test phase is searched for in the PMA database, where there is one only MFCC unit related to it. This MFCC unit is then associated as the closest candidate for the test unit. This process is repeated for every unit to predict the sequence of MFCC units that corresponds with the PMA data. Distance evaluation methods are used, and not only a single metric is considered. Further description will build on this further on.

Training phase

Once the processed PMA signals and the voice MFCC's are available, the training data base is constructed. As stated, it is composed of a big collection of pairs of PMA-MFCC units. The creation of each type of unit is performed separately, as they are different in nature.

Data base creation For each of the two sets that make up the data base, the following list details the process needed for each of them:

- **MFCC's:** Once the audio signals are passed through WORLD VoCoder, the voice MFCC's are obtained with a analysis frame size of 5 or 10 ms (the two values accepted as input when executing WORLD). The standard value used for this project will be 10 ms. One important thing to take into consideration is that this parameter must be in accordance with the sampling frequency of the PMA biosignals, such that the samples and units represent the same time interval. As explained before, they were obtained synchronously and the very functioning on Unit Selection relies in this synchronism. For example, a 100 Hz sampling frequency corresponds with a 10 ms frame size (hence, this will be the configuration). Each array that the WORLD VoCoder outputs consists of the Units that we need, so there is no further processing needed.
- **PMA:** For creating the PMA units of the data base, a function was created that performs the conversion of each individual file to units. This functions takes as input parameters a file that contains PMA biosignals, its associated sampling frequency, the size of the window and the frameshift. What here is called frameshift is the parameter that has to match with the analysis frame size from WORLD VoCoder. The units obtained from the file are short portions of itself (according to the window size) and successively shifted. The following figure 3.3 shows the process of unit creation from the PMA files.

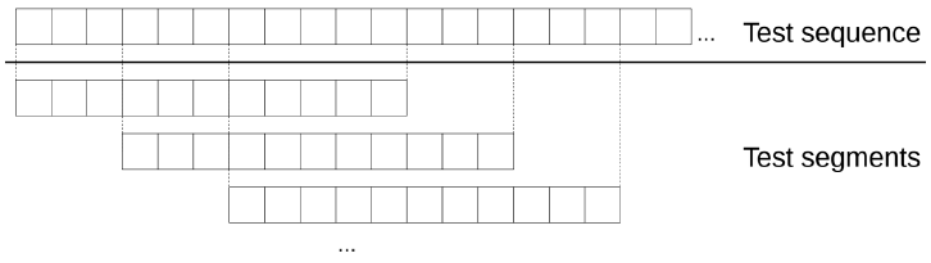

Figure 3.3: Creation of units from PMA files. Source: [33]

The upper section *Test Sequence* conforms the original PMA file, while the inferior part *Test Sequence* is composed by the different individual units. The window (unit) size can be seen noted as w_u, as well as the frameshift, noted as s_u. In the example figure 3.3 the window size is $w_u = 11$ samples and the frameshift $s_u = 3$ samples. As disclosed, s_u is pre-given by the WORLD VoCoder, but w_u is a parameter that can be modified and that will affect the final result of the speech synthesis. The consequence of widening the window size w_u is that each unit has a larger amount of information (this results in greater temporal context). Greater temporal information can be benefitial for the quality of the final result, although up to a certain point. The modification of this parameter was evaluated and put into context in the last investigation performed by the student.

Once all the units are obtained from a file, the process is repeated for all the files that make up the training and test sequence.

Unit normalization For the distance metrics to work correctly, it is highly advisable to normalize all the parameters that we are working with. This normalization process has a great importance in these type of speech synthesis algorithms for two main reasons:

- Differences in scale: The various values inside of a unit can be very different and uneven. This disparity can misrepresent the calculation when evaluating distances (sometimes, big distance values coexist with small distance values, due to the features having a wide range of possible values. Reducing this range (normalization) results in more robust distance calculation.

- Difference between PMA and MFCC ranges: For distance calculation, distances evaluated among PMA signals and MFCC signals get mixed. Because they are incorporated into the same expression, a normalization prevents one of the distances having an abnormal prevalence over one another.

For this algorithm implementation, the decision was made to normalize the range of every unit (both PMA and MFCC) into the [0,1] range, using the python library from *sklearn*, *MinMaxScaler*. The library helps transform the units by means of a scaling to the specified range.

To perform the normalization, first the training database (which is much larger than the test one) is used to perform the scaling based on its statistical properties. For the test units, the exact same scaling is performed (based in the properties of the training database). This can be done because it is safe to assume that the properties from the bigger set apply also to the

smaller one. Each parameter is scaled individually, such that they correspond with the limits for the dataset. This transformation is typically used as an alternative to *null mean, unit variance* scaling.

Speech Synthesis

This section will focus in detailing the behaviour of the Unit Selection created for speech synthesis. The following URL [1] links to a *Github* repository where the main scripts of the legacy algorithm are implemented.

Unit selection has been the standard for a long time for synthesizing speech from biosignals, since it is capable of synthesizing audible voice with a workable quality, all while keeping the algorithm computationally lighter than many of the more powerful counterparts. Algorithm complexity often gets in the way of achieving real time capabilities, key for real world, commercial implementations.

Unit Selection is based on the premise that natural sounding utterances by arranging small units (portions of utterances) obtained from a real voice data base. The following figure 3.4 illustrates the process:

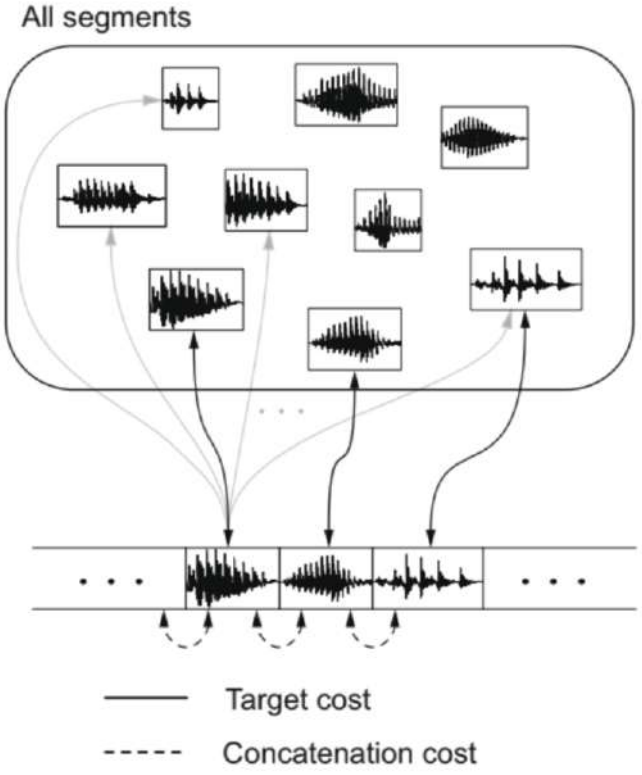

Figure 3.4: Unit Selection visual overview. Source: [34]

[1] https://github.com/javilobato/speech-synthesis.git

The main metric used for selecting the optimal unit from the database is the *Target Cost*. It indicates the level of similarity between the unit being assessed and the corresponding unit from the database. The result derived from using only this metric may not be optimal, because it fails to tackle the similarities between adjacent units. That is why there is an additional cost in the implemented algorithm: *Concatenation Cost*. This added metric measures how two adjacent units are compared. The effect is that abrupt transitions are penalised, while smooth, natural sounding transitions are promoted. *Target Cost* is defined in 3.1

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) \tag{3.1}$$

Where j is iterated along all the components that make up the unit in question, t stands for *Target* and u stands for the specific unit in the database. w is a weight that can be used to prioritise any of the two weights. *Concatenation Cost* is defined in 3.2.

$$C^c(u_{i-1}, u_i) = \sum_{k=1}^{q} w_k^c C_k^c(u_{i-1}, u_i) \tag{3.2}$$

Where k iterates along the components of the unit being assessed. There is also a weight w to establish priorities.

Both costs shall be optimized in order to obtain the unit sequence $u_{1:n} = u_1, ..., u_n$ from the database that optimizes total cost, such that 3.3:

$$\hat{u}_{1:n} = argmin_{1:n} C(t_{1:n}, u_{1:n}) \tag{3.3}$$

Where, $C(t_{1:n}, u_{1:n})$ notes the total cost function. The result of the upper equation 3.3 shape the *Viterbi Path* in the sense that, for each file, indicates the most likely sequence of units. The total cost function is described in the following equation 3.4

$$C(t_{1:n}, u_{1:n}) = \sum_{i=1}^{n} C^{(t)}(t_i, u_i) + \sum_{i=2}^{n} C^{(c)}(u_{i-1}, u_i) \tag{3.4}$$

The task of choosing the weights is arbitrary and depends on the exact case of study. There is not a fixed criteria to determine them. The window length is also important, as commented before. The bigger the unit size, the larger the database needed to cover the whole domain (thus, higher computational load). Smaller units can offer better performance in given situation, as they offer more potential joining points.

Following the workflow of the Unit Selection Algorithm, once finished the process of creating the database, the results for the training phase are two arrays of units (PMA units and MFCC units) and only PMA units for

the test phase. The corresponding predicted MFCC units for the test phase will be calculated by the Unit Selection Algorithm.

Distance assessment (Target Cost) Equation 3.1 implements the distance assessment. This metric of resemblance origins from the Euclidean distance between to arrays, represented in the following equation 3.5.

$$C^t = \sum_{k=1}^{w_u} \sqrt{\sum_{d=1}^{D_s} (s^t_{test}(k,d) - s^t_{train}(k,d))^2}, \tag{3.5}$$

Where k iterated along all the frames that make up the unit and d iterates along each dimension of the PMA frame. D_s denotes the PMA signal dimensionality (s is for source). The term *train* refers to the number of training units in the database.

The calculation of the *Target Cost* in 3.5 in the test phase can be a computationally heavy process: For each test unit, distance has to be measured for each and every unit that compose the training database. This is why, with the aim of optimizing the calculation, the decision was made to use a tree structure for partitioning the database, allowing for a much faster and efficient distance evaluation. More in depth, the chosen structure is known as *Ball Tree* and allows to arrange data arrays in a multidimensional space. The search for the closest neighbor is more efficient in this kind of structure [37]. The closest neighbour is the candidate unit that is sought after when minimising distance. As illustrated in figure 3.5, the algorithm constructs a hierarchy tree that originates from a point cloud that contains the database elements. The tree is constructed by building *spheres*: First, a random point is chosen as the center of a sphere. This sphere has a given size (*leaf size*). The size of the sphere determines the speed in the search process and the time it takes to create the structure, but the final result stays the same. Next, the points are divided into two groups: those inside the sphere and those outside of it. In the following step, two points are chosen: one inside and one outside of the sphere. this process is repeated until a termination criteria is met (for example, a given sphere size). For the implemented structure, the N closest neighbours of a given unit can be efficiently calculated. Evaluating only the N closest neighbours allows to change the limits in equation 3.4, the resulting equation would be as shown below 3.6:

$$C(t_{1:N}, u_{1:N}) = \sum_{i=1}^{N} C^{(t)}(t_i, u_i) + \sum_{i=2}^{N} C^{(c)}(u_{i-1}, u_i) \tag{3.6}$$

For this algorithm, a *python* package provided in the *scikit-learn* library: *BallTree*

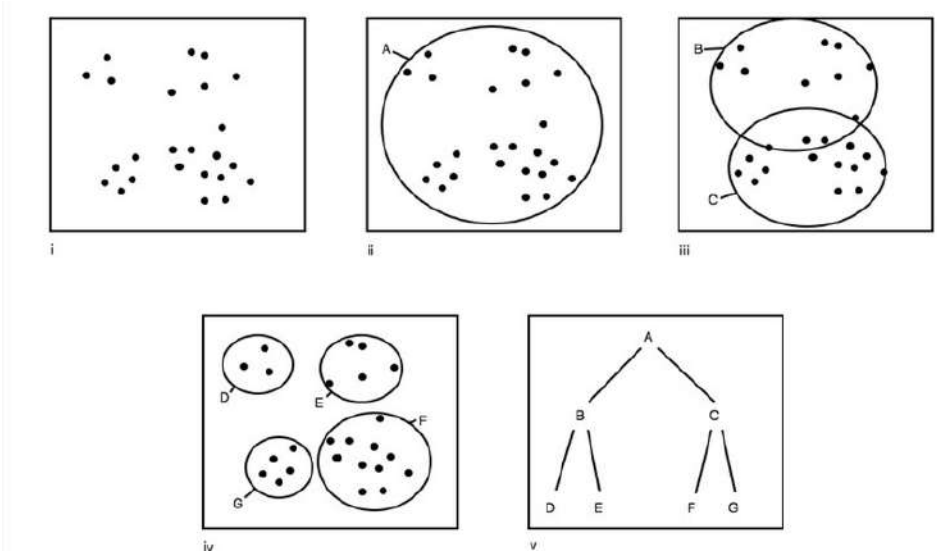

Figure 3.5: Tree construction in *Ball Tree method*. Source: [50]

Concatenation Cost For computing the *Concatenation Cost* metric, cepstral distance is used. For evaluating the smoothness of the transition, similarity between adjacent MFCC's units is obtained (distance among PMA units is used for *Target Cost*). Cepstral Distance can be calculated as the euclidean distance between two MFCC units, as shown below 3.7

$$C_c(t_{1:N}) = \sum_{k=1}^{w_{u(MFCC)}} \sqrt{\sum_{d=1}^{DA} (t_{train}^{t+1}(k,d) - t_{train}^{t}(k,d))^2} \tag{3.7}$$

Where k is iterated along the MFCC unit size and d is iterated along the unit features (25 for MFCC's)

Incorporating metrics When evaluating the *Target Cost* in one test unit, the nearest N units are drawn from the training data base. Choosing N units instead of only one is preferred because when incorporating bot target and concatenation cost, the best candidate unit can be one that is not the nearest neighbour. For example, it is common to find units that are the closest candidate in terms of target cost but have a very abrupt transition from one MFCC unit to another. This improved approach allows for various candidates by evaluating target cost and then the concatenation cost is incorporated for the N candidates. The lowest total cost will mean that the unit is the closest and smoothest-transitioned option possible inside the training database. The weight w includes a multiplying factor to the concatenation cost that can adjust its relevance when determining the optimal unit.

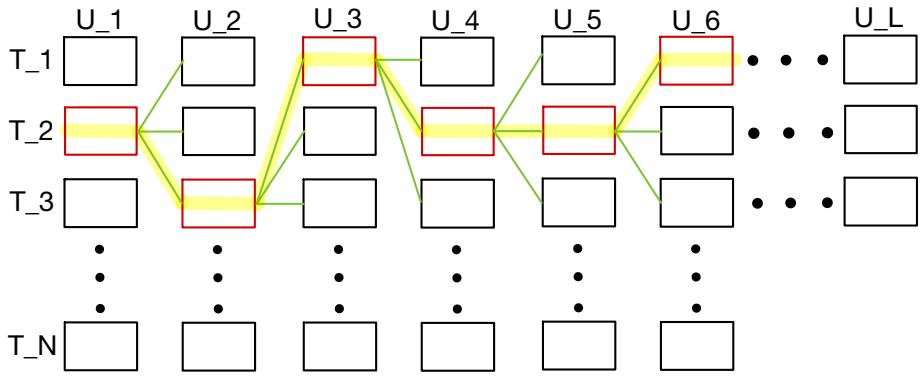

Figure 3.6: Optimal unit search process.

The process of unveiling the optimal units is performed in a cost matrix that is progressively filled with the various costs. The immediately preceding MFCC's units are taken into consideration for computing the concatenation cost of each frame. The very nature of the algorithm's design is programmed such that when all the test units are evaluated and costs computed, the MFCC units array that minimizes the overall cost function is already available.

The lowest cost path is determined as the cost matrix is filled. In the first iteration, the optimal unit is the one with lowest target cost (there is no preceding unit to compute any concatenation cost). For the second unit, target cost is computed for each of the N neighbours and concatenation cost is calculated using the anterior MFCC unit. Final result is the optimal unit (minizes both costs). this process is repeated iteratively to build the optimal path. This approach for obtaining optimal MFCC's is known as *backtracking*.

It should be noted that the cost minimization process can be done by using Viterbi algorithm. As stated in previous sections, the final output of the algorithm is formed by the sequence of units known as the Viterbi path, understood as the unit sequence with lowest associated costs.

Figure 3.6 illustrates the process of choosing the optimal unit each frame:

Figure 3.6 shows a series of candidate units. For each PMA unit there are N MFCC candidate units (arrange in the vertical axis of the figure). These units are in descending order according to the target cost. For the first unit, only the target cost is used (appears red in the figure). The green lines indicate that there is a concatenation cost calculated for that connection. This green lines always go from an optimal unit (in red) and the following N candidate units. Each new optimal unit is determined considering both costs. This process is repeated until all L units are evaluated. The optimal path (Viterbi Path) is shown in yellow in the figure.

Once the process has come to an end, we are left with a sequence of MFCC units (as many as PMA units). This sequence of concatenated MFCC coefficients is then fed to the WORLD Vocoder, specified in section 2.4.2.

As an example, figure 3.7 shows the temporal representation of an audio synthesized by Unit Selection, as well as the associated spectrogram. The signal came from the data base mentioned in 3.2.1.

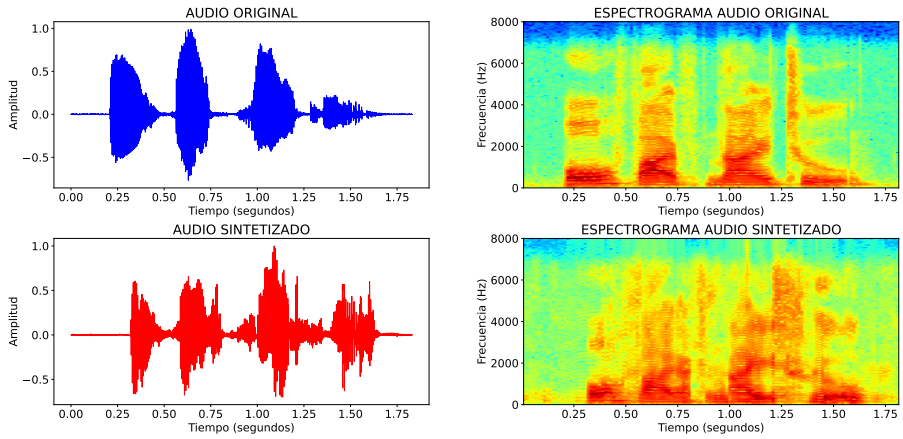

Figure 3.7: Example of an audio synthesized by means of Unit Selection. Temporal (left) and Spectrogram (right)

Example in 3.7 corresponds to a series of uttered digits, where the following sequence is pronounced in english: *zero five nine two*.

3.2.3 Algorithm Advantages

Unit Selection was chosen over other methods because it has a series of advantages over the rest of implementations collected in 2 that make it appropriate to use in this specific circumstances. They are listed below:

- As commented in previous sections, the relationship between biosignal parameters and voice parameters is almost always of a nonlinear nature. Unit Selection is not based in any linear model to treat data (temporally-wise), so the results are better than those of more simple, linear techniques such as linear regression.

- The Unit Selection algorithm implemented is non-parametric: it uses real MFCC coefficients from individuals speaking. This is in contrast to parametric models, that synthesize voice by mathematical and statistical models. Parametric models typically require large data bases for obtaining high-quality results. Unit Selection can achieve good quality results with relatively small sized data bases.

- Smaller datasets for implementing the algorithm means that the data base is smaller, easier and faster to work with. Computational effort is also lower compared to other methods, allowing its use in real time applications.

- As the synthesized audio is composed of small, concatenated portions of real audio, voice distortion is lower compared to other methods that synthesize voice by using parametric methods.

3.3 Dimensionality Reduction using Canonical Correlation Analysis (CCA)

Speech synthesis algorithms are synonymous to processing large amounts of data: training corpuses include hundreds if not thousands of different spoken digits or sentences that must be processed for their use in synthesis algorithms. In previous sections, the relevance of computational demand has been mentioned. Faster approaches that can achieve speech synthesis should be sought after for various reasons:

- Less data to process reduces energy consumption of algorithms. This is always positive, but it is specially the case for battery-powered applications (i.e a portable speech synthesizer) where autonomy is key.

- Less demanding algorithms take less time to train and execute. This speeds up the design and test of the algorithm (which is of great relevance in the investigation process) and also allow for real-time processing, ideal for real world implementations.

- Naturally, larger data corpuses tend to offer better results than smaller ones. With ever-increasing data availability, maintaining performance requirements at a reasonable level is key for successful deployments.

The conclusion was that it would be of interest to test a method that reduced the number of dimensions (features) of the units in the dataset, while maintaining a similar level of performance. For that purpose, there are mainly two methods: Principal Component Analysis (PCA) and CCA:

- PCA: This method consists in transforming one set of probably correlated variables into a smaller one composed of non correlated variables. Using the information provided by the variance, the dispersion in de data can be measured [9]. PCA tries to retain the highest amount of variance possible with the least amount of variables. This method is applied to one set of variables at a time.

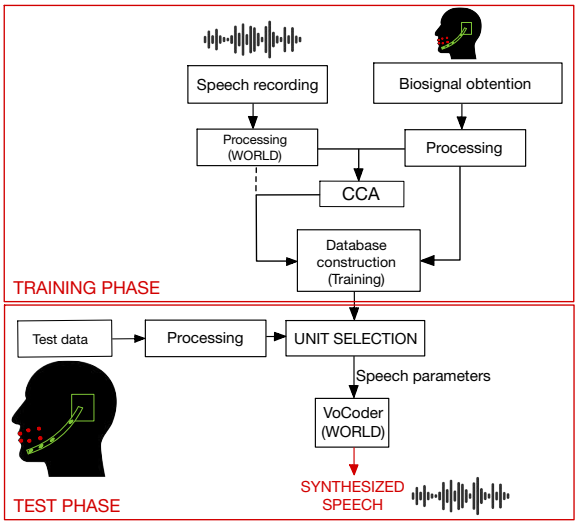

Figure 3.8: Illustration of Canonical Correlation Analysis Algorithm for speech synthesis

- CCA: It has some resemblance with PCA. This technique is used to identify linear relationships between two variables. Its task it's to find linear combinations of variables in each dataset that maximize the correlation between the two. This method is very useful when the need is to reduce the dimensionality of two groups of data while maintaining the relationship among them [61]. This is exactly what we are looking for in this investigation, so CCA is the best fit for our needs.

Intuitively, the algorithm works as illustrated in the following figure 3.8. For dimensionality reduction calculation, both sets of variables are used, however, it is only applied to PMA units. We need that the MFCC units stay intact, otherwise the information would not be useful for the VoCoder for it to be turned into audible speech.

CCA works with two sets of variables at a time X, Y, with p and q independent and dependent variables, respectively. The idea is to find two different variables, V_i and U_i in such manner that the correlation between both of them is maximized. The only functions chosen are the ones that best express correlation between the two sets. These linear functions are called *canonical variables*, while the correlations are called *canonical correlations*. The following figure 3.9 illustrates this concept.

The expression that CCA seeks is one a linear combination that can be expressed using weights, as shown in equation 3.8:

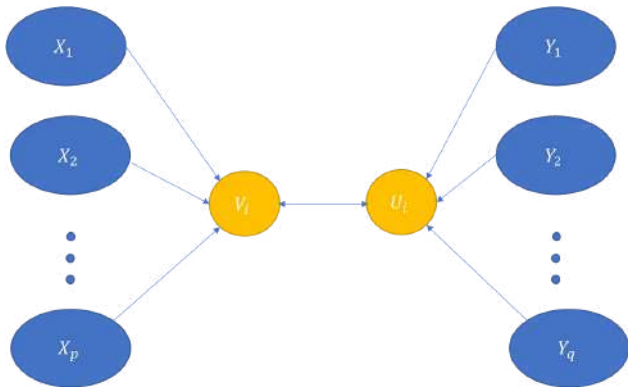

Figure 3.9: Illustration of Canonical Correlation Analysis. Source: *Medium*

$$\begin{aligned} U_i &= a_1 Y_1 + a_2 Y_2 + \cdots + a_q Y_q \\ V_i &= b_1 X_1 + b_2 X_2 + \cdots + b_p X_p \end{aligned} \tag{3.8}$$

The correlation between the variables U and V can be calculated. The correlations are proved to follow the equation below 3.9:

$$\begin{aligned} P_X &= X(X^T X)^{-1} X^T \\ P_Y &= Y(Y^T Y)^{-1} Y^T \end{aligned} \tag{3.9}$$

The eigenvalue and eigenvectors of the correlations show the canonical correlation and canonical variables U and V. The first canonical variables are the most important ones: they are the ones with the greatest level of correlation. When dimensionality reduction is the objective, a number of canonical variables are chosen, being this number lower than the minimum number of features of the two original variables: $n \leq min(p, q)$. For the current investigation, CCA will be included into a version of Unit Selection and a different number of eliminated dimensions will be tested and compared against legacy Unit Selection.

In python there are multiple packages that allow to implement CCA. In this case, the decision was made to go with *sklearn*'s package for cross decomposition called *CCA*. This package is of special interest because not only allows us to perform dimensionality reduction on our dataset, it also includes functionalities to test linear regression using CCA, which can be seen as a multiple regression method. This can be later compared with the legacy method of linear regression.

3.3.1 Algorithm Advantages

The main advantages of improving the original Unit Selection implementation by using CCA is that the variables managed by the algorithm will

be of a lower complexity, making it possible to improve processing speeds. The results will depend on the number of dimensions eliminated from the variables. Additionally, eliminating information from PMA data that does is not related with speech itself can result in an improvement of the results, because there is less 'noise' in the data (understanding noise as the part of the signal that is not useful for the algorithm's objective).

3.4 Direct Speech synthesis

One of the conclusions that was drawn out from previous investigation on speech synthesis algorithms was that predicting MFCC's instead of voice directly was an intermediate step that increased code complexity, processing power needed and execution time. The usage of MFCC parameters is helpful because it works in a different domain than voice, and general features from a portion of audio can be extracted and represented in a compact way. The fact that the algorithm was a two step process with the need of a VoCoder and the results obtained in previous investigation (results did not vary much when changing algorithm configuration) sparked the curiosity to try a different method of synthesizing speech. The conclusion was that a Direct Speech Synthesis, data-based algorithm was to be implemented. The aim of this implementation is to skip a step in the legacy Unit Selection and directly obtain voice right out of the algorithm's prediction. This approach is one of the many available for speech synthesis and many studies have achieved intelligible voice with this technique [57], [63] . Figure 3.10 shows an graphical overview of the solution:

In this algorithm, the audio recordings are directly converted into units to form the database. There is no need whatsoever to use a Vocoder, or to take into consideration further voice parameters such as Fundamental Frequency (F0), aperiodicity, etc. For the conversion into units, the same approach as in legacy Unit Selection is used 3.2.2. Taking into consideration the sampling frequency of each kind of signal, the creator function adapts in order for the units to represent the same time interval. In the same fashion as in the case of the PMA signals, the units are also normalized for the reasons commented in previous sections.

Another key aspect is that the distance calculation changes from what we had in the previous Unit Selection method. While the target cost stays unchanged, the same cannot be said for concatenation cost, which now has to be calculated using audio sections instead of MFCC's. In the legacy Unit Selection, the whole MFCC unit was used to compute the concatenation cost. The reason behind this was that a single set of MFCC's is obtained for a given amount of time, meaning they are the same for the whole section, so there is no way to only get the MFCC's of the 'final' part of the unit. With the novel approach, it is now possible to compute concatenation cost

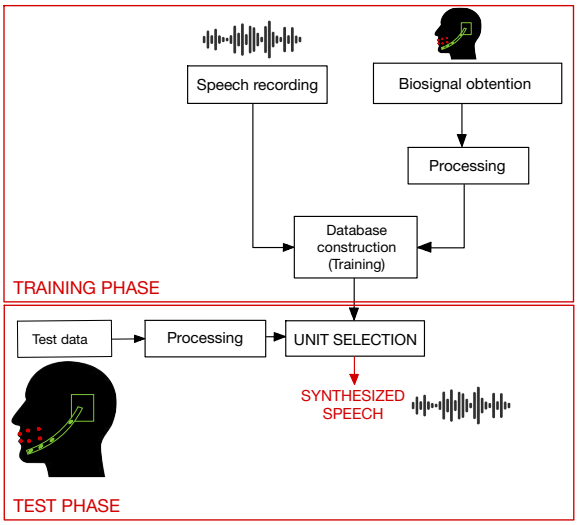

Figure 3.10: Illustration of Direct Speech Synthesis Algorithm

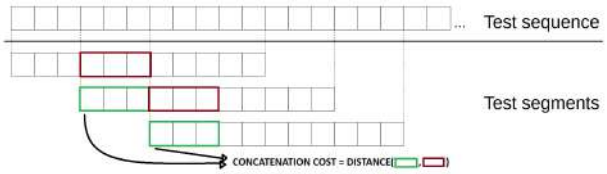

Figure 3.11: Illustration of Concatenation Cost calculation in Direct Speech synthesis

using only the overlapping frames of the audios, thus improving precision. Figure 3.11 illustrates this concept. The calculated metric is the Euclidean Distance, the same as in the target cost, explained in 3.5.

In legacy Unit Selection, the final set of units was created by just concatenating every optimal MFCC unit with no need to overlap them. This was the correct way for the VoCoder to work. Now, for the novel DSS approach, the final result is constructed as a continuous sum of concatenated audio sections, as shown in 3.11. This process is known as *overlapp-and-add*, as multiple overlapping units are added to create a single, intelligible voice sequence. Depending on the chosen frameshift, multiple speech units may overlap in given frames, likely interfering with one another and worsening the final result. For that, a smoothing function was included, implementing a weight function that was first proposed in [27].

For the smoothing function, a variable n is declared as the number of units that are overlapped on a given frame. For example, in 3.12, the marked

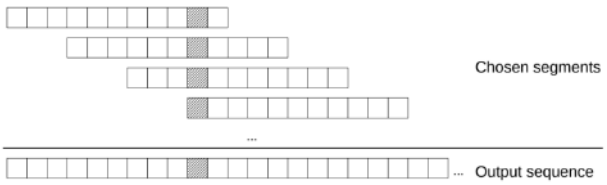

Figure 3.12: Illustration of shared frame in direct speech synthesis. Source: [33]

shared frame has an overlap such that $n = 4$. As said, this number depends on the frameshift chosen.

The weight for each unit's frame is calculated as stated in equation 3.10:

$$w[i] = \frac{\exp(-0.2 \cdot a[i])}{\hat{w}}, \quad i = 1 \dots n \tag{3.10}$$

The term $a[i]$ can be found in following equation:

$$\text{a[i]} = \begin{cases} \left[\frac{n}{2}, \frac{n}{2} - 1, \dots, 1, 1, \dots, \frac{n}{2} - 1, \frac{n}{2}\right], & \text{if } n \text{ is even} \\ \left[\frac{n}{2}\right], \left[\frac{n}{2}\right] - 1, \dots, 1, \dots, \left[\frac{n}{2}\right] - 1, \left[\frac{n}{2}\right], & \text{if } n \text{ is odd} \end{cases} \tag{3.11}$$

Where $\hat{w} = \sum_{i=1}^{n} exp(-0.2)a[i]$ normalizes the weights such that their sum equals 1.

In the previous Unit Selection method, frameshift was given as a constant by WORLD Vocoder, but this is no longer the case with Direct Speech Synthesis. The frameshift can be arbitrarily changed, adapting to different needs and affecting the final result. This is a very clear improvement over the original version, because now units can be overlapped in such a way that one unit represents a single phoneme. This is of special interest in Unit Selection: If the database that is being used contains single phonemes, they can be arranged as preferred to synthesize multiple type of words that did not need to mentioned *per se*, so new words can be synthesized that did not appear in the training audios. Figure 3.13 shows how this works.

3.4.1 Algorithm Advantages

The following are some of the key benefits of our DSS approach.

- One of the steps of legacy Unit Selection is avoided. Saving time and improving computational efficiency.

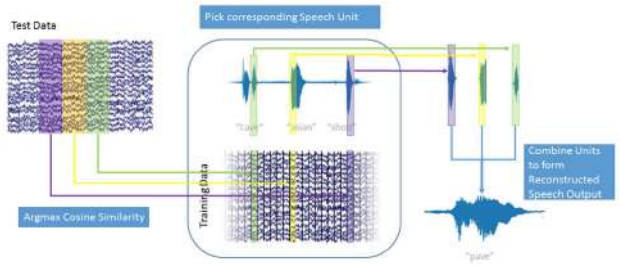

Figure 3.13: Illustration of phoneme-concatenating by Unit Selection. Source: [57]

- Frameshift can be arbitrarily varied, improving flexibility and freedom when synthesizing.

- Algorithm can be tuned to fill the database with single phonemes, allowing for good quality synthesized speech and synthesis of not-seen-before words.

3.5 Neural Network Speech Synthesis

In recent years, Neural Networks have experienced an impressive comeback, completely transforming the Artificial Intelligence landscape. As stated in previous sections 2.6.2, computational hardware advances, new efficient algorithms and access to huge amounts of data have made this possible. Numerous Neural Networks applications can be found on the topic of Speech Synthesis [38], [39], [58]. This field of research is ever-increasing in relevance due to the expectation that it helps solve the most complex challenges of speech synthesis. One of the conclusions drawn out from the previous investigation on the topic was that it would be of great interest to implement a speech synthesis system algorithm based in Neural Networks that could predict voice from the available PMA biosignals. One of the limiting factors of Neural Networks is that the computational cost of training and testing the algorithm is very high (and quite time consuming), so the idea was to design two simple Neural Networks that could prove the feasibility of intelligible speech synthesis using Neural Networks and PMA biosignals: one DNN (as a base model) and one GRU (as a more advanced, appropriate method for the specific case of use). Recurrent Neural Networks (mainly LSTM and GRU) have the capability of retaining long term context information from previous frames, which is a feature to look for in a speech synthesis algorithm, as utterances are very much context-dependent.

Figure 3.14 illustrates how a neural network based speech synthesis sytem works. The scheme is the same for a DNN and a GRU

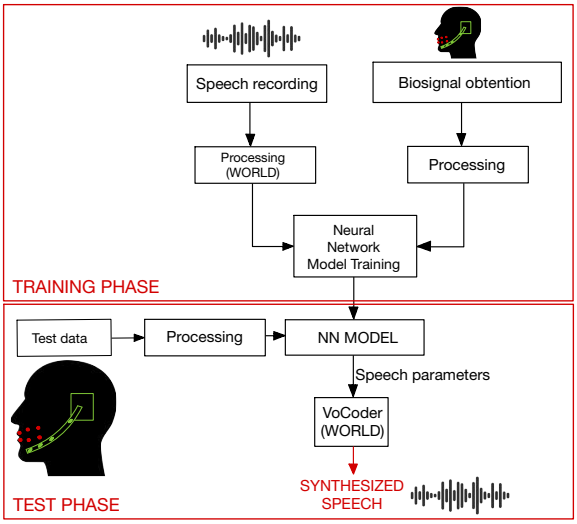

Figure 3.14: Illustration of Speech Synthesis by means of a Neural Network

The intention is to use the Neural Networks to synthesize MFCC's that represent voice, so the WORLD VoCoder is still present. The algorithm works by linking each PMA unit (9 dimensions) frame with the most suitable MFCC unit (25 dimensions). In this case, the synthesized MFCC's are not directly drawn out from a database, instead they are synthesized from the information that the Neural Network has learned. This improves flexibility with respect to legacy Unit Selection (which used units of real MFCC's), but on the other hand, the system has to perform correctly for the synthesized voice to sound natural (as Unit Selection used real recorded information, naturalness was much less o a problem). The algorithm is divided into two steps: First, there is a training phase, where PMA frames and its corresponding true MFCC frames are available. The Neural Network is trained by trying to predict the MFCC and then gradually correct its errors through *backpropagation* (see 2.6.2). Each time the algorithm passes through the whole dataset is called an *epoch*. We determine a given number of them for the training phase. This number cannot be too high in order to avoid *overfitting* (the model learns well the training data but fails to address new data). The learning rate also has to be adjusted. For this investigation, it is kept constant at $l_r = 0.001$. We have to avoid excessive complexity when dimensioning the Neural Networks for the following reasons:

- Training time grows as Net complexity does. The available hardware is limited, so complexity is to be kept at a low level to avoid excessive compilation times.
- The databases used do not have a great size. In neural networks,

typically a big database is needed in order to get high quality results. The digit dataset contains 308 elements (to be divided between train and test), so it makes no sense to create a very complex Neural Network to later feed it with a relatively small dataset.

- It is possible to achieve, intelligible voice using a Neural Network of medium complexity. This is going to be the objective of this part of the project.

Considering all the premises, and evaluating different levels of complexity in the Neural Networks, the decision was made to implement a Neural Network with the following features:

- 4 hidden, fully connected layers. Good compromise between complexity and results.
- MSE loss function. Suitable function for speech synthesis.
- Normalization of parameters to optimize the Network performance.
- ReLu activation function.
- ADAM type optimizer.

There are multiple options to implement Neural Networks in Python, we chose to use the *PyTorch* library, which includes functionalities for both DNN's and GRU's. Its modular design allows for an easy implementation.

As stated previously, the input parameters (PMA) have dimension 9 (64 in figure 3.15 as a unit is composed in this case of 6 PMA frames), while the output parameters have dimension 25 (MFCC). Figure 3.15 shows a graphical example of the Neural Networks that are going to be implemented. Depending on the type of Neural Network (DNN or GRU) the neurons will be standard ones or GRU ones (with memory).

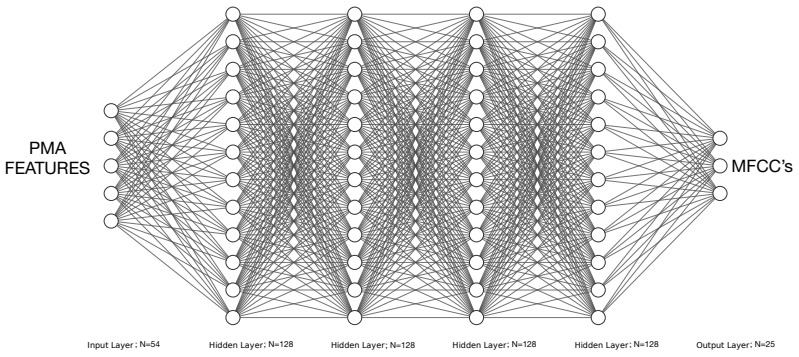

Figure 3.15: Illustration of implemented Neural Network

In the same fashion as in legacy Unit Selection, once the Neural Network has predicted the MFCC units, the information is passed to the WORLD VoCoder, where it is transformed into audible voice.

3.5.1 Algorithm advantages

- Neural Networks are a powerful tool for speech synthesis, as they can adapt to the latent nonlinearities in the relationship between biosignals and voice.
- Generalization. A well trained Neural Network should be able to adapt to new situations not seen before (such as new words or phrases).
- Results can be improved by increasing the number of hidden layers and tuning of other parameters, along with using bigger datasets as soon as they are available.
- GRU Neural Networks make possible to implement a solution that takes into consideration long-term dependencies, which is ideal for speech synthesis applications, where parts of the utterances may be highly correlated.
- The modular nature of the implementation makes the creation of the Neural Network a practical task and allows for real time modifications.

Chapter 4

Obtained Results

This chapter will be focused in presenting the experimental results obtained after evaluating the different proposed methods for synthesizing speech from biosignals 3. Objective metrics for quality and intelligibility will be used for evaluation, also using figures for better understanding. With the main goal of having a reference to compare against the different algorithms, two base methods were devised for this project: Standard Linear Regression (developed in the student's previous project) and Linear Regression via CCA. There will be one section dedicated to each dataset, where the results for each different implemented method will be arranged in subsections.

4.0.1 Evaluated Methods

As stated in previous sections, multiple different algorithms were implemented in this project. We also have available the information from the previous study on the topic, which are called 'legacy' methods. Its results will also be shown as a reference.

For the results presentation, the calculated metrics and the different parameters that are evaluated will depend on the specific algorithm created, as the implementation is different in each case. As for available metrics, we have MCD, which evaluates similarity among original MFCC units and synthesized ones. We also have STOI, which is an objective intelligibility measure for audios. In depth descriptions can be found in 2.7. In 4.4, the following results will be discussed, in order to perform an analysis as a whole.

Below we can find a breakdown of each different algorithm:

- Legacy Linear Regression: The only parameter that can be tuned in Legacy Linear Regression is the length of the Unit. In terms of metrics, as we are working with MFCC's, both STOI and MCD will be computed.

- Legacy Unit Selection: There are 2 parameters that can be tuned:

Unit Length and Concatenation Cost Weight. As for metrics, STOI and MCD will be computed.

- CCA Linear Regression: The case for CCA Linear Regression is the same as for the Legacy implementation: STOI and MCD will be computed.

- CCA Unit Selection: The case for CCA Unit Selection is the same as for the legacy one, both in terms of parameters and metrics computed.

- DSS: As stated in the dedicated section 3.4, this novel implementation allow for a new parameter to be tuned: unit shift. Thus, in total 3 parameters will be tested: Unit Length, Unit shift and Concatenation Cost Weight. As for metrics, only STOI can be computed, as the VoCoder part of the algorithm is eliminated in this approach, so there are no MFCC's to perform measures on.

- DNN and GRU: Taking into consideration that the nature of Neural Networks is a highly time consuming task in the available hardware, paired to the fact that the objective is to test feasibility of the implementation, the focus will be in finding a configuration that achieves a good result and then testing this configuration in both a DNN and a GRU. In terms of the metrics, both STOI and MCD will be computed.

The presentation of results will be divided into various sections, where parameters that affect final results will be modified and tested in suitable algorithms. As a disclaimer, even though all designed algorithms were tested in all datasets available, in order to avoid redundancy, the results will only be shown for one of each datasets (one for digits and one for full sentences). The results are almost identical in the two digits datasets (TiDigit - LC and TiDigit - TP), the same happens for the sentences datasets (Arctic - RM and Arctic - JG), so it does not make sense to show redundant information twice, as the conclusions drawn will be the same for both.

4.0.2 Data Processing

As stated, the information from the databases is comprised of PMA biosignals and voice recordings. The processing that is applied to each one of them is shown below:

- **PMA Biosignals**: As specified in 3.2.1, the biosignals used for the algorithm endure a processing that eliminates the earth's magnetic field by means of a reference sensor, thus eliminating possible external influences.

- **Voice Recordings**: As stated in 3.2.2, the voice audios are processed using the WORLD VoCoder, whose functioning is described in 2.4.2. This way, the MFCC's are obtained, which are then used to make up the units used by the algorithm. In DSS, voice is used untreated.

4.0.3 Evaluation and Metrics

In the investigation carried out, the partitioning of the data folder of each patient is carried out following the Cross Validation (*K-Fold*) philosophy. This technique is used in order to have a robust estimation of the performance of the algorithm created when working with unknown data (*K-Fold* data). The working basis is simple: Each patient's folder is partitioned into K equal subsets. The algorithm is then evaluated K times. This way, most of the subsets are used to create the training database, while the remaining subsets form the test database, by which the performance of the algorithm will be evaluated. By performing K iterations, the procedure will have been completed, and all the components of the folder will have formed part of the training database and will have been used for testing. This fact allows the results obtained from the metrics associated with the results of the algorithm to be much more robust than if a single partition had been performed, since the complete distribution of the data is represented. The following figure 4.1 shows how *K-Fold* partition works.

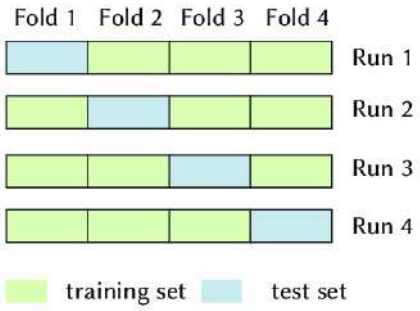

Figure 4.1: *K-fold* working example with $K = 4$. Source: [64]

For this project, the decision was made to use $K = 10$, in such a way that in each iteration, 9 *folds* are used for the training phase and 1 *fold* is used for testing.

4.1 Base Methods: Linear Regression and CCA Linear Regression

The results obtained through Unit Selection will be evaluated by means of objective metrics widely used in speech synthesis applications. In order to have another level of context and a better understanding, two alternative base methods have been implemented in order to compare the results against the more complex methods. The legacy linear regression method was chosen for its simplicity of implementation and compatibility with unit analysis. CCA linear regression was also chosen because it works as a base method for the CCA approach. It is also enriching to be able to compare two base methods one against each other.

For the creation of both methods, the original Unit Selection code was taken as a reference, seeking the maximum similarity possible for comparability reasons. The implementation was done in *Python* using the *LinearRegression* and *sklearn.cross.decomposition.CCA* libraries. The model execution is as it follows:

1. **Model Creation:** For standard Linear Regression, the objective is to fit a linear model with certain coefficients to the training data, in order to minimize the residual quadratic sum between the units in the database and those predicted by the linear approximation. Once the training database is available, the linear fit model is created. The formulas governing the creation of the linear model are presented in 2.8, 2.9, 2.10. The process is equivalent for CCA Linear Regression, where all the details can be found in 3.3

2. **MFCC Unit prediction:** The chosen libraries allow to make predictions based on the model created. To obtain the result for the test units, the *predict* method of both libraries is used, which returns a prediction of the MFCC units using the model created. As with the Unit Selection algorithm, WORLD VoCoder is used to synthesize audio from the results obtained.

4.2 TiDigit - LC

This section is focused on displaying the results obtained by each the different methods, for the first of the two digits datasets.

4.2.1 Base Method: Legacy Linear Regression

Both linear regression methods, by design, only have a single parameter that can be modified: unit length. Results will be shown for one Digit dataset and one Sentence dataset for each of the two base methods, modifying unit

length. A graphical comparison will also be shown, allowing us to compare the temporal and frequency behaviour of original and synthesized audios. The configuration with the best result will be marked in **bold**.

The results for legacy Linear Regression can be found below:

Variable Unit Length

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	10.950	**10.909**	10.937	11.034
STOI	**0.529**	0.517	0.514	0.502

Best performing configuration is 0.04s unit length of for MCD and 0.02s for STOI

The following figure 4.2 shows a comparison between the original signal and the one synthesized via legacy Linear Regression. Temporal representation and spectrogram are included.

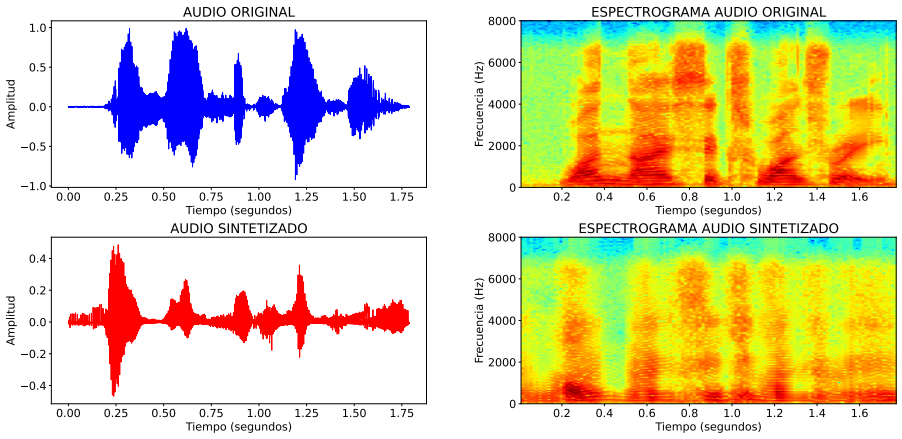

Figure 4.2: Comparison between original signal and synthesized with legacy Linear Regression. TiDigit - LC

Figure 4.2 contains original and synthesized signal for the English digit sequence 'One Nine Six One Three' (19613). The limitations of the legacy Linear Regression method are made evident in the figure. It can be seen in the time representation, as well as in the spectrogram, that in the synthesized signal there is low frequency noise throughout the audio. Similarly, the spectrogram shows a clear lack of detail in the areas where the digits are pronounced (they appear diffuse). These two phenomena are due to the limitations of the linear regression method itself.

4.2.2 Base Method: CCA Linear Regression

Variable Unit Length

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	**12.33**	12.33	12.33	12.33
STOI	**0.432**	0.432	0.432	0.431

As we can see, the results are the same, no matter the chosen Unit Length. The results are substantially worse than for Legacy Linear Regression.

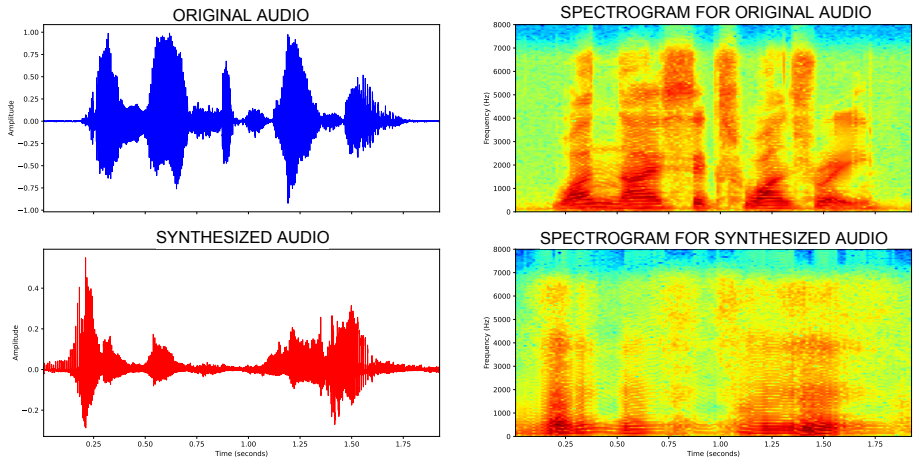

Figure 4.3: Comparison between original signal and synthesized with CCA Linear Regression. TiDigit - LC

Figure 4.3 contains original and synthesized signal for the English digit sequence 'One Nine Six One Three' (19613). As we can see, the limitations of this method are even more clear with this results. The low frequency noise is still present in CCA method. Moreover, there are single digits which are not correctly synthesized, judging from the spectrogram representation.

4.2.3 Legacy Unit Selection

Legacy Unit Selection was the algorithm designed and implemented by the student in the last study on the topic. The results shown can serve as a reference for the newly implemented methods. Legacy Unit Selection demonstrated that the algorithm was capable of synthesizing intelligible speech for

all datasets. As mentioned previously in 3, there are two parameters that can affect the performance of the system:

- Unit Length
- Concatenation cost

In order to avoid redundancy, only one table of results will be shown for each type of dataset: Digits or Sentences. The results for the two speakers in each dataset are very similar, so it is not necessary to duplicate the information, considering that this algorithm has already been studied. The best performing configurations will be shown in **bold**.

Below we can find the results for legacy Unit Selection. Independent tables are shown to illustrate the result of the individual modification of each parameter.

Variable Unit Length ; **Weight** = 0.1

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	11.389	11.363	**11.032**	10.819
STOI	0.494	0.517	**0.521**	0.506

Best overall performance was found for a Unit Length of 0.08 s.

Unit Length = 0.08 ; **Variable Weight**

Concatenation Weight	0	0.01	0.1	0.5	1	2
MCD	11.195	11.111	**10.819**	10.912	10.938	10.945
STOI	**0.572**	0.566	0.521	0.504	0.500	0.491

In terms of MCD, the best result is for 0.1 concatenation weight, while no weight is the best option for intelligibility.

The following figure 4.4 shows representations for original audio and synthesized through Unit Selection.

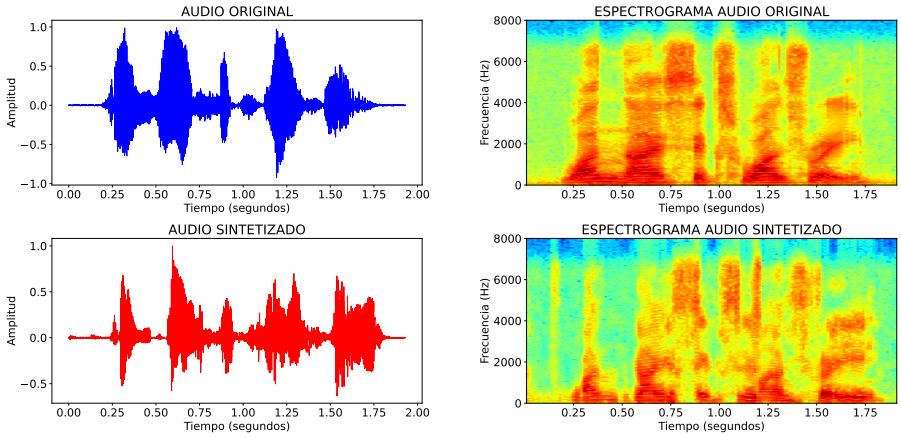

Figure 4.4: Comparison between original-synthesized audio via legacy Unit Selection. TiDigit - LC

Figure 4.4 shows the comparison between original and synthesized signals for the English digit sequence 'One Nine Six One Three' (19613) . The figure shows the advantages of Unit Selection over Linear Regression: In the spectrogram it is clear how the digits are differentiated and not fuzzy. The low frequency noise that appeared even in the 'silence' zones does not occur using the Unit Selection method. In general, the time representation and spectrogram show that the algorithm works better than the two base methods.

4.2.4 CCA Unit Selection

This subsection show the results for the novel method that combines CCA with the Unit Selection algorithm. The algorithm has the same parameters that can be tuned as legacy Unit Selection: Unit Length and Concatenation weight. The following two tables display the results:

Variable Unit Length ; Weight = 0.1

Unit Length(s)	0.02	0.04	0.08	0.16
MCD	11.459	11.385	**11.073**	11.256
STOI	0.413	0.485	**0.492**	0.491

Best performance in terms of both distortion and intelligibility happens for a Unit Length of 0.08 s.

Results for variable concatenation weight:

Unit Length = 0.08 ; **Variable Weight**

Concatenation Weight	0	0.01	0.1	0.5	1	2
MCD	11.519	11.073	**10.919**	10.956	11.002	11.174
STOI	0.435	0.492	0.490	**0.495**	0.491	0.489

In terms of MCD, the best result is for 0.1 concatenation weight, while $w = 0.5$ is the best option for intelligibility.

The following figure 4.5 shows representations for original audio and synthesized through CCA Unit Selection.

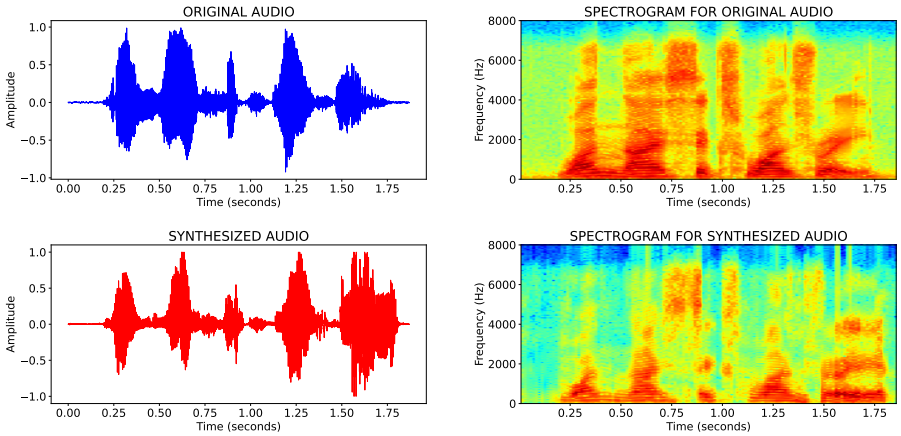

Figure 4.5: Comparison between original-synthesized audio via CCA Unit Selection. TiDigit - LC

Figure 4.5 shows a direct comparison between original and synthesized signals for the English digit sequence 'One Nine Six One Three' (19613). The conclusions that can be extracted from the figure are quite similar as those from legacy Unit Selection. The individual digits are easily distinguishable and there is no noise whatsoever in the silence parts. The discussion section 4.4 will dive into the results analysis.

4.2.5 Direct Speech Synthesis (DSS)

This subsection will present the results obtained for the Direct Speech Synthesis novel method. As commented in previous sections, this method has three main tunable parameters: Unit Length, Concatenation Cost and Frameshift. Frameshift is shown as a percentage % of the Unit Length, 4 different levels of overlap are tested each time: 10%, 25%, 50%, 75%. The only measurable objective metric that can be computed is STOI The following two tables show a breakdown of the results, one with and one without concatenation cost:

STOI for Direct Speech Synthesis ; TiDigit - LC ; Concatenation Weight = 0

Frameshift (%) \ Unit Length (s)	0.01	0.05	0.1	0.15	0.20	0.25
10	0.418	0.522	0.563	0.589	0.600	**0.606**
25	0.426	0.521	0.559	0.577	0.584	0.584
50	0.426	0.506	0.530	0.549	0.544	0.536
75	0.415	0.470	0.488	0.491	0.494	0.444

The best result in terms of STOI is found for a combination of Unit Length = 0.25 s, and a Frameshift = 0.025 s.

The following table shows the same analysis but with Concatenation Weight = 1:

STOI for Direct Speech Synthesis ; TiDigit - LC ; Concatenation Weight = 1

Frameshift (%) \ Unit Length (s)	0.01	0.05	0.1	0.15	0.20	0.25
10	0.383	0.499	0.544	0.566	0.574	**0.589**
25	0.375	0.469	0.511	0.543	0.551	0.558
50	0.354	0.422	0.473	0.492	0.508	0.499
75	0.321	0.357	0.402	0.441	0.442	0.402

The best result in terms of STOI is found, again, for a combination of Unit Length = 0.25 s, and a Frameshift = 0.025 s.

The following figure 4.6 is a comparison between original and synthesized audio for Direct Speech Synthesis:

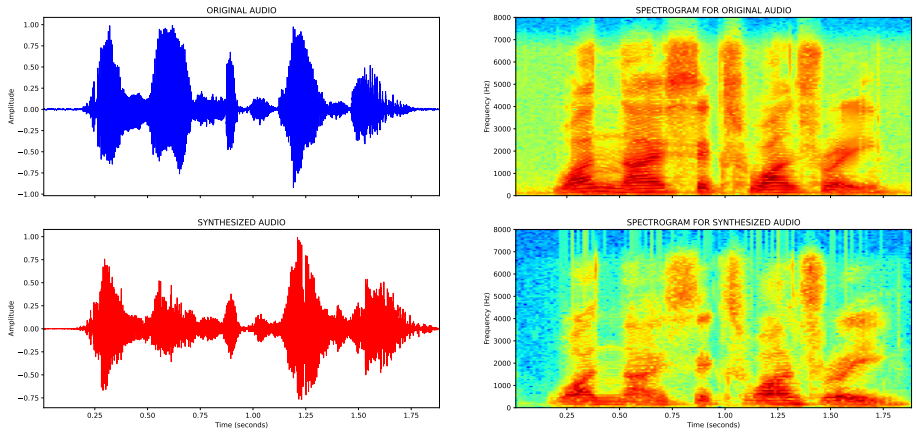

Figure 4.6: Comparison between original-synthesized audio via Direct Speech Synthesis. TiDigit - LC

As we can see, the algorithm performs very good for synthesizing digits. Both the temporal representation and spectrogram show that digits are close to the original version and there is no noise whatsoever.

4.2.6 Deep Neural Network (DNN)

Regarding the results for Neural Networks, the objective was to test different configurations until one that worked correctly was found, and then compute the results with that same configuration. Multiple tests were performed and the best configuration (also considering the computational cost) was found to be the one stated in 3. The following table shows the mean results obtained for the TiDigit database:

Métrica	**MCD (dB)**	**STOI (adim)**
Resultado	11.189	0.508

Table 4.1: Table for objective metrics for Deep Neural Network. TiDigit - RM

Values are slightly higher than for legacy Unit Selection regarding distortion, with an acceptable value for STOI.

The following figure 4.7 shows a comparison between original and synthesized audio.

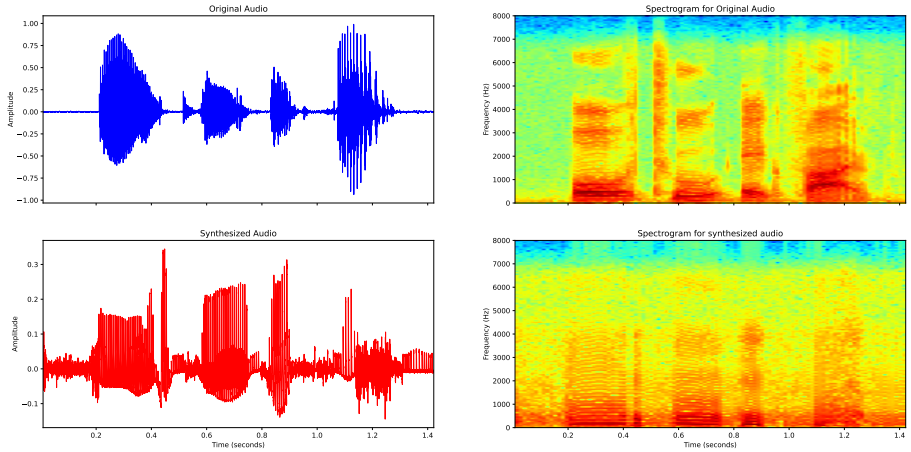

Figure 4.7: Comparison between original-synthesized audio via Deep Neural Network. TiDigit - LC

As we can see, the results are not great. The DNN correctly identifies and synthesizes the digits, but fails to do so without noise and a clear differentiation between silence and utterances. The spectrogram looks a little fuzzy and the temporal representation suggest there is noise throughout the audio. Subjective listen confirms this premise. The speech is intelligible but sounds quite noisy. The synthesized audio does not sound very natural. This will be analyzed in the 4.4 section.

4.2.7 Gated Recurrent Unit (GRU) Neural Network

The GRU Neural Network is an evolution of the DNN, so the same process was followed. The general configuration of the Neural Network is equivalent as theDNN case. The main difference, as commented in 3, is that the neurons include a memory cell within themselves that allow to consider temporal context into the network. The following table shows the mean results obtained for the TiDigit database:

Métrica	**MCD (dB)**	**STOI (adim)**
Resultado	9.41	0.56

Table 4.2: Table for objective metrics for GRU Neural Network. TiDigit - LC

Values are much better, actually, the best distortion metric for all the evaluated methods, and certainly better than legacy Unit Selection, with a good STOI metric.

The following figure 4.8 shows a comparison between original and synthesized audio.

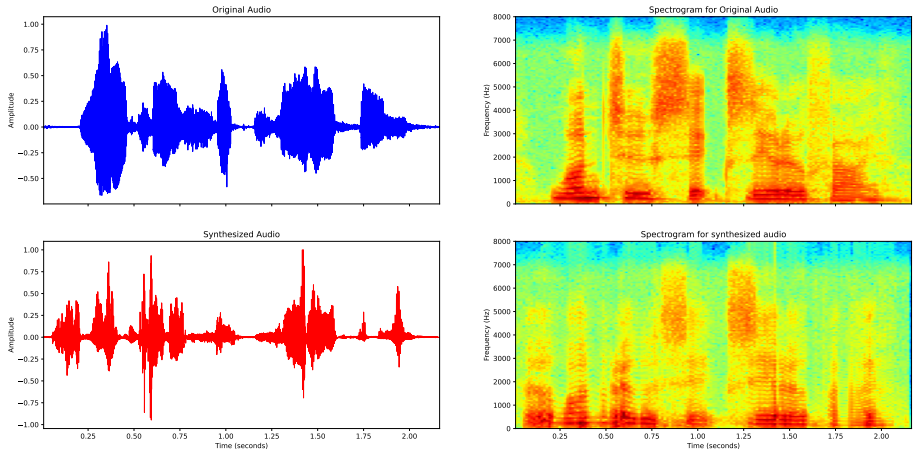

Figure 4.8: Comparison between original-synthesized audio via GRU Neural Network. TiDigit - LC

As we can see, the results are very good. The audio does not seem to have any noise, the synthesized digits are very clearly represented and are very close to the original version of the audio. Subjective listen confirms that the result is positive. Intelligibility is very good, although naturalness is not as polished as in Unit Selection derived methods.

4.3 Arctic - RM

This section is dedicated to showing the results for one of the two datasets containing complete sentences. synthesizing speech in this circumstances is much more difficult than with single digits, so results should be different using the same algorithm.

4.3.1 Base Method: Legacy Linear Regression

The results for legacy Linear Regression varying the unit length are below:

Variable Unit Length

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	11.175	11.145	**11.116**	11.123
STOI	0.468	0.477	0.481	**0.487**

0.08 s gives the best result in terms of distortion, while the best intelligibility is found for a unit length of 0.16 s.

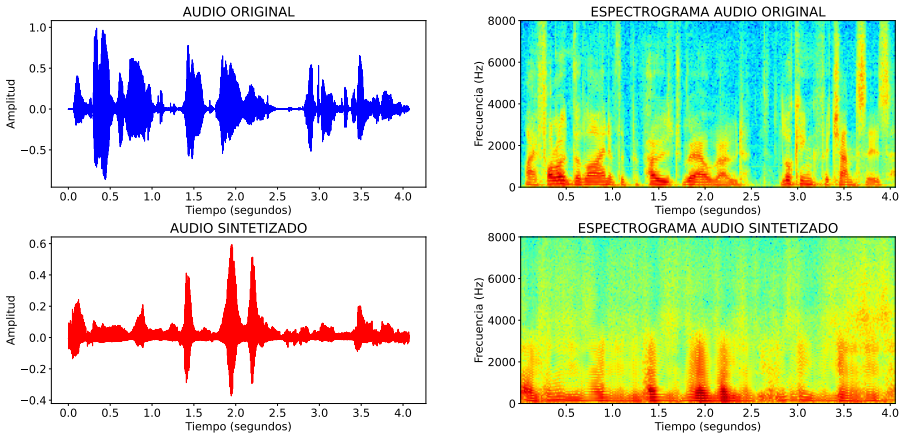

Figure 4.9: Comparison between original and synthesized audios using legacy Linear Regression. Arctic - RM

In figure 4.9 we can check the differences between original and synthesized audio for a complete sentence: 'I followed the line of the proposed railroad, looking for chances'. The same problems are observed as for individual digits synthesis. The areas corresponding to words in the spectrogram appear fuzzy, and there is also noise throughout the audio. The intelligibility problems of Linear Regression are, indeed, accentuated in a more complex situation such as speech synthesis for complete sentences.

4.3.2 Base Method: CCA Linear Regression

The results for CCA Linear Regression are below:

Variable Unit Length

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	**12.029**	12.044	12.044	12.044
STOI	**0.367**	0.365	0.364	0.364

0.02 s gives the best result in terms of bioth distortion and intelligibility, though results are almost identical.

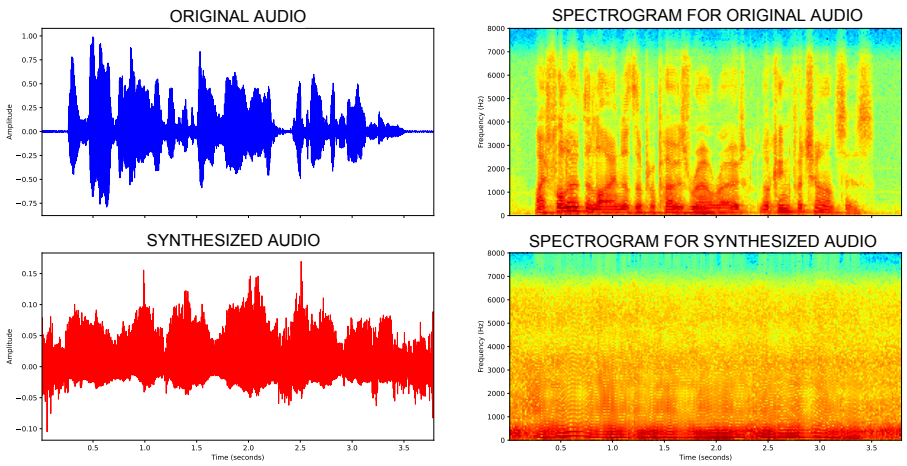

Figure 4.10: Comparison between original and synthesized audios using CCA Linear Regression. Arctic - RM

Figure 4.10 shows us again that the result is worse in CCA Linear Regression than in standard Linear Regression. Words are almost impossible to differentiate from each other and the audio figure shows extreme levels of noise.

4.3.3 Legacy Unit Selection

Below we can find the results for synthesis of sentences using the legacy Unit-Selection method.

For variable Unit Length:

Variable Unit Length ; Concatenation Weight = 0.1

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	12.255	12.057	**12.060**	12.638
STOI	0.396	0.404	**0.405**	0.373

Best overall performance was found for a Unit Length of 0.08 s.

For a variable concatenation weight:

Unit Length = 0.08 ; **Variable Concatenation Weight**

Concatenation Weight	0	0.01	0.1	0.5	1	2
MCD	12.640	12.447	**12.068**	12.130	12.142	12.144
STOI	0.413	**0.414**	0.404	0.389	0.387	0.387

Best distortion figure is found for a 0.1 weight, while best intelligibility is for 0.01 weight.

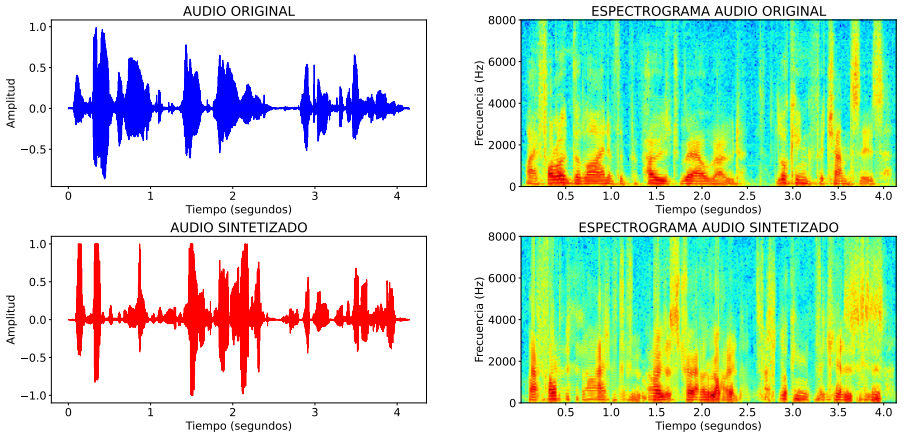

Figure 4.11: Comparison between original an synthesized sentences using legacy Unit Selection. Arctic - RM

In figure 4.11 we can check the performance of the Unit Selection algorithm for speech synthesis using a complete sentences dataset. We can see the differences between original and synthesized audio for a complete sentence: 'I followed the line of the proposed railroad, looking for chances'. In the temporal representation we can see that there is a correspondence between the original and the synthesized signal, as well as in the spectrogram, where we can see that the phonemes are much more defined than in the base methods.

4.3.4 CCA Unit Selection

This subsection show the results for the novel method that combines CCA with the Unit Selection algorithm, but now for a dataset with complete sentences. The algorithm has the same parameters that can be tuned as legacy Unit Selection: Unit Length and Concatenation weight. The following two tables state the results:

Variable Unit Length ; Concatenation Weight = 0.1

Unit Length (s)	0.02	0.04	0.08	0.16
MCD	12.897	12.745	**12.689**	12.731
STOI	0.339	0.342	**0.387**	0.312

The best result overall is found for a Unit Length = 0.08 s.
Results when varying concatenation weight:

Unit Length = 0.08 ; **Variable Concatenation Weight**

Concatenation Weight	0	0.01	0.1	0.5	1	2
MCD	12.793	12.823	**12.689**	12.731	12.784	12.740
STOI	0.332	0.345	0.387	**0.394**	0.398	0.404

In terms of MCD, the best result is for w = 0.1 concatenation weight, while $w = 0.5$ is the best option for intelligibility.

The following figure 4.12 shows representations for original audio and synthesized through CCA Unit Selection.

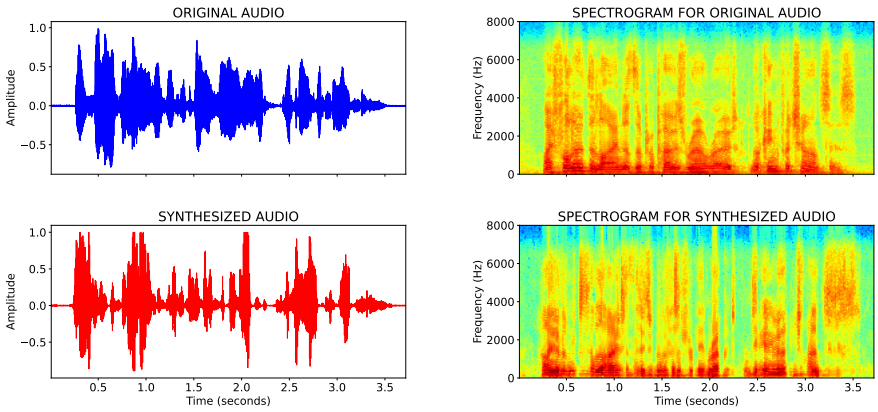

Figure 4.12: Comparison between original-synthesized audio via CCA Unit Selection. Arctic - LC

Figure 4.12 shows the differences between original and synthesized audio for a complete sentence: 'I followed the line of the proposed railroad, looking

for chances'. The figure shows that CCA Unit Selection gives a good performance for complete sentences. The conclusions are similar than those of Unit Selection, however, the digits seem more defined and with a decisively higher power than in the legacy Unit Selection Method.

4.3.5 Direct Speech Synthesis (DSS)

This subsection will present the results obtained for the Direct Speech Synthesis novel method for a dataset with complete sentences. The following two tables show a breakdown of the results, one with and one without concatenation cost:

STOI for Direct Speech Synthesis ; Arctic - RM ; Concatenation Weight = 0

Frameshift (%) \ Unit Length (s)	0.01	0.05	0.1	0.15	0.20	0.25
10	0.314	0.438	0.457	0.461	**0.542**	0.442
25	0.333	0.432	0.446	0.441	0.424	0.409
50	0.337	0.405	0.408	0.395	0.372	0.357
75	0.330	0.368	0.360	0.342	0.328	0.309

The best result in terms of STOI is found for a combination of Unit Length = 0.2 s, and a Frameshift = 0.02 s.

The following table shows the same analysis but with Concatenation Weight = 1:

STOI for Direct Speech Synthesis ; Arctic - RM ; Concatenation Weight = 1

Frameshift (%) \ Unit Length (s)	0.01	0.05	0.1	0.15	0.20	0.25
10	0.2918	0.407	0.431	**0.433**	0.422	0.414
25	0.283	0.363	0.389	0.383	0.375	0.366
50	0.271	0.305	0.330	0.321	0.309	0.298
75	0.233	0.265	0.273	0.271	0.273	0.261

The best result in terms of STOI is found for a combination of Unit Length = 0.15 s, and a Frameshift = 0.015 s.

The following figure 4.6 is a comparison between original and synthesized audio for Direct Speech Synthesis:

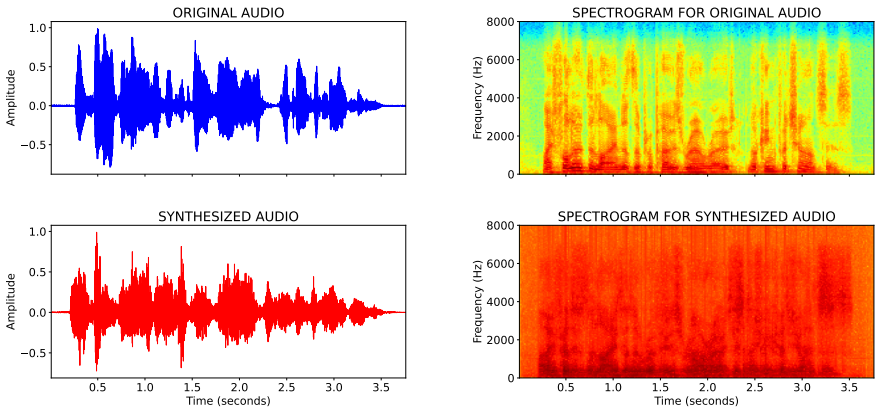

Figure 4.13: Comparison between original-synthesized audio via Direct Speech Synthesis. Arctic - RM

As we can see, the algorithm performs quite poorly at the task of performing complete sentences. Spectrogram shows a very noisy representation for the synthesized audio. Reasons will be discussed in 4.4. Subjective listen confirms that intelligibility is very low and that the audio sections that compose the sentence are either overlapped or abruptly changed.

4.3.6 Deep Neural Network (DNN)

The following table shows the mean results obtained for the Arctic database using a DNN:

Métrica	**MCD (dB)**	**STOI (adim)**
Resultado	11.83	0.421

Table 4.3: Table for objective metrics for Deep Neural Network. Arctic - RM

Values are decisively higher than for legacy Unit Selection, with an low STOI, far from the results obtained from the Unit Selection based methods. Subjective evaluation shows that intelligibility is practically null, with lots of noise and a robotic-like voice result.

The following figure 4.14 shows a comparison between original and synthesized audio.

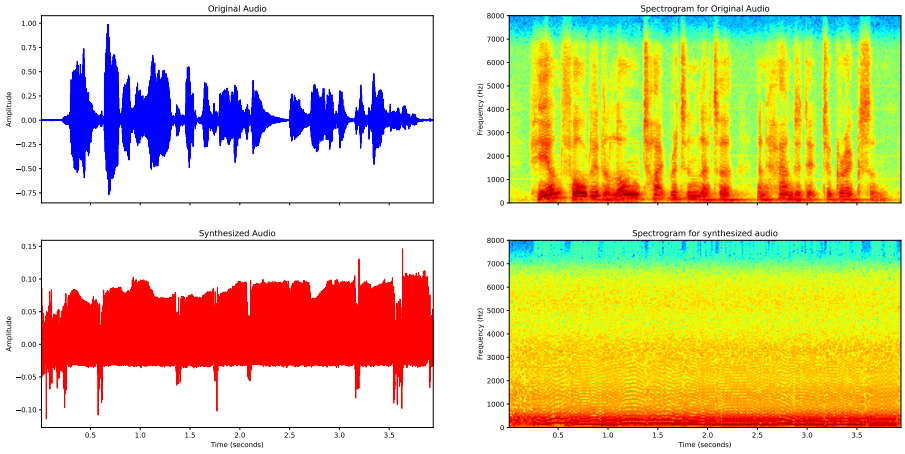

Figure 4.14: Comparison between original-synthesized audio via Deep Neural Network. Arctic - RM

As we can see, the results are very bad. Neither the temporal representation nor the spectrogram have any resemblance whatsoever to the original sentence. The audios are full of noise.

4.3.7 Gated Recurrent Unit (GRU) Neural Network

The following table shows the mean results obtained for the Arctic database using a GRU Neural Network:

Métrica	**MCD (dB)**	**STOI (adim)**
Resultado	10.44	0.49

Table 4.4: Table for objective metrics for GRU Neural Network. Arctic - RM

Values are better than for the DNN, both in terms of distortion and objective intelligibility.

The following figure 4.8 shows a comparison between original and synthesized audio.

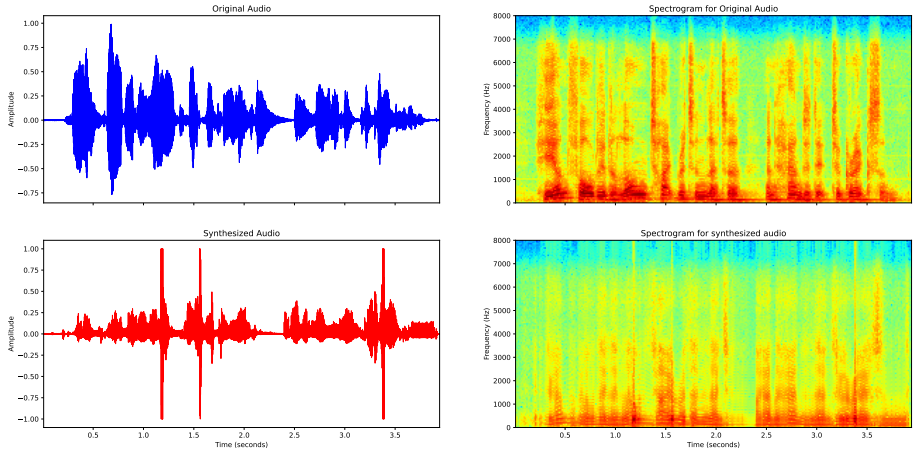

Figure 4.15: Comparison between original-synthesized audio via GRU Neural Network. Arctic - RM

As we can see, the results are much better than in the DNN. The temporal representation is not very close to the original one, but the spectrogram confirms that the waveform is very well achieved and that utterances are correctly represented, although at a slightly lower power. Subjective listen reveals that the result is intelligible. The synthesized voice sounds a bit robotic but sentences can be identified. Noise seems to be present but the overall result is satisfactory.

4.4 Discussion

The scope of this section is to discuss the obtained results in terms of metrics and subjective listenings for all of the different proposed methods.

4.4.1 General overview by subjective Evaluation

With the aim of evaluating the quality of the synthesized speech signals, informal listenings of the various synthesized audios were arranged. As far as the main objective of the project is concerned, this was achieved satisfactorily. Intelligible speech synthesis can be achieved with **all** of the novel proposed algorithms with varying levels of quality. Evaluation determines that all the novel approaches perform significantly better than base methods. Intelligibility is good for all novel methods, although quite poor for the two base methods that were evaluated, where even single digit identification can be challenging at times. It is proved, then, the superior effectiveness of proposed methods when it comes to achieving intelligible speech.

By subjectively listening to the different results for each dataset, it is

concluded that the intelligibility achieved for the *TiDigit* database is superior to that of *Arctic*. This is to be expected, as the synthesis of complete sentences is a more complex process, which also requires larger databases. The phonetic variability of the sentences in the *Arctic* database, where the number of phonemes pronounced is much higher, must also be taken into account. Another factor, identified in other works such as [48], is the limitations of the PMA technique itself. In [48] it is concluded that there are phonemes that the PMA technique is not able to capture, making it *de facto* impossible to synthesise them correctly without additional capture tools or extra context information. The scope of this work is not specific enough to analyse which phonemes are not captured. The intelligibility for the synthesised sentences in the *Arctic* database varies depending on the sentence being uttered, whereas for *TiDigit* total intelligibility was achieved for virtually all files and implemented methods.

4.4.2 Review of proposed methods

When it comes to speech synthesis from biosignals, the proposed algorithms offer a better performance than base methods because they are a more powerful option. This is because they are non-parametric methods that, unlike linear regression methods, do not distort the speech significantly. Speech is synthesized with unaltered portions of the speech from the database in Unit Selection variants and distortion in Neural Network methods is kept to a minimum. As discussed in previous sections, the relationship between PMA biosignals and speech often does not follow a linear logic, so this is an additional advantage of the proposed methods over Linear Regression. Another advantage is that proposed solutions are flexible, as they can be used with a variety of biosignals, e.g. EEG [56]. In theory, it would also be possible with Linear Regression, but the non-linear nature is more pronounced for a complex case such as EEG-speech synthesis.

While the metrics do not show a visible difference in performance of the Unit Selection algorithm versus linear regression, the differences are noticeable when reviewing the differences in the figures and in subjective listening. Following, there is a breakdown of the performance of each proposed method:

- CCA Unit Selection: The results obtained for this novel proposed method were, in general, similar to those found in Unit Selection in terms of objective metrics (STOI seems to be lower across the datasets). Metrics do not seem to be significantly affected by the variation of parameters. This was an issue in the last investigation using legacy Unit Selection that seems to affect this algorithm too. The figures 4.5, 4.12 show a clean audio, where utterances can be easily differentiated from the silences. In general, there is a good resemblance to the original audio. However, subjective listening shows

that the synthesized audio quality is slightly lower than that of legacy Unit Selection. The main target of applying CCA was to concentrate the features of PMA and MFCC's in less dimensions with the hope that the algorithm could better establish the relationship between features. This has not been the case for this specific implementation. One plausible reason for this is that CCA is based in the identification of **linear combinations** between two variables. The fact that what is being considered are just linear relationships may be the issue that hinders the performance of the system. As stated along this report, this relationship has a nonlinear nature.

- DSS: The results obtained for this method were quite encouraging for digits, where real improvement over the legacy method in terms of intelligibility could be achieved. The result is natural sounding, as speech is synthesized by concatenating real portions of audio (there is no synthesis process as we had in the methods dependant on VoCoder). The best configuration seems to be longer units (0.25 s) with a small frameshift (10%). As a reference, phonemes have a duration of more or less 0.1 s. No concatenation cost seems to perform slightly better than concatenation weight, so further investigation is needed on this cost calculation, as it is intended to improve the results, not the other way round. This phenomena was investigated by entering into the values of the concatenation weight along the can be caused by the high variability of the weight for each different unit evaluated. Even though the values are normalized and kept between a specified range [0,1], they variate a lot with respect to the target cost. The objective of the concatenation cost is to influence the target cost into choosing one unit or other. In this case it does influence it, just in a way that does not seem to improve performance. The difference in subjective listening is negligible. As for the results for the Arctic dataset, the performance is much worse. Depending on the exact configuration, the results worsened with respect to legacy Unit Selection. The sample spectrogram shown in figure 4.13, shows the synthesized audio is full of noise and phonemes appear to be fuzzy. Objective listening shows little intelligibility. The audios show that the main problem lies in overlapping between phonemes as well as concatenation of different phonemes that do not coordinate among them to form a sentence that makes sense. Additionally, transition between phonemes seems too abrupt at times. The problem stays the same with and without concatenation cost. The task of synthesizing complete sentences is of a much higher complexity than single digits. It is possible that constructing complete sentences of high quality using phonemes is not viable using this exact approach. Future possible evolution for this method will be proposed in the conclusions of this report.

- DNN: The results obtained for Deep Neural Networks were good for digits, with acceptable intelligibility but limited naturalness. However, the results were quite disappointing for synthesizing complete sentences, where the network failed to identify and correctly synthesize the utterances. It is possible that the complexity of complete sentence speech synthesis is too high to be addressed by the number of neurons in the Network. It is not clear if a higher number of hidden layers or a bigger size can help with this problem (this was not tested because the increase in complexity in the Network was not manageable by the available hardware). Another issue that came up during the investigation was that the naturalness of the synthesized speech was worse than for the Unit Selection based options. The reason behind this is that Neural Networks are a parametric method, whereas Unit Selection based methods are non-parametric, meaning the work with real portions of voice (either voice in DSS of MFCC's in the rest of methods). The use of real voice (or real voice parameters) means that the result, if correct, will have a very high naturalness due to it being composed of small real voice fractions. Parametric methods such as Neural Networks often fail to achieve good naturalness in speech synthesis applications.
- GRU Neural Network: In general, the results obtained for the GRU Neural Network were very encouraging. Performance of this method was significantly better than the DNN algorithm. Both in terms of Mel Cepstral Distortion and STOI, the results were very good. Subjective listening confirmed that the synthesized voice was of good quality and with no problems of intelligibility whatsoever. The only issues were reduced naturalness with respect to the Unit Selection based methods. Additionally, the results with complete sentences were also very promising, with all metrics and subjective listening being correct. It is demonstrated that a Recurrent Neural Network of intermediate complexity can successfully perform speech synthesis with PMA biosignals. It is also established that the consideration of temporal context in Neural Networks is a differentiating factor and affects decisively the quality of the results, to the point where it makes no sense to continue investigating speech synthesis with plain Deep Neural Networks. One consideration that needs to be taken into account is the huge computational cost of the GRU Neural Network. A standard solution like the one implemented in this project took more than 20 hours to train for the sentences dataset. Considering that the size of the dataset is not really big when compared with the amount of data treated this days, computational complexity remains a topic to keep an eye on.

In general, the values obtained for MCD are high compared to the literature [41] [47], where this value is between 4.4 and 6 dB. A possible reason

for this difference lies in the fact that the VoCoder used in the aforementioned literature is the STRAIGHT VoCoder, while in this work the VoCoder WORLD VoCoder has been used. The operation of both is different and could have affected the performance of the synthesis system. DSS, was the best performing algorithm in terms of intelligibility, achieving a very good result for speech synthesis in the digits dataset, also it did not rely on the use of a VoCoder. The best result in terms of distortion was achieved by the GRU Neural Network. At the same time, it has been observed that in the algorithm derived directly from legacy Unit Selection (CCA Unit Selection), the modification of the parameters did not report significant variations in the results obtained by objective metrics (also for subjective listening). In addition, the differences observed in the objective metrics between datasets, as well as against the base method, are not very big. It can be inferred that the objective metrics used in this work are not a definitive source of information regarding the performance of the algorithm, since in subjective listening (as well as in images and spectrograms) the differences are more visible. DSS associated intelligibility metrics did perform correctly when it comes to reflecting differences in synthesized speech quality.

Chapter 5

Conclusions and Future Lines of Action

The main goal of this work has been the implementation of various speech synthesis algorithms that used voice and biosignals as inputs. To meet the objectives, three different approaches (CCA, DSS and Neural Networks) were implemented to take information from a database with PMA biosignals and speech records for digits and sentences. From this database, speech synthesis was performed. The results have shown that it is possible to synthesise intelligible speech from PMA biosignals using every proposed algorithm.

As for the metrics associated with the results obtained, in terms of mel cepstral distortion (MCD) they are between 9.41 dB and 12.4 dB for all datasets, while for intelligibility (STOI) they are between 0.32 and 0.606 (Being the upper limit of this value around 0.9). Subjective listening determines that the algorithms created for this purpose have a superior intelligibility to the implemented baseline linear regression methods. Direct Speech Synthesis showed a superior performance than the legacy Unit Selection method for synthesizing single digits.

After concluding the work carried out on the algorithms that make up this project, the main conclusion is that it is possible to synthesise speech by means of PMA biosignals using any of the three main different proposed methods. The information provided by the PMA signals is sufficiently correlated with the voice to obtain an intelligible voice of acceptable quality. The best results were, as expected, found for the most simple case of use: single digit synthesis.

Superiority of methods based on Unit Selection (CCA, DSS) is mainly understood to be due to the fact that they use a non-parametric approach, which allows acceptable results (speech synthesis) to be achieved with databases of manageable sizes, making it possible to implement and experiment with common hardware. Additionally, the abundance in the literature of speech synthesis implementations that make use of Unit-Selection was a factor to

be taken into account.

The good results obtained for Neural Networks (DNN and GRU) are mainly attributed to the huge power of these approaches, where nonlinear correspondence can be learnt using a manageable amount of hidden layers. This is specially true for the GRU Neural Network, which can take advantage of the consideration of temporal context in the training phase. Temporal context is found to be of a high importance when it comes to speech synthesis.

In this project, all different methods that have been implemented to synthesise speech from PMA biosignals have achieved the set objective. The implementation of two base methods of simple operation and theoretical basis has served as a basis for the implementation and debugging of the main algorithm, by providing the opportunity to check first-hand the performance obtained when synthesising speech by means of a simpler algorithm under the same conditions and for the same datasets. A baseline was set also in terms of intelligibility and metrics in order to have a starting point in the evaluation of the results of the different proposed methods.

The proposed methods are slightly different versions of what can be found in the literature due to the specific working environment: available datasets and computing power. There is, consequently, an added value in the work carried out, where speech synthesis feasibility was tested for new variants of the available algorithms and the specific datasets used.

As for the results obtained, synthetic speech was obtained for all the datasets available. However, the metrics used (mainly MCD) do not justify the results obtained, especially in terms of the improvement with respect to the base algorithm and intelligibility of the different proposed method. As commented in previous sections, MCD values are higher than the norm in the available literature, while intelligibility was very good depending on the specific algorithm implementation and configuration. In addition, in the case of Unit Selection related methods, the variation of the different parameters has not resulted in significant alterations to the results obtained. Beyond the results obtained in the metrics, the subjective listening shows that the synthesised audio is intelligible for all methods and that the algorithm can be valid for use in speech synthesis systems for PMA biosignals.

In terms of future lines of action that can be adopted for the scope of this investigation, it worth noting that it would be of a high interest to keep studying the DSS approach, as the results obtained were encouraging, specially for digits. One specific line of improvement would be to re-evaluate the concatenation cost, as it would be a feasible way of improving results in digits, but specially full sentences. One possible approach can be to use features like MFCC's to compute the concatenation cost. This parameters can be computed on demand using different libraries and help give a better context of the concatenation between units.

Another future line of action that has a great potential is to keep evolv-

ing the Neural Network approach, specially for the RNN methods. Results were promising in the methods proposed, considering the limited computing power available, as well as the size of the datasets. The results obtained in the extensive literature indicate that there is much room for improvement in terms of high quality speech synthesis [42]. This algorithms are highly adaptable and can easily be tuned to work with different datasets and 'power' configuration, so they make a great tool for speech synthesis investigation. Availability of an extensive, high quality and accessible dataset stays a small challenge that needs to be overcame.

Additionally, a hybrid approach can be followed and Neural Networks can be combined with the DSS approach. Judging by the results obtained, this seems to be a promising method that needs further investigation on the topic, by means of literature research and algorithm testing. This would also increase efficiency, as the need of a VoCoder such as STRAIGHT or WORLD would be no more. This approach could also address the issue of speech naturalness in Neural Networks methods, as the Network would be trained directly on real speech portions, not parameters derived from it.

Another field of study of great interest would be that of speech synthesis from brain biosignals, especially by means of Neural Networks. Speech synthesis by using of brain signals means that they can be used in patients affected by all kinds of injuries or diseases, even those with no joint mobility at all, so they could be used in virtually every conceivable situation. This would be like a silver bullet to the problem of communication disabilities. However, the challenges associated with this branch are also significant, the main one being the quality of the biosignals themselves. Often the acquisition of such biosignals depends on medical (non-technical) criteria, so it happens that in many situations there is zero flexibility in terms of biosignal acquisition. Additionally, the field of study that relates brain signals to voice production is a branch of knowledge that still does not have a solid answer to how to obtain highly correlated biosignals with voice. A very high inter-patient variability is very common to this day, considering that it is impossible to precisely replicate sensor placement in two people and that every brain works in a slightly different way.

Many challenges are still to be addressed in this exciting field of investigation. With such a commendable endeavour, every minute invested in this area is for sure worth-it.

Appendix A

Project Timing and budget

A.1 Project Timing

To show the timing of the project, a Gantt chart has been created, as shown in the following figure A.1.

	MONTH							
Task	**December**	**January**	**February**	**March**	**April**	**May**	**June**	**July**
General Documentation								
Neural Networks								
Direct Speech Synthesis								
CCA Unit Selection								
Algorithms adjustments and improvements								
Report elaboration								

Figure A.1: Project timing according to a Gantt Diagram

The first months were mainly dedicated to bibliographic documentation: studying the different methods used in the available literature that could be implemented as part of this project, taking into consideration performance, computational cost and availability of detailed explanations on the topic. Synthesizing speech using Neural Networks was one of the conclusions of the last investigation, so work on that area started right away. The following months were invested in simultaneously work on the proposed methods and the elaboration of the report. In the last months, the results were computed and algorithms were adjusted as the final report was elaborated.

A.2 Project Budget

This appendix will show the estimated budget for the completion of the work. As indicated in A.1, in total, there are 8 full months of work, with the addition of a portion of the last month (July). The total time spent on the completion of the project is estimated to be 750 hours along the 8 months. An engineer working full time works 160 hours per month, so the number of months a telecommunications engineer would need to carry out the project:

$$N^{\circ}months = \frac{Hours\_worked}{Monthly\_hours\_engineer} = \frac{750h}{160h} = 4.68months \quad \text{(A.1)}$$

Taking as a common salary for a junior telecommunication engineer the figure of 1500 euros (net figure), and taking into account that this would have been the only associated monetary cost (all the software and material used is free), we obtain the following.

Position	Months	Salary/month(€)	Total cost (€)
Junior Telecommunication engineer	4.68	1500	7020

To estimate the total cost for a hypothetical employer that wanted to carry out the project, the total cost (including all taxes associated) would increase at least 40% to an amount of around 10000€.

Apart from the time invested in the project, the project has been carried out on a laptop computer at a cost of approximately 1000 euros, as well as on a desktop computer at a cost of approximately 800 euros. The software used for the implementation of the algorithm is *Python*, with no associated cost. Additionally, all the articles listed in the bibliography have been consulted, as well as the two biosignal/voice databases indicated throughout the report, with no associated cost for obtaining any of the resources.

Bibliography

[1] Aubrey J Yates. "Delayed auditory feedback." In: *Psychological bulletin* 60.3 (1963), p. 213.

[2] Bishnu S Atal and Suzanne L Hanauer. "Speech analysis and synthesis by linear prediction of the speech wave". In: *The journal of the acoustical society of America* 50.2B (1971), pp. 637–655.

[3] William J Hardcastle and Fiona Gibbon. "Electropalatography and its clinical applications". In: *Instrumental clinical phonetics* (1997), pp. 149–193.

[4] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[5] Ingo R Titze and Daniel W Martin. *Principles of voice production*. 1998.

[6] Songun Na and Seungwha Yoo. "Allowable Propagation Delay for VoIP Calls of Acceptable Quality". In: Aug. 2002, pp. 47–56. ISBN: 978-3-540-43968-4. DOI: 10.1007/3-540-45639-2_6.

[7] Antti Karjalainen Didier Dupré. "Employment of disabled people in Europe in 2002". In: *Eurostat* 1.1 (2003), pp. 1–4.

[8] Olov Engwall. "Combining MRI, EMA and EPG measurements in a three-dimensional tongue model". In: *Speech Communication* 41.2-3 (2003), pp. 303–329.

[9] Amaro Lima et al. "On the use of kernel PCA for feature extraction in speech recognition". In: *IEICE TRANSACTIONS on Information and Systems* 87.12 (2004), pp. 2802–2811.

[10] Yuet-Ming Lam, Man-Wai Mak, and Philip Heng-Wai Leong. "Speech synthesis from surface electromyogram signal". In: (2005), pp. 749–754.

[11] Maureen Stone. "A guide to analysing tongue motion from ultrasound images". In: *Clinical linguistics & phonetics* 19.6-7 (2005), pp. 455–501.

[12] Mikhail Lebedev and Miguel Nicolelis. "Brain-Machine Interfaces: Past, Present and Future". In: *Trends Neurosci.* 29 (Oct. 2006), pp. 536–. DOI: 10.1016/j.tins.2006.07.004.

[13] Stuart N Baker. "Oscillatory interactions between sensorimotor cortex and the periphery". In: *Current opinion in neurobiology* 17.6 (2007), pp. 649–655.

[14] Gregory Hickok and David Poeppel. "The cortical organization of speech processing". In: *Nature reviews neuroscience* 8.5 (2007), pp. 393–402.

[15] Jochen Baumeister et al. "Influence of phosphatidylserine on cognitive performance and cortical activity after induced stress". In: *Nutritional neuroscience* 11 (June 2008), pp. 103–10. DOI: 10.1179/147683008X301478.

[16] Paul Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

[17] Paul Taylor. "The text-to-speech problem". In: (2009), pp. 26–51. DOI: 10.1017/CBO9780511816338.005.

[18] Jonathan Brumberg et al. "Brain-Computer Interfaces for Speech Communication". In: *Speech communication* 52 (Apr. 2010), pp. 367–379. DOI: 10.1016/j.specom.2010.01.001.

[19] Bruce Denby et al. "Silent speech interfaces". In: *Speech Communication* 52.4 (2010), pp. 270–287.

[20] Bruce Denby et al. "Silent speech interfaces". In: *Speech Communication* 52.4 (2010), pp. 270–287.

[21] Thomas Hueber et al. "Development of a Silent Speech Interface Driven by Ultrasound and Optical Images of the Tongue and Lips". In: *Speech Communication* 52 (Apr. 2010), pp. 288–300. DOI: 10.1016/j.specom.2009.11.004.

[22] Cees Taal et al. "A short-time objective intelligibility measure for time-frequency weighted noisy speech". In: Apr. 2010, pp. 4214–4217. DOI: 10.1109/ICASSP.2010.5495701.

[23] Jeffrey J Berry. "Accuracy of the NDI wave speech research system". In: (2011).

[24] Robin Hofe et al. "Speech Synthesis Parameter Generation for the Assistive Silent Speech Interface MVOCA." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (Aug. 2011), pp. 3009–3012. DOI: 10.21437/Interspeech.2011-753.

[25] World Health Organization et al. "World report on disability." In: *World report on disability.* (2011).

[26] Asterios Toutios and Shrikanth S Narayanan. "Articulatory synthesis of French connected speech from EMA data." In: (2013), pp. 2738–2742.

[27] Zhizheng Wu et al. "Exemplar-based unit selection for voice conversion utilizing temporal information." In: *INTERSPEECH*. Lyon. 2013, pp. 3057–3061.

[28] Kyunghyun Cho et al. "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259* (2014).

[29] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[30] Laura Marzetti et al. "Magnetoencephalographic alpha band connectivity reveals differential Default Mode Network interactions during focused attention and open monitoring meditation". In: *Frontiers in Human Neuroscience* 8 (Sept. 2014). DOI: 10.3389/fnhum.2014.00832.

[31] Masanori Morise. "CheapTrick, a spectral envelope estimator for high-quality speech synthesis". In: *Speech Communication* 67 (Jan. 2014). DOI: 10.1016/j.specom.2014.09.003.

[32] Chris Neufeld and Pascal van Lieshout. "Tongue kinematics in palate relative coordinate spaces for electro-magnetic articulography". In: *The Journal of the Acoustical Society of America* 135.1 (2014), pp. 352–361.

[33] M. Zahner et al. "Conversion from facial myoelectric signals to speech: A unit selection approach". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (Jan. 2014), pp. 1184–1188.

[34] Heiga Zen. "Statistical Parametric Speech Synthesis". In: (2014). Tutorial.

[35] Sandesh Aryal and Ricardo Gutierrez-Osuna. "Data driven articulatory synthesis with deep neural networks". In: *Computer Speech Language* 36 (Mar. 2015). DOI: 10.1016/j.csl.2015.02.003.

[36] Lorenz Diener, Matthias Janke, and Tanja Schultz. "Direct conversion from facial myoelectric signals to speech using Deep Neural Networks". In: (July 2015), pp. 1–7. DOI: 10.1109/IJCNN.2015.7280404.

[37] Mohamad Dolatshah, Ali Hadian, and Behrouz Minaei-Bidgoli. "Ball*-tree: Efficient spatial indexing for constrained nearest-neighbor search in metric spaces". In: *arXiv preprint arXiv:1511.00628* (2015).

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[39] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[40] Marc Arnela et al. "Influence of lips on the production of vowels based on finite element simulations and experiments". In: *The Journal of the Acoustical Society of America* 139 (May 2016), pp. 2852–2859. DOI: 10.1121/1.4950698.

[41] Jose Gonzalez Lopez et al. "A Silent Speech System based on Permanent Magnet Articulography and Direct Synthesis". In: *Computer Speech Language* 39 (Mar. 2016). DOI: 10.1016/j.csl.2016.02.002.

[42] Thomas Merritt et al. "Deep neural network-guided unit selection synthesis". In: (Mar. 2016), pp. 5145–5149. DOI: 10.1109/ICASSP.2016.7472658.

[43] Masanori MORISE, Fumiya YOKOMORI, and Kenji Ozawa. "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications". In: *IEICE Transactions on Information and Systems* E99.D (July 2016), pp. 1877–1884. DOI: 10.1587/transinf.2015EDP7457.

[44] R.B. Randall. "A History of Cepstrum Analysis and its Application to Mechanical Problems". In: *Mechanical Systems and Signal Processing* 97 (Dec. 2016). DOI: 10.1016/j.ymssp.2016.12.026.

[45] Zhizheng Wu, Oliver Watts, and Simon King. "Merlin: An open source neural network speech synthesis system". In: *9th ISCA Speech Synthesis Workshop*. 2016, pp. 202–207.

[46] Sri Harsha Dumpala and K N R K Alluri. "An Algorithm for Detection of Breath Sounds in Spontaneous Speech with Application to Speaker Recognition". In: (Aug. 2017), pp. 98–108. DOI: 10.1007/978-3-319-66429-3_9.

[47] Jose Gonzalez Lopez et al. "Direct Speech Reconstruction From Articulatory Sensor Data by Machine Learning". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (Dec. 2017), pp. 2362–2374. DOI: 10.1109/TASLP.2017.2757263.

[48] Jose Gonzalez Lopez et al. "Evaluation of a Silent Speech Interface Based on Magnetic Sensing and Deep Learning for a Phonetically Rich Vocabulary". In: (Aug. 2017), pp. 3986–3990. DOI: 10.21437/Interspeech.2017-802.

[49] Myungjong Kim et al. "Speaker-Independent Silent Speech Recognition From Flesh-Point Articulatory Movements Using an LSTM Neural Network". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (Dec. 2017), pp. 2323–2336. DOI: 10.1109/TASLP.2017.2758999.

[50] Conor Ransome and C Ransome. "The Fermi Paradox and Galactic Habitability (Masters thesis)". PhD thesis. Apr. 2017.

[51] Hojjat Salehinejad et al. "Recent advances in recurrent neural networks". In: *arXiv preprint arXiv:1801.01078* (2017).

[52] Sandra Vieira, Walter Pinaya, and Andrea Mechelli. "Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications". In: *Neuroscience Biobehavioral Reviews* 74 (Jan. 2017). DOI: 10.1016/j.neubiorev.2017.01.002.

[53] Yuxuan Wang et al. "Tacotron: A fully end-to-end text-to-speech synthesis model". In: *arXiv preprint arXiv:1703.10135* 164 (2017).

[54] Miguel Angrick et al. "Speech Synthesis from ECoG using Densely Connected 3D Convolutional Neural Networks:" in: (Nov. 2018). DOI: 10.1101/478644.

[55] Gopala Anumanchipalli, Josh Chartier, and Edward Chang. "Speech synthesis from neural decoding of spoken sentences". In: *Nature* 568 (Apr. 2019), pp. 493–498. DOI: 10.1038/s41586-019-1119-1.

[56] Christian Herff et al. "Generating Natural, Intelligible Speech From Brain Activity in Motor, Premotor, and Inferior Frontal Cortices". In: *Frontiers in Neuroscience* 13 (Nov. 2019), p. 1267. DOI: 10.3389/fnins.2019.01267.

[57] Christian Herff et al. "Generating natural, intelligible speech from brain activity in motor, premotor, and inferior frontal cortices". In: *Frontiers in neuroscience* 13 (2019), p. 469935.

[58] Mohaiminul Islam, Guorong Chen, and Shangzhu Jin. "An overview of neural network". In: *American Journal of Neural Networks and Applications* 5.1 (2019), pp. 7–11.

[59] Sebastian Nagel. "Towards a home-use BCI: fast asynchronous control and robust non-control state detection". In: (Dec. 2019). DOI: 10.15496/publikation-37739.

[60] Han Gyol Yi, Matthew K Leonard, and Edward F Chang. "The encoding of speech sounds in the superior temporal gyrus". In: *Neuron* 102.6 (2019), pp. 1096–1110.

[61] Jose A Gonzalez-Lopez et al. "Multi-view temporal alignment for non-parallel articulatory-to-acoustic speech synthesis". In: *arXiv preprint arXiv:2012.15184* (2020).

[62] Jose A Gonzalez-Lopez et al. "Silent speech interfaces for speech restoration: A review". In: *IEEE access* 8 (2020), pp. 177995–178021.

[63] Miguel Angrick et al. "Towards closed-loop speech synthesis from stereotactic eeg: a unit selection approach". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 1296–1300.

[64] Filip Boltuzic. "Computational methods for argumentation mining of claims in internet discussions". PhD thesis. June 2022.

[65] Matthew E Foster, Ai Leen Choo, and Sara A Smith. "Speech-language disorder severity, academic success, and socioemotional functioning among multilingual and English children in the United States: The National Survey of Children's Health". In: *Frontiers in Psychology* 14 (2023), p. 1096145.

[66] J Anthony Seikel, David G Drumright, and Daniel J Hudock. *Anatomy & physiology for speech, language, and hearing*. Plural Publishing, 2023.