# Interpolation of Perceived Gender in Speech Signals

## Master's thesis

**Authors:**
**Alexander Hagelborn & Jack Hulme Geber**

**Supervisors:**
Filip Elvander
**Assisting supervisors:**
Andreas Jakobsson
Susanna Whitling

A thesis presented for the degree of
Master of Science

Division of mathematical statistics
Lunds Tekniska högskola
Sweden
November 23, 2020

# Abstract

For individuals with gender dysphoria, voice therapy can be an important tool to change characteristics about their voice to align better with their gender identity. This is often done by practising with a speech therapist and can be a long and difficult process. A useful tool in this setting would be software that can generate a voice, based on the patients voice, which lies slightly closer to their desired voice. The patient could then mimic the generated voice in order to train their voice.

The purpose of this thesis is to explore how voices can be digitally modified in order to change how their gender is perceived. The aim is to find a method of interpolation where a voice could gradually be modified to sound like a target voice, and where all intermediate points on the path sound natural. Two methods were evaluated, but only one produced adequate results that were evaluated with a participant survey.

Survey participants listened to voices that are a mix of female and male voices, and rated on a scale how they perceived the gender and if the voices sounded natural. The results show that there is a decrease in how natural the modified voices sound. On average there is a consensus that the perceived gender is changed, however the individual participant results showed that there is a need for improvement.

**Keywords:** speech modelling, speech morph, interpolation, gender dysphoria

# Contents

# Chapter 1

# Introduction

In this thesis we have studied methods of morphing voices to sound more male or female, with applications to voice therapy for people with gender dysphoria. Our report is structured as follows: First, we give a background to the problem and state the project goal. After the goal is stated, some relevant mathematical theory is presented. This is followed by two chapter that present the deep learning and the source-filter approaches to the problem respectively and their results. Chapter 5 treats the survey evaluation of the morphed voices and a chapter follows with a discussion of the results and suggestions for future research.

## 1.1 Gender dysphoria

*Transgender* refers to individuals whose gender identity doesn't match their assigned sex at birth and is the opposite of *cisgender/cis*. A transgender woman is a woman who was assigned man at birth, and a transgender man a man who was assigned woman at birth [1].

Transgender individuals can experience a lot of distress due to the mismatch in gender identity and assigned sex and this is known as **gender dysphoria** (GD). GD is associated with anxiety, stress, social isolation and an increased risk for suicide, and the American Psychiatric Association deems the presence of *clinically significant distress* a symptom central to the condition. It is common for people with gender dysphoria to experience disconnect and discomfort with their body and attributes [2, 3].

According to the American Psychiatric Association, the needs of individuals with gender dysphoria range from wanting support in their gender identity to a desire to transition to the opposite sex [2]. Accommodating treatments include counseling, hormone therapy, sex reassignment surgery and voice therapy [2].

### Voice therapy

The voice is an important tool for communication and we use it to communicate ideas and emotions, but also our identities. A testimony to this are technological systems that can verify speaker identity just by listening to a voice, like the Amazon Alexa®. Because of this, it can be an important attribute to change for a person with gender dysphoria [4].

Voice therapy refers to techniques for modifying the way a person sounds when speaking, and strives to change characteristics like quality, pitch and resonance [4]. These traits can be changed with treatments that involve hormones and even surgical procedures, but adjustments can also be made by practising to speak in a different way [4]. For transgender men, hormonal therapy is effective, as testosterone naturally induces male characteristics, by enlarging the larynx [5]. In contrast, for transgender women, female hormones do not affect the size of the larynx and thus they are the majority of people seeking speech therapy for gender nonconformity [4]. The research shows that it is not enough to only change the pitch into the female range, but that other aspects such as resonance and intonation is also important [4].

### Tools for voice therapy assistance

A tool where a patient could record their voice and then play it back slightly modified in a direction towards a target voice would be of use for speech therapists working with people with gender dysphoria. This could aid by letting the patient imitate the modified voice, and thus practise in smaller steps. There are existing smartphone applications that track the pitch of the user, but not other characteristics of the voice [6].

## 1.2 The human voice

In order to properly describe what a voice is, a brief background in the anatomy of the speech production system is needed. A diagram of the vocal tract and its constituent parts is shown in Figure 1.1.

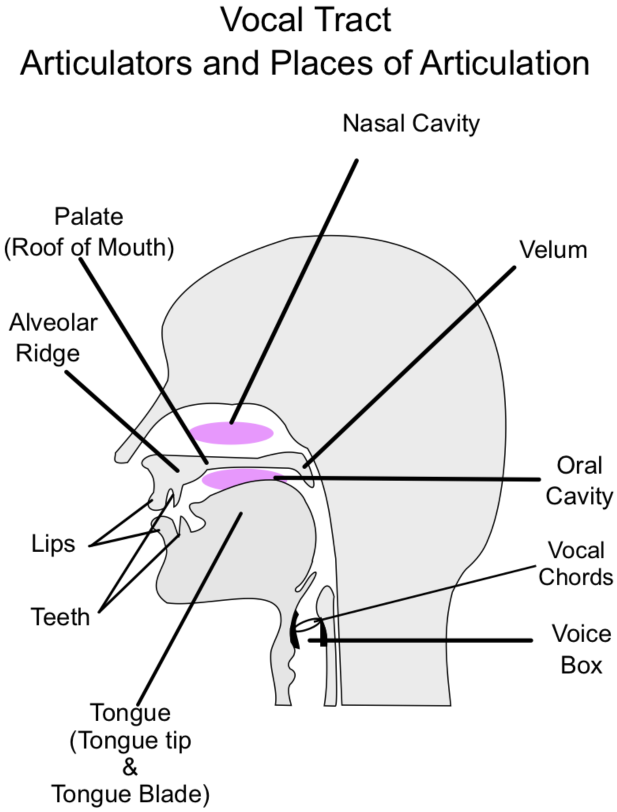


*Figure 1.1: Schematics of human vocal tract. Created by Tavin / Wikimedia Commons / CC BY 3.0*

The human voice production system can produce many different sounds, such as whisper, scream and laughter. The voice is also used to communicate using language; what we call speech. There are two predominant classes of sounds used in speech; voiced and unvoiced speech [7]. Voiced speech is produced when the vocal cords are taut, air from the lungs then cause them to vibrate, which in turn results in *glottal pulses* of air traveling through the vocal tract [8]. The frequency of the vocal cord vibration and in extension the frequency of the glottal pulses is what determines the *fundamental frequency*, or *pitch*, in the produced speech [8]. The acoustics of the resulting speech is further altered when the glottal pulses travel further along the vocal tract, through the mouth and nasal cavities [8]. In unvoiced speech there are no periodic pulses generated from the vocal cords [8]. Unvoiced sounds include fricatives and consonant sounds whereas voiced speech is consists of vowel sounds or singing [8].

The shape of the vocal tract impacts the sound of the voice and adjusting the shape, by moving the tongue, lips, jaw or larynx is what allows humans to produce different vowel sounds [8]. The vocal tract acts as a resonator for the sound produced at the vocal cords, and changing the vocal tract shape changes its resonating characteristics [8]. The resonant frequencies tend to be about 20% lower for cis men than for cis women [9]. It is hypothesised that this difference is due to differences in physical dimensions of the vocal tract [8]. However, the difference in dimensions does not fully explain the gender difference in the resonances [8]. Accordingly, another hypothesis is that there are gender dialects that cause different articulation of vowels [8].

The voice production also differs at the vocal cords; the vocal cords tends to be longer in cis men than in cis women, which is the predominant reason males tend have lower pitch voices than females [10]. Further, the dynamics of the vocal cords also affect the shape of the pulses, resulting in further differences in speech [8]. For example, if the vocal cords do not fully close, the voice will sound leaky or breathy [8].

## Features of a voice

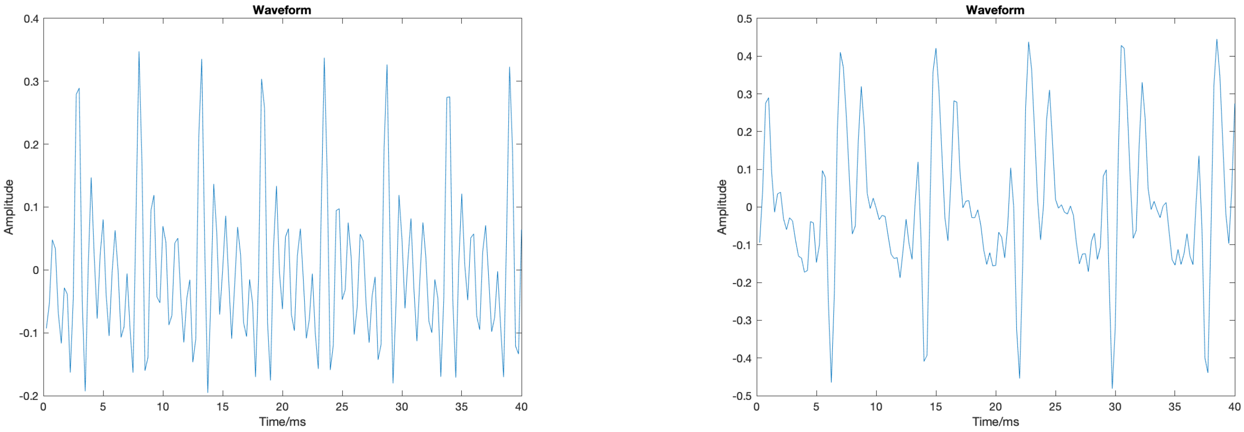
In figures 1.2 and 1.3 speech signals are shown in time and frequency respectively for both a female and a male. In the time domain, one can discern a periodic structure and some variations in the amplitude. In the frequency domain, four important features are visible:

- The fundamental frequency, $f_0$: The frequency we perceive, and usually the first strong peak in the spectrum [8].
- Overtones/Harmonics: Multiples of the fundamental frequency that arise when the vocal cords vibrate [8].
- The smooth spectral shape: The overarching shape of the spectrum.
- Formants: The frequency bands where there are peaks in the smooth spectral shape due to the vocal tract resonances [8].

If we were to listen to these signals, they would immediately reveal two different people. They would also likely be perceived as being of different gender. In the frequency domain it is obvious that all of the previously mentioned frequency characteristics (fundamental frequency, overtones, spectral shape and formants) are different.

The formants are a result of the vocal tract resonances and there are four to five formants of importance. The frequency at the peak value of a formant is known as the formant frequency. The formant's effect on the sound is to amplify the overtones that are close to the formant frequency, which affects the shape of the smooth spectrum [8].

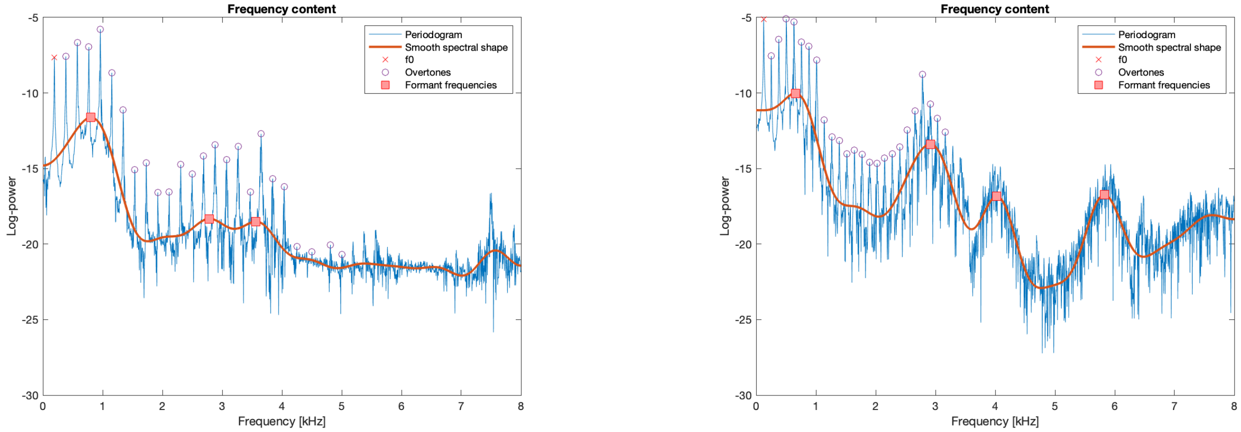
In appendix B more examples of the frequency representations are shown for a more complete picture of the individual differences between voices.



(a) Female

(b) Male

*Figure 1.2: Audio waveforms*



(a) Female

(b) Male

*Figure 1.3: Power spectral density for the waveforms in Figure (1.2). Fundamental frequency, overtones and estimated formant frequencies are marked.*

There are other features that describe a voice. Jitter and shimmer, small variations in the fundamental frequency and the amplitude of the fundamental frequency respectively for example. These features have been used in order to improve the performance of voice identification systems [11], which shows that the features contain important information regarding the identity of the speaker.

## 1.3 Project goal

The goal of this project is to explore different methods of digitally morphing voices to change how their gender identity is perceived. The research questions we study are:

- Can spectral methods be used to find a transformation of a voice from female to male and vice versa?
- Is it possible to find an interpolation between two voices of different gender?

We believe that it is important in a voice therapy setting that the morphed voices sound natural. Synthesized voices can be perceived as unnatural and in the article "Measuring the naturalness of synthetic speech", the authors write:

> "There is no extant objective definition of naturalness that we are aware of - it is a voice quality that is purely subjective. Thus there is no filter or signal processing algorithm that we can apply to a sample of speech that will yield a measure of naturalness." [12]

The lack of a measure presents problems not only for evaluating final results, but also for comparing how well different methods work and subjective evaluation has been used in order to compare results. In this thesis, we will consider a voice natural if it is perceived as coming from a real human.

By interpolation of a voice we mean a system capable of generating voices at different points in between two voices with respect to their perceived identity. We believe that it is reasonable to expect the formants to be placed part way between the formants of the original voices, in both frequency and amplitude. Further, we think that the pitch of an intermediate voice ought to lie in between the pitch of the original signals. However, exactly how these formants and fundamental frequency should be altered is not clear.

Voice data was provided for the project, recorded with mobile phone at patient sessions with a voice therapist, where the patient was voicing an "Ah". Some of this data was discarded as some patients had undergone vocal tract surgery. We also collected some additional voice data from family and friends. In total, the data set consisted of 123 samples: 56 females and 67 males. Our goals of the project were restricted to finding interpolation on the provided data.

## 1.4 Contributions from this work

In this thesis we explore two methods for solving the voice interpolation problem. The first method uses deep learning. It explores the idea of generating a voice from a speaker embedding in a latent space, as previous research in image generation has shown that interpolation in such spaces can yield realistic results [13].

The second method we tried is based on the commonly used source-filter model for speech. In this system we implemented some existing filter estimation and interpolation techniques, and provide a new technique for filter morphing. The methods were integrated into a well known pitch shifting pipeline. The quality of the resulting audio was assessed by the authors using the different combinations of estimation and interpolation.

We created a survey containing the audio of many different morphs using the combination of techniques we thought produced the highest quality interpolations. The participants in the survey were asked to rate the perceived gender of the speaker and the naturalness of the voice. The survey was created to give an idea of the interpolation quality and to give insights of how future surveys may be constructed.

# Chapter 2

# Mathematical background

## 2.1 Stationarity of signals

For modelling signals, the notion of stationarity is important. In *An Introduction to Time Series Analysis* by A. Jakobsson a stochastic process or signal is defined as *wide sense stationary* (WSS) if [14]

- The mean of the process is constant and finite.
- The autocovariance depends only on the time lag and not on the time itself.
- The variance of the process is finite.

The frequency characteristics are determined by the autocovariance [15]. Which in conjunction with the second condition means that the frequency characteristics cannot change over time. Many of the spectral analysis methods used in speech analysis assume that the signals in question are WSS, but speech signals typically are not as they change over time [7]. However, they can be approximated as such on a short time scale of 20-30 ms [7].

## 2.2 Frequency representations

The frequency content of a time domain signal can be estimated using the *Discrete Fourier Transform* or (DFT). For a discrete signal $s(n)$ with $N$ samples this transform can be defined as

$$\hat{s}(\omega) = \mathscr{F}[s(n)](\omega) = \sum_{n=0}^{N-1} s(n)e^{-i\omega n/N} \tag{2.1}$$

with $i$ as the imaginary unit [16].

The absolute value of the Fourier transform $|\hat{s}(\omega)|$ and the argument $\arg(\hat{s}(\omega))$ correspond to the amplitude and phase of the sinusoidal component of the signal with the angular frequency $\omega$ [17]. In order to convert angular frequency to Hertz we use the relationship

$$f = \frac{f_s}{2\pi}\omega \tag{2.2}$$

where $f_s$ is the sampling frequency [17].

For finite length signals the transform often contain artifacts known as lobes [14]. Mainlobes which smear and smooth the frequency content among nearby points in the transform and sidelobes which causes energy to leak from frequencies where there is actual energy to frequencies where there is less energy [14]. In order to reduce the impact of the lobes for a given application, the signal can be multiplied by a window function $w(n)$ [18]. For voice modeling purposes the Hanning window is a common choice [7]. Zeros can be appended to the end of a signal before it is Fourier transformed, a practice which is called *padding* [16]. Padding results in a finer frequency grid which may yield more accurate representation of peaks in the transform [16].

Often, the periodogram $P(\omega)$ is used as an estimate of the the power spectral density (PSD), or the energy contained in each frequency of the signal, and is defined as [18]

$$P(\omega) = \frac{1}{N}|\hat{s}(\omega)|^2. \tag{2.3}$$

As mentioned previously, speech is considered to be stationary only on short time frames. For such signals the *Short Time Fourier Transform* or *STFT* can be used. The transform is performed by dividing the signal into short segments of equal length. The frames often overlap and there is a set length $d$ between the beginning of each window [19]. A window function is applied to each of the frames and the Fourier transform is performed. For discrete signals the STFT, $S(n,\omega)$, can be written as

$$S(n,\omega) = \sum_{k=0}^{N-1} x(k)w(k-nd)e^{-iwk}, \tag{2.4}$$

describing the amplitude and phase in the $n'$th frame, at the angular frequency $\omega$ [19]. There is a trade-of concerning the length of the windows; shorter frames result in more smeared frequency resolution and longer windows leads to a poor time resolution [20].

The time-frequency information of the STFT can be visualized as a spectrogram. The spectrogram visualizes the relative power of the frequencies (thus discarding all phase information) and is defined as [20]

$$\hat{S}(n,\omega) = |S(n,\omega)|^2. \tag{2.5}$$

### The Mel spectrogram

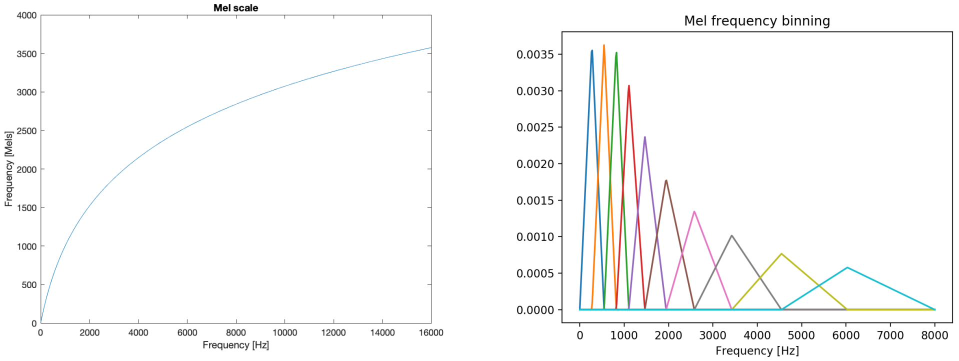
The human auditory perception is logarithmic for changes in amplitude, via the well known decibel scale [21]. Likewise, we perceive changes in frequency in a non linear fashion [22].

In an attempt to capture how we perceive frequencies, the mel scale was invented based on listening experiments [23]. The mel scale is a quasi-logarithmic frequency scale constructed so that pitch increments are perceived the same anywhere on the scale and conversion from Hertz is often calculated with the formula

$$f_{mel} = 2595\log_{10}(1 + \frac{f_{hz}}{700}),$$

which is also visualized in Figure 2.1a [24].

In the python audio package *Librosa*, a mel filter bank can be created by specifying the sample rate and the number of filters and this can be used to bin spectrograms [25]. The filters are spaced at equal perceptive distance and are normalized so that their area is equal to 1 and can be seen in Figure 2.1b. As can be seen, projecting onto a mel frequency basis compresses frequency information more at higher frequencies. The mel filter bank is applied to the spectrogram in order to transform the linear frequency scale spectrogram to a mel-scale spectrogram. The bins are also referred to as channels. The mel spectrogram is frequently used as input feature in deep learning, see [26, 27, 28].



(a) Approximate mel frequency scale.

(b) Mel filter bank with 10 filters

*Figure 2.1: Mapping from linear frequency to mel frequency (left) and mel filter bank (right)*

## 2.3 Linear predictive speech coding

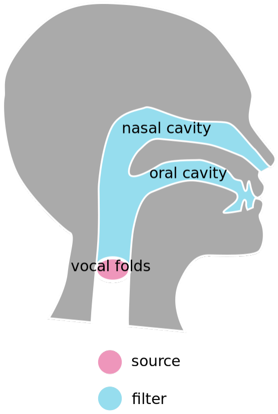


*Figure 2.2: Image showing how the source filter models try to separate the effects of the anatomy of the vocal tract. Created by Emflazie / Wikimedia Commons / CC-BY-SA-3.0*

The bio mechanical production of speech was used as an inspiration for the Linear prediction coefficient or LPC speech coder proposed in the 1970s [29]. In the analysis stage of the LPC pipeline the vocal tract acoustics are modelled as an autoregressive filter on a short time interval, typically overlapping windows that are $20 - 30$ ms [7, 29]. In the synthesis stage, an excitation signal representing the pulses created by the vocal folds is passed through the time varying filters. The pink area in Figure 2.2 shows where the excitation signal is created and the blue area is modeled by the autoregressive filters.

In chapter 4 we make use of the source-filter model. The first objective is to separate the source and filter characteristics so that they can be modified independently. We think that ideally, the spectrum of the residual is a series of Dirac pulses at multiples of the fundamental frequency on a noise floor representing the glottal excitation and the flow of air, which are then amplified by the vocal tract filter.

The roots of the filter polynomial $A(z)$ are often called the poles of the system and have an important relationship to the acoustics. For one, if the poles of the polynomial lie outside the unit circle there will be feedback in the system causing the magnitude of the output to tend to infinity [14]. Filters with all poles inside the unit circle are called *stable* whereas filters with poles outside the unit circle are called *unstable* [14]. Another relation between the roots of the polynomial and the effect of the filter is a relation of the power spectral density of the excitation and the output of the system. Under the assumption that the signal is wide sense stationary the following relation holds

$$\phi_y = \left| \frac{1}{A(\omega)} \right|^2 \phi_x \tag{2.6}$$

where $\phi_y$, $\phi_x$ are the power spectral densities of $y$, $x$ and $A(\omega) = A(z)|_{z=e^{iw}}$ [14]. The filter polynomial is monic [14], and therefore we get

$$\left| \frac{1}{A(\omega)} \right| = \left| \frac{1}{\prod\limits_{re^{i\theta}\in R} (1 - \frac{e^{-i\omega}}{re^{-i\theta}})} \right| = \prod_{re^{i\theta}\in R} \frac{1}{\left|(1 - \frac{e^{i(\theta-\omega)}}{r})\right|} \tag{2.7}$$

where $R$ is the set of roots of $A(z)$. Note that for a given root the denominator is small when the angle $\theta - \omega$ is small, and large when $\theta - \omega$ is large. Note also that the effect is increased when the pole is close to the unit circle. This relationship shows that the power spectral density is amplified at frequencies close to poles in the filter polynomial and attenuated at frequencies with large angles to the poles [18].

As seen above the roots can give important information about the acoustics of a filter, however, the problem of finding roots of a polynomial of order 5 or higher does not have a general solution which means an approximation has to be used. In this thesis the eigenvalues of the companion matrix are used when finding roots of polynomials [30].

## 2.4 Optimal mass transport

The optimal mass transport (OMT) distances are a set of metrics for computing the distance between two probability distributions [31]. An OMT metric gives the total distance that all the particles have traveled when the particles in one of the distributions are reassigned to the other, under the condition that the reassignment plan minimizes the total distance [32]. Here 'particles' refers to each infinitesimal mass of the distribution. In a future chapter, the optimal mass transport formulation is used to interpolate the vocal tract characteristics of two voices.

The *Earth Mover's distance* or *Wasserstein metric* is the special case where the $L^1$ metric is used to measure the distance each particle has traveled [31]. The name comes from the fact that it can be thought of as giving a distance for moving an initial mound of earth to another shape and place under the conditions that each grain is moved in an optimal way [33]. A plan can be denoted as a function $\pi(x,y)$ defining how much mass, is moved between distributions $X(x)$ and $Y(y)$. The definition of the Wasserstein metric plan can be stated more formally as follows: For the discrete probability distributions $X(x), Y(y)$ the optimal plan in the Wasserstein metric, $\pi^*$, is defined as [31, 34]

$$\pi^* = \underset{\pi}{\operatorname{argmin}} \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |x-y|\pi(x,y) \tag{2.8}$$

subject to the mass conservation constraint

$$\sum_{y=-\infty}^{\infty} \pi(x,y) = X(x) \text{ and } \sum_{-\infty}^{\infty} \pi(x,y) = Y(y). \tag{2.9}$$

The optimal mass transport formulation has previously been used to slide between the STFTs of two signals for musical applications [32]. For one dimensional distributions $X$ and $Y$, no mass can pass over any other mass in the optimal plan, since a lower cost could be achieved by swapping the reassignments [32]. This also means that the optimal Wasserstein plan is equivalent to any optimal mass transport plan using an $L^p$ norm. Further, the restriction results in a fast recursive algorithm to find the optimal path $\pi^*$ between the two discrete distributions $X$ and $Y$, see Algorithm 1 [32].

**Algorithm 1** Optimal transport plan

```
    Input Probability distributions X,Y
    Output Optimal transport plan $\pi^*$
$i$,$j$ ← 0,0
$\pi^*$ ← 0                                         ▷ Initialize transport matrix
$\rho_X, \rho_Y \leftarrow |X_0|, |Y_0|$            ▷ Assign the initial masses
while True do
    if $\rho_X < \rho_Y$ then
        $\pi^*_{i,j} \leftarrow \rho_X$             ▷ Assign as much as mass as possible
        $i \leftarrow i + 1$
        if $i$ = length($X$) then
            break                                   ▷ End the algorithm
        else
            $\rho_X \leftarrow X_i$                 ▷ Refill the bin
            $\rho_Y \leftarrow \rho_Y - \rho_X$     ▷ Complete the assignment
        end if
        Symmetric to other case.
    end if
end while
```

Given a transport plan $\pi^*$ and any interpolation coefficient $\tau \in [0,1]$ an interpolated probability distribution, $Z_\tau(w)$, can be generated by

$$Z_\tau(z) = \sum_{x,y \in I(z,\tau)} \pi^*(x,y) \tag{2.10}$$

where the set $I(z,\tau) = \{x,y|(1-\tau)x + \tau y = z\}$ [32]. This interpolation places a mass reassigned from $x$ to $y$ at the point $(1-\tau)x+\tau y$. There is no guarantee that the displaced masses are placed on query points and we use a linear interpolation to split the masses between the bins.

Figure 2.3 shows the optimal mass transport interpolation and linear interpolation of two distributions. The distributions are horizontally shifted copies. The blue and orange graphs depict the source and target distributions $X$ and $Y$. The Figure shows that a linear interpolation creates two objects while the optimal mass transport shifts the object existing in the first image and transports all the points to the goal function.

Figure 2.4 displays another optimal transport interpolation. This time the input distributions are not simply shifted along the horizontal axis. As can be seen in the middle three graphs the interpolation is not smooth even though both the input distributions are smooth. As we will use the optimal plan to interpolate between smoothed spectra representing vocal tract characteristics, we expect the result to be relatively smooth as well. A smoothing filter could be used to fix this problem but we propose our own solution in chapter 4.

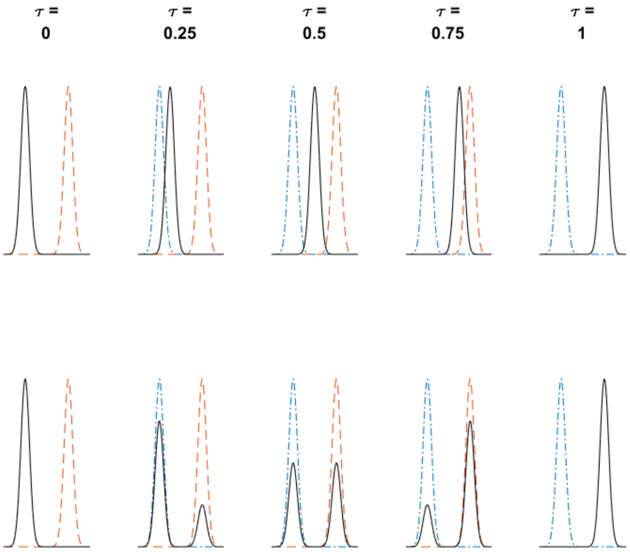


*Figure 2.3: Optimal transport interpolation (top) and linear interpolation (bottom) between the blue and orange functions.*

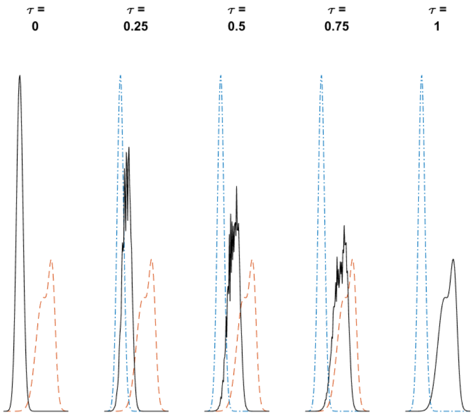


*Figure 2.4: Optimal transport interpolation between two smooth functions.*

# Chapter 3

# Deep learning

In this chapter we explore deep learning as an approach to the interpolation problem. The approach is based on the idea of extracting unique speaker embeddings from an audio file and being able to reconstruct a mel spectrogram from this embedding. If such a model could be trained, a speaker embedding can be perturbed to produce new mel spectrograms. These can then be converted to audio using the Griffin-Lim algorithm [19].

## Artificial neural networks

Artificial neural networks (ANNs) are network like architectures that have the ability to learn a function from experience and the early models were invented based on a a simple model of how neurons in the brain work [35].

The simplest neural network, the one layer perceptron, shown in Figure 3.1, simply computes a weighted linear combination of the input and then passes it through an activation function $a$ [35]. The output $\hat{y}$ for this network becomes

$$\hat{y} = a(\sum_{k=1}^{4} w_k x_k)$$

where $w_k$ denotes weight $k$. Normally, a bias term $b$ is also included [35].

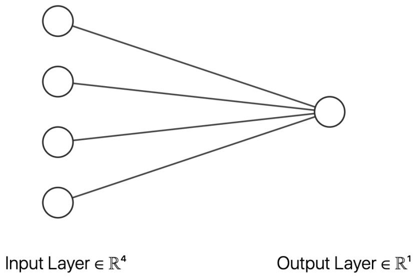


*Figure 3.1: One layer perceptron*

The ANN's function is to transform the input $\mathbf{x}$ to an output $\hat{\mathbf{y}}$ and its objective is to tune its weights so that it can do it well; this process is referred to as training [35]. To do this, the network has to be given a mathematical objective, a *loss function* $L(\hat{\mathbf{y}})$, which measures how well a given forward pass is, compared to a target $\mathbf{y}$ [35]. The network keeps track of all operations so that the gradient of $L(\hat{\mathbf{y}})$ with respect to all weights is known, making it possible to use an iterative optimization algorithm to reduce the loss function, hence training the network [35].

For an in-depth description of deep learning architectures like convolutional and recurrent blocks, optimization algorithms and loss functions we refer to [35].

### The promise of deep learning

In the past years, generative networks have been more deeply explored. For natural language processing there are now famous models that can produce coherent articles from just an input sentence [36]. One kind of model architecture known as Variational Auto Encoders, or VAEs for short, are trained to compress(or encode) data into a latent space and then regenerate the original sample. VAEs typically learn a Gaussian distribution in the latent space, which can then be sampled from to generate new data samples [35].

This type of model has been used to generate remarkably real looking human faces [13]. But what truly makes the idea of sampling from a latent space interesting is that this research has shown that the latent space can support vector arithmetic and interpolation [13]. On the author's github page, they show how they can transform faces of men into faces of women [13].

If we can create a model that encodes and decodes voices through a latent space, then we can experiment with perturbing a specific individuals representation in the latent space to modify their voice.

Deep learning generally requires large data sets, but we justified this approach since the data we had was very simple as it was only the voicing of one vowel.

## 3.1 Previous research in speech generation

In 2017, a Google research team presented a text-to-speech(TTS) model called Tacotron2. The system consists of two parts: a synthesizer and a vocoder. The synthesizer is trained on a data set of speech from a specific person, and tries to predict a mel spectrogram from a string of text. A vocoder is trained to produce a waveform from these mel spectrograms and is trained on a single person [26].

In 2018, this model was extended to include a speaker encoder, making it possible to generate speech from different target voices. The extension is simple enough: the aforementioned synthesizer now takes a second input in the form of a speaker embedding. This model is known as SV2TTS [27].

The speaker embeddings are produced by an encoder. The encoder is trained separately to compress utterances into a latent space using a loss function invented for speaker verification, known as Generalised End-to-End Loss (GE2E) [37]. The loss function forms a cosine similarity matrix, where it tries to group utterances of the same person and separate them from the utterances of the closest person [37].

What especially caught our eye in the paper was the fact that after a non-linear dimension reduction to two dimension using UMAP the male and female speaker embeddings were linearly separated. And since the embeddings contained enough information to generate any text from a voice, it seemed plausible to perform interpolation in the latent space.

In his master's thesis, Corentin presented a pretrained, open source version of this system with some slight modifications to make it computationally lighter [38]. Since we had limited access to data and computation resources, we decided to investigate if we could use the pretrained encoder from his open source system.

## 3.2 SV2TTS

This section describes the open source version of the SV2TTS model in short since its architecture provided a start for our own model. SV2TTS consists of three different parts, all trained separately:

1. Speaker encoder
2. Synthesizer
3. Vocoder

### 3.2.1 Speaker encoder

The role of the speaker encoder is to capture a meaningful representation of an individuals voice; to do this it takes a mel spectrogram as input and outputs a 256 dimensional vector [27]. The architecture is simple: the mel spectrogram is fed into a three layer LSTM block, after which the last hidden layer is projected onto the 256 dimensional vector [38]. This is shown in Figure 3.2.

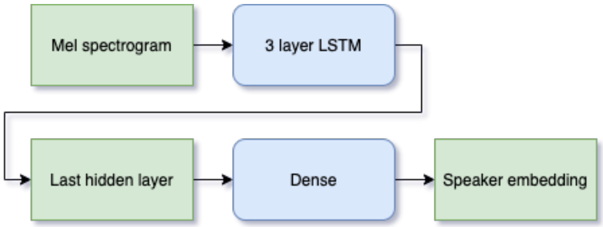


*Figure 3.2: Speaker encoder architecture in SV2TTS*

The three LSTM layers all contain 256 hidden nodes each and the fully connected layer has a ReLU activation function [38]:

$$ReLU(x) = \begin{cases} x & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases}$$

Due to the threshholding of the ReLU, the embeddings are sparsely activated [38]. This output is then $L2$ normalized [38].

The inputs are 40 channel mel spectrograms computed on 16kHz audio files [38]. The window size and step length for the spectrogram estimation are 25 and 10 ms respectively and the number of FFT points is 2048 [38]. The open source version of the encoder was trained 1.56M steps (20 days with a single GPU) with a batch size of 64 [38].

### Generalized End-to-End loss

This is short presentation of the loss function used in the encoder, as originally presented in the paper *Generalized End-to-End loss for Speaker Verification*, 2016, by Wan et al [28]. It was originally developed for speaker verification tasks but was later incorporated in the encoder of SV2TTS.

During training, the encoder takes $M$ utterances from $N$ different speakers and produces embeddings $\mathbf{e}_{ij}$, $1 \leq i \leq N$, $1 \leq j \leq M$ , and for each speaker $i$ a centroid can be computed as [28]

$$\boldsymbol{c}_i = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{e}_{ij} \tag{3.1}$$

.

To compare distances in the latent space, a scaled cosine similarity is used to define the similarity matrix $\mathbf{S}$(see equation (3.2)) which compares each embedding $\mathbf{e}_{ij}$ with each centroid $\mathbf{c}_k$ [28]. The weight $w$ and bias $b$ in this equation are both trainable parameters [28].

$$\boldsymbol{S}_{ij,k} = w \cdot \cos(\boldsymbol{e}_{ij}, \boldsymbol{c}_k) + b, \qquad w > 0 \tag{3.2}$$

The authors of [28] propose defining the loss for one embedding by penalizing the distance to its own centroid, and rewarding the distance to the closest other centroid. When minimizing this loss, it should result in a force that pulls it close to its own centroid while pushing it away from the closest other centroid [28]. It is formulated as

$$L(\boldsymbol{e}_{ij}) = 1 - \sigma(\boldsymbol{S}_{ij,j}) + \max_{\substack{1\leq k\leq N\\ k\neq i}} \sigma(\boldsymbol{S}_{ij,k}) \tag{3.3}$$

where $\sigma$ denotes the sigmoid function, $\sigma(y) = \frac{1}{1+e^{-y}}$.

In [28] the authors note that removing $\mathbf{e}_{ij}$ in the computation of $\mathbf{c}_i$ gives desirable effects during training and therefore reformulated the similarity matrix as

$$\boldsymbol{S}_{ij,k} = \begin{cases} w \cdot \cos(\boldsymbol{e}_{ij}, \boldsymbol{c}_i^{(-j)}) + b, & \text{if } k = j \\ w \cdot \cos(\boldsymbol{e}_{ij}, \boldsymbol{c}_k) + b, & \text{otherwise} \end{cases} \tag{3.4}$$

where $\mathbf{c}_i^{(-j)}$ are defined as

$$\boldsymbol{c}_i^{(-j)} = \frac{1}{M-1}\sum_{\substack{m=1\\ m\neq j}}^{M}\boldsymbol{e}_{im}$$

The loss over the entire training batch $\mathbf{x}$ is the sum of the individual losses, i.e.,[28]

$$L_{\text{GE2E}}(\boldsymbol{x}) = \sum_{i=1}^{N}\sum_{j=1}^{M} L(\boldsymbol{e}_{ij}). \tag{3.5}$$

### 3.2.2 Synthesizer

The SV2TTS synthesizer takes as input a string of text and a speaker embedding and produces a log-mel spectrogram [27]. A log-mel spectrogram simply refers to a mel spectrogram where the amplitude is compressed by some logarithmic function. A graph showing the flow of information is displayed in Figure 3.3.

The first three blocks(shown in blue) are used for text encoding [38]. They output a sequence of features that is then concatenated with the speaker embeddings before going into an attention block [38].

The blocks shown in the upper part of the figure(coloured orange) form the decoder, and produce the actual mel spectrogram. It predicts a coarse frame of the mel spectrogram at a time, and uses the prediction as feedback for the next frame prediction [27]. To aid the prediction, a convolutional post-net adds finer details to the predicted frame [27]. We refer to the entire coarse spectrogram as $\tilde{\mathbf{y}}$ and the entire output spectrogram as $\hat{\mathbf{y}}$.

The attention block(shown in gray) is what connects the encoder and decoder of the synthesizer. It is a mechanism that learns relationships between the events in time in the input sequence and output sequence [38]. Without an attention mechanism, the decoder would only see the encoded feature vector at each time step.

To allow for dynamic duration of generated speech, the authors of [27] include a prediction of a stop token which is used at inference to determine if it is time to stop predicting spectrogram frames. This will not be used in our model.

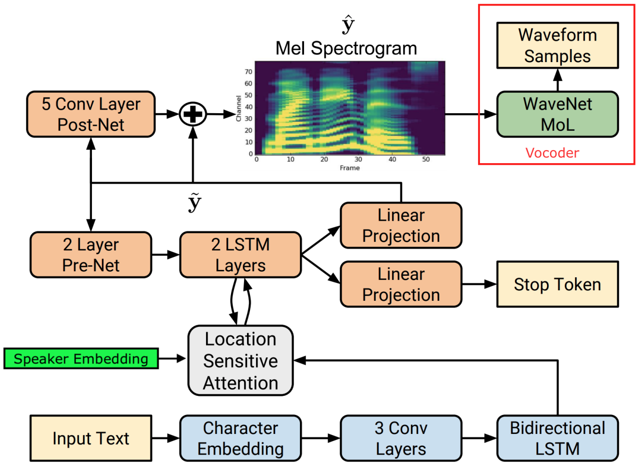


*Figure 3.3: SV2TTS synthesizer and vocoder architecture. Vocoder highlighted in red. Image taken from original Tacotron2 paper with author's consent.*

As loss function for the synthesizer the authors of [27] propose the sum of the mean square error(MSE) of the predicted spectrogram before ($\tilde{\mathbf{y}}$) and after the postnet ($\hat{\mathbf{y}}$):

$$L_{MSE} = \text{MSE}(\tilde{\mathbf{y}},\mathbf{y}) + \text{MSE}(\hat{\mathbf{y}},\mathbf{y})$$

where

$$\text{MSE}(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{n}\sum_{k=1}^{n}(\hat{y}_k - y_k)^2.$$

### 3.2.3 Vocoder

SV2TTS uses an existing neural vocoder architecture known as WaveNet which synthesizes a waveform from a mel spectrogram [39]. An alternative is to use the iterative Griffin-Lim (GL) algorithm, which estimates the time domain signal using only the spectrogram, without needing any training [19]. Wavenet has the advantage of producing better quality audio, but needs to be specifically trained for outputs from a given synthesizer [39]. We found that GL produced results sufficient to determine if the spectrogram corresponded to a voice. Comparing 16kHz audio and audio generated from its mel spectrogram using GL, we found no perceptible loss of quality when using more than 200 mel channels.

## 3.3 Experiments

For detailed information on the model architectures we refer to our git repository at `https://bitbucket.org/Hitmonlundgren/tacotron3`.

### 3.3.1 Preprocessing

Since pretrained encoder was trained on 40 channel mel spectrograms from audio sampled at 16 kHz, we down-sampled all of our audio files to 16kHz. The files were also trimmed so that they were five seconds long in such a way that there were no silent periods at the beginning or end.

In the synthesizer part of the SV2TTS network, the authors use log dynamic compression to limit the amplitude range, likely to make the weight optimization easier [27]. Equation (3.6) shows how this is performed, where $\gamma$ is a tunable parameter. Since the authors of the original paper did not reveal their choice of gamma we investigated the effect on a few mel spectrograms. The results are shown in Figure 3.4, with $\gamma$ chosen as 10,100,1000 and 10000 respectively.

$$y_{compressed} = \ln(1 + \gamma * y) \tag{3.6}$$

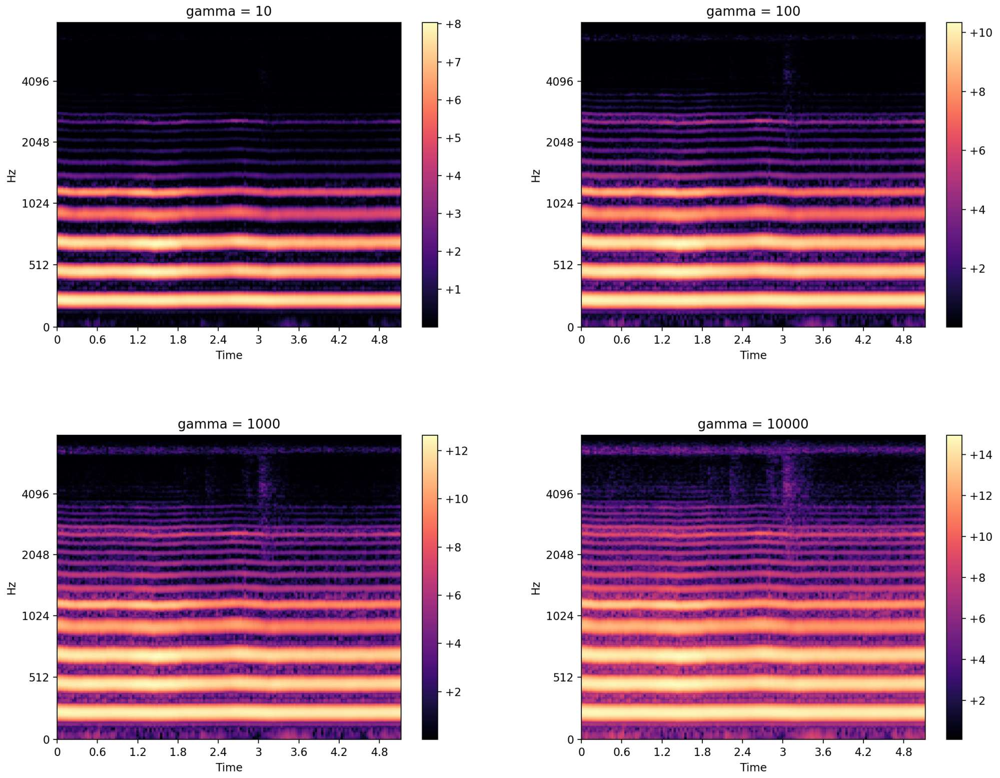


*Figure 3.4: Log dynamic compression of mel spectrograms using four different values of $\gamma$*

Immediately clear is that a higher value of $\gamma$ amplifies noise, whereas a lower value attenuates both noise and other information. The horizontal lines in the mel spectrograms are the overtones of the voice, and thus carry important information about the tone of the voice. Therefore it should be easy for the network to realize their importance; they should be visible in the images.

To choose a good value of $\gamma$ we want to balance compression against noise amplification. We ruled out both $\gamma = 10$ and $\gamma = 10000$ due to the first not compressing the dynamic range enough, and the latter too much. In the end, $\gamma = 1000$ was chosen, after training some easier models revealed this choice to yield lower validation losses after compensating for the compression.

### 3.3.2 Modifying SV2TTS decoder

Starting from the SV2TTS synthesizer described in section 3.2.2, we clearly needed to modify the blocks that were meant to decode text since our objective was to create a model that could reconstruct a log-mel spectrogram.

Although a deep learning model theoretically can learn any abstraction, in practise it is important to exploit the structure of the data to obtain good results [35]. ANNs used for image related tasks use convolutional layers, since neighbouring pixels in an image have a close spacial dependency [35]. In sequence models, recurrent layers like the LSTM and GRU architectures are used since the parsed data has dependency in time [35].

Mel spectrograms are a two dimensional representation of a waveform, frequency vs time. They can thus be seen as either a gray scale image or a multidimensional sequence. Although there is previous work with spectrograms where convolutional networks are used (see [40]), the axes in a spectrogram do not carry the same spatial relationship as in a normal image. Another disadvantageous fact is that frequencies far from each other can be related. Take for example the relationship between the fundamental frequency of a voice and its overtones; a convolutional architecture would have a hard time understanding the relationship between the frequency 440 Hz and and its third overtone at 1760 Hz.

It is therefore more intuitive with a recurrent architecture, as it should should look at each frequency channel developing in time. The voices in our data have approximately constant pitch and seem stationary to some degree, but in reality there is always some degree of both jitter (pitch variation) and shimmer (amplitude variation). These variations are important to capture for the network to reconstruct the mel spectrogram. We exploit the already existing architecture of SV2TTS by removing the text parser with an LSTM block to detect time variations.

To choose the number of features for this LSTM block we have to be cautious: the idea is to use the final model and perturb the speaker embeddings to generate new voices. Thus we do not want the introduced LSTM block to capture any speaker identity, only jitter and shimmer. We restrict the features and decide on five hidden features. The model thus extracts a few time features over the length of the mel spectrogram, and then concatenates the speaker embedding to each of these, making the input to the decoder. This is shown in Figure 3.5.

The synthesizer is left unmodified, apart from removing the stop token prediction as we have data of fixed length. We also try training on mel spectrograms with 80, 256 and 512 channels to see if the frequency resolution was of any importance for the training. For the actual implementation, we use an open source version of the Tacotron2 synthesizer implemented in Pytorch by NVIDIA, and modify it accordingly [41].

Training is done both with the originally proposed MSE loss

$$L_{MSE} = \mathrm{MSE}(\tilde{\mathbf{y}},\mathbf{y}) + \mathrm{MSE}(\hat{\mathbf{y}},\mathbf{y})$$

and with Huber loss which is less sensitive to outliers [42].

$$L_{Huber} = \mathrm{Huber}(\tilde{\mathbf{y}},\mathbf{y}) + \mathrm{Huber}(\hat{\mathbf{y}},\mathbf{y})$$

The Huber loss is a scaled MSE for small errors, and a scaled $L_1$ loss for larger errors and is defined as [42]:

$$\mathrm{Huber}(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} h_i$$

$$h_i = \begin{cases} 0.5(\hat{y}_i - y_i)^2, & \text{if} \quad |\hat{y}_i - y_i| < 1 \\ 0.5|\hat{y}_i - y_i|, & \text{otherwise} \end{cases} \qquad .$$

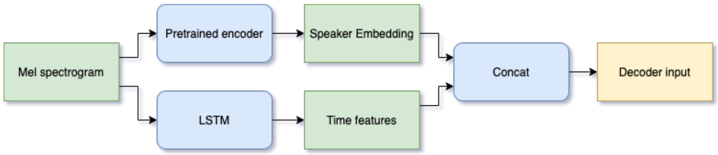


*Figure 3.5: Modified encoder architecture*

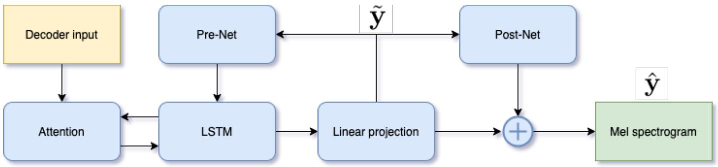


*Figure 3.6: Modified decoder architecture*

### 3.3.3 Exploring pretrained embeddings

We retrieved embeddings for the entire data set in order to explore the potential of the latent space, and we use principal component analysis (PCA) to reduce the dimensionality of the embeddings [43]. A two dimensional PCA projection of the speaker embeddings retrieved using the pretrained encoder are shown in Figure 3.7. We see that there is some separation between males and females, but it is far from total.

To verify how well they were separated in the full 256 dimensional space we fitted a simple perceptron classifier with a linear activation function, which could separate the points completely.

As mentioned before, the ReLU function in the encoder leads to sparsely activated embeddings. When investigating the embeddings, we find that in many cases, as few as 80 out of 256 dimensions are activated. This is likely due to many features that exist in general speech does not exist in our data of voiced vowels. Because of this, we also try training our model with embeddings reduced to 64 dimensions using PCA.

### 3.3.4 Gaussian speaker embeddings

The pretrained encoder was trained on a data set consisting of general speech, unlike ours consisting of one voiced vowel only. We therefore investigate training our own embeddings. Because of how small our data set is, we rule out using the same independent encoder structure as the pretrained encoder, due to how the GE2E loss is implemented. For GE2E loss to be efficient, it is of importance that each batch contains many speakers $N$ to push speaker centroids from each other [28].

Instead we opt for an architecture that uses an LSTM block and then learns a Gaussian distribution by sampling the speaker embedding from it. This architecture is shown in

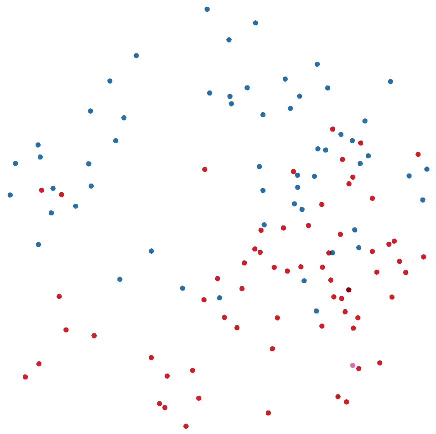


*Figure 3.7: Pretrained encoder embeddings of male(red) and female(blue) voices projected onto 2D with PCA.*

Figure (3.8). The mel spectrogram passes through a three layer LSTM and then the last hidden state is projected onto a mean and a standard deviation, from which the embedding is then sampled. This is a common strategy in VAEs [35].

This encoder is trained together with the synthesizer described in section 3.3.2. The embeddings after training for 1000 epochs are shown in Figure 3.9. They are well separated even in the two dimensional plane. They also have less in group variance compared to the pretrained embeddings, something we think could be problematic when performing an interpolation if the model does not understand the space in between these groups.

### Training

All models train for a maximum of 1000 epochs using the ADAM optimizer. The data set was split into training and validation at a 85:15 ratio. The Griffin-Lim algorithm is then used to evaluate if the predicted mel spectrograms on the validation set sound like voices.

## 3.4 Results

We only present the results from the two best models:

- *Tacotron3*: Non-reduced pretrained embeddings - 256 mel channels - MSE loss
- *Gausstron*: 64 dimensional Gaussian embeddings - 256 mel channels - MSE loss

The models that trained on the pretrained embeddings reduced to 64 dimensions did not show any improvement in validation scores compared to the full 256 dimensional embeddings.

In regards to the number of mel channels in the input spectrograms, the models trained with 80 and 256 have no significant difference in MSE on the predictions, while 512 mel channels. We thus opted for only training with 256 since the Griffin Lim algorithm output benefits from a higher resolution spectrum.

In Figure 3.10a we see a predicted mel spectrogram from the validation set and in Figure 3.10b the real spectrogram by the *Tacotron3* model. The output from *Gausstron*

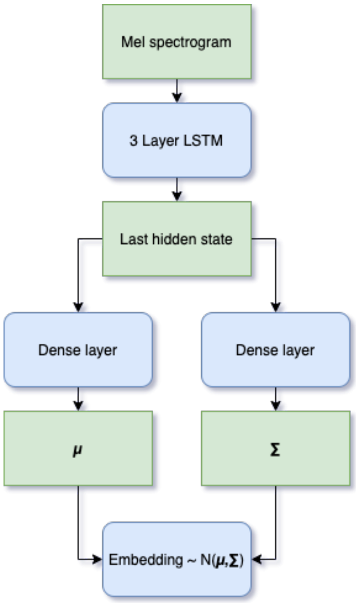


*Figure 3.8: Chart over our modified speaker embedding which samples from a Gaussian distribution in the latent space.*

are shown in Figure 3.11. Both model outputs show some understanding of the spectrogram structure. After generating waveforms from these spectrograms, they sometimes resemble the original voices, but were induced with a lot of noise and sometimes strange artifacts. The predictions are not robust and interpolation only yields noisy audio which does not sound human.

The predicted mel spectrograms on the validation set show that the network has learnt some structure. It seems to understand the general harmonic structure by looking at the images. Unfortunately, it seems to introduce frequency content in between the harmonics, making the evaluated audio extremely noisy. The models had stagnated in training and no experiment with hyper parameters and/or choice of loss function could improve it significantly.

The models do not generalize well enough to reconstruct validation or test data. Because of this, interpolation in the latent space is meaningless due to the over-fitting in the system. We conclude that the data set is too small to spend more time investigating this approach, but believe that it can be revived at a later point in time.

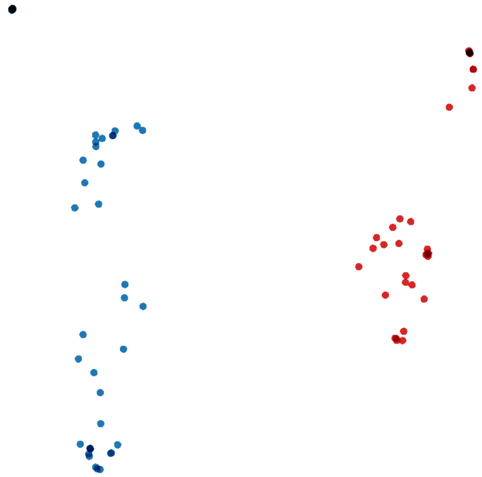
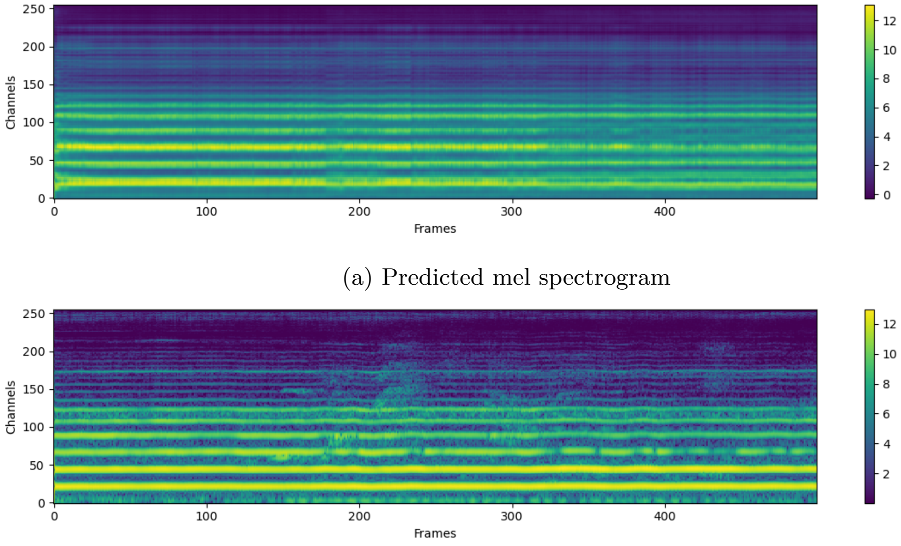


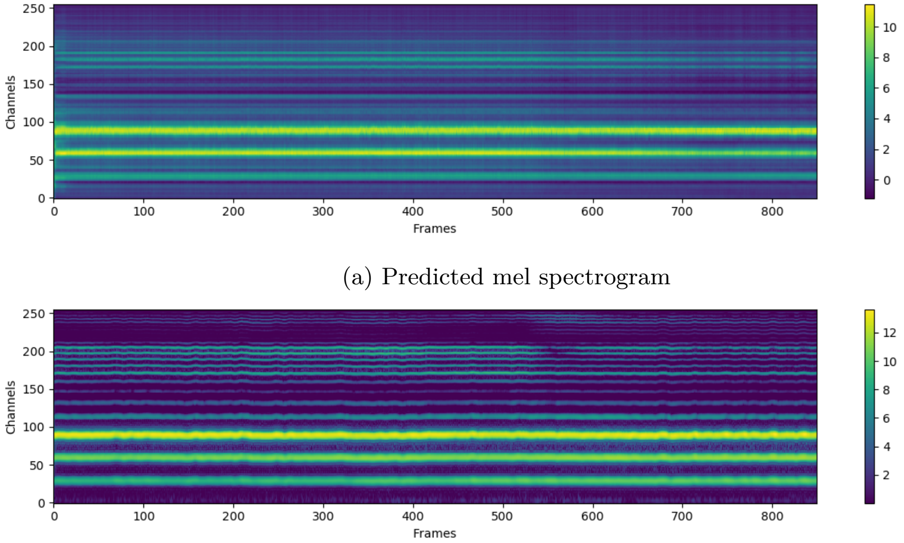
*Figure 3.9: Gaussian encoder embeddings of male(red) and female(blue) voices projected onto 2D with PCA*



(a) Predicted mel spectrogram

(b) True mel spectrogram

*Figure 3.10: Tacotron3 output on validation data*



(a) Predicted mel spectrogram

(b) True mel spectrogram

*Figure 3.11: Gausstron output on validation data*

# Chapter 4

# Source-filter models

In this chapter a method for voice morphing is investigated which uses the linear predictive modeling described in the mathematical background. As previously mentioned, the method consists of modeling the linear predictive filters on short time frames in order to capture the vocal tract acoustics. These filters are altered in order to change the characteristics towards a target voice and the residual signal is altered to change the fundamental frequency. In the next section the estimation techniques used to model the filter are discussed. Subsequently, methods for altering the filters are presented. Following this the pitch shifting work is presented and finally we describe how the methods fit together as parts in a voice morphing pipeline.

## 4.1 Estimating a linear predictive filter

In time series analysis the input signal is assumed to be generated from independent and identically distributed (iid) Gaussian noise. Assuming that the true model order is known there are several ways of estimating the filter coefficients with slightly different constraints. Examples of these are the autocorrelation method and the covariance method [14]. The aforementioned methods have been widely used in speech communications modelling and transmission even though the excitation of voiced speech is more accurately described as a semi periodic impulse train with some added noise [44]. In this section, we present methods for estimating a smoothed spectrum and estimating a digital filter from it.

### 4.1.1 Periodogram and smoothed spectrum estimates

The periodogram autocorrelation estimate of the LP filter is found by first calculating the windowed periodogram $P(w) = \frac{1}{M}((\mathscr{F}[y(n)W(n)](\omega))^2$, where $M$ is the number of samples in the frame. We use a Hanning window, $W(n)$, as a compromise between large and small scale feature resolution. Subsequently the periodogram autocorrelation estimate $r^*(\tau)$ can be calculated as [15]

$$r^*(\tau) = \mathscr{F}^{-1}[P(\omega)]. \tag{4.1}$$

The filter coefficients $\{a_k\}_{k=1}^{N}$ can then be estimated by solving the Yule-Walker equations, corresponding to $r(\tau)$ of order $N$ [16].

A drawback of the periodogram estimated LPC coefficients is that it is difficult to select a good model order. Parts of the glottal excitation will be modeled by the filter if the order is set too high [45]. In contrast, we have found that if the model order is set too low the filter will be unable to capture all the vocal tract information resulting in a poor separation of glottal excitation and filter. A previously used method for mitigating the problem of selecting the LPC order is to estimate the autocorrelation based on a smoothed spectrum [45]. This smoothed spectrum ideally corresponds to the vocal tract acoustics. Given an smoothed spectral magnitude, $\hat{S}$, the filter coefficients can be estimated by first calculating the autocorrelation sequence corresponding to the smoothed spectrum as [45]

$$\hat{r}(\tau) = \frac{\mathscr{F}^{-1}[\hat{S}(w)^2]}{M}. \tag{4.2}$$

The filter coefficients can then be found using the Levinson-Durbin recursion. Since $\hat{S}$ ideally contains less glottal source information than $P(\omega)$ the Levinson-Durbin recursion can be solved for a high order filter without overfitting the filter [45]. We will call this way of estimating autocorrelation and LP coefficient the *smoothed spectrum estimation*.

The periodogram and smoothed spectrum estimates of LP coefficients are based on the assumption that the smoothed spectrum represents the acoustics of the vocal tract. In reality it may be harder to separate the spectrum generated from the glottal pulses and the amplification due to the vocal tract. The vocal tract acoustics have positive spectral slope and that the resulting negative slope is due the fact that the harmonics of the glottal pulses have an even steeper negative slope [8]. This discrepancy may cause the filters estimated from smoothed spectra to be a poor representation of the vocal tract acoustics. This can be seen in appendix B where some periodograms and smoothed spectra are shown. Globally the spectra all have negative spectral slope, with most of the power situated at low frequencies. Nonetheless, three techniques for obtaining smoothed spectra are presented below and the results are evaluated. The techniques work by estimating a smoothed log-spectrum which means the result has to be exponentiated before generating a corresponding filter.

### 4.1.2 Quefrency smoothing

Quefrency analysis can be used to estimate the smooth magnitude response of the vocal tract by modelling the periodicities in the log magnitude spectrum of a signal. The analysis is performed on the *real cepstrum* which is the time lag representation corresponding to the log magnitude spectrum [7]. Given a signal, $s(n)$, a smooth spectral shape can be calculated by first calculating the log magnitude spectrum

$$L(w) = \ln |(\mathscr{F}[s(n)]|. \tag{4.3}$$

Subsequently the real cepstrum is calculated as [46, 17]

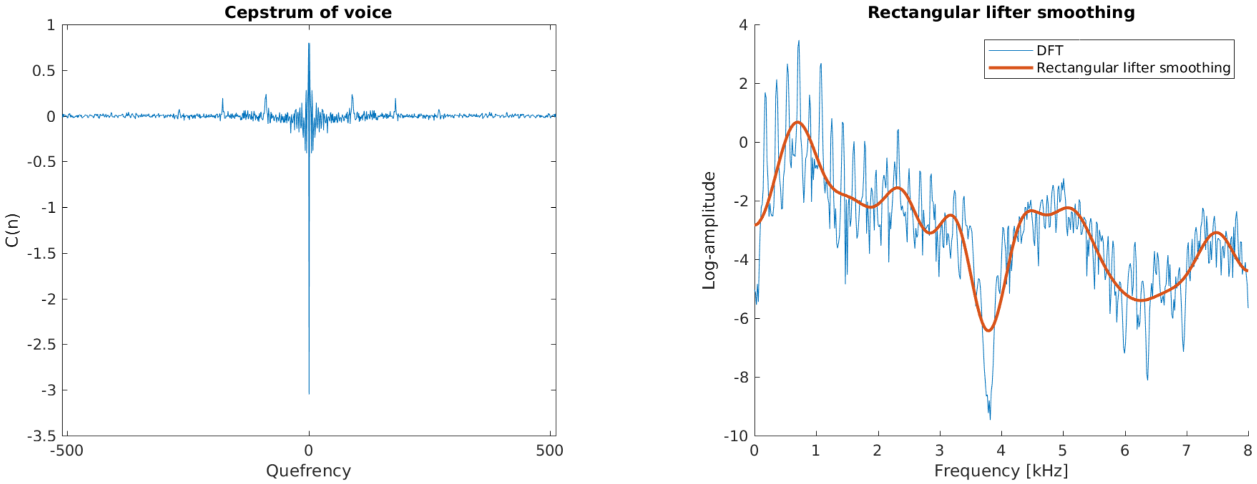
$$c(n) = \mathrm{Re}(\mathscr{F}^{-1}[L(w)]). \tag{4.4}$$

The smooth spectral shape can then be estimated as

$$\hat{S} = \mathrm{Re}(\mathscr{F}[c(n)W(n)]) \tag{4.5}$$

where $W(n)$ is a window function referred to as the liftering function [46]. Often, $W(n)$, is set to a rectangular window centered at zero [7]. A multiplication in the time domain corresponds to a convolution in the frequency domain and the Fourier transform of a rectangular window is a sinc function [8]. This means that the rectangular liftering corresponds to filtering the log spectral magnitude with a sinc function.

An example of a cepstrum for a voiced speech segment can be seen in Figure 4.1a. We see that there is a lot of information in the low time lags and periodic peaks of higher lags. The periodic peaks correspond to information about the excitation of the signal [47], whereas the low lag values correspond to the overarching structure of the log-magnitude



(a) Cepstrum

(b) Rectangular lifter envelope

*Figure 4.1: Cepstrum of a voiced speech signal (left) and log-magnitude spectrum of the DFT and the smoothed spectrum extracted via a rectangular liftering of the same cepstrum (right).*

spectrum [7]. Figure 4.1b displays the DFT of the same frame as 4.1a and the smooth spectrum calculated using equation (4.5) with a rectangular window of length 39 centered at zero. As we can see the low order lags capture a smoothed estimate of the DFT.

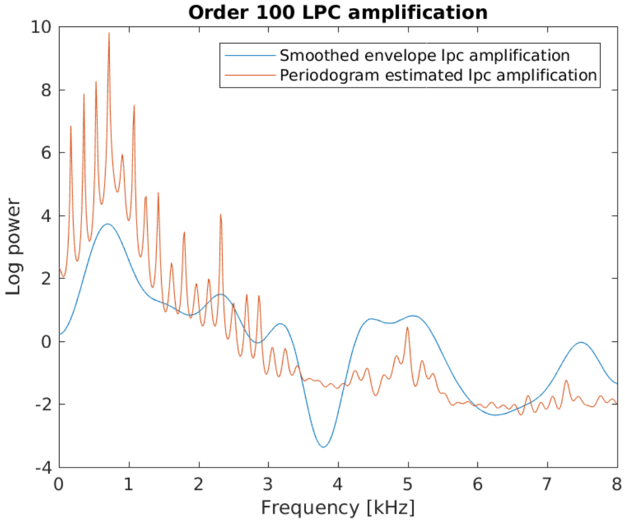


*Figure 4.2: Spectral amplification of two filters. The blue is estimated from a smoothed spectrum and the red from the periodogram.*

Figure 4.2 shows the the amplification due to two LP filters of order 100. The blue graph is the amplification corresponding to the smoothed spectrum in Figure 4.1b and the red graph is the filter amplification corresponding to the periodogram. The filters are estimated using equation (4.2). In the periodogram estimate we see spikes corresponding to the harmonic peaks of the voice whereas the smoothed spectrum amplification is not as affected by the harmonic content of the signal.

### 4.1.3 True Envelope

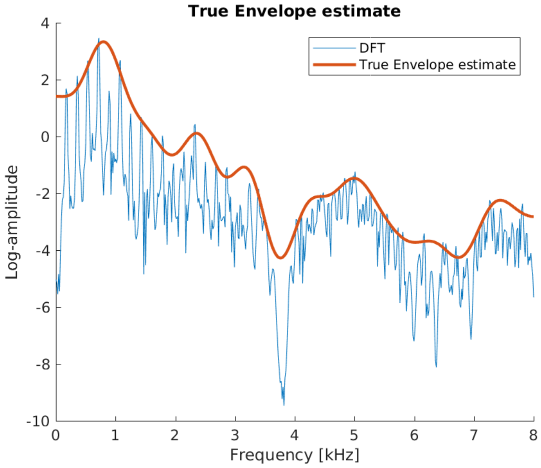


*Figure 4.3: Envelope estimated using the True Envelope algorithm and Log-amplitude spectrum (DFT)*

The spectral envelope is a smooth function tangent to the peaks in the spectrum and can be used to estimate a filter [48, 45]. The True Envelope estimation is a quefrency analysis method for estimating the spectral envelope [45]. The method is based on iteratively re-estimating the envelope in order to achieve a more accurate description of the vocal tract acoustics [45]. The estimated envelope is then used to compute an all pole filter from the biased auto correlation estimate [45].

The re-estimation is performed by first calculating the cepstrum and using a rectangular window in order to generate the smoothed spectral envelope of the signal [45]. The algorithm for generating the True Envelope estimate is described in Algorithm 2. When using the algorithm the lifter window was chosen as a rectangular window of order $10 - 40$. The tolerance was chosen to correspond to 2 db as suggested by the authors of the algorithm [45].

Figure 4.3 shows the result of a True Envelope estimate. As can be seen the resulting smooth spectrum from this technique follows the peaks of the Log-spectrum. This is in contrast to the rectangular lifter smoothing where the signal lies in the middle of the peaks.

To show the difference, Figure 4.4 displays the magnitude response of three different filters: a low order LPC, a high order LPC and high order filter from the True Envelope estimated filter.

**Algorithm 2** True Envelope estimation

```
**Input** Signal: $y$, Lifter window: $W$, tolerance: tol
**Output** True Envelope estimate: $S$
$A_0 \leftarrow \ln(|\mathscr{F}(y)|)$                ▷ Compute the log-amplitude
$a_0 \leftarrow \mathrm{Re}(\mathscr{F}^{-1}(A_0))$    ▷ Compute the cepstrum
$v_0 \leftarrow Wa_0$                                  ▷ Perform the liftering
$V_0 \leftarrow \mathrm{Re}(\mathscr{F}(v))$           ▷ Compute the envelope
**while** d > tol **do**
    $A_i \leftarrow \max(A_{i-1},V_{i-1})$             ▷ Iterate modified log amplitude
    $a_i \leftarrow \mathrm{Re}(\mathscr{F}^{-1}(A_i))$ ▷ Compute new cepstrum
    $v_i \leftarrow Wa_i$                              ▷ Perform the liftering
    $V_i \leftarrow \mathrm{Re}(\mathscr{F}(v_i))$     ▷ Compute new envelope
    $d \leftarrow \max(A_0 - V_i)$                     ▷ Compare envelope to log amplitude
**end while**
$S \leftarrow \exp(V_i)$
```

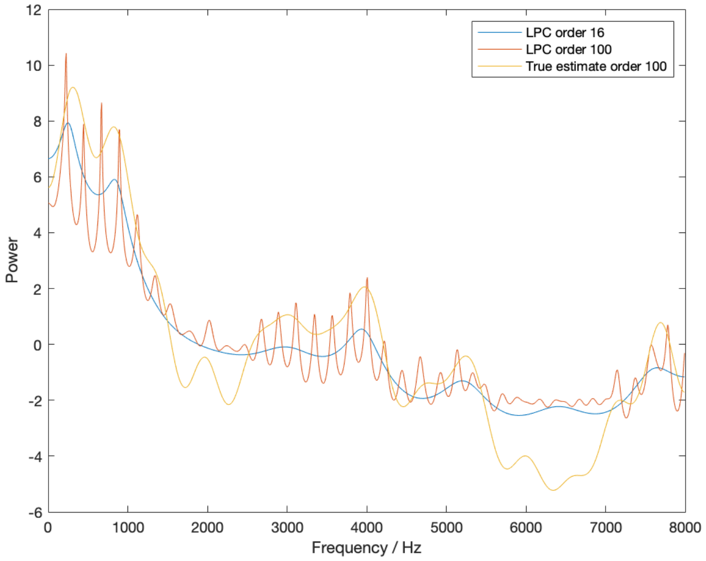
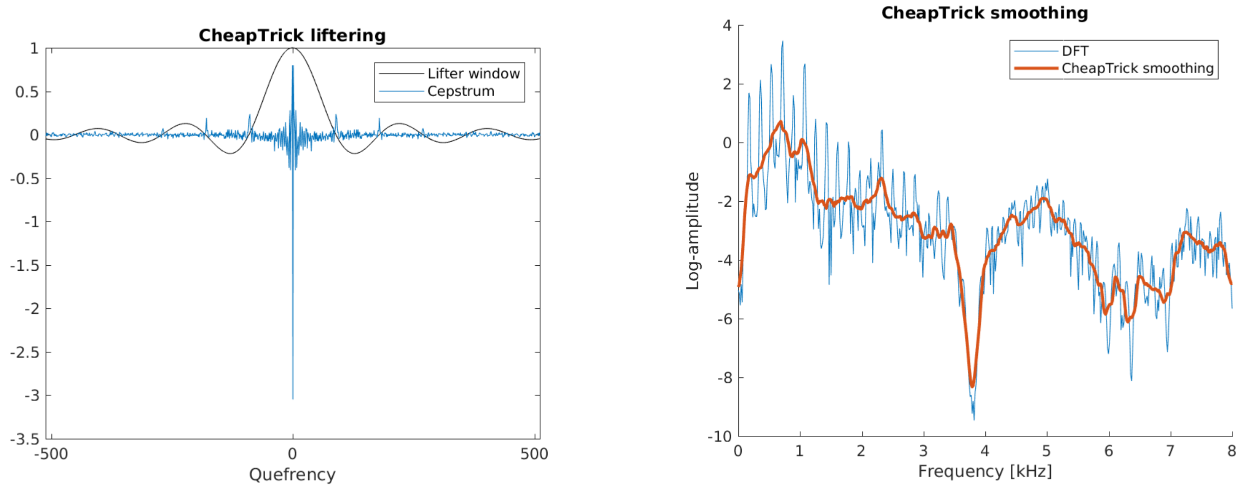


*Figure 4.4: Magnitude response for three LPC polynomials: two using the periodogram estimate of the LPC, and the third by first estimating the spectral envelope using the True Envelope algorithm*

### 4.1.4 CheapTrick envelope

In [47] the authors suggest using a pitch varying lifter function $W(n) = \frac{\sin(\pi f_0 n)}{f_0 n}$ in order to smooth a log-spectrum. The lifter function cancels the harmonic information of the signal by aligning the zeros of the lifter function with the periodic peaks in the cepstrum. This liftering also corresponds to filtering the log magnitude spectrum with a rectangular window since the Fourier transform of a sinc function is a rectangular window.



(a) Cepstrum and CheapTrick lifter.

(b) DFT and CheapTrick smoothed spectrum

*Figure 4.5: CheapTrick smoothing of spectrum.*

Figure 4.5a displays the previously shown cepstrum, and the CheapTrick lifter function. As can be seen in the figure the periodic high lag peaks coincide with the zeros of the lifter function. In Figure 4.5b the resulting smoothed spectrum is graphed together with the periodogram of the signal. Although more jagged than the rectangular lifter smoothing the CheapTrick smoothed spectrum seems to capture the overall shape of the log magnitude spectrum while containing little information about the harmonics.

## 4.2 Interpolating Filters

In this section we will present methods for interpolating the linear predictive filter. As we have previously stated we believe that a good interpolation of voices consists of smoothly moving the peaks and troughs of the smooth spectrum in frequency from the source to the goal spectrum. This corresponds to moving the poles of the filter polynomial $A(z)$ in some smooth way starting with one filter and ending up in the other. Previous methods have been proposed for interpolating directly in the poles of the filter. The interpolation is then produced by moving the poles continuously between the representations [49]. However, this requires finding an assignment of the poles which requires a large amount of computations and moving a pole affects the power spectral density across all frequencies which means that it is difficult to choose the path the poles are to take in the interpolation. With this in mind we chose to work with simpler methods for interpolating the filters. A description of the four methods of filter interpolation are described below.

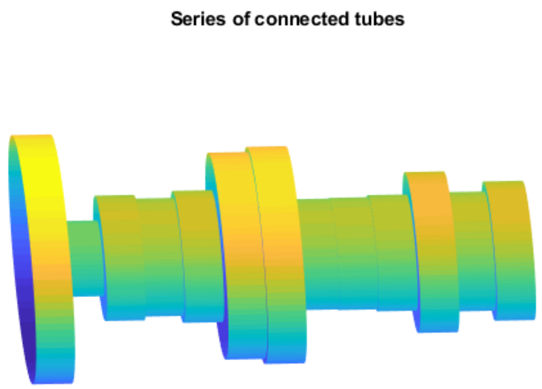


*Figure 4.6: Series of connected tubes corresponding to an autoregressive filter.*

### 4.2.1 Interpolating in reflection coefficients

The first method of interpolating between autoregressive filters draws inspiration from a physical interpretation of the LPC filter. Specifically the fact that the autoregressive process is equivalent to passing the excitation signal through a series of joined, reflecting, cylindrical tubes of equal length and varying radius. Figure 4.6 shows a series of tubes corresponding to an autoregressive filter. In the speech production model the tubes can be thought of as an approximation of the vocal tract even though the throat is neither lossless nor does it visually resemble a series of tubes [50].

The amount of energy reflected at each joint in the system of tubes is related to the relative areas of the tubes at the joint. The proportion of reflected energy is quantified by the reflection coefficients [7]. The coefficients are an alternate representation of the LPC filter in the sense that each linear predictive filter is equivalent to a set of reflection coefficients. Further, sets of reflection coefficients in the box $(-1,1)^n$ correspond to stable AR filters, where $n$ is the order of the filter [7]. This means that a straight line interpolation of reflection coefficients will also correspond to a stable filter.

The reflection coefficient $k_m$ can be calculated from the filter coefficients by first setting $b_{n,i} = a_i$ and then iteratively solving

$$\begin{aligned} b_{n-1,m} &= \frac{b_{n,m} - k_n b_{n,n-m}}{1-k_n^2} \\ k_m &= b_{m,m}, \text{ for } m = 1,2,\ldots,n-1 \end{aligned} \tag{4.6}$$

The relationship can also be used in reverse in order to compute the filter coefficients given only the reflection coefficients [7].

Reflection coefficient interpolation has previously been used to interpolate between the spectral contents of sound signals [51]. An interpolated filter, $A^\tau(z)$, in between two filters with interpolation coefficient, $\tau$, is generated by the following procedure:

- The reflection coefficients of the source and target filter $\vec{k}^s$, $\vec{k}^t$ are calculated using relation (4.6).
- A new set of reflection coefficients is generated using $\vec{k}_\tau = (1-\tau)\vec{k}_s + \tau\vec{k}_t$.

- The interpolated filter is found by solving equation (4.6) backwards using the $\vec{k}_\tau$.

### 4.2.2 Interpolation in tube area

The second interpolation we test is also based on the joint tube representation of the autoregressive process, but this time the areas of the tubes are used as a representation. The relation between the tube areas, $T_i$, and the reflection coefficients, $k_i$, is [50]

$$T_i = T_{i-1}\frac{k_i - 1}{k_i + 1}. \tag{4.7}$$

The highest order area has to be specified and is set to one. This tube area representation of LPC filters has previously been used in order to perform interpolations between filters [52]. The interpolation for a given interpolation coefficient, $\tau$, is produced by the following steps

- The source and target filters, $A^s(z)$, $A^t(z)$, are converted to reflection coefficients and subsequently to tube area vectors, $\vec{T}^s$, $\vec{T}^t$, using (4.7).
- A morphed tube area vector is created as $\vec{T}^\tau = (1-\tau)\vec{T}^s + \tau\vec{T}^t$.
- The synthesised tube vector is subsequently converted to reflection coefficients and finally these are converted to filter coefficients.

### 4.2.3 Line spectral frequency interpolation

Another representation of the filter polynomial $A(z)$ is the line spectral pairs. In [7] the line spectral pairs are defined as the polynomials $P(z)$, $Q(z)$ that satisfy

$$P(z) = A(z) - z^{-(n+1)}A(z^{-1}) \tag{4.8}$$

$$Q(z) = A(z) + z^{-(n+1)}A(z^{-1}). \tag{4.9}$$

The LPC polynomial can then be written as the mean of the two polynomials

$$A(z) = \frac{P(z) + Q(z)}{2}. \tag{4.10}$$

The poles of the line spectral pairs have roots on the unit circle implying that the pairs are fully described by the angles of their roots [7]. The set of angles are called the *line spectral frequencies* or LSFs. The LFSs have previously been used to perform voice conversions [53]. The line spectral frequencies of stable polynomials have the property that if the LSFs are sorted according to the angle then the poles alternate in originating from $P(z)$ and $Q(z)$ and the first angle always corresponds to a root of $P(z)$ [53]. This structure implies that a straight line interpolation of two sets of LSFs from stable filters will also correspond to a stable filter [53]. Closely spaced line spectral frequencies corresponds to a filter $A(z)$ which has a small bandwidth and large amplification at the corresponding frequency [53].

In practice the roots of the polynomials $P(z)$,$Q(z)$ are estimated, and the angles extracted yielding an approximation of the line spectral frequencies. The LSF interpolations, with interpolation coefficient $\tau$ can then be generated by linearly interpolating the ordered line spectral frequencies of the source and target filters, $\vec{\theta}^s$, $\vec{\theta}^t$ that is,

$$\vec{\theta}^\tau = (1-\tau)\vec{\theta}^s + \tau\vec{\theta}^t. \tag{4.11}$$

The interpolated frequencies can then be converted back to the line spectral pairs by

$$\begin{aligned} P(z) &= (1 - z^{-1})\Pi_{k \in E}(1 - 2z^{-1}\cos\theta_k^\tau + z^{-2}) \\ Q(z) &= (1 + z^{-1})\Pi_{k \in O}(1 - 2z^{-1}\cos\theta_k^\tau + z^{-2}) \end{aligned} \tag{4.12}$$

where $E$ is the set of even positive integers $\leq n$ and $O$ is the set of odd positive integers $\leq n$ [53]. The interpolated filter is finally calculated using equation (4.10).

### 4.2.4 Optimal mass transport filter interpolation

We finally propose our own method for interpolating the vocal tract acoustics using the optimal mass transport formulation from section 2.4. Since the interpolation described in equation (2.10) often produces a non-smooth distribution we decided to construct a new way of interpolating, this time interpolating both in frequency and in amplitude according to the optimal mass transport assignment. This approach alone misses some of the query points in a discrete setting which motivates using weights to create an interpolation which is defined on all the query points. After normalizing the smoothed spectra to sum to one the optimal transport plan, $\pi$ is calculated as explained in algorithm 1. For a source and target smooth spectrum, $\hat{S}_s$, $\hat{S}_t$, an interpolated smoothed spectrum $\hat{S}_\tau(z)$ with interpolation coefficient $\tau$ is generated. For each query point, $z$, let

$$J(\tau,z) = \{x,y | \pi(x,y) \neq 0 \text{ and } |(1-\tau)x + \tau y \rfloor - z| < 1\} \tag{4.13}$$

which is the set of all points within unit distance to $z$ after linearly interpolating the points according to the optimal mass transport assignment. For each element $x,y \in J(\tau,z)$ let the weight

$$w(x,y) = 1 - |(1-\tau)x + \tau y - z|$$

and

$$W(\tau,z) = \sum_{x,y \in J(\tau,z)} w(x,y).$$

The proposed interpolation is then calculated as

$$\hat{S}_\tau(z) = \sum_{x,y \in J(\tau,z)} \frac{w(x,y)((1-\tau)\hat{S}_s(x) + \tau\hat{S}_t(y))}{W(\tau,z)}. \tag{4.14}$$

The Matlab code is seen in appendix A for further details on the implementation. In Figure 4.7 we see a comparison between the original optimal mass transport interpolation and the proposed interpolation. For the given distributions the new interpolation produces smooth results where the original method fails. As of yet we have no proof that smooth input distributions always result in smooth results. But after trying some different inputs we have not seen any distribution where the proposed method fails.

The new method no longer conserves mass and the interpolated smoothed spectrum is normalized to sum to one. The resulting interpolation is then multiplied by a linear interpolation of the energies in the source and target spectra. A corresponding LPC filter was then generated by using the periodogram method described in section 4.1.1. We would also like to point out that the interpolation may work with other methods of reassignment than optimal mass transport. For example it may be used with dynamic frequency warping.

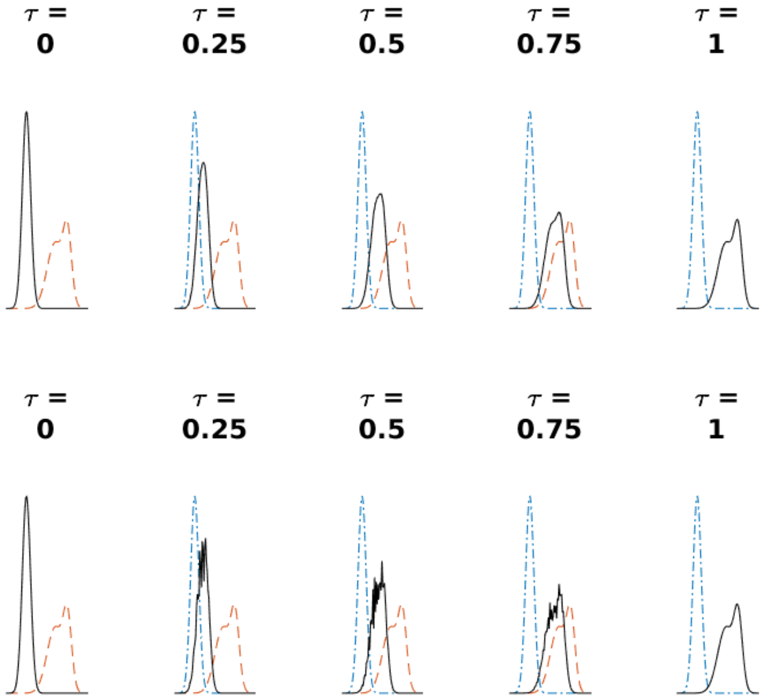


*Figure 4.7: Proposed interpolation (top) and Optimal mass transport interpolation (bottom).*

## 4.3 Pitch correction

In order to change the fundamental frequency of the voices we implement a method called resampled pitch synchronous overlap add (re-PSOLA). As the name suggest re-PSOLA is largely based on the original PSOLA. In the original PSOLA the duration and location of each period in the voice is found by estimating each period in the signal [54]. Some of the periods are then deleted or repeated and are overlap added into a synthesised signal at a new frequency [55]. The result is a signal with a new fundamental frequency of the same number of samples as the original signal [55].

### 4.3.1 Period picking

In order to track the pitch of the signal we use the autocorrelation method on short time frames [56]. The signals are divided into 40 ms frames. For each of the $k$ frames the period of the fundamental frequency is estimated by finding the lag, $T$, maximizing the periodogram estimate of the autocorrelation function $r_k^*(T)$, for lags, $T$ in the range corresponding to a frequency between $100-300$ Hz, encompassing the fundamental frequencies of most human speech and singing, i.e [56]

$$\begin{aligned} T_k^* = &\underset{T}{\text{argmax}} \quad r_k^*(T) \\ &\text{such that} \quad T \in [T_{min}, T_{max}]. \end{aligned} \tag{4.15}$$

For a periodic signal the autocorrelation estimate is periodic [56]. We find that this results in the autocorrelation estimate often finding multiples of the fundamental period. In order to combat this a re-estimation is performed. The median period, $\tilde{T}$, over all the frames is calculated. The period in each frame is then re-estimated using the same autocorrelation maximization step but this time only optimizing over lags with small deviation from the median, $T_{min} = (1 - \epsilon)\tilde{T}$ to $T_{max} = (1 + \epsilon)\tilde{T}$. A subjective evaluation of the results suggests that $\epsilon = 0.05$ give the best results. Such a small epsilon, however, means that the method only works for signals with very small variation in fundamental frequency.

The maximal points in each period is estimated by first finding the maximum value of the signal in the first $T$ samples. The peaks, $j_i^*$, in the signal, $s(n)$, are found iteratively by

$$\begin{aligned} j_i^* = &\underset{j}{\text{argmax}} \quad s(j_{i-1}^* + j) \\ &\text{such that} \quad j \in [(1 - \epsilon)T_k^*, (1 + \epsilon)T_k^*] \end{aligned} \tag{4.16}$$

where $T_k$ is the estimated period in the frame corresponding to the sample $j_{i-1}^*$.

### 4.3.2 Resampled PSOLA

The PSOLA works well provided that the desired pitch changes are small [55]. In our data set we find that some of the male voices had a fundamental frequency half that of some females. We therefore think it is important that our pitch shifting method can accommodate such changes and extend the PSOLA to the re-PSOLA algorithm. As previously stated the idea of the algorithm is to allow for large pitch shifts by resampling each pitch synchronous frame before overlap adding the result into the synthesised signal. This is done by choosing the length of the resampling such that there is always a 50% overlap between successive frames.

For the resampling we tested three methods

- Linear interpolation
- Linear interpolation with filtering to avoid aliasing
- Fast Fourier transform resampling method [57]

These were compared by listening to a few signals generated using the different methods of resampling. We perceived that the linear interpolation with filtering produced the most natural sounding voices.

The re-PSOLA produces voices which are rescaled equally in pitch and overarching shape of the spectrum resulting in voices which sounds like cartoon chipmunks or giants. The chipmunk or giant effect can also be achieved by simply resampling the signal or playing the signal at a different sampling frequency [58]. However, the re-PSOLA keeps the duration of the signal constant and keeps the sampling frequency constant. Keeping the signal duration constant was not essential for investigating interpolations of stationary voices and a simpler pitch shifting algorithm may have been used. Although for more general speech conversion systems the duration of each syllable is important to conserve. The constant sampling rate is essential for this method since the magnitude response of a filter is relative to the sampling frequency. Another advantage of the re-PSOLA is that it may be extended to time varying pitch changes on very short time frames the uses of which will be discussed in later sections.

### 4.3.3 Voice interpolation pipeline

The source filter voice conversion method is constructed by combining the previously suggested linear predictive models, in order to change the vocal tract characteristics, with the re-PSOLA, in order to change the pitch of the residual signal.

- First, the pitch markers of a source and target voice are extracted.
- The vocal tract characteristics are then modeled by extracting a frame of set length centered at each of the period peaks and, for a specified order, modeling an LP filter using one of the methods presented in section 4.1. The modeling is performed for both the signals and results in a collection of autoregressive filters for each speaker. The sets generally contain a different number of elements since the voices contain a different number of period peaks.
- Filters in the target signal are deleted or repeated until there are the same number of source and target filters. A set of interpolated filters is then created for each pair of source-target filters using one of the methods described in section 4.2.
- The median pitch, $\tilde{f^s}_0$, $\tilde{f^t}_0$, of the source and target is calculated using the period markers and the interpolation pitch ratio was set to

  $$r^\tau = \frac{(1-\tau)\tilde{f}_0^s + \tau\tilde{f}_0^t}{\tilde{f}_0^s}. \tag{4.17}$$

  New period peaks are then calculated such that the new signal has the instantaneous fundamental frequency $r^\tau f_0^s$.
- In contrast to the re-PSOLA, each frame is inverse filtered with the corresponding source filter before being resampled, ideally removing the vocal tract characteristics. The corresponding interpolated filter is then applied to the resampled residual, with the aim of introducing interpolated vocal tract characteristics.
- The result is then overlap added to the synthetic signal at the corresponding pitch marker.

The last two steps are repeated until a synthesized frame has been added to all of the new pitch markers.

## 4.4 Linear predictive modeling evaluation

The methods for extracting smoothed spectra, $\hat{S}$, discussed in section 4.1, were used to estimate LPC filters for a short frame of voiced speech. Residuals of filters estimated using the periodogram, rectangular liftering, True Envelope and CheapTrick method are seen in Figure 4.8 and the log-amplitude spectra of the residuals are seen in Figure 4.9. The periodogram estimated filter uses 12 linear prediction coefficients, the spectral smoothing estimation models uses 50 linear prediction coefficients and a lifter order of 20 for the rectangular lifter and True Envelope.

The features of the residuals vary between the voices being modeled but the residuals shown here share some common characteristics with most other residuals we have looked at. For example we see that to some degree all four residuals in Figure 4.8 have periodic peaks. These are the peaks which ideally represent the glottal pulses in the biomechanical model. The residuals are also noisy and may even contain some structure in between the peaks. The structure in between the peaks suggest that none of the models fully separate

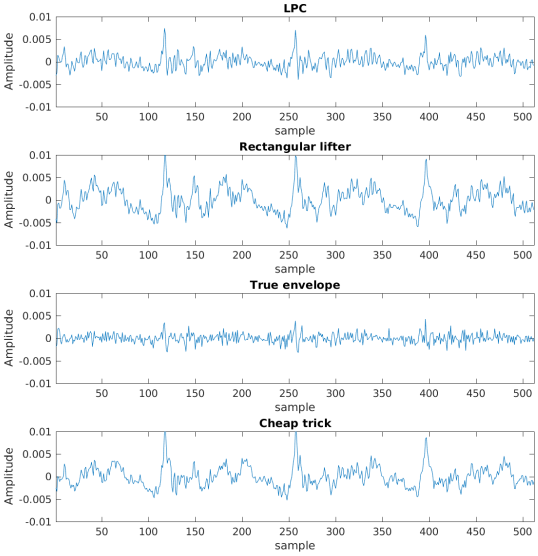


*Figure 4.8: Residuals of a signal using different filter estimation methods.*

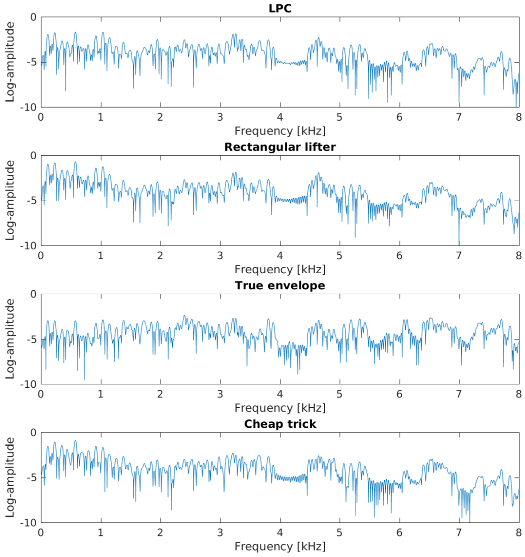


*Figure 4.9: Log-amplitude spectra of a residual using different filter estimation methods.*

the glottal pulses from the vocal tract characteristics. We also think that the ratio of peak to noise in the residuals is stronger than an ideal model would produce. The signals do contain measurement noise and any voice is in part driven by noise but the level found in the residual does not reflect what is heard in the voice.

The amplitude spectrum of the residual obtained from the True Envelope was often flatter than the other spectra and the time domain residual had comparatively weak peaks. The rectangular lifter and the CheapTrick method produced residuals similar to each other but the structure in between the pulses often had slightly less amplitude in the CheapTrick residual. This result is also seen in the figure.

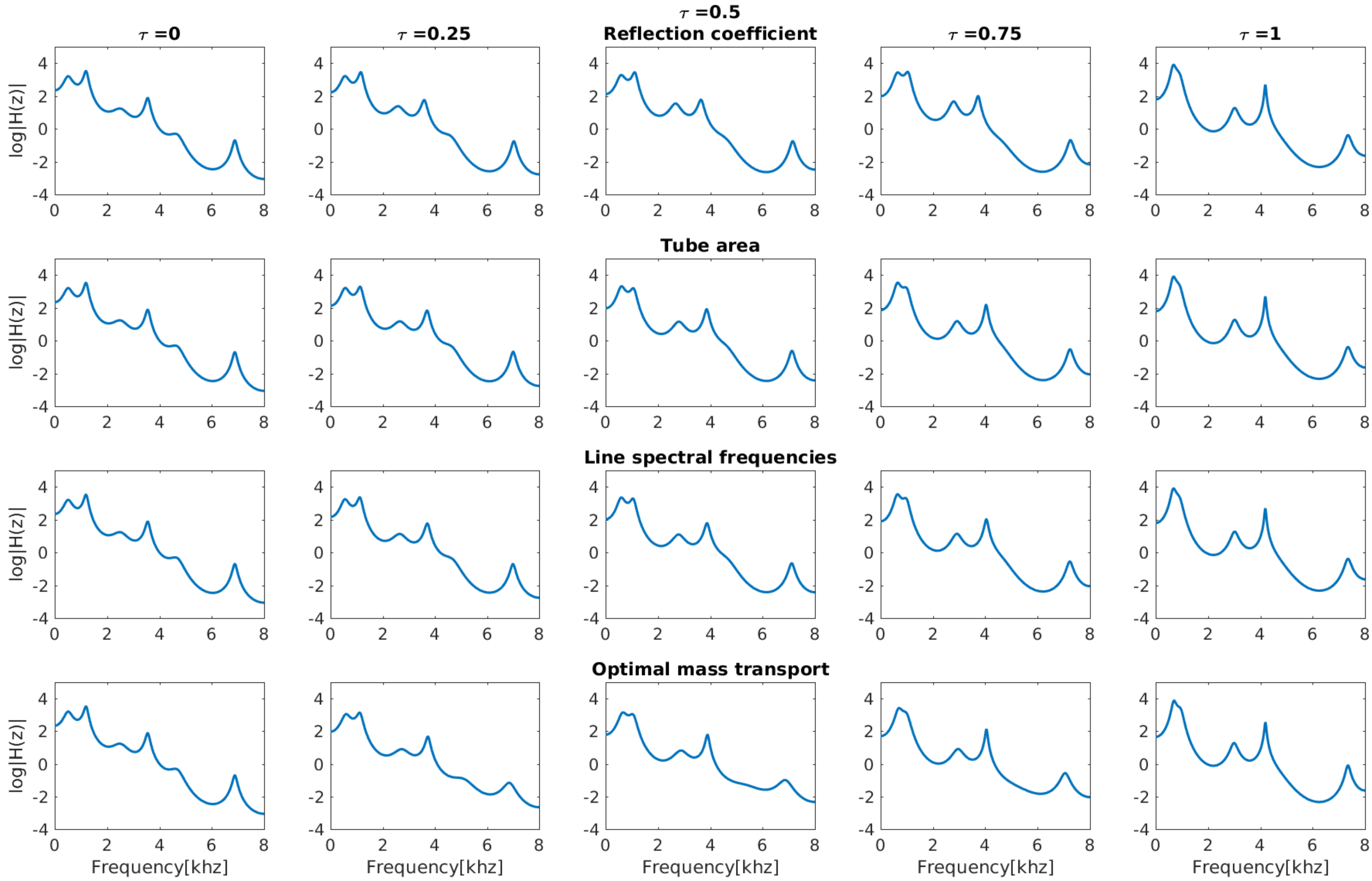
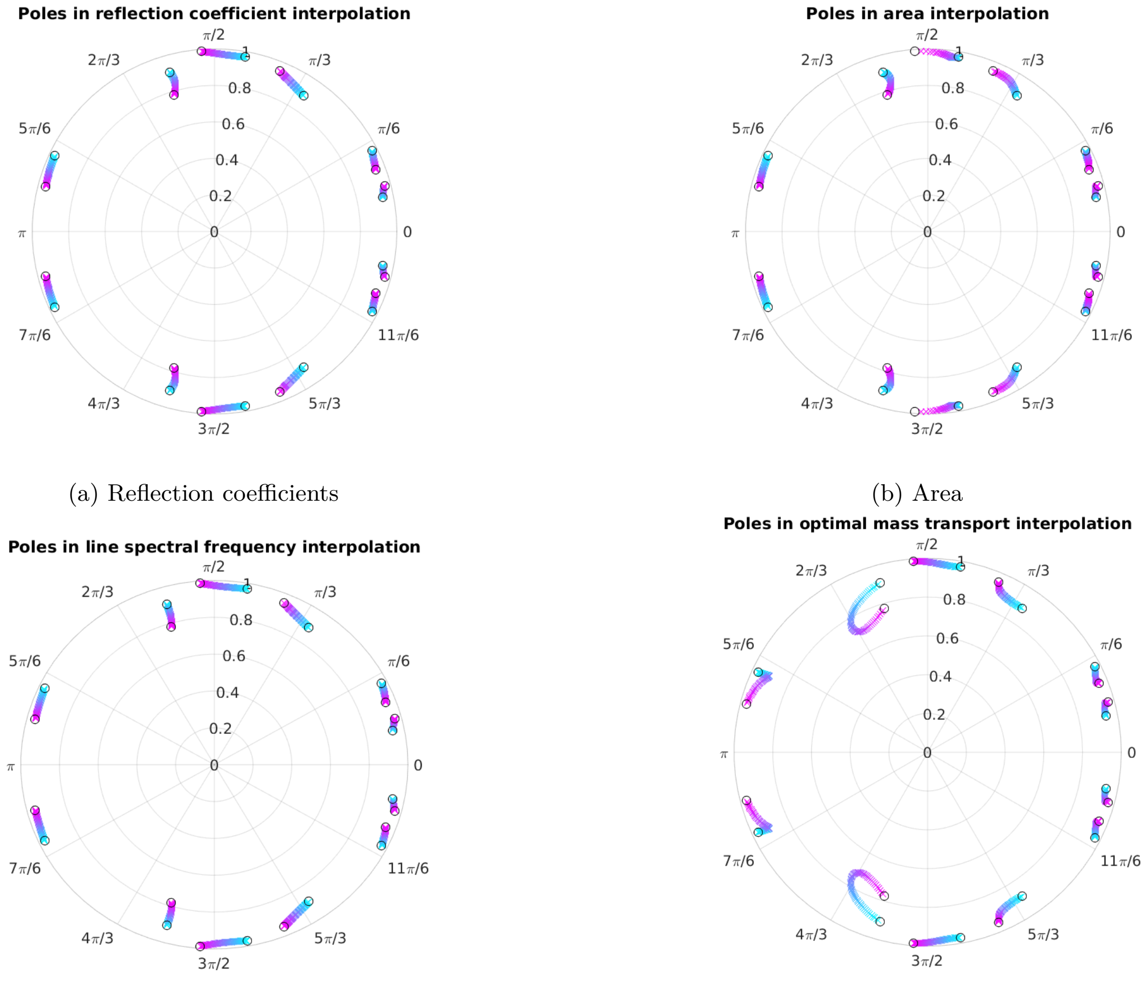


*Figure 4.10: Grid of log magnitude responses of interpolated filters. The different rows show different methods of interpolation and the columns show different interpolation coefficients $\tau$.*

## 4.5 Filter interpolation evaluation

In Figure 4.10 each row corresponds to one of the interpolation methods proposed in section 4.2 and each column is represents different interpolation coefficients. The figures were generated by extracting the smoothed spectra of frames from a female speaker and a male speaker. The frames were 40 ms long and the smoothed spectra are estimated from order 12 LPC filters. All of the methods of filter interpolation seem to shift the acoustic characteristics rather than fading in the new objects. Visually, there seems to be no major differences between the log amplitude spectra of the interpolation methods. However, the intermediate magnitudes are not identical. For example the highest frequency peak is smoother in the optimal mass transport interpolations than any of the other methods. We cannot visually exclude any of the interpolated filters as valid vocal tract acoustics based solely on the magnitude response.


Poles in reflection coefficient interpolation

(a) Reflection coefficients

Poles in area interpolation

(b) Area

Poles in line spectral frequency interpolation

(c) Line spectral frequency

Poles in optimal mass transport interpolation

(d) Optimal mass transport

*Figure 4.11: Poles of 64 interpolation filters using reflection coefficient, area interpolation, line spectral frequency and optimal mass transport interpolation.*

Figure 4.11 shows how the poles of the different interpolation methods between using the same source and target frames as in the previous image but with 64 interpolation points. The poles of the source and target filters are marked as black circles. In these figures the poles move continuously between the source and target poles on slightly different trajectories and speeds. We noted that in the reflection coefficient, area and LSF interpolations increasing $\tau$ reduces the distance of each pole to the target of that pole whereas this is not true for the optimal mass transport interpolation. It is at the time not clear if this is true in general or if it has a major impact on the resulting sound of the voice interpolations.

## 4.6 Subjective evaluation of the morphed voices

We performed a subjective evaluation of the voice conversion system, comparing different combinations of methods and model orders. We found that the most natural sounding interpolations were achieved when using the CheapTrick method for modelling, using the line spectral frequency interpolation, and an LPC order of 50. We used a 100 ms window for filter modeling. Long analysis window were found to reduce buzzing sound in the resulting voice. However, for general speech with faster changes in spectral properties a shorter analysis window needs to be used.

When using a morphing factor greater than around $\tau = 0.5$ many of the generated voices sounded unnatural, containing a buzzing sound and artifacts, and this problem seemed to be worse for interpolations from a male source voice. Further, many of the male to female morphs sounded nasal regardless of the nasality of the input voices. Sometimes the nasality was so pronounced that the voices did not sound like humans. In general we though the interpolations generated with a female source voice sounded like male voices even for interpolation coefficients around $\tau = 0.5$.

Morphs were also created using interpolation coefficient $\tau = 1$ which corresponds to changing the source filters to the target filters and pitch shifting to around the target speaker fundamental frequency. Regardless of the source and target speakers these signals were dominated by buzzing synthetic sounds. We can thus say that the final pipeline does not accommodate a full conversion of a voice.

We also created some morphs using separate interpolation factors for the pitch and formant shifting. The resulting voices revealed that most of the perceived voice change came from the pitch shifting system. These results suggest that most of the information about the speaker is not contained in the autoregressive filters but in the residual.

# Chapter 5

# Survey evaluation

## 5.1 Method

In order to evaluate the source-filter pipeline, we create a survey where participants can listen to audio samples and then rate how they perceive them. Voice-expert listeners are not explicitly chosen as participants of the survey.

The survey contains 28 audio samples of voices. Out of these, 24 are modified and four are unmodified (two male and two female). The 24 modified voices are constructed from four male-female pairs, each pair interpolated with coefficients $\tau = 0.3$, $\tau = 0.5$ $\tau = 0.7$. Each pair of voices are interpolated in both directions, i.e., using both the male and female as source voice resulting in six voices per pair. The voices generated from the same source will be referred to as interpolation triplets. To avoid any selection bias from our part, we randomize the male-female pairs from our data set.

Before the actual survey, participants are presented with two voices, one female and one male, and are told to see these as reference voices. In the survey, the unmodified voices are placed at indices 6,12,18 and 24 to act as hidden anchors and as a way to compare the synthetic and unmodified voices. The modified voices are placed at random.

For each audio sample, the participants are asked to rate how they perceived the voice on an unnumbered scale with ten ticks from female to male, see Figure 5.1. They are also asked if the voice sounds natural, i.e., if it sounds like a human voice.

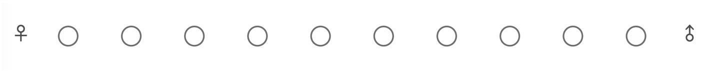


*Figure 5.1: Scale used in survey for expressing how the gender of a voice was perceived*

For analysis, we divide the voices into eight classes:

- Female to male: $\tau = 0.3$
- Female to male: $\tau = 0.5$
- Female to male: $\tau = 0.7$
- Male to female: $\tau = 0.3$
- Male to female: $\tau = 0.5$
- Male to female: $\tau = 0.7$
- Unmodified female

- Unmodified male

The analysis consists of computing histograms over class ratings, naturalness per class, and how many interpolation triplets were rated in a consecutive order. The interpolation triplets were assessed on both a global level and by individual survey responses. On a global level, the mean rating for all audio samples are calculated and the analysis script checks if they are ordered according to the interpolation order. It also counts the percentage of individual survey respondents who order the triplets in consecutive order.

## 5.2 Results

In total, the survey got 61 replies. Although the gender perception scale looked unnumbered for the survey participants, it was numbered from 0(female)-9(male). Histograms of the perceived gender rating for the unmodified classes are shown in Figure 5.2 and for all modified classes are shown in Figure 5.3. As the interpolation constant increases, the histograms show that the perceived gender is changed in the intended direction, although the variance of the answers is high.

Looking at the histograms for the unmodified females and males, Figure 5.2, some survey participants guessed completely opposite to the truth. This surprises us since the unmodified voices, to us, represents typical male and female voices. Two participants had for all four unmodified voices guessed on the wrong side of the scale. It is unclear if this was due to technical difficulties or their actual perception. Keeping this in mind, there are potential error sources in the results.

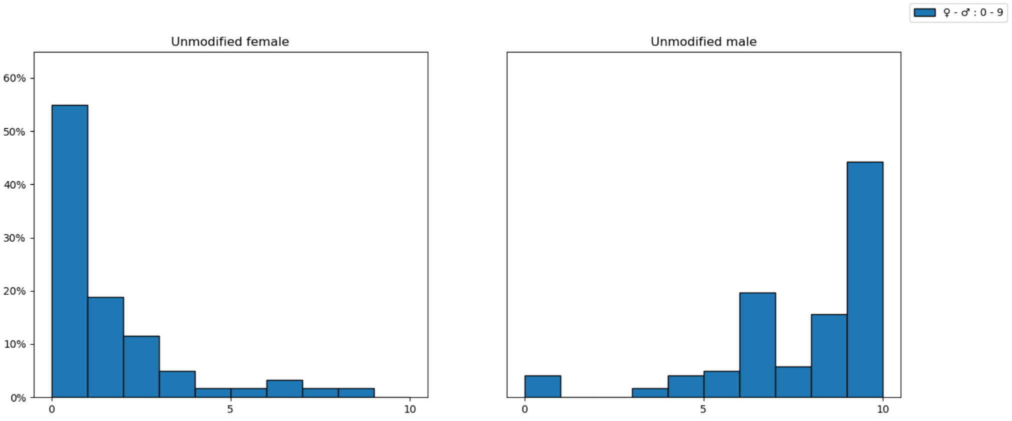


*Figure 5.2: Histogram of perceived gender for the unmodified classes.*

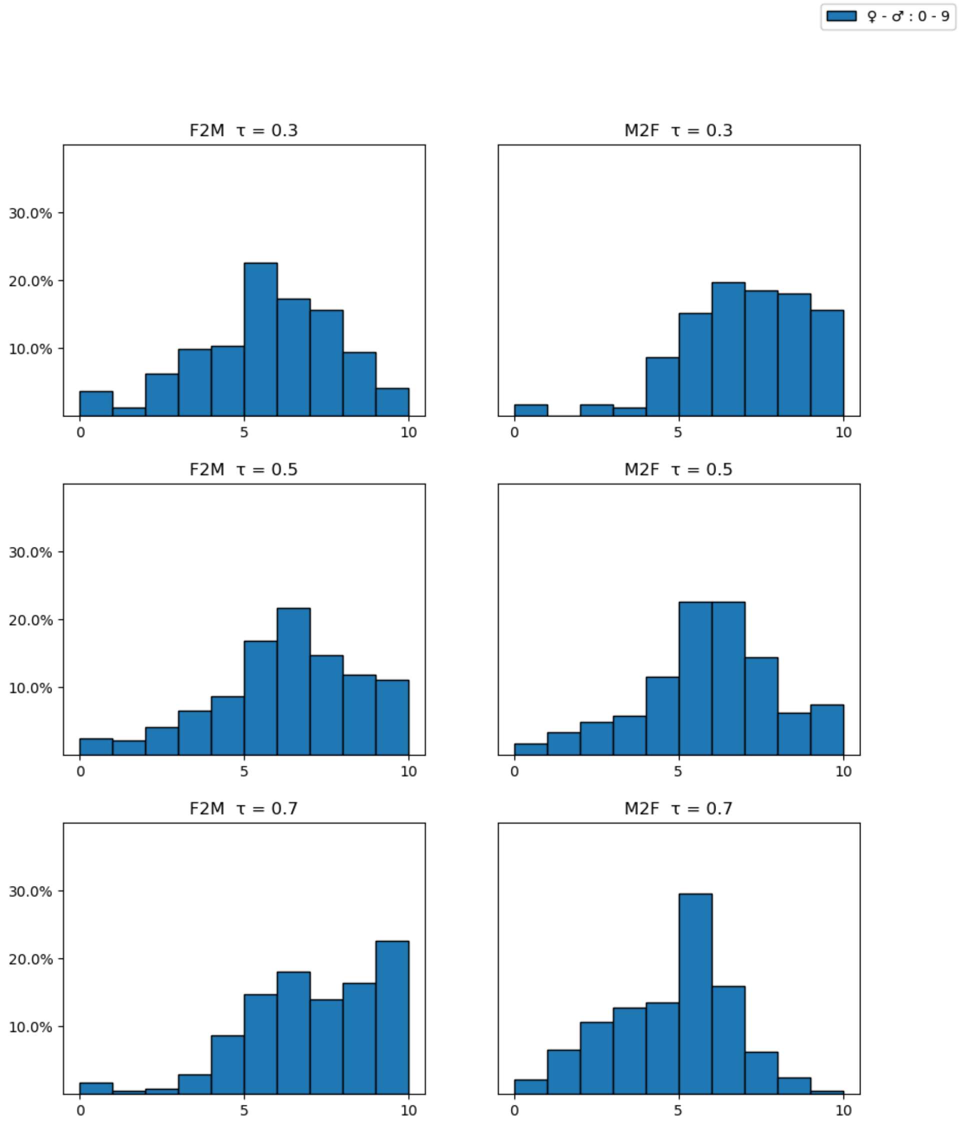


*Figure 5.3: Histograms of perceived gender for the six modified classes of audio samples. $\tau$ denotes the interpolation constant.*

Table 5.1 displays mean and standard deviation for all eight classes, and the percentage of survey participants who found the voices in the classes sounding natural. Looking at the mean ratings we can see that it correlates with the value of $\tau$ for both the female-to-male(F2M) and male-to-female(M2F) classes, but the standard deviation is large, which can be seen in the spread in the histograms as well. Note that the variation of the unmodified classes is not lower.

Looking at the naturalness however, there is a large difference between the modified classes and the unmodified ones.

The interpolation triplet results are shown in Table 5.2. The "Individuals" column

| Class | $\tau$ | $\mu$ | Rated natural |
|---|---|---|---|
| F2M | 0.3 | $5.2(\sigma = 2.1)$ | 63% |
| F2M | 0.5 | $5.8(\sigma = 2.2)$ | 54% |
| F2M | 0.7 | $6.6(\sigma = 2.0)$ | 58% |
| | | | |
| M2F | 0.3 | $6.5(\sigma = 1.9)$ | 76% |
| M2F | 0.5 | $5.4(\sigma = 2.0)$ | 66% |
| M2F | 0.7 | $4.3(\sigma = 1.8)$ | 29% |
| | | | |
| Female | - | $1.2(\sigma = 1.9)$ | 93% |
| Male | - | $7.3(\sigma = 2.2)$ | 90% |

*Table 5.1: Survey results showing mean class rating $\mu$ (and standard deviation $\sigma$) as well as percentage of answers that found them sounding natural. $\tau$: interpolation coefficient*

displays the proportion of participants who perceived the triplets to be in the intended order. The global median and mean columns show if the median or mean where ordered correctly for the triplet. Looking at the global ratings, there seems to be consensus that the triplets reflect the desired effect of the interpolation for seven and six classes respectively. However, at the individual level this is not seen to a similar extent, with a maximum of 31% of correspondents ordering a triplet correctly.

A noteworthy relationship seen in the individual ratings of triplets is the ordering of the male-female pairs. M2F 4 has the highest percentage out of the M2F transitions, while F2M 4 has the lowest in F2M. This relationship is the same for all other pairs: ordering from best-to-worst in M2F is mirrored in worst-to-best in F2M. This could suggest that for some transitions, interpolating in one direction is easier than the other.

| Triplet | Individuals | Global median | Global mean |
|---|---|---|---|
| F2M 1 | 19.7 % | Yes | Yes |
| F2M 2 | 8.2 % | Yes | No |
| F2M 3 | 24.6 % | Yes | Yes |
| F2M 4 | 1.6 % | No | No |
| | | | |
| M2F 1 | 26.2 % | Yes | Yes |
| M2F 2 | 29.5 % | Yes | Yes |
| M2F 3 | 19.7 % | Yes | Yes |
| M2F 4 | 31.1 % | Yes | Yes |

*Table 5.2: Survey results for interpolation triplets for if they were rated in consecutive order. Showing percentage of individual survey respondents and ordering by global median and mean.*

# Chapter 6

# Discussion

In the following chapter we discuss results from the previous chapters. We also discuss how they could be improved and suggest methods for future research.

### Survey

The survey results indicate that the interpolation does change the perceived gender in the direction of the interpolation. They also clearly show that the voices sound less natural than an unmodified voice.

Comparing the transitions F2M and M2F, the results show that the interpolation coefficient affects the voices differently. The mean rating of class F2M with $\tau = 0.3$ is $\mu = 5.2$ which is already closer to the male end of the scale. It is even clearer looking at the histograms in Figure (5.3) where all six modified classes are skewed towards the male/right end. We had previously noticed that F2M morphed voices often sounded more male than M2F sounded female, so this does not surprise us.

The problem of how fast the interpolation should be changed in order to create an interpolation which sounds linear was not taken into account when constructing the interpolation. In future research a function which regulates the speed of the interpolation may be investigated. The shape of such a function may be estimated using survey results. Alternatively, the pipeline may be extended to accommodate time varying interpolation coefficients. The function can then be tuned such that if the interpolation coefficient varies linearly in time it results in a voice which sounds like it varies linearly in perceived voice.

The survey result regarding the speed of interpolation is also interesting from the point of view of the mel scale and how we perceive pitch. Since we are more sensitive to frequency changes at lower frequencies, interpolating linearly in pitch should result in a perceived higher change going from a lower to a higher frequency than vice versa if pitch is the most important indicator. The results suggest that the mel frequency scale might not translate well to how humans perceive gender in voices, and that pitch is not the only important factor in determining gender.

Some participants suggested using a scale for the naturalness of the voices. This could give a more accurate picture of the naturalness, as a scale might more accurately describe how the participants perceived the voices. It would also be much easier to assess naturalness of general speech, as a short snippet of a vowel is very little information to make a judgement based on.

The histograms of the ratings for the unmodified classes were more varied than expected and this could be a sign of weakness in the survey format. It is realistic to assume that when making an assessment of a voice, it is influenced by the most recent voices one has heard, which can lead to undesirable effects in the results. On the other hand, we thought it was more important to avoid any construction bias on our part. By randomizing the placements, we hoped for some independence within the interpolation triplet ratings. The survey format is also somewhat vague as the question of which gender is perceived leaves some room for interpretation. It could mean "How sure are you of the gender of the speaker?" or "How do you perceive the gender of this voice on a scale?". This type of ambiguity can also affect the way the participants answered and could explain some of the variability.

Regarding the percentages of participants who rated the triplets in consecutive order, the results leave much to be desired. We believe that also these results are affected by the survey format. This makes it hard to say what a good result is. One could imagine solving this problem by asking the participants to order the perceived gender of one triplet at a time. However, the triplets are easily sorted based solely on the pitch and most of the vocal tract information about gender would be disregarded. In the current survey format we think that a global consensus in mean and median on all triplets as well as a percentage of at least 50 % would be a required. The triplet results are also connected to the problem of finding a linear speed of interpolation. The perceived gender of the triplets may be further separated by choosing a better speed of interpolation. Further, we believe that these results could be improved by increasing the naturalness of the interpolated voices.

Another idea for the survey format is to always calibrate results against the same anchor, and letting the participant reset their reference after each voice in the following format:

1. Play and listen to anchor voice
2. Play and listen to a modified voice
3. Rate it on a scale
4. Repeat

The ordering of the modified voices could still be randomized, but with the repeated anchor reducing the aforementioned undesirable effects.

## Source-filter

From our survey and subjective evaluation of the voices we have produced, we can conclude that increasing the naturalness is an important subject for future research.

The nasality and low naturalness found in many of the male to female interpolations may be caused by some unintended effect of resampling the residuals or it may be due to a poor separation of glottal excitation and the vocal tract acoustics. A poor separation would mean that the pitch shifting method will affect more than the pitch. This suggest that the system may be improved by finding a better way of modeling the source and filter decomposition. One possibility is to use a sparse linear predictor as presented in [44]. It minimizes the one norm error of the residual, promoting sparse residuals which are dominated by the glottal pulses.

While a better filter modeling technique may result in filters which contain more information about the speaker identity this would increase the need for investigating different filter interpolation paths. Another interesting way of performing the smoothed spectrum morph is proposed in [59] where dynamic warping techniques are used to find a reassignment between smoothed spectra. The reassignment could be tested with our smoothed spectrum interpolation proposed in equation (4.14).

Our current system of voice morphing cannot produce a full voice conversion; a morph with interpolation coefficient $\tau = 1$ does not return a voice sounding like the target voice. This is because the residual is simply pitch shifted to the target frequency instead of being interpolated between the signals. A method for interpolating between the residuals of voices may increase the naturalness of signals synthesised with large interpolation factors. It may also circumvent the fact that the source-filter decomposition is poor. The problem of how such an interpolation should be constructed is not trivial.

In [60] a method is proposed where dynamic time warping techniques are used to construct an interpolated residual. The system uses an interpolation similar to the one we proposed in equation (4.14). The author, however, notes that the voices which are investigated share some characteristics which are not universal, and the framework might not work for general voices.

Another alternative for interpolating the residuals of voices is proposed in [52], where the waveform interpolation techniques are used to extract a three dimensional representation of the excitation. Two residuals are then interpolated by a linear interpolation between two such three dimensional representations. Other alternatives for interpolating the excitation signals may include using a parametric model and interpolating the model parameters.

The period picking algorithm has some drawbacks. One of the restrictions is a result of limiting the search for fundamental frequencies close to the median fundamental frequency. This limits the method to marking periods in signals with fairly uniform fundamental frequencies. Another weakness is as previously stated that the autocorrelation method often finds multiples the fundamental frequency and there is no guarantee that the re-estimation solves this problem. Further, finding maximum values in the signal is heavily influenced by noise and relies on the assumption that there is one unique maximum point of the signal in each period. There may exist pitch tracking algorithms which allow for more widely varying fundamental frequencies and are more robust. We think that a more robust period picking method may increase the naturalness of the voices.

## Deep learning

Under the assumption that the main bottleneck in the deep learning experiments was the size of the data set, a technical solution could be to construct a matched data set of female and male voices saying the same sentence using dynamic time warping for alignment. There exists such data sets online and they could also be created by extracting audio from videos on streaming sites.

It would also be interesting to explore a GAN like architecture like the one presented in [61]. GAN networks are set up like a game where a generator is trying to fool a discriminator [61]. The generator's objective would be to transform the gender identity of a voice, and the discriminator's to detect if it is a real sample from the set of male/female voices.

## General speech

As of now the pipeline generates a voice saying "ah". From this alone it might be hard to asses how natural the voice sounds. The framework can be extended to work with general speech, such as full sentences. For the current method this would require a data set where voices are perfectly aligned in time, which is challenging to collect. However, in previous research, dynamic time warping has been used in order to align the rhythm of different speakers which could solve this problem [62].

The period picking in the PSOLA would also have to be modified as it currently assumes a stationary pitch, which is not true for general speech.

# Concluding remarks

With the data provided for the project only the source-filter method yielded results that were worth investigating in a survey. After evaluating the results in the survey we can conclude that our interpolation did affect the perceived gender of the voices. However, the results suggest that there is a need to improve the naturalness of the morphed voices, and are not satisfactory for any implementation in a voice therapy tool.

Our initial research questions were:

- Can spectral methods be used to find a transformation of a voice from female to male and vice versa?
- Is it possible to find an interpolation between two voices of different gender?

In regards to the first one, we can not conclude that there is a full conversion with the methods we tried. Even still our results indicate that the voice interpolation does change the perceived gender as desired. However, since we could not find the complete transform, the interpolation did not sound good for interpolations close to the target voice. We think that either better separation of vocal tract characteristics and excitation signal, or interpolation in excitation signal is required for this.

We have suggested potential methods for improving the naturalness and modifications to the survey format for better evaluation. We have also suggested methods for extending the source-filter framework to more general speech.

# Chapter 7

# Bibliography

[1] National Centre for Transgender Equality, "Frequently asked questions about transgender people." https://transequality.org/issues/resources/frequently-asked-questions-about-transgender-people. Accessed: 6-10-2020.

[2] American Psychiatric Association, "Gender dysphoria." https://www.psychiatry.org/File%20Library/Psychiatrists/Practice/DSM/APA_DSM-5-Gender-Dysphoria.pdf, 2013.

[3] American Psychiatric Association, "What is gender dysphoria?." https://www.psychiatry.org/patients-families/gender-dysphoria/what-is-gender-dysphoria, 2016.

[4] S. Davies, V. G. Papp, and C. Antoni, "Voice and communication change for gender nonconforming individuals: Giving voice to the person inside," *International Journal of Transgenderism*, vol. 16, no. 3, pp. 117–159, 2015.

[5] Abitbol, "Sex hormones and the female voice," *Journal of Voice*, vol. 13, no. 3, p. 424–446, 1999.

[6] Purr Programming, "Voice pitch analyzer." https://play.google.com/store/apps/details?id=de.lilithwittmann.voicepitchanalyzer. Accessed: 5-10-2020.

[7] I. McLoughlin, *Applied Speech and Audio Processing: With Matlab Examples.* USA: Cambridge University Press, 1st ed., 2009.

[8] J. Sundberg, *Röstlära - Fakta om rösten i tal och sång.* Malmö: Prinfo Team Offfset Media, 2001.

[9] L. Carew, G. Dacakis, and J. Oates, "The effectiveness of oral resonance therapy on the perception of femininity of voice in male-to-female transsexuals," *Journal of Voice*, vol. 21, no. 5, pp. 591 – 603, 2007.

[10] I. R. Titze, "Physiologic and acoustic differences between male and female voices," *Acoustical Society of America Journal*, vol. 85, pp. 1699–1707, Apr. 1989.

[11] M. Farrús, J. Hernando, and P. Ejarque, "Jitter and shimmer measurements for speaker recognition," pp. 778–781, 01 2007.

[12] A. S. H. Howard C. Nusbaum, Alexander L. Francis, “Measuring the naturalness of synthetic speech,” *Int J Speech Technol*, no. 2, pp. 7 – 19, 1997.

[13] X. Hou, K. Sun, L. Shen, and G. Qiu, “Improving variational autoencoder with deep feature consistent and generative adversarial training,” *Neurocomputing*, vol. 341, pp. 183–194, 2019. `https://github.com/houxianxu/DFC-VAE`.

[14] A. Jakobsson, *An Introduction to Time Series Modeling.* Malmö: Studentliteratur, 2015.

[15] D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications.* Cambridge: Cambridge University Press, 1993.

[16] P. Stoica and R. Moses, *Spectral Analysis of Signals.* Upper Saddle River, New Jersey: Prentice Hall, 1997.

[17] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing.* Pearson Education, 4 ed., 1996.

[18] G. Lindgren, H. Rootzén, and M. Sandsten", *"Stationary stochastic processes for scientists and engineers".* "Chapman and Hall", 2013.

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

[20] M. Sandsten, “Time-frequency analysis oftime-varying signals and non-stationary processes - an introduction.” `http://www.maths.lu.se/fileadmin/maths/personal_staff/mariasandsten/TFkompver4.pdf`, 2020.

[21] Encyclopedia Brittanica, “The decibel scale.”
`https://www.britannica.com/science/sound-physics/The-decibel-scale`.

[22] Encyclopedia Brittanica, “Dynamic range of the ear.”
`https://www.britannica.com/science/sound-physics/Noise#ref64015`.

[23] S. S. Stevens, J. Volkmann, and E. B. Newman, “A scale for the measurement of the psychological magnitude pitch,” *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[24] D. O’Shaughnessy, “Speech communication: Human and machine.,” *Journal of the International Phonetic Association*, vol. 20, no. 2, p. 52–54, 1990.

[25] Librosa development team, “librosa.filter.mel.” `https://librosa.org/doc/latest/generated/librosa.filters.mel.html`. Accessed: 7-10-2020.

[26] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” 2017.

[27] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, “Transfer learning from speaker verification to multispeaker text-to-speech synthesis,” 2018.

[28] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” 2017.

[29] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *The Journal of the Acoustical Society of America*, vol. 50, no. 2B, pp. 637–655, 1971.

[30] A. Edelman and H. Murakami, "Polynomial roots from companion matrix eigenvalues," *Mathematics of Computation*, vol. 64, no. 210, pp. 763–776, 1995.

[31] C. Villani, *Optimal Transport.* Berlin, Heidelberg: Springer, 2009.

[32] T. Henderson and J. Solomon, "Audio transport: A generalized portamento via optimal transport," 2019.

[33] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, 01 2000.

[34] X. Nguyen, "Wasserstein distances for discrete measures and convergence in nonparametric mixture models," 2011.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[36] OpenAI, "Better language models." `https://openai.com/blog/better-language-models/`.

[37] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," 2017.

[38] J. Corentin, "Automatic multispeaker voice cloning," Master's thesis, Université de Liège, 2019. `https://github.com/CorentinJ/Real-Time-Voice-Cloning`.

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

[40] L. Zhang, L. Wang, J. Dang, L. Guo, and H. Guan, "Convolutional neural network with spectrogram and perceptual features for speech emotion recognition," in *Neural Information Processing* (L. Cheng, A. C. S. Leung, and S. Ozawa, eds.), (Cham), pp. 62–71, Springer International Publishing, 2018.

[41] NVIDIA, "Tacotron 2 (without wavenet)." `https://github.com/NVIDIA/tacotron2`, 2018.

[42] R. Girshick, "Fast r-cnn," 2015.

[43] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[44] D. Giacobello, M. Christensen, M. Murthi, S. Jensen, and M. Moonen, "Sparse linear prediction and its applications to speech processing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 1644 –1657, 07 2012.

[45] F. Villavicencio, A. Roebel, and X. Rodet, "Improving lpc spectral envelope extraction of voiced speech by true-envelope estimation," vol. 1, pp. I – I, 06 2006.

[46] J. O. S. III, "Cepstral smoothing." `https://ccrma.stanford.edu/~jos/SpecEnv/Cepstral_Smoothing.html`. Accessed: 06-10-2020.

[47] M. Morise, “Cheaptrick, a spectral envelope estimator for high-quality speech synthesis,” *Speech Communication*, vol. 67, 01 2014.

[48] D. Schwarz, “Spectral envelopes in sound analysis and synthesis.” `http://recherche.ircam.fr/anasyn/schwarz/da/specenv/Spectral_Envelopes.html`. Accessed: 19-10-2020.

[49] V. Goncharoff and M. Kaine-Krolak, “Interpolation of lpc spectra via pole shifting,” in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 780–783 vol.1, 1995.

[50] S. Krstulovic, “Lpc modeling with speech production constraints,” 2000.

[51] M. Caetano and X. Rodet, “Automatic timbral morphing of musical instruments sounds by high-level descriptors,” *International Computer Music Conference, ICMC 2010*, 06 2010.

[52] Y. Lavner and G. Porat, “Voice morphing using 3d waveform interpolation surfaces and lossless tube area functions,” *EURASIP Journal on Advances in Signal Processing*, 05 2005.

[53] I. V. McLoughlin, “Line spectral pairs,” *Signal Processing*, vol. 88, no. 3, pp. 448 – 467, 2008.

[54] G. Upperman, M. Hutchinson, B. V. Osdol, and J. Chen, *Methods for voice conversion.* OpenStax CNX, 2010.

[55] A. Mousa, “Voice conversion using pitch shifting algorithm by time stretching with psola and re-sampling,” *Journal of Electrical Engineering*, vol. 61, p. 2011, 06 2011.

[56] L. Tan and M. Karnjanadecha, “Pitch detection algorithm: autocorrelation method and amdf,” 01 2003.

[57] Mathworks, “Interpft.” `https://www.mathworks.com/help/matlab/ref/interpft.html`. Accessed: 07-10-2020.

[58] B. Lawlor, “A novel efficient algorithm for voice gender conversion,” 1999.

[59] H. Pfitzinger, “Dfw-based spectral smoothing for concatenative speech synthesis,” in *INTERSPEECH*, 2004.

[60] H. R. Pfitzinger, “Unsupervised speech morphing between utterances of any speakers,” in *In proc. of the 10th australian int. conf. on speech science and technology (SST 2004*, pp. 545–550, 2004.

[61] S. Pidhorskyi, D. Adjeroh, and G. Doretto, “Adversarial latent autoencoders,” 2020.

[62] V. L. Latsch and S. L. Netto, “Pitch-synchronous time alignment of speech signals for prosody transplantation,” in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 2405–2408, 2011.

# Appendix A

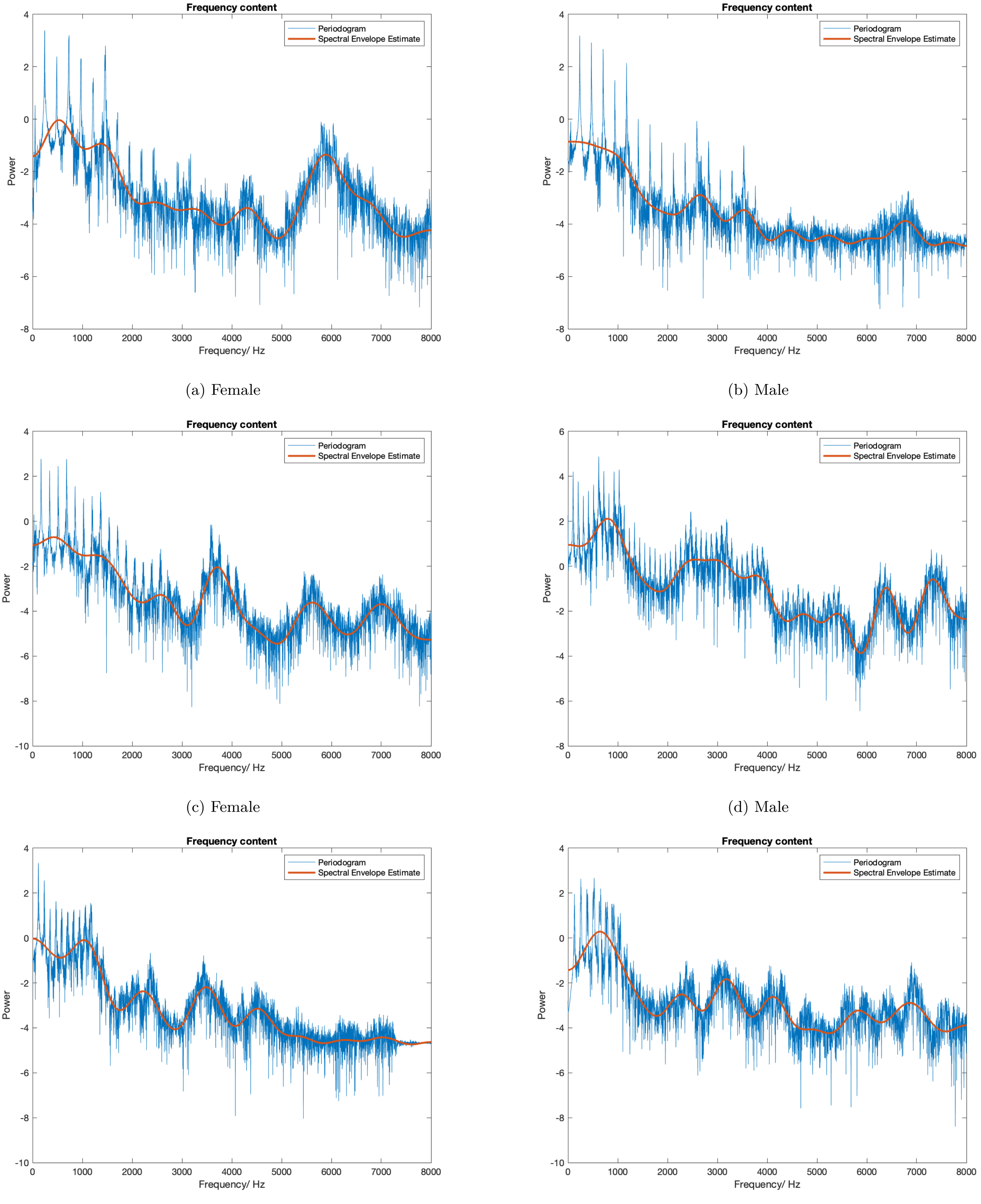
# Envelope interpolation code

```
function [interp] = envelope_interpolation(left,left_ind,right,right_ind,k)
%Interpolation between the left and right signal using the assignment specified
%in left_ind and right_ind. k is the interpolationcoefficient.
path_len = length(left_ind);
%Initialize interpolation with an extra bin due to roundoff error.
len = max([length(left),length(right)])+1;
interp = zeros(len,1);
% vector to keep track of the weights of a given bin.
line_vec = zeros(len,1);
for i = 1:path_len
    ind = (1-k)*left_ind(i) + k*right_ind(i);
    less = floor(ind);
    mor = ceil(ind);
    weight = ind-less;
    if weight == 0
        interp(less) = interp(less) + (1-k)*left(left_ind(i))...
        + k*right(right_ind(i));
        line_vec(less) = line_vec(less) +1;
    else
        interp(less) = interp(less) + (1-weight)*...
        ((1-k)*left(left_ind(i))+ k*right(right_ind(i)));

        interp(mor) = interp(mor) + weight*
        ((1-k)*left(left_ind(i))+ k*right(right_ind(i)));

        line_vec(less) = line_vec(less) + (1-weight);
        line_vec(mor) = line_vec(mor) + weight;
    end
end
% Fix roundoff error.
interp(end-1) = interp(end-1)+interp(end);
interp = interp(1:end-1);
ind = (line_vec(1:end-1) ~= 0);
interp(ind) = interp(ind)./line_vec(ind);
end
```

# Appendix B

# Spectral representations of voices

Additional spectral representation of male and female voices. Envelopes estimated with a rectangular lifter of order 20.



(a) Female

(b) Male

(c) Female

(d) Male

(e) Female

(f) Male

*Figure B.1: Some examples of periodograms and estimated spectral envelopes.*