New speech-inspired tools for exploring timbre in computer-based composition and music production

Michele Pizzi

PhD

University of York

Music

September 2018

Abstract

Speech processing can be used in music to transfer the qualities of the human voice to another recorded sound. The use of technologies employed in phonetics and telecommunications provide the creative potential to use speech to design sound transformations over time and to use the features of the human voice to control other sound sources. Computer music tools are often viewed as treating the voice as a time-varying resonator, and this is true, but it could also be argued that the voice and its time-varying anti-resonant capabilities have been under-emphasised in the tools and in the literature. This thesis explores possibilities of including anti-resonance and examines types of vocal sound that are characterised at least partly by anti-resonance. The aim of the research presented in this thesis is to explore technology as a means of transferring the timbre of human voice to another sound source. The estimation of the resonant structures of the vocal tract, through the audio analysis of recorded speech, provides a way to extract the features of voice and combine its timbre with another sound. This enables the extraction of the colour of speech sounds to control sound synthesis, in order to design transformation over time informed by knowledge of the acoustics of the human voice.

List of Contents

List of Figures

List of Accompanying Material

Digital copy of Audio examples:

Example 1.1 - Example vowels cascade formant synthesizer

Example 1.2 - Isolated formants vowel a

Example 1.3 - Example of diphthong

Example 1.4 - Example of synthetic vocal fry voice source

Digital copy of accompanying Audio tracks.

Track n.1	*Vocal Fry* (2015)	10’12’’
Track n.2	*Fricatives* (2016)	5’01’’
Track n.3	*Whispers* (2018)	3’16’

Acknowledgements

Thank you to all the wonderful people of the Music Department who supported me during this journey.

A big heartfelt thank you goes to my supervisor Jez Wells whose expertise, guidance and patience have been invaluable.

Paulina, your immense support has been of great comfort during difficult moments.

Finally, this work is dedicated to my family. You are my pillar. I love you all very much. Thank you.

Author’s Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Chapter 1: Introduction

1.1 Motivation

I discovered my interest in speech processing during my Masters degree at the University of York. The structure of the course encouraged me to explore technology I wasn't familiar with in a creative way. I was captivated by an unusual piece of software named Praat that is employed in phonetics for the analysis and re-synthesis of speech. The idea of using a tool employed in phonetics to create music was incredibly fascinating to me and I started exploring the different functions offered by Praat. By learning the different features of Praat I was at the same time investigating the features of the human voice and the connections between speech processing and music. In particular, what captured my attention was the idea of the human voice as a filter. Filters are used in music production to shape the frequency content of sounds and I was excited to investigate how filters could be controlled with your own voice. Maybe I could create those robotic voices I heard many times on songs by Kraftwerk and Daft Punk. How about creating a 'talking' guitar or a 'singing' saxophone? The idea that the resonances of the vocal tract could be converted into a filter inspired many experiments with sound that I undertook. With Praat I could convert recordings of speech into audio filters changing over time and use this data to control any other sound source. To me, Praat was (and still is) like a workbench to dismantle, re-assemble and combine any recorded sound. I could design vowel-like sounds by taking advantage of audio analysis of voice. This thesis aims to continue this journey into exploring how the human voice can be investigated in music through technology. The idea was to study different ways to produce, model or approximate the resonant structures of the vocal tract. It was a joy to me to study the properties of vowels, how they are different from consonants and how it is possible to synthesize them. The human voice is able to produce a wide range of sounds, and technology offers the possibility to combine these sounds with other sources. Vowels and consonants have a specific set of resonances that can be explored. Resonances are frequency bands that are enhanced and more prominent within the timbre of a sound and are also called formants. Technologies for telecommunications and sound synthesis in music and phonetics have successfully provided the ability to model these resonances as a filter to re-create the human voice or to transfer them to another sound source. However, particularly in nasal consonants, the resonant structures of speech display frequency bands that are attenuated. These attenuations result in dips or valleys in the spectral envelope of sound and are called anti-formants. I am interested in studying their effect on the processing of sounds. I aim to explore the potential of anti-formants approximation and its creative use in music. The goal is to investigate the qualities of the very building blocks of speech and to discover these apparently simple sound sources by approximating their features with different techniques and to study their effect and perception. My main idea is to take an accurate 'photograph' of two different speech sounds and perform interpolations that take into account the effect of anti-resonances, as well as resonances, on the resulting time-varying spectral envelope and to explore how this morphing can be used creatively to assist the design of transformations of sounds. With this thesis, I aim to offer new digital tools that make use of developments in digital signal processing to successfully transfer the qualities of recorded voice to other sounds using knowledge of the features of the human voice and related technology. The techniques and audio examples described in this thesis were used to explore sound transformations informed by the qualities of speech. I developed these techniques as my personal way to shape the timbre of sounds and their purpose is to combine existing knowledge in music, speech processing, filter design and acoustics in order to identify the parameters of sound I used to implement my tools.

The research challenge of this thesis is to understand the implications of mathematical manipulation on the features of sound through digital filter design and signal processing techniques. The aim is to create a set of tools that brings to the recording studio an intuitive method to perform sound transformations. The curiosity of discovering new opportunities to control sound synthesis and processing recorded sounds has been the catalyst driving this research. This wouldn’t have been possible without understanding what mathematical manipulations mean in terms of variations in frequency, bandwidth or amplitude. For example, to understand how moving poles and zeros within a unit circle affects timbre and can be used to design dramatic changes in character and what it means sonically. I encapsulated into intuitive instruments a combination of techniques and concepts drawn from different disciplines, and I hope they can be useful tools to composers, music producers, sound designers and anyone working with audio.

1.2 Research questions

The hypotheses of this thesis are expressed in the form of the following research aims:

- How can digital signal processing techniques enable the creation of tools to better represent the qualities of the voice in order to extend, improve or generate new opportunities for timbre control?
- In what new ways can the mathematical manipulation of poles and zeros provide novel techniques to create meaningful timbre transformations?
- To what extent can speech processing be used creatively to emulate any existing audio processing techniques?
- What is the role of the vocalist (or speaker) when the required performance is designed for sound analysis and audio processing?

This dissertation adopts existing digital signal processing ideas and brings them into new tools. The results offer new opportunities or extend existing ones, enhancing their quality across a broader range of sounds. These new opportunities are introduced by the development of new audio-specific, and voice-specific, processing techniques for use by composers.

This research looks at ways in which the current tools can be extended by adopting additional signal processing techniques which have been typically deployed, for example, in telecommunications. Composers such as Paul Lansky use Linear Predictive Coding (LPC) that deals with characterising resonances but completely ignores anti-resonances that are an important element of speech. In fact, they play a bigger part in certain languages than they do in English. This is because anti-resonances are present in nasal consonants and vowels that certain languages use more than others. The Polish or French languages, for instance, include more nasalized sounds than English. Therefore, tools that take this into account and also apply the idea of poles and zeros in new ways have the potential to extend and augment the sonic palette. This is important, especially when considering sound design and composition techniques based on speech. This research also demonstrates how these tools can be useful by generating new pieces that feature innovative techniques of timbre manipulation based on speech. The resulting compositions have meanings which can be explained in a similar way to the work of the pioneers of speech synthesis whose works are described in the following chapters.

1.3 Overview

Chapter 2 contains a review of the literature concerned with the investigation of different aspects and features of the human voice. The acoustic properties of the voice organ are described in order to study the vocal tract and to explore how the vocal tract as a time-varying filter enhances or attenuates certain frequency bands during the production of certain speech sounds. Chapter 2 then describes the technology and techniques employed in speech processing to model the acoustic properties of speech and to extract information from recorded speech. An overview is provided to describe how the features of the voice and speech processing tools have been employed in computer music.

Chapter 3 provides an overview of how an approach to Prony's method enables the approximation of anti-resonances as well as resonances from recorded speech. This chapter outlines some of the differences between linear predictive coding (LPC) and Prony's method.

Chapter 4 investigates how vowels emerge from resonance and how the perception of speech sounds can be altered with a collection of techniques developed during the creation of the composition *Vocal Fry*.

Chapter 5 provides a collection of techniques to perform transformation of fricative consonants into vowel sounds. Chapter 6 demonstrates a collection of techniques to perform sound interpolation from the analysis of vowels and nasal consonants. The chapter explores an approach to capture the anti-resonant qualities of speech with new tools.

Chapter 7 outlines the main findings and compares the result of the investigations included in each chapter with the research aims described here.

Chapter 2

2.1 Context, scope and area of research

This chapter describes the literature that is presented by investigating links between different areas of research concerned with various aspects of the human voice. In order to investigate the multi-disciplinary nature of the topics described in this thesis, the literature is organised into a framework consisting of three areas of research:

- Acoustic features of the voice organ;
- Music production, computer music and sound design;
- Speech coding and synthesis.

This chapter outlines how these three areas of research are explored to find connections between them and includes a proposed methodology to answer the research questions.

From the outset, it explores the possibilities of software for speech analysis and re-synthesis (Praat) with the idea of making music with technology designed for phonetics.

Much information can be extracted from recorded speech through audio analysis and the features extracted can be used to explore the creation of hybrid sounds controlled by speech itself. A ‘talking’ guitar is an example of a sound that it is possible to create.

The study of how the voice organ works, and how its features can be represented on the computer, provides a means to create an equivalent model and to replace the vocal folds or the vocal tract with any sound file. It is possible to capture or 'sample' the parameters for subtractive synthesis from speech to perform sound transformations. All the actions described above share the common idea of representing the voice organ with the source-filter model of speech production as opposed to other approaches such as concatenation of speech sounds. This idea is the main feature behind the design of the tools built to explore the research questions.

For the scope of this thesis, acoustic features of the voice organ, speech coding and music are described and used to inform design and development of the digital tools and audio processing techniques that form the output of this work. The concepts analysed in the literature are explored to design audio processing tools; the resulting software is then tested in musical examples and music production techniques to identify the parameters that provide meaningful control over sound.

2.2 The vocal tract: a time varying filter

The following section describes the features of the voice organ and outlines how these affect the type of sounds that can be produced and are used to communicate as speech. Vowels and consonants, for example, are building blocks of spoken language and possess individual characteristics that the speaker can control. Through the movements of the jaw, tongue and pharynx it is possible to control the qualities of the sound being produced.

The voice organ has three main parts and each part plays its role in the production of speech. As described by Sundberg, each unit of the voice organ has a function: the lungs act as compressor, the vocal folds act as an oscillator and the vocal tract as a resonator (filter).[1] The lungs act as a power source forcing the flow of air through the vocal folds, their vibration generates a sound that is called voice source. The sound of the voice source is then modified by the vocal tract that acts as a resonator(filter).

[1] Johan Sundberg, *The science of the singing voice* (Dekalb, Illinois: Northern Illinois University Press,1987), 10.

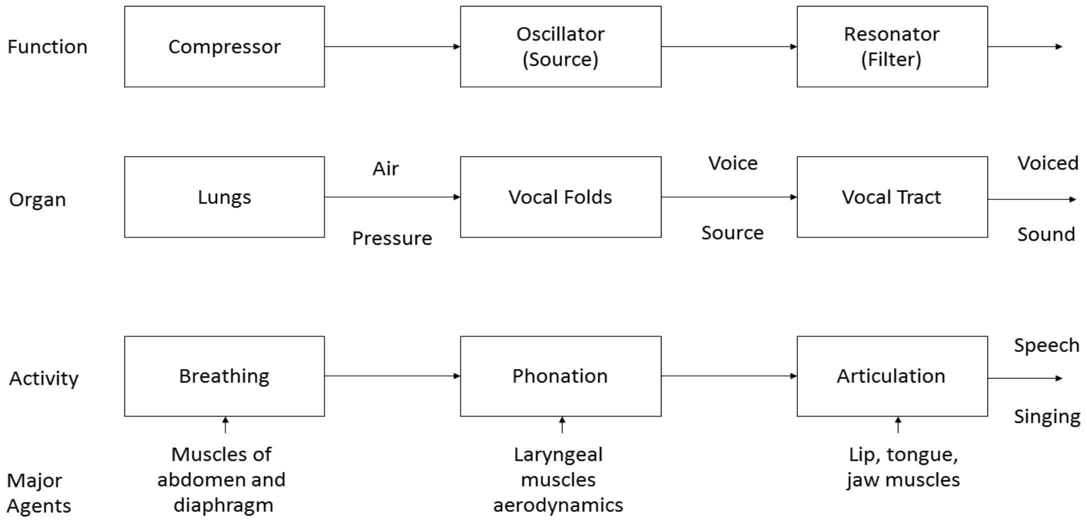


Figure 1. Chart outlining the functional constituents of the voice organ. After Sundberg.[2]

Figure 1 maps the process in a chart that assigns to each organ an activity and a function, this arrangement describes the vocal folds as an oscillator and the vocal tract as a filter. This approach is very helpful in dividing the vocal tract into separate features that can be studied and considered separately. It is a simplified but intuitive model to present something as complicated as the human voice in three stages. This thesis focuses on the idea of the vocal tract as a filter, and on the human voice as a model that provides independent control over the spectral content of a sound source (oscillator) and the spectral envelope of a resonator (filter).

Spoken language involves the use of a variety of sounds and not all sounds produced by the voice organ involve the vibration of the vocal folds. There are three types of sound sources that can be generated by the voice organ:

- Voiced: Sources that involve the vibration of the vocal folds. The resulting signal has a pitch.

[2] Sundberg, *The science of the singing voice*, 10.

- Voiceless or unvoiced: To produce this type of source the air is forced through a narrowing within the vocal tract, creating turbulence and generating a noise source lacking in pitch.
- Mixed: this involves the production of both voiced and unvoiced sources.

Interestingly, Howard and Murphy and Fant describe a similar classification of sound sources as voiceless, voiced and mixed and as no source, voice source only, the mixed voice (as noise sources and voiced source) and noise source (or several).[3]

The vibration of the vocal folds is the result of the Bernoulli effect that causes the closure and opening the vocal folds.[4] The pitch of the voice source is given by how many times the vocal folds open and close per second, the speaker or singer can control the pitch contour of the voice source by controlling the tension of the vocal folds. Sundberg describes how the laryngeal musculature can be used to control the tension of the vocal folds. The resulting sound can be compared to a pulse train, in which frequency is controlled by the speaker; a pulse train represents an approximation of the waveform produced by the vocal fold vibration.[5] Therefore, the sound source is spectrally bright. The Bernoulli effect causes glottal pulses, the vocal fry register is an example of one type of voice source that makes this effect more obvious. Vocal fry is a type of voice source that has a very low frequency of phonation; this register of voice allows one to perceive each individual voice pulse.[6]

There exists a pair of folds similar to vocal folds called ventricular folds; they are placed a few millimetres above (see Figure 2). These can be used for the production of specific sound sources that are described in section 2.3 and in chapter 4.

The vocal tract is considered a resonator (filter) as it is the unit of the voice organ that shapes the frequency content of the voice source during sound production.

[3] Gunnar Fant, *Acoustic theory of speech production: With calculations based on X-ray studies of Russian articulations* (Paris; The Hague: Mouton,1970): 18; David M. Howard and Damian Murphy, *Voice science, acoustics and recording* (San Diego; Oxford: Plural Publishing, 2008):30-39.

[4] Sundberg, *The science of the singing voice,* 64.

[5] Ibid.*, 16.*

[6] Ibid.*,* 50.

The resonant (or preferred) frequencies appear as peaks in the frequency response of the vocal tract.[7]These peaks are called formants and are peaks in the magnitude response of the vocal tract. They are not dependent on the fundamental frequency of vibration of the vocal folds (voice source).

Changes in the shapes of the vocal tract, through the use of articulators such as the jaw, tongue, lip, muscles, also change the position of formants in the magnitude response of the vocal tract. The shape of the vocal tract changes constantly during spoken language, therefore, the vocal tract acts as a filter that changes over time. Formant frequencies depend on the length of the vocal tract.[8] The features of the oral and nasal cavities, as well as the function of moving parts in the voice organ, can be defined as articulators.[9]

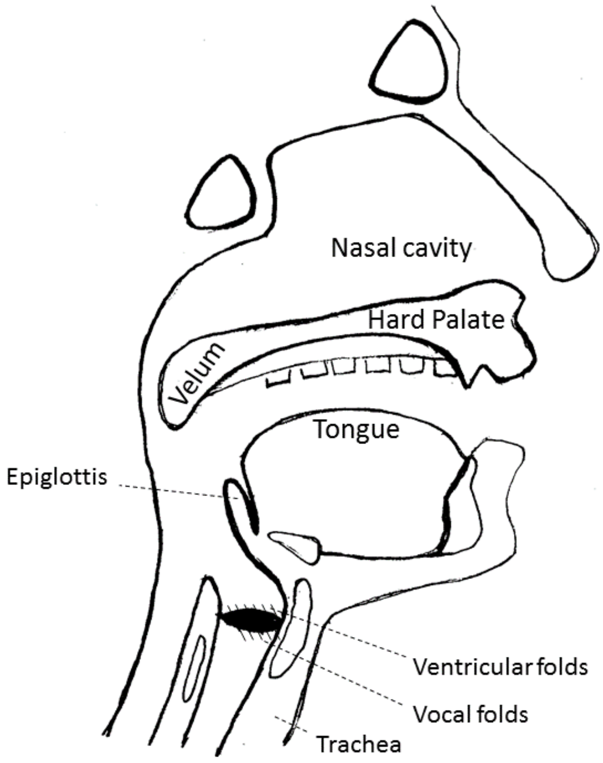


Figure 2. Schematic representation of the voice organ (mid-sagittal profile). After Sundberg[10]

[7] David M. Howard and Damian Murphy, *Voice science, acoustics and recording* (San Diego; Oxford: Plural Publishing, 2008), 45.
[8] Sundberg, *The science of the singing voice,* 20.
[9] Howard and Murphy, *Voice science, acoustics and recording,* 33.
[10] Sundberg, *The science of the singing voice,* 7.

There are two cavities in the voice organ that affect the frequencies of formants:

- The oral cavity is defined by the distance between the glottis and the lips.
- The nasal cavity is defined by the distance between the velum (see Figure 2) and the nostrils.

As the formant frequencies are related to the length of the vocal tract they vary between different individuals and with age. For example, men and women have different formant frequencies and children have different formant patterns from adult individuals; this is caused by the difference in size between men and women and adults and children. Formant frequencies extend across the audible frequency range, however for voiced sounds the energy output of the voice source decreases with increasing frequency. Therefore, the combination of formants and voice source leads to a pattern of formants which in general is diminishing with increasing frequency. The first three formants are usually considered for speech production and the perception of vowels. Higher formants (for example the fourth, fifth and sixth formants) are relevant to the specific features of the individual's vocal tract and display idiosyncratic acoustic signatures between individuals.[11] Also, Sundberg describes how the fourth formant in particular is important to define the personal vocal timbre.[12] This suggests the need to consider how many formants to use in order to model the timbre of voice through sound synthesis to explore the qualities of speech. A model which considers more formant frequencies will allow one to represent more personal features of the speaker.

This and the following chapters make use of the IPA (International Phonetic Alphabet) phoneme notation to refer to specific speech sounds. It is a convenient notation to express through text the sound of vowels and consonants itself. It is also used in the discussion of the creation of original compositions, speech synthesis and timbre manipulation techniques in further chapters. There exist useful resources that outline the IPA notation with reference to English keywords, for example, to intuitively see the relationship between letters and sounds.[13]

[11] Howard and Murphy, *Voice science, acoustics and recording,* 45.

[12] Sundberg, *The science of the singing voice,* 101.

[13] Ray D. Kent and Charles Read, *Acoustic analysis of speech* (Albany, NY: Singular Press, 2002), 248;

The nasal cavity acts as an anti-resonator as well, in nasalized vowels and nasal consonants like /m/ and /n/. Interestingly its effect reduces the amplitude of certain bands of frequencies, thereby introducing valleys in the vocal tract's magnitude response. The resulting valleys are called anti-formants.

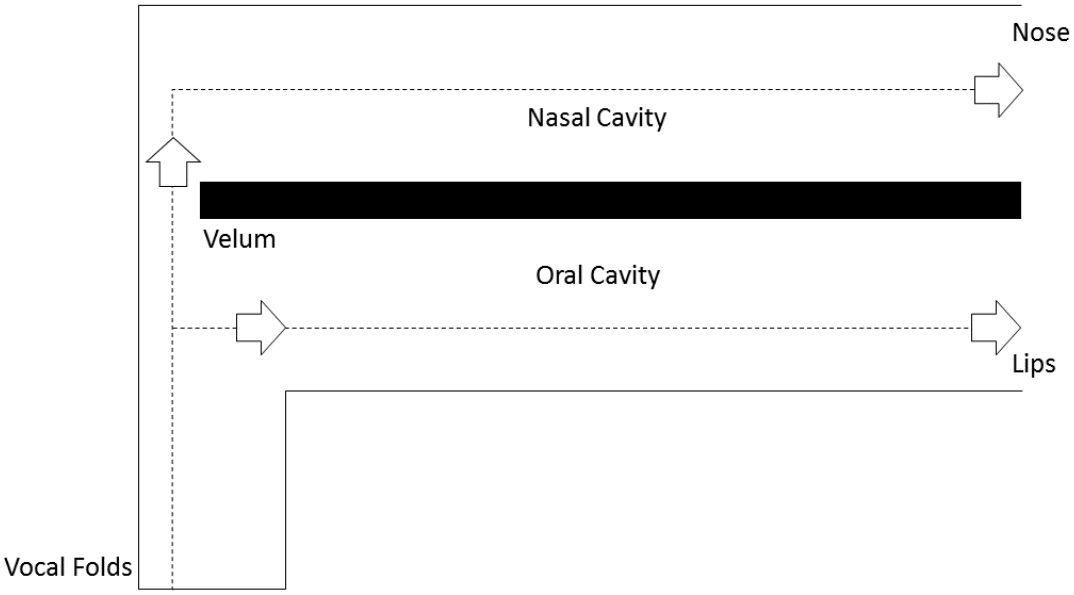


Figure 3. Effect of the nasal cavity in conjunction with the oral cavity: bifurcation in the resonating system.

Figure 3 shows an approximation of the mechanism of production of nasal vowels and nasal consonants. When the velum is lowered it allows sound to travel through both the nasal cavity and the oral cavity (nasalised vowels) or only the nasal tract, which introduces a bifurcation into the resonating system. Interestingly, in nasal consonants the oral cavity is blocked and has the effect of a side-branch (or splitting) introducing anti-resonances as a result of the energy absorbed by the side-branch of the nasal tract (the oral cavity). This is due to a feed-forward delay that causes a destructive interference between the output of the nose and mouth. This has an

Vivian J., Cook, "IPA Transcription of English Phonemes," In *The English Writing System* (Routledge, 2014): 215.

effect on the voice source, resulting in an overall amplitude lower than that of the vowel sounds.[14]

The effect of formants explains how the vocal tract acts as a filter that is independent from the source. It is possible to represent the voice organ with the source-filter model of speech production comprised of a source (voice source) and a filter (vocal tract). In spoken language we produce a variety of speech sounds by moving the articulators and producing different combinations of formant patterns that are peculiar to a specific type of sound. For example, each vowel sound is associated with a specific pattern of articulator adjustments which also produce a specific combination of formant frequencies.[15] Ladefoged explains how the difficulty of hearing formant frequencies individually leads us to think about vowels as a single meaningful entity rather than as a sound with separable elements.[16] Whispered (unvoiced) or very low pitch sounds (creaky voice) can be used as excitation of the vocal tract to produce different vowels and this helps to isolate the different formants highlighting the resonances and making them more obvious.[17] Whispering can be associated with white noise, a signal that contains a wide range of frequency content. Whispering doesn't involve the oscillation of the vocal folds and produces a noisy sound that makes the resonances of the vocal tract easier to perceive. For example, when whispering an /a/ and slowly making a continuous change towards /u/. By repeating the same slow change at a very low pitch (fry or creaky voice) it is possible to hear changes in vowel timbre more clearly. This suggests the idea of exploring the perception of vowels by isolating the individual resonances and providing control over the separable elements of speech sounds through sound synthesis.

Another type of sound related to vowels are diphthongs; these are not characterised by a single formant pattern, as the vocal tract changes shape during their

[14] Howard and Murphy, *Voice science, acoustics and recording,* 49.

[15] Sundberg, *The science of the singing voice*, 23.

[16] Peter Ladefoged, *Vowels and consonants: An introduction to the sounds of languages*. (Malden, MA; Oxford: Blackwell, 2005), 34.

[17] -Ibid., 34.

production.[18] Kent and Read explain how a single pulse can theoretically be used to define a vowel as it would produce the formant pattern associated with that vowel.[19]

During the production of nasal consonants certain regions of frequencies are attenuated by the effect of anti-formants. The nasal cavity produces a nasal formant (or nasal murmur) at approximately 300 Hz and consonants like /m/ and /n/ possess a series of anti-formants located around 1kHz, 3.5 kHz, and 5 kHz.[20]

Anti-formants are also called dips or spectral valleys due to the shape they create in the frequency response of the vocal tract. This means that the magnitude response of the transfer function displays gain values closer to zero where dips are present, resulting in an attenuation of the energy of the frequencies within the band of the anti-formant. Anti-formants are produced any time a bifurcation is created within the vocal tract; sounds produced by splitting the vocal tract in two parts due to the position of the tongue can display anti-formants in their frequency response (see Figure 3). Similar to nasal sounds, the consonant /l/ produces anti-formants in the transfer function of the vocal tract.[21] The human ear is very sensitive to the perception of peaks, in particular the first three formants that are associated with vowel sounds.[22] It is more difficult for the human ear to perceive the effect of a single anti-resonance; however, Cook explains that when many zeros (anti-formants) are changing over time in groups, their effect becomes noticeable and zeros provide clues about the distance and position of a sound source.[23] A feed-forward flanger is an example of an audio processor used in music that takes advantage of groups of anti-formants (or notch filters) by changing their position over time to create modulation effects.[24] Interestingly, according to Risset and Wessel, anti-resonances affect the quality of timbre by reducing roughness, for example, in string

[18] Kent and Read, *Acoustic analysis of speech*, 135.
[19] Ibid., 105.
[20] Howard and Murphy, *Voice science, acoustics and recording*, 49.
[21] Kent and Read, *Acoustic analysis of speech*, 181.
[22] Perry R. Cook, "Formant Peaks and Spectral Valleys" in *Music, cognition, and computerized sound: An introduction to psychoacoustics*, ed. Perry R. Cook (Cambridge, Mass: MIT Press, 1999): 136-136.
[23] Ibid., 136-137.
[24] Ibid., 137.

instruments.[25] Harmonics are equally spaced but the auditory filter broadens with frequency, so interactions between adjacent upper harmonics produce the beating effect on the basilar membrane that is associated with the sensation of roughness. Therefore, the introduction of anti-resonances attenuates the magnitude of certain frequency bands. These attenuations in the magnitude response might affect these high frequency partials that are close between one another, reducing harshness in timbre. This suggests the possibility of exploring creative transformations of sounds over time by moving the position of anti-formants estimated from consonant sounds and investigating their interaction with the position and bandwidths of the peaks in the magnitude response.

2.2 Speech coding and synthesis: formants and computers

The features of the vocal tract and formants have been explored in speech science to study the perception of speech sounds through the synthesis of speech; in speech coding the analysis and re-synthesis of signals is used to represent them for transmission in telecommunications.

The time-varying filter of the vocal tract can be tracked and approximated by extracting information about the formant frequencies from audio analysis of recorded speech. The resulting magnitude response can be used to shape any audio signal (source) fed to the filter. The goal is to use the resulting data about formants to control digital filters in order to perform transformations based on the analysis of speech. This will provide the means to use the human voice as a starting point for the synthesis of hybrid sounds and to inform the development of audio processors.

This section outlines how the vocal tract can be computer-modelled as filter, and how data about formants can be extracted from speech through the use of linear predictive coding.

Filters are used in subtractive sound synthesis to shape the frequency content of a sound by enhancing or attenuating certain frequency bands of a spectrum. A variety

[25] Jean-Claude Risset and David L. Wessel, "Exploration of timbre by Analysis and Synthesis". In *Psychology of Music*, 2nd edition, ed. Diana Deutsh (San Diego: Academic Press, 1999): 28.

of sound sources can be approximated by sculpting the frequency content of a sound source with a rich spectrum, from musical instruments to the human voice. A feed-forward digital filter finite impulse response (FIR) can be implemented by mixing a delayed copy of a given input sound source with the output of the filter while a feed-back digital filter infinite impulse response (IIR) can be implemented by mixing a delayed copy of the output of the filter with its input.[26] The sum or difference with the input or output of the filter with the delayed copies produces a new sound with a new waveform and spectrum. A variety of filters with different or more complex magnitude responses can be computed by combining multiple delay lines.

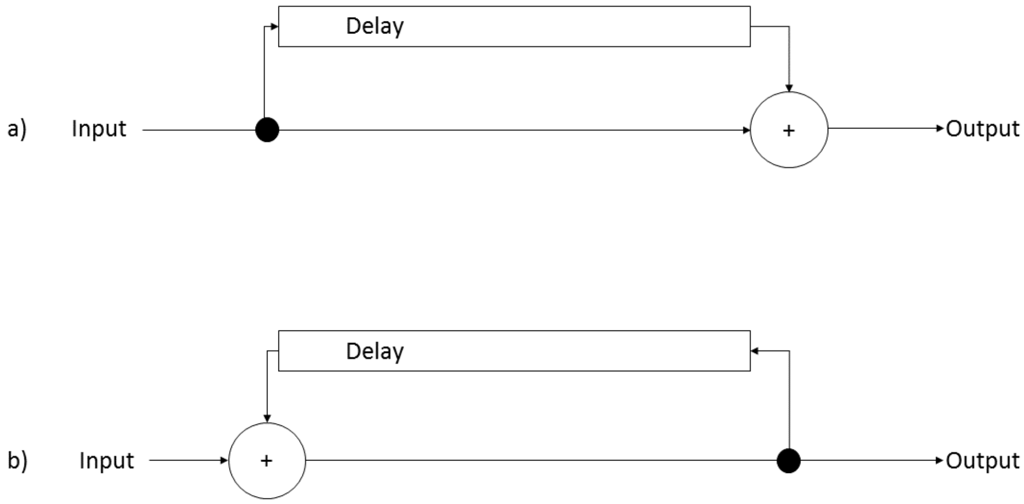


Figure 4. Two types of filter: a) feed-forward delayed input, finite impulse response (FIR) and b) feed-back delayed input, infinite impulse response (IIR).

Feed-back filters produce resonances or peaks while feed-forward filters produce attenuations (anti-resonances or valleys) in the frequency response of the filter. Poles and zeros can be used to describe bandpass or band-reject (notch) filters.[27] Interestingly, any bifurcation within the vocal tract caused by the nasal tract or the tongue can be compared to a feed-forward process, also anti-formants or zeros are

[26] Roads, Curtis. *The computer music tutorial* (Cambridge, Mass.; London: MIT Press, 1996), 185.
[27] Roads, *The computer music tutorial*, 201-202.

produced as result of such bifurcation. The mechanism shown in Figure 3 is the acoustic equivalent of a feed-forward filter (see Figure 4a).

The synthesis of speech sounds involves the use of pulse trains and white noise to mimic the properties of the voice source. A pulse train is a periodic signal, consisting of the repetition of a pulse at a frequency F_0 and has a broad range of harmonic partials. White noise is a signal that has a flat spectrum, with equal energy when measured over a long period of time.

A formant synthesizer uses banks of filters to model the effect of peaks in the magnitude response of the vocal tract. Klatt provides a model of a synthesizer consisting of five resonators and explains how filters can be used in series or in parallel to approximate the effect of the vocal tract to synthesize speech-like sounds.[28] The parameters of the synthesizer can be controlled to generate a variety of speech sounds by controlling the source generators, the frequency and bandwidth of filters. This process is equivalent to changing the shape of the vocal tract during the production of speech. Changing the source generators is the equivalent of changing the voice source: white noise for whispering, pulse train for voiced speech.

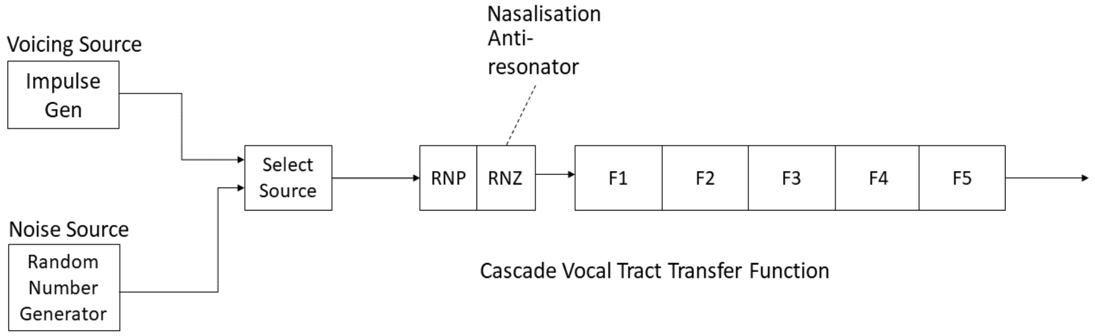


Figure 5.Example of cascade vocal tract model.

The ability to model the vocal tract through different filtering techniques prompts the investigation of filter design to control timbre. This idea has inspired the creation of the novel audio processing tools used in the production of the pieces *Vocal Fry*, *Fricatives* and *Whispers* described in the following chapters.

[28] D.H. Klatt, "Software for a Cascade/Parallel Formant Synthesizer," *J. Acoust. Soc. Amer.,* no. 67 (1980): 975.

As shown in Figure 5 the output of a pulse train generator is fed to the parallel or cascade bank of filters to model voiced sources while white noise can be used to mimic unvoiced sounds and whispering. This design shows how the source-filter model of speech production can be used to design technology which approximates sound production in the voice organ.

Much like the oral cavity and the nasal tract generates formants and anti-formants, digital filters can be used to introduce peaks and valleys in the magnitude response of the resonator modelled. In the Figure 5 the effect of nasalisation is approximated with an additional resonator RNP and an anti-resonator RNZ.

Fant describes the features of the OVE II synthesizer which employs three banks of cascaded filters fed by a pulse train and a noise generator.[29] Interestingly the OVE II models the effect of anti-resonances for the synthesis of nasals and non-vowel sounds and expands the capabilities of a cascade formant synthesizer. One bank is comprised of four poles and one zero for the synthesis of nasal sounds, while the other bank consists in two poles and one zero to model unvoiced (fricatives) sounds and short consonant sounds (stops).

Formant synthesizers can be used to explore how formants are perceived and how, by using specific parameters and different input signals, they can display speech-like features. It is possible to analyse recorded speech in order to track how formants change over time. The resulting data sets represent the formant trajectories and allow one to control the filter coefficients of a speech synthesizer and to approximate the input spectrum.[30] Linear predicting coding (or LPC) is a well-known technique for the analysis and re-synthesis of speech signals and is employed in telecommunications to encode speech at a lower bit-rate. Linear prediction makes an estimation of the output signal by comparing the sums and differences of the values of the previous sequence of samples in input and, through the analysis of

[29] Gunnar Fant, *Speech Acoustics and Phonetics* (Dordrecht; London: Kluwer Academic, 2004), 68.

[30] X. Rodet and P. Depalle, "Synthesis by rule: LPC diphones and calculation of formant trajectories," *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tampa, FL, USA (1985): 736-739.

several snapshots of the signal, it is possible to estimate the spectrum of the input signal.

Filter estimation in speech coding can be divided into two stages:

- Encoding: derives the filter coefficients from a smaller segment of a speech signal called a frame. The result of this process is then transmitted to a decoder.
- Decoding: constructs the filter that approximates the original speech frame to perform operations on the coefficients received.

LPC can estimate the approximation of the spectrum analysed by performing calculations on the samples of the speech frame to determine how the resulting linear prediction coefficients can be used to generate a signal with a similar spectral content.[31] By repeating this process on multiple segments (frames) of the same speech signal it is possible to approximate a time-varying filter.

Linear prediction refers to the source-filter model and this is able to model the effect of the resonant characteristics of the cavities of the vocal tract by using an excitation signal (or voice source) and an autoregressive filter varying over time (vocal tract).[32] The main feature of LPC analysis deals with the detection of peaks in the magnitude response of the signal analysed; this is ideal to find where the formant frequencies of the vocal tract are, and how they change over time, extracting the position of peaks from recorded speech. In an LPC model the vocal tract is represented by an all-pole filter. The all-pole filter can model the effect of formants on the voice source by enhancing the bands of frequency that match the formants analysed with LPC in the resulting spectrum.[33] The main difference between an all-pole filter and an all-zero filter is that an all-pole filter is feedback whereas an all-zero filter is feed-forward. An all-pole filter is a filter consisting of a series of peaks in its frequency response, conversely an all-zero filter displays a series of valleys or null points. Interestingly,

[31] Wai C. Chu, *Speech coding algorithms: Foundation and evolution of standardized coders* (Hoboken, N.J.: Wiley, 2003), 93.

[32] Y. Stylianou, "Voice Transformation" in *Springer Handbook of speech processing.* Ed. Benesty, Sohndi, Huang (Berlin: Springer.2007): 490.

[33] Thomas P. Barnwell III and Kambiz Nayebi and Craig H. Richardson, *Speech coding: A computer laboratory textbook* (New York; Chichester: Wiley, 1996), 90.

the algorithm of linear prediction approximates an inverted version of the spectrum in input. This is due to the way the filter coefficients are computed from the values of the current and past samples analysed. The result of this process is an all-zero filter that displays a series of notches and is usually converted into an all-pole filter by inverting the sign of the filter coefficients.[34] This feature of linear prediction can be used to extract the sound of voice source in a process called inverse filtering; the all-zero filter estimated is applied to recorded speech in order to cancel the effect of the poles removing the effect of the vocal tract from the output signal.[35] A time-varying all-pole filter can be approximated by dividing the speech signal into frames and by performing LPC analysis on each frame. The result of the analysis of each frame is then concatenated in succession to model the variations in time as well as the effect of the movements of the articulators. This process can be implemented through a method which consists of using moving and overlapping windows to divide the speech signal into frames. Each window includes a speech frame to be concatenated.[36] This technique is used in speech coding for the re-synthesis of speech by filtering a pulse train or white noise and, much like in the synthesizer developed by Klatt, voiced and unvoiced sources can be fed to the time-varying filter computed with LPC. In this arrangement speech controls the features of the synthetic sound.[37]

Another way to model the effect of the vocal tract on the voice source is through the use of digital waveguides. They reproduce how the acoustic sound wave propagates to create a physical model of the sound. Waveguide filters employ delay lines with feedback, similar to all-pole infinite impulse response filters (see Figure 4). And, similar to feedback (IIR) filters, when the coefficients of the feedback of the delay line is greater than 1 this results in an unstable model.

Fant describes a model of a vocal tract consisting of two tube resonators that produce different formant patterns in the magnitude response of the apparatus by

[34] Roads, *The computer music tutorial*, 202-203

[35] Sundberg, *The science of singing voice,* 66

[36] Barnwell III, Nayebi and Richardson, *Speech coding: a computer laboratory textbook,* 91-92-93.

[37] Ibid., 91; Klatt, "*Software for a Cascade/Parallel Formant Synthesizer*", 975.

varying the length and area of the tubes. Digital delay lines can be used to produce the effect of tube resonators to model the vocal tract.[38] Cook presents the vocal tract as a filter, using a chain of scattering junctions to build a network of acoustic tube segments. By varying the parameters of the delay lines, it is possible to model the characteristic impedance given by the difference in area between the tubes, thus modelling the effect that the shape of the vocal tract has on the voice source.[39] Therefore, different vowel sounds can be created using this method. More detail on the use of models of acoustic tubes to build physical models of the vocal tract for the synthesis of vowel-like sounds will be provided in chapter 5.

2.3 Music and speech coding

The idea of representing the human voice as a filter and controlling formants to shape the spectral envelope of sounds captured the interest of composers. It allows the exploration of the sonic possibilities that speech technologies can offer to musical applications.

The position of formants provides control over the frequency content of sounds in music and this is particularly relevant in computer music and in audio processing in the recording studio. The partials of the voice source and the position of formants are controlled independently but interact in order to produce the output sound source. The source-filter model of speech production can be used to identify what actions performers can take to transform sounds with their voice organ. Wishart describes a similar division into oscillators (glottis, windpipe, tongue, ventricular folds) to produce pitched sounds through vibration; control over cavities (oral and nasal) produces resonances that can be changed by moving the tongue or rounding the lips.[40] The performer can shape the resulting sound by affecting resonances with their articulation. Or they can generate percussion-like sounds by exaggerating the production of consonants. This prompts the exploration of control over the voice organ to produce sonic material that can be used in sound synthesis through audio

[38] Fant, *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations,* 66.
[39] Cook, Perry R. *Real sound synthesis for interactive applications*. (Natick, MA: AK Peters, 2002), 230.
[40] Trevor Wishart, *On sonic art* (Abingdon: Routledge Taylor & Francis group,1996), 264-268.

analysis of the voice. Furthermore, it can also be manipulated by computer models of the vocal tract to extend the physiology of the performer using technology.

By switching between voiced and unvoiced sources a performer can sing, speak or whisper, thus controlling the source over time. This separation between the control over the source and control over the filter allows the human voice to produce a wide range of sounds. Wishart provides a classification of sounds produced by the human voice and how control over the vocal tract affects the production of sonic objects.[41] A variety of sound sources are generated by oscillations produced by vocal folds and the articulators (like the tongue): Pulses, clicks, noises along with vowels. The resulting classification provides an intuitive description of the mechanics of the vocal tract during the production of certain sounds. The performance aspects of the sonic objects are explained according to the source-filter model. This is relevant to the development of voice-specific audio processing tools and to the recording and performance techniques of speech sounds.

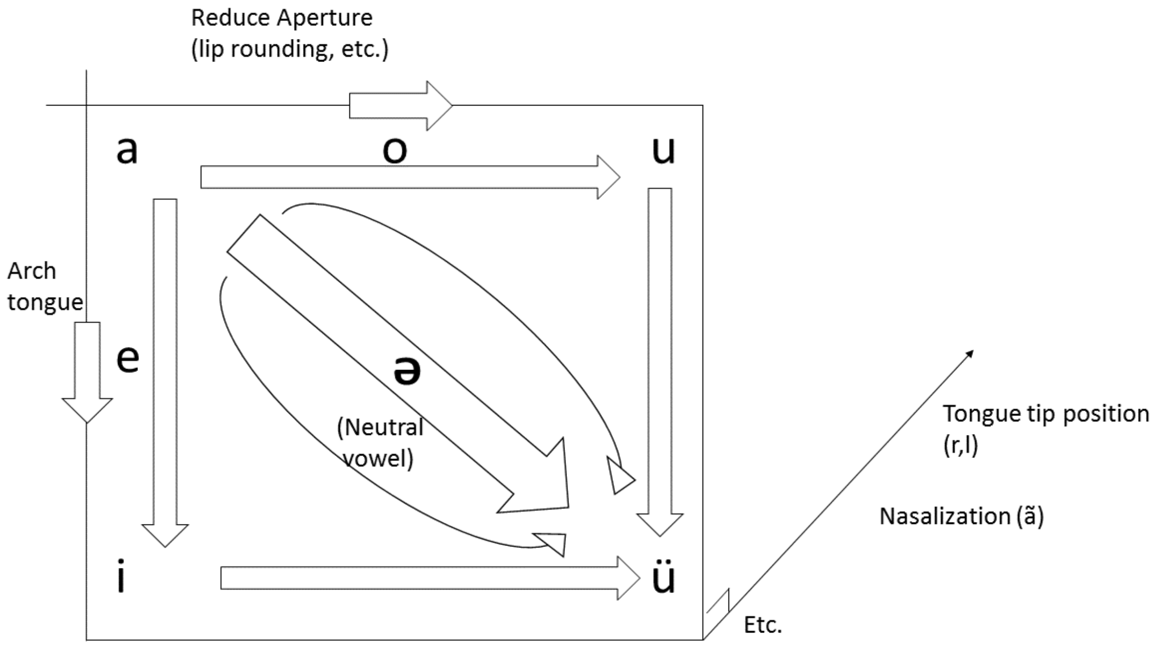


Figure 6. Map of the 'formant space' of vowels. Adapted from Wishart.[42]

[41] Wishart, *On sonic art,* 263-266.
[42] Ibid*.,* 266.

A map of the transformations over formants with reference to the action the performer can perform with the articulator on vowels is provided by Wishart and is shown in Figure 6.

Vowel sounds can be used as a starting point to perform transformations over the filter by moving the articulators and, therefore, slowly moving the position of formants. Each vowel has a signature colour that can be explored in music. Slawson presents a theory of 'Sound Color' in which changes in the first two formants of vowels define changes in sound colour and explains how this can be used to create transformations using filters. According to Slawson, in a source-filter model the sound colour is associated with the filter rather than the source; keeping the formant frequencies and bandwidth of the filter constant enables the sound colour to be kept constant.[43]

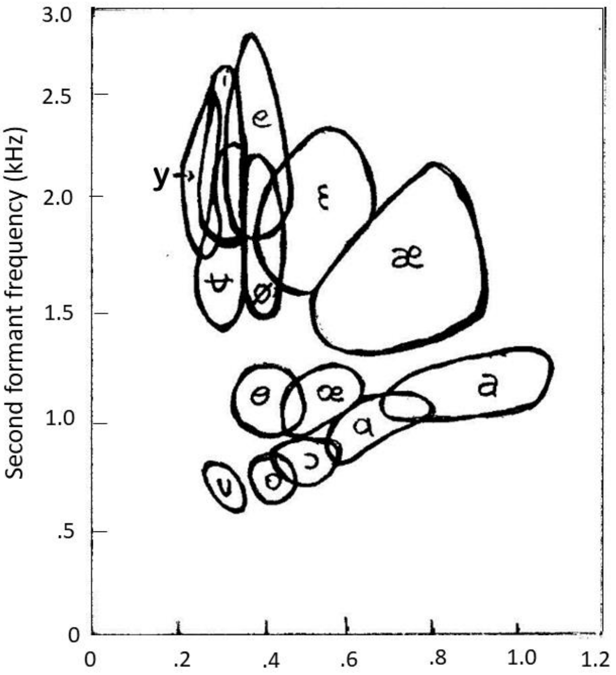


Figure 7. Vowel regions of Swedish vowels mapped according to the first and second formant frequencies. After Sundberg.[44]

[43] Wayne Slawson, *Sound Color* (Berkeley: University of California Press, 1985), 23-43.
[44] Sundberg, *The science of the singing voice,* 24.

Figure 7 shows how the positions of the first two formants in particular allow the representation of a vowel in an x and y two-dimensional chart to approximate an area (or island) in which the slight variation between formant frequencies between individuals still allows the user to define the identity of that vowel in Swedish.[45] The islands represent how the position of the first and second formant define a vowel colour. The data about the position of first formants can be extracted from a collection of recorded sounds of the same vowel pronounced from different speakers to approximate the area of the island of that particular vowel. The variation in morphology (physical shape of the vocal tract) and pronunciation between a variety of speakers means that the same vowel can be represented with an area that covers different values for the first and second formants.

Figure 8 shows the dimensions of sound colour defined by Slawson that are named openness, acuteness, laxness and smallness. Each dimension of sound colour is characterised by the position of the first and second formant of the filter, by increasing or decreasing the position of formants it is possible to change the dimension of sound colour.[46] Charts of the first two formant frequencies of different vowel sounds can provide guidance to control formant synthesizers to generate different vowels in order to transfer the desired sound colour to another source.[47] This suggests using this idea of changing the position of formants over time, to control the dimensions of sound colour as a starting point, to investigate how formant structures of vowels can be used to control digital filters. Furthermore, the idea of sound colour can be explored by including the effect of anti-formants, experimenting with transitions from vowels to nasal consonants.

As discussed in section 2.1, the perception of anti-resonances is more noticeable when there are more than one zero that are moving their position over time. This feature can be combined with the idea of interpolation of sound colour extracted from recorded speech. Extracting both the frequency and bandwidths of more than two resonances and including more than one anti-resonance provides creative

[45] Ibid., 24.
[46] Slawson, *Sound Color,* 57.
[47] Ibid.,55.

potential to investigate the interaction between groups of poles and zeros moving over time and resulting in gradual sound colour changes. Furthermore, it would allow the representation of the timbre of nasal and lateral consonants with more accuracy and to explore how they can be morphed into vowels. For instance, consonants /m/, /n/, /l/ could be slowly morphed into vowels like /a/, /e/, /o/.

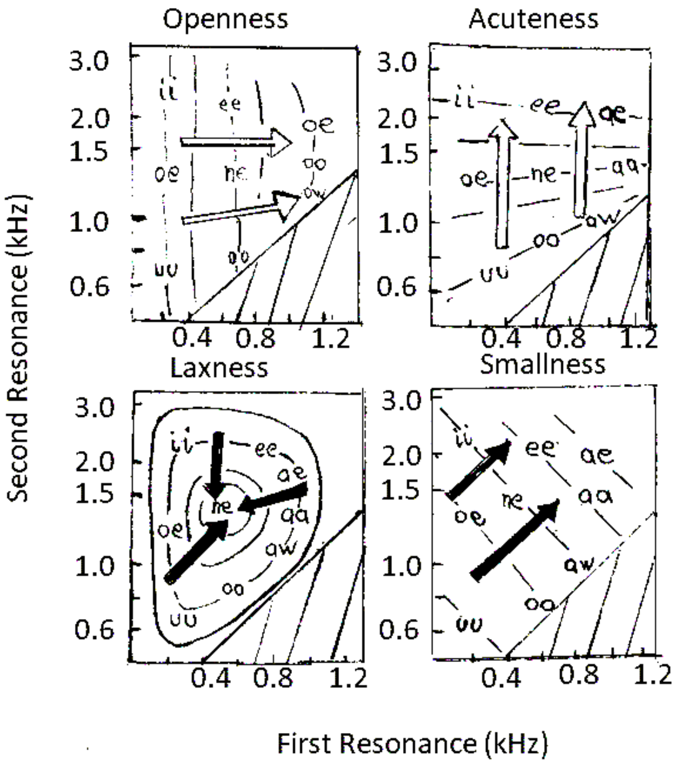


Figure 8. Dimensions of sound colour relating to the formants of vowels. After Slawson[48].

According to Sundberg, the positions of formant frequencies are affected by the speaker's unique morphology and contributes to the voice colour[49]. This suggests a need to explore how a more accurate approximation of the spectral envelope can affect the perception of voice colour transferred to other sound sources.

Wishart describes how consonants like /m/, /n/ and /l/ are used in speech during transitions and can be used to perform transitions from one formant state to

[48] Slawson, *Sound Color*, 55.

[49] Sundberg, *The science of the singing voice,* 24.

another[50]. This suggests that consonants are meaningful in creating transformation of the harmonic content of the voice source over time. Techniques to derive more accurate approximation of anti-resonant features from recorded speech can be explored, to design interpolations between different formant states in order to investigate new ways to use the human voice to control timbre in music.

It is possible to combine the repertoire of the voice with other sounds by using speech synthesis techniques to mix sources and filters. For example, Klatt provides data about formant frequencies and bandwidths to use as input to a formant synthesizer to produce a collection of vowel sounds.[51]

The combination of a variety of sound sources and filters allows us to explore the extraction of the formant structure of recorded speech to superimpose the colour of vowels, for example, onto another sound. Some cross-synthesis approaches explore this concept by extracting the time-varying contour of formants from speech and using them to filter the frequency content of another sound (recorded or synthesized). This allows one to design new sounds that possess the features of two existing sounds. Varying the spectral envelope of sounds over time can be used to perform transformations in sound colour. LPC can be used to track the contour of formants from speech and then use these to filter another sound; it is possible to modify the pitch of the source independently from the filter approximated by LPC.

Formant synthesizers and some LPC vocoders make use of pulse trains and white noise to mimic the sound of voice source. However, speech sounds can be explored by generating sounds with a wide range of frequencies that are ideal to be recorded and employed with subtractive synthesis techniques. For instance, stops (or stop consonants) can be used to produce pulse sounds by creating impulses with the glottis or the tongue, and may or not involve the action of lungs to produce a variety of short sounds.[52] The recorded sounds can be used to mimic different types of voice sources and fed to filters that model the effect of the vocal tract. This process

[50] Wishart, *On sonic art*, 278.
[51] Klatt, "*Software for a Cascade/Parallel Formant Synthesizer,"* 986.
[52] Wishart, *On sonic art,* 276.

suggests the possibility to explore the perception of unvoiced speech sounds by superimposing the timbre of vowel sounds to their frequency content.

Charles Dodge and Paul Lansky used LPC to explore the use of voice filters in musical applications. Lansky describes an approach to sound manipulation close to photography as the machine becomes a tool (like a camera) used to capture and manipulate information taken from the real world: in *Six Fantasies on a Poem by Thomas Campion* (1979) taking information from speech sounds and, in *Folk Images* (1981), from the sounds of string instruments.[53] *Six Fantasies* will be discussed and analysed in a later section. Any recorded sound can be used in an LPC synthesis system to provide the musical texture features of speech, as long as the chosen source has enough frequency content to excite the resonances estimated by the LPC analysis.[54]

The voice offers a great amount of control over the amplitude of partials of the source (frequency content) and the position of the formants (spectral envelope) compared to many musical instruments that do not possess the same level of independence; technology allows one to apply the idea of formant control for timbre manipulation of recorded instruments, introducing variation over time.[55] Alternatively, the use of speech technology provides a way to extract the fixed formant pattern of a collection of musical instruments and introduce movement by performing interpolations over the filter coefficients extracted with audio analysis.

Moore describes a theory of opposites describing how working in the opposite direction from a sound's main feature (descriptor) might create meaningful transformation of sound: different processes (like filtering and cross-synthesis) affect particular descriptors, providing control over the articulation and the frequency content of sounds.[56] A sustained sound might acquire the rhythmic features and

[53] C. Roads and Paul Lansky, "Interview with Paul Lansky", *Computer Music Journal, vol*.7, no.3 (1983): 18-19.

[54] Charles Dodge and Thomas A. Jerse, *Computer music: synthesis, composition, and performance,* 2nd ed. (Boston; Schirmer. 1997), 234.

[55] Trevor Wishart, *Audible Design* (Orpheus and Phantomime, 1994), 36.

[56] Adrian Moore, *Sonic Art: An introduction to Electroacoustic Music Composition* (New York: Routledge, 2016), 96-99.

resonances of speech through cross-synthesis with recorded voice, or rhythmic percussive sources can become a drone through convolution with a sustained sound. Alternatively, a white noise source can become sinusoidal by passing it through a filter with a very narrow bandwidth.

Writing in Mathews and Pierce, Lansky explains how, being an all-pole synthesis, LPC fails to capture nasal sounds and suggests a pole-zero arrangement for improved results.[57] This suggests the need to explore the possibilities of a system that can provide more control over the filter extracted from speech and other sound sources to represent the spectral envelopes of sounds including zeros in the filter. This means taking into account the anti-resonances in the magnitude response of the speech signal analysed.

2.4 Composers and Speech: different approaches to control timbre in sound synthesis

It is fascinating to explore what the features of human voice, the sounds employed in spoken languages and speech synthesis, mean to composers. This section begins by describing composers' approaches to music through the sound of the human voice, and what technologies for speech synthesis have brought to their experience of sound synthesis and composing music. The analysis continues by investigating what features and possibilities are available for composers to control timbre, focusing especially on the work of Paul Lansky. Finally, it explores some of the ideas presented in relation to the techniques that will be demonstrated in the following chapters of this thesis. Whether used as a reference to arrange and design percussion sounds or to manipulate the timbre of different sound sources, speech informs, in a fascinating way, different methods to organise the musical material in electronic music.

There exist examples of musical applications where speech sounds have been used as a means to create families of timbre during the compositional process. Stockhausen explains how in *Kontakte*, electronic sounds are grouped according to

[57] Paul Lansky, "Compositional Applications of Linear Predictive Coding" in *Current directions in computer music research*. ed. Max V. Mathews, John R. Pierce. (Cambridge, Massachusetts: MIT Press, 1989): 2-8.

the material of percussion and other musical instruments (such as wood, metal and skin) and refers to consonants to organise categories of sounds such as noise (consonants), noise-pitched sounds (half consonants) and pitch determined sounds.[58] Interestingly, Stockhausen's references to consonant sounds (/s/, /f/, /p/, /t/, /g/, /k/) display similarities to the types of voice sources outlined in section 2.1 (voiced, unvoiced, mixed) that in this case are used to manage the sound sources involved in the musical material. A similar approach to investigating the features of speech, in particular vowel sounds, for timbral organization is described by David Evan Jones in *Still Life Dancing,* a piece for percussion players and computer tape. Jones uses synthetic percussion sounds to present the resonant structures of the vowels. The approach involves using the CHANT synthesizer to synthesize families of impulsive vowel-like sounds with features similar to and recognisable as percussion sounds (struck wood, struck metal, bowed metal). The resulting sonic objects display the resonant characteristics of nine vowel sounds to blend speech and non-speech sounds.[59] CHANT is a program for sound synthesis and processing. In order to model the singing voice, CHANT generates the resonances of the vocal tract by synthesizing the waveform of the formants in parallel for every period of the voice source. The fundamental frequency is determined by how many times per second the formant waveforms are repeated.[60] Struck percussion-like sounds can be generated from recorded consonant sounds that acquire vowel-like features when processed with a physical model of the vocal tract. The gradual modification of the parameters to extreme settings can produce sounds with non-speech qualities that resemble more closely struck cymbals. This approach is explored in *Fricatives* in Chapter 5. It is an example of technology used to go beyond the body of the performer in order to achieve several transformations over timbre, for example, vowels derived from consonant sounds. Also, *Fricatives* is an exploration of sound synthesis featuring

[58] David Felder and Karlheinz Stockhausen, "An Interview with Karlheinz Stockhausen," *Perspectives of New Music* 16, no. 1 (1977): 90.

[59] David Evan Jones, "Speech Extrapolated," *Perspectives of New Music* 28, no. 1 (1990): 113-114.

[60] Gerald Bennet and Xavier Rodet, "Synthesis of the singing Voice," in *Current Directions in Computer Music Research,* ed. Max V. Mathews and John R. Pierce (Cambridge, Massachusetts: MIT Press, 1989): 19-44.

sonic objects that are not perceivable as a human voice but, in fact, are drawn from speech sounds (specifically consonant sounds).

In *Six Fantasies on a Poem by Thomas Campion,* Paul Lansky explores the perception of contours, shapes and rhythm of spoken words as musical objects by using technology to select and isolate the features of speech (for example, the pitch contour).[61] Analysis and re-synthesis through LPC provide the ability to gather information from recorded speech regarding the resonant structures of the sound. At the same time, it allows the independent control of both the filter and the sound source processed through the resulting spectral envelopes, altering musical reality. For instance, using the analysis of speech to control the spectral envelope of a saxophone improvising a jazz solo might create the impression of the instrument singing as the phrasing is affected by the articulation, rhythm and formants of speech. This approach to gathering information from the voice in order to control sound synthesis can be compared to photography. Lansky describes an approach to using the computer to emphasize and bring to the front the sound of words in music. He also explains how employing technology in this way developed as a concept consisting of taking 'photographs' of sound to transform and manipulate the musical reality.[62] The computer becomes a means to capture information from the real world. Such information is then processed as coefficients, frames and a pitch contour of a voice source, providing an expressive tool to combine or alter the nature of the sounds analysed. Chapter 3 provides an overview of how LPC works. It outlines how LPC assumes a model of production consisting of a source (excitation) and a filter and how the algorithm works to find the coefficients that best represent the sound analysed. Composers might build their own sound world observed (heard) through the loudspeakers in which somebody is making noise, somebody is putting effort into production of sound, even if the sound sources are electronically generated.[63] This idea suggests exploring how technology confers the qualities of the performance

[61] C. Roads and Paul Lansky, "Interview with Paul Lansky," *Computer Music Journal* 7, no. 3 (1983): 18-19.

[62] Ibid., 18-19.

[63] Joshua Cody and Paul Lansky, "An Interview with Paul Lansky," *Computer Music Journal* 20, no. 1 (1996): 22.

analysed with LPC to synthetic (or recorded) sources transferring the actions and physiology of the performer's vocal tract to sound synthesis.

The use of speech analysis and re-synthesis draws the attention of composers as it provides a way to control speech-like sounds that are interesting to human attention. It makes it possible to separate the source from the filter, enabling cross-synthesis. Moreover, it gives access to a collection of techniques that allow for the manipulation of the result of the analysis in order to control the evolution in time of sound.[64]

Slawson defines the importance of the spectral envelope as the driving force of the sound colour of vowels. Slawson's theory of sound colour focuses on the first two formants of vowel sounds and doesn't feature anti-resonances.[65] As such, the first two formants are a crucial feature to extract, or model, in order to transfer their timbre to other sound sources. This constitutes the basis for the theory of sound colour that provides composers with a framework with which to work with vowel-like sounds and organise them meaningfully.[66] Computer programs exist for musical speech synthesis. These early synthesis tools acknowledge the absence of anti-resonances from the spectral envelopes generated, as they provide control over resonances only[67]. This type of computer program offers the flexibility of a speech synthesis model and allows composers to control its resonance in order to design vowel-like sounds, a variety of formant transitions and even consonants.[68]

Speech Songs by Charles Dodge is one of the first computer music compositions to incorporate analysis of recorded speech into sound synthesis. Synthesis-by-analysis techniques offer new opportunities to manipulate the human voice. For example, one can shape the pitch contour of the fundamental frequency independently from the articulation of speech. The result has paved the way for a new type of music based on speech, one that seems to talk. Dodge employs synthesized speech to explore the ambivalence of the words of the poem while the synthetic character of

[64] Perry R. Cook, "Singing Voice Synthesis: History, Current Work, and Future Directions," *Computer Music Journal* 20, no. 3 (1996): 40.

[65] Wayne Slawson, *Sound Color* (Berkeley: University of California Press, c1985): 40-43.

[66] A. Wayne Slawson, "The musical control of Sound Color," *Canadian University Music Review/Revue de musique des universities canadiennes*, no. 3 (1982): 67-71.

[67] Wayne Slawson, "A Speech-Oriented Synthesizer of Computer Music," *Journal of Music Theory 13*, no. 1 (1969): 108.

[68] Ibid., 109.

the voice serves to express the humour of the poem. A formant tracking system is used for the first three songs of *Speech Songs.* The fourth song instead is created using linear predictive coding.[69] The computer programs enable the composer to extract the parameters of the synthesis directly from speech. Then it is possible to manipulate the parameters within each frame of the synthesis and to repeatedly hear the result. *Speech Songs* is an example of a new way of making music through a new technology that allows one to edit the data derived from recorded sound before synthesizing it.[70] Consequently, the voice becomes part of the sound synthesis, in this case defining the all-pole filter of LPC or the centre frequency of band-pass filters.

Evidence exists of the importance of timbre for the perception of musical tension. Stephen McAdams describes how a collection of timbres grouped according to their features (e.g. spectral centroid) can be used to predict the musical effect and perception of tension by the listener. This prompts exploration of timbre transformations over time to create musical interest.[71]

LPC analysis offers the possibility to manipulate the perception and features of speech. For instance, by editing the pitch of the voice source in the frames of the analysis, one can morph speech into a singing voice or create polyphony by doubling and manipulating several voice sources controlled by spectral envelopes of the same vowel. Also, it is possible to time-stretch the parameters of a single analysis frame in order to design a longer speech-like sound. A similar approach is used in *Six Fantasies* where Lansky uses LPC to manipulate and exaggerate the musical features of speech and to control its duration, pitch contours and number of performers to make it sound like singing.[72]

The rhythm as well as pitch can be manipulated with LPC to produce the effect of a crowded room from speech sounds by creating several parts of rhythmic patterns and exploring different combinations of positioning each part within the stereo

[69] Charles Dodge,” On *Speech Songs,*” in *Current Directions in Computer Music Research,* ed. Max V. Mathews and John R. Pierce (Cambridge, Massachusetts: MIT Press, 1989): 9-17.

[70] Ibid., 9-17.

[71] Stephen McAdams, “Perspectives on the Contribuition of Timbre to Musical Structure,” *Computer Music Journal* 23, no. 3 (1999): 96-99.

[72] Paul Lansky and Jeffrey Perry, "The Inner Voices of Simple Things: A Conversation with Paul Lansky," *Perspectives of New Music* 34, no. 2 (1996): 48.

image. For instance, in the piece *Notjustmoreidlechatter*, Lansky seems to build a foreground where wide and fast changes in the pitch of the voice source, in combination with rhythmic patterns, generate arpeggios of chattering voices with different panning within the staging of the stereo image.[73] The piece is based on the LPC and granular synthesis of recorded speech sounds. The frames of the analysis from LPC are repeated several times. The pitch of the repetitions varies at regular intervals of time. The composition displays several layers of these repetitions of vowel and speech segments filtering a voice source. The latter one is then transposed over several pitches and octaves to create the impression of a crowded place. Very low transpositions of the voice source sound almost like a bass line (e.g. the segment at 58’’). In the background, granular synthesis is used to produce sustained sounds that resemble a choir singing in counterpoint. Interestingly, the pitch of the chattering arpeggios matches the background sustained sounds. The rhythmic pulse of vowel and diphthong sounds have been achieved through stochastic mixing techniques to operate when a part (or voice) is chattering. The sustained tones, in turn, are generated through granular synthesis that employs recorded vowel sounds.[74] The overall effect applied to the different voices uses LPC to confer on speech a machine-like rhythmic pattern with wide pitch transpositions. The voice source sounds like an arpeggiator-sequencer of a synthesizer that creates a pattern extended to multiple octaves. It is then filtered by formants derived with LPC that blend human voice performance and machine (non-human) performance. The density of the chattering recreates the effect of a crowded place where words are not intelligible, perceived as many people having a conversation. Only occasionally certain speech sounds become recognisable, in particular when focusing the attention on a specific pattern or when the density is reduced (e.g. in the central section of the piece, around 2’50’’). For instance, in this section the chattering concentrates on a narrower range of intervals. As the section progresses, at 3’37’’ the pitch of the chattering descends to a lower register. Also, the speech sounds are repeated at different pitches and panned differently left and right in the stereo

[73] Paul Lansky, "Notjustmoreidlechatter," *Electro Acoustic Music 1.* Neuma Records, 1990. CD.

[74] Dodge, and Jerse, *Computer Music: synthesis, composition and performance,* 273-274.

image. Here, technology is used to estimate the frequency and bandwidth of audio filters from the real world. It also allows the composer to manipulate the information to perform extreme changes in pitch and create dense rhythmic textures. Lansky describes LPC as a bank of formant filters that display fast variations, and how in the *Idle Chatter* series of pieces this technique has been used to isolate the words and flatten and transpose their pitch contours.[75]

Six Fantasies includes creative uses of LPC to capture the sound of vowels and control computer sound synthesis with a human voice. The six fantasies are all based on the recording of a single reading of a poem by Thomas Campion. Each fantasy explores different features of speech production and provides variations of the same performance.[76] This section investigates some examples of the possibilities and artistic effects explored in *Six Fantasies* for the transformation of speech through technology. What follows is an analysis of Lansky's treatments of sonic material, focusing on those particular features that are relevant to the techniques discussed in later chapters. Particularly, there is an explanation of how the independence between the voice source and the resonant structures of the vocal tract can enable the synthesis of hybrid sounds to be generated from the human voice; furthermore, how it can provide a means to combine human performance with machine performance (audio processing and sound synthesis). Lansky states that in this poem Campion plays with the sound of speech as if it is an instrument, freely manipulating a collection of vowels and balancing repetitions.[77]

The way Lansky performs treatments on the sonic material captures information from recorded human performance. Instead of being the vocal tract of the performer that shapes the vowel sound, it is Lansky who artificially manipulates vowels through technology and uses them to control sound synthesis. The access to audio processing technology allows exploration beyond the sound of the words of the poem. It has been discussed previously in this chapter how bifurcations within the vocal tract generate anti-resonances within its magnitude response during the production of nasals and lateral sounds. The poem chosen for the realization of the fantasies is *Rose*

[75] Lansky and Perry, "The Inner Voices of Simple Things: A Conversation with Paul Lansky," 50.

[76] Paul Lansky, Fantasies and Tableaux, NWCR 683, 2007, CD sleeve, 1.

[77] Ibid., 1.

cheekt Lawra.[78] The following text is the Poem used by Lansky to compose each of the six fantasies; the nasals and lateral consonants sounds are underlined (my undelining).[79]

Rose –cheekt Lawra, come,
Sing thou smoothly with thy beawties
Silent musick, either other
Sweetly gracing.

Lovely forms do flowe
From concent divinely framed;
Heav'n is musick, and thy beawties
Birth is heavenly.

These dull notes we sing
Discord neede for helps to grace them;
*O**nl**y beawty purely loving*
Knows no discord;

*But **still** moves delight,*
Like cleare springs renu'd by flowing,
Ever perfect, ever in them-
Selves enternall

The underlined consonants will be relevant to describe the sounds synthesized in 'her song'.

[78] Thomas Campion, "Observations in the Art of English Poesie," The Work of Thomas Campion, ed. Walter R. Davis (London: Faber and Faber, 1967): 310.

[79] Lansky, Fantasies and Tableaux, 2.

2.4.1 ‘her voice’: changing voice source pitch-contour

In the first fantasy, ‘her voice’*,* it is possible to identify three types of sound sources used throughout the composition:

- Re-synthesized voiced speech where words are intelligible.
- Sustained sound sources providing a background of homophonic textures.
- Percussive low-pitched synthetic sounds. They are short in duration and their timbre and amplitude envelope appear similar to an electric bass played with a pick. Throughout the piece they are used sparsely to highlight certain moments.

The voice in the foreground is re-synthesized with wider variation in pitch than the performance from the last fantasy ‘her self’*.* It shows a different pitch contour of the voice source and, furthermore, it sounds as if two performers are speaking. The two re-syntheses of the same performance with different pitch contours would provide the effect of more than one performer reading. This suggests the use of LPC as a means to create variations of the same performance by manipulating the pitch contour but retaining an intelligible articulation of words.

2.4.2 ‘her presence’: interplay between foreground and background

In ‘her presence’ two types of sounds are interacting. The first one is intelligible re-synthesized voiced speech and the second one comprises sustained notes (and later homophonic textures). The piece displays interesting techniques of duplicating the pitch contours in order to create the impression of an ensemble of performers. For instance, in the first half of the piece, copies of the re-synthesized reading of the poem are playing against each other. These lines of melody are at times creating a call and response echo effect or developing interaction resembling polyphonic voicing by altering the frequency of the voice sources in different ways. The resonances of certain vowels are captured and moved to the background in order to provide the spectral envelope of synthesized sustained notes. This process 'freezes' the timbre of the speech in time, becoming the background of the reading of the poem. Lansky describes the use of vowels in this poem as if Campion is playing a musical instrument.[80] In this piece, sound synthesis plays with vowels as if they were an instrument in order to explore the interplay between the foreground and the background sounds. Transferring the filter from one sound source to another produces smooth transitions from foreground to background. For instance, at 0'23'' the /i/ of 'musick' is captured and prolonged. Or at 1'44'' when the /o/ of 'discord' is prolonged in a fading sound. The choir-like sounds generate the sustained notes and later homophonic textures. They use different voice sources with the pitch of the desired note filtered with the formant structures of vowels. This technique creates layers of sustained notes developing in homophonic material. For instance, at 2'57'' LPC produces a choir-like synthesized sound. The resulting sonic objects give the impression of an increasing number of voice sources. The latter ones capture vowel resonances from the voice to acquire their timbre. Here the possibility of controlling the source and the filter of the synthesis independently allows the manipulation of the human performance with technology, in order to play with vowels as part of the synthesis instrument.

2.4.3 ‘her reflection’: blurring articulation

The articulation of speech in ‘her reflection’ is blurred through the extension and exaggeration of resonance. Delays and sounds similar to comb filters with a high value of feedback extend the resonance of speech. It appears that Lansky uses a higher feedback value to make the filter ring for a longer time, producing sharper harmonically related peaks in the magnitude response. The result generates longer tails to words which take a longer time to fade out and consequently smooth the articulation of words. At the same time, the frequency content is more focused, featuring sharper equidistant peaks in the magnitude response. Delay generates different rhythmic patterns from articulation of words (2'11'') used as input. LPC is used to resynthesize a wide range of low or high pitch sustained sounds (e.g. 3'28') filtered by more static vowels. Interestingly, the title ‘her reflection’ plays with the word reflection and the use of delays (reflections/echoes) and reverb-like effects as a way to alter the perception of speech.

[80] Lansky, Fantasies and Tableaux, 1.

2.4.4 ‘her song’: LPC homophonic textures and consonants

LPC is employed in ‘her song’ to alter the duration of words for the re-synthesis of voiced speech. Transpositions in pitch and layers of voice sources of speech create homophonic textures. The articulation of the words is clear and stays the same in every layer of the texture. The variation in pitch of the voice source of the different layers matches the words rhythmically resulting in a homophonic texture. Every layer is filtered with the same LPC analysis which determines the layer’s timbre and articulation. In ‘her song’ the re-synthesis of speech slows down the rhythm of the reading. This results in certain sounds being stressed (or prolonged), sometimes nasals and laterals. These sounds contain zeroes (anti-resonances) in their magnitude response. For example, at 0'15'' an /m/ seems to be prolonged and used as a filter. In a similar way at 1'08'' the /m/ in the re-synthesis of 'framed' appear to be time-stretched. Interestingly, at 1’55’’ the /l/ and /n/ consonants are also stressed and stretched in the re-synthesis of the word 'loving'. Another example of nasal and lateral sound stressed is displayed at 2'15'' in 'still' and at 2'28'' in 'springs' with /l/ and /ng/. The presentation of the text of the poem at the beginning of this section includes underlining of all consonants that feature anti-formants. The use of a pole-zero system to synthesize nasals (as they produce anti-formants as well as formants) has already been discussed. *Six Fantasies* has been realized with LPC synthesis which does not approximate anti-resonances as it is an all-pole model.[81] This suggests that a system able to approximate more accurately the anti-resonant features of nasal and lateral sounds would better allow the exploration of the timbral qualities of these consonants in sound synthesis. Therefore, it would also enable us to go beyond the features of LPC and explore the manipulation of zeros as well as poles derived from recorded speech. This would offer new possibilities to creatively use the information and features gathered from the voice.

[81] Lansky, Paul. Fantasies and Tableaux, 3.

2.4.5 'her ritual': unvoiced speech

The ability to control the source and the filter separately allows the use of noise as the voice source of the LPC filter generating unvoiced speech. In 'her ritual' the voice source of the performance is changed to noise in order to provide an unvoiced quality to the re-synthesized speech (e.g. 2'48''), maintaining the intelligibility of words as if they were whispered. Comb-filter like processing with an apparently high feedback value is then applied to the resulting unvoiced speech. This confers a more resonant and metallic character to speech, smoothing and blending the articulation of words.

2.4.6 'her self': original reading

'her self' includes the original reading of the poem as also described by Lansky.[82] The composition includes two main musical materials:

- The original reading of the poem.
- A re-synthesized background, consisting of sustained vowel-like sounds that increase the number of voices. It evolves into strata of vowels and chordal materials. A voiced source is used as excitation of the filter.

The synthesized sounds continuously vary the timbre of vowels used for the analysis and resynthesis. The resulting sound resembles the effect of a flanger or phaser with a slow rate of modulation. The spectral envelope of these evolving sustained sounds is derived from the timbre of speech rather than displaying evenly spaced peaks or notches. Sometimes the timbre of the re-synthesized sounds in the background changes in response to the vowels accentuated in the foreground poem reading.

'her self' includes the original reading with the voice of the performer. This suggests exploration of the audio analysis of consonants to investigate how LPC and Prony's method compare in the analysis of nasal sounds. Figures 9 and 10 show the comparison between the LPC and Prony analysis of the vowel /o/ of the word 'discord'. This vowel is used, for instance, in the re-synthesis of background sustained sounds in 'her presence' *at* 1'44''. Figures 11 and 12 present the analysis of the nasal

[82] Ibid., 1.

sound /m/ of the word 'musick'. The spectral envelope derived through Prony's method is more detailed, yet still intuitive. The audio analysis shows how anti-formants can affect the overall shape of the spectral envelope, especially of nasal sounds. This prompts exploration of the use of Prony's method as a technique to capture the spectral envelope of speech and how the data approximated by the analysis can be creatively manipulated in order to provide control of timbre over time. For example, in *her song* (at 0'15'' and again 1'08'') nasal sounds are stressed and time-stretched through sound synthesis. A better representation of anti-resonances present in nasal and lateral consonants would offer more options for processing techniques in pieces such as 'her song'.

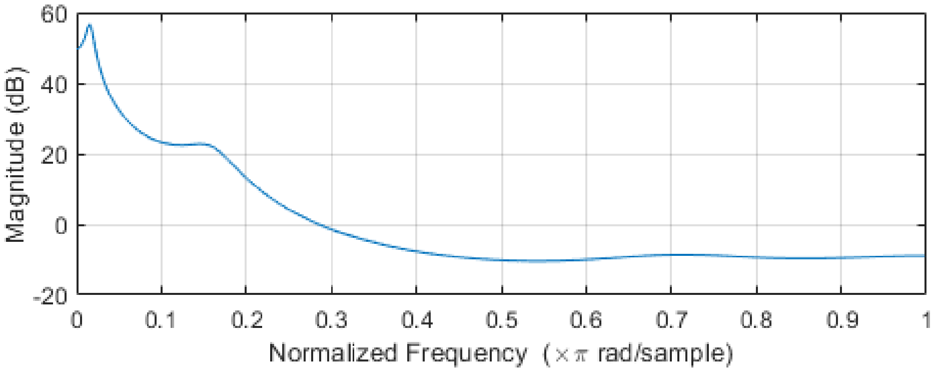


Figure 9. Analysis of the sound /o/ of 'disc<u>o</u>rd'. LPC with 5 poles.

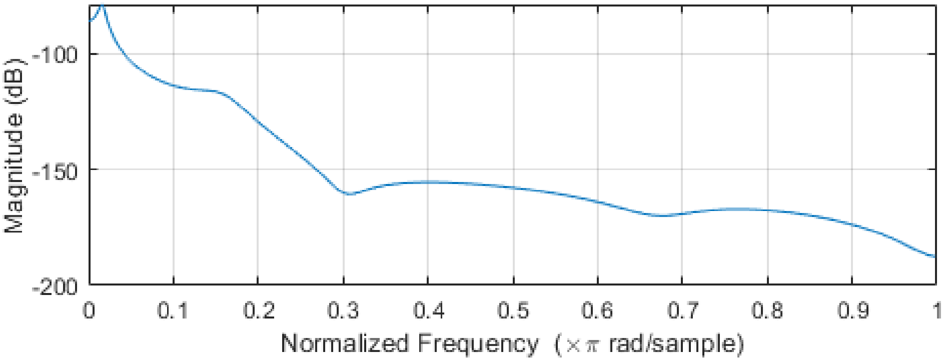


Figure 10. Analysis of the sound /o/ of 'disc<u>o</u>rd'. Prony's method with 5 poles and 3 zeroes.

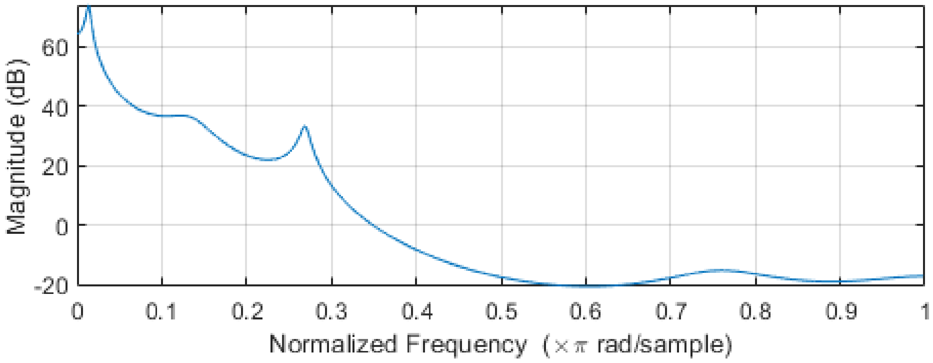


Figure 11. /m/ of musick LPC analysis 5 poles.

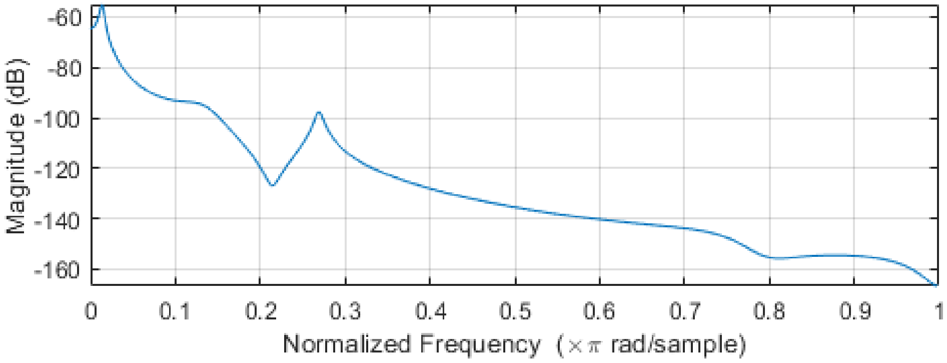


Figure 12. /m/ of 'musick' analysis with Prony's method. 5 poles and 3 zeroes.

Idle chatter and *Six fantasies* exaggerate rhythm, shapes of the pitch contour and envelopes of speech.[83] It is possible to expand on this approach in order to design interpolations through the 'sampling' of different spectral envelopes. Formant extraction techniques have been used in music to achieve different artistic effects, for instance, *Maentwrog, Music for Soleil* and *Alillia* by Richard Cann. In the former, whispers are generated by creating unvoiced sources through scraping an electric guitar; in the latter, the formants derived from speech are changed very slowly to filter recorded piano sounds, acquiring a cello-like quality.[84] This idea of slowly changing formants over time provides a fascinating starting point to explore speech-

[83] Lansky and Perry, "The Inner Voices of Simple Things: A Conversation with Paul Lansky," 48-50.

[84] Richard Cann, "An Analysis / Synthesis Tutorial, Part 2," *Computer Music Journal* 3, no. 4 (1979): 13.

based interpolations over a longer time-frame. This approach appears to focus less on exaggerating the qualities of words (articulation, intelligibility, rhythm) to synthesize talking instruments, and more on transferring the resonances of the vocal tract in order to shape, in a more macro time-scale, the timbre of a sound transformation. Filters can be used in a creative way for the interpolation of sounds. This might be achieved through changes over time of the features of the filter or by the creation of hybrid sounds by means of convolution between two sound files.[85] The vocal tract can be represented as a time-varying filter which can be achieved in sound synthesis on a frame by frame basis (see section 2.2). Interestingly, this concept can be applied to perform interpolation over sound based on the human voice that does not include the rhythm or articulation of speech. Spectral morphing can be obtained by creating a seamless transition over time between the audio analysis of two sounds. Interpolations and concatenation can be used to achieve this effect through cross synthesis, spectral analysis and re-synthesis. Hugill observes that very often the intermediate state (in-between) is the most interesting section of the transformation as it can be isolated and processed again.[86] For example, one could perform interpolation techniques to seamlessly transition from a recorded vowel sound /a/ to /u/ over five seconds. The resulting intermediate state would be a combination of the timbral qualities of both /a/ and /u/ vowels. One could capture this intermediate sound and apply further processing. This is just one example of how sound synthesis can be explored to build a bridge between two recorded sounds from human performance.

As stated before, audio analysis of speech can be used to take ‘photographs’ of the resonant structures of a recorded performance. The machine would then synthesize the intermediate states to move over time from the starting ‘photograph’ of actual speech to the target state captured. This approach to sound synthesis prompts exploration of techniques that blend the interaction between human and machine performance, one complementing the other. This approach described above has been explored in chapter 6 during the design of PZeroSynth and the creation of

[85] Andrew Hugill, *The Digital Musician* (New York and London: Routledge, 2008): 90-91.

[86] Ibid., 92-93

Whispers. Another method is to hide the articulation of speech and transfer only the resonant structures of vowels. This can be found in the techniques explored in chapter 4 where convolution between synthetic vocal fry sources produced with LPC analysis transfers their time-varying features to synthetic stereo white noise. Consequently, it generates resonant strata of sounds or confers a vowel timbre to reverberation-like sources.

One of the spaces through which morphing trajectories are conducted in this thesis is that described by what is known in signal processing terms as the Z-plane. And it will be seen in Chapter 6 that different trajectories through the Z-plane have different implications for formant bandwidth and level. See Chapter 6 for further details.

The challenge of using speech to control sound transformation through audio analysis and in building tools for audio morphing is to explore what implications mathematical and filter design techniques have on the manipulation of sound. For instance, poles and zeros closer to the unit circle of the Z-plane (see chapter 6 for details) result in a narrower formant and anti-formant bandwidth. Therefore, moving the poles and zeros in a circular movement to perform a transition would result in a sharper and more resonant transformation where the filter sweeps are more audible. The opposite is also true; moving the poles and zeros close to the centre of the circle will result in peaks and notches with a wider bandwidth. If we consider an audio interpolation over a given duration, a linear movement would generally keep the poles and zeros farther from the unit circle. This will produce a less exaggerated resonance for poles. When dealing with nasal and lateral sounds, anti-resonances provide a more accurate representation of their spectral envelope. Valleys (zeros) are more perceptible when their bandwidth increases.[87] However, a wider band for zeros might introduce drops in amplitude when performing transformations between vowel and consonant sound. The option to perform interpolations with either linear or circular movements of poles and zeros allows the better design of interpolations and offers the user more flexibility to adapt the synthesis tool to the recordings one is working with.

[87] E. Floyd Toole and Sean E. Olive, "The Modification of Timbre by Resonances: Perception and Measurement," *J. Audio Eng. Soc*. 36, no. 3 (1988): 122-123.

2.4 Proposal for a pole-zero system to explore voice colour

The timbre of voice and the variety of timbre it can produce can be made available to audio processing and sound synthesis in music. LPC vocoders have been used by composers in order to apply cross-synthesis techniques and thus combine the qualities two different sound sources. For example, Jean-Claude Risset blends natural sounds (e.g. birds) and musical instruments (e.g. cello) in his composition *Sud.*[88] The source-filter model of speech production considers the vocal tract as a filter that shapes the frequency content of a sound source, which is the vibration of vocal folds. In particular, this filter is considered a separate entity from the output of the sound source; a feature that can be extracted from speech and transferred to a different source, a musical instrument for example.

This concept has been employed in speech processing to design digital filters that approximate the effect of the vocal tract for the transmission and synthesis of speech. Audio filters have been used in music to shape the timbre of sound sources by altering their frequency content. LPC allows the user to control a digital filter with speech, this feature captured the interest of composers and it has been explored successfully in music by Paul Lansky and Charles Dodge to transfer the qualities of speech to different sound sources. LPC still plays a crucial role within the algorithms of current audio processors to perform cross-synthesis. The advantage of linear predictive coding is that it can estimate and provide control over an intuitive parametric model of the spectral information of the timbre of voice. It can successfully detect resonances (formants) within the magnitude response of the vocal tract by taking advantage of audio analysis to compute a spectral envelope, for example, from recorded speech sounds. However linear predictive coding is an all-pole model, this means it can detect peaks (poles) within the magnitude response of the vocal tract very efficiently but fails to detect spectral valleys (zeros).

The stated aim of this thesis is to explore and develop new digital tools for the successful transfer of qualities of speech onto other sounds by using knowledge of

[88] Risset, Jean-Claude, "Examples of the Musical Use of Digital Audio Effects." *Journal of New Music Research* 31, no. 2 (2002): 95.

the human voice. Vowel and consonant sounds have a specific set of resonances (poles) and anti-resonances (zeros), their spectral envelope represents an intuitive parametric model of the colour of speech sounds. A pole/zero model is explored in this research in order to provide access to a more accurate estimation of this parametric model for cross-synthesis applications. Cross-synthesis has been used by composers to perform transformations of timbre over time (morphing) as a way of altering the perception of two separate sound sources. Linear predictive coding (LPC) or techniques based on the Discrete Fourier Transform (DFT) can be used in order to perform these transformations. However, the DFT does not provide as intuitive a model as LPC does; on the other hand LPC does not detect the position of anti-resonances. Both processes involve a frame-by-frame reconstruction of the interpolation by cross-fading static filters or spectra. The DFT is a transform with particular properties. It is energy preserving, invertible and it is a way of representing a signal in the frequency domain which has previously been available in the time-domain.[89] LPC, on the other hand, attempts to describe the spectrum as a series of resonances (i.e. it is an 'all-pole' spectrum) with specific parameters.[90] Those parameters are derived from an error signal minimization technique which is based on the autocorrelation function. The autocorrelation function is the square of the Fourier transform.[91] Therefore, the possibilities are either an approach to processing sounds which works directly on the Fourier transform data (e.g. magnitude and phase) or the use of the Fourier transform data to derive a parametric model one of which, for instance, can be LPC.

This thesis presents new tools that implement a pole-zero representation of recorded speech sounds, providing the user with a means to perform spectral interpolation and achieve a more accurate representation of anti-resonances. The thesis includes a proposed method to manipulate the coefficients of analysis that implements the interaction between poles and zeros during spectral interpolations. Furthermore, it explores techniques of digital filter design to offer the user the ability

[89] Jaffe, David A. "Spectrum Analysis Tutorial, Part 1: The Discrete Fourier Transform." *Computer Music Journal* 11, no. 2 (1987): 9.
[90] Moorer, James A. "The Use of Linear Prediction of Speech in Computer Music Applications." In *Journal of the Audio Engineering Society* 27.3 (1979): 134-135.
[91] Makhoul, J. "Linear prediction: A tutorial review," in *Proceedings of the IEEE*, vol. 63, no. 4: 569.

to avoid the cross-fade of static filters by implementing a save-state method combined with a rectangular window for the synthesis of sound. The aim is to design a technique to approximate two spectral 'photographs' that provides the user with an intuitive way to perform meaningful sound transformations based on the sound colour of vowels and consonants. The techniques and audio examples that form the output of this thesis represent the personal journey taken by the author to explore the qualities of speech in order to identify and implement useful parameters in the tools developed.

The previous sections of this chapter discussed how different areas of research employ the source-filter model of speech production for the study of speech acoustics, the development of speech coders and in music as a model to perform creative transformations. This thesis is focussed on exploring the filter part of this model to answer the research questions. This section outlines the approach to the investigation of technologies and techniques able to successfully transfer the timbre of speech to different sound sources by employing knowledge of the human voice. It also provides an overview of the literature explored to design a novel sound design tool with the ability to perform morphing techniques based on the audio analysis of recorded sounds. The proposed model aims to extract and model anti-resonances.

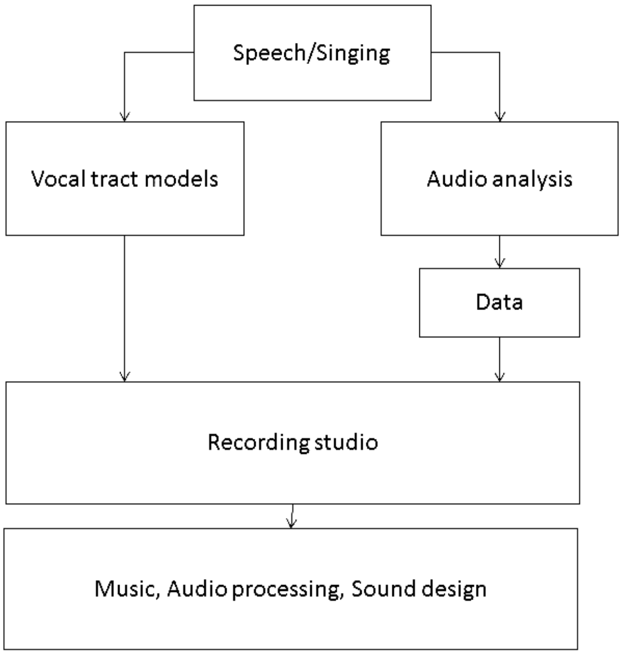


Figure 13. Approach to audio processing and sound synthesis based on the human voice.

The approach to audio processing shown in Figure 13 has been used in the recording studio to investigate how features of the human voice can be explored in this thesis. Two main ways to apply the source-filter model of speech production are considered to guide the development of sound synthesis strategies and audio processing techniques. The new audio processing techniques for voice manipulation included in this thesis will be described in detail in the following chapters and demonstrated through audio examples and original compositions to illustrate the effects that speech coding and synthesis can bring to the creative process. Well known music production techniques have provided inspiration in the design of some of the processes and have affected the features of the audio processing tools developed during the research process.

The human voice is used to provide data to control digital filters or to generate sound sources that can be filtered by models of the vocal tract (e.g. formant synthesizers, physical models). The proposed approach is to identify different ways the filter can be controlled to affect the musical result of these operations, starting with the processing and synthesis of the building blocks of speech (e.g. vowels, consonants). According to Fant, the nasal cavities can add zeros, and sometimes extra poles, to the vocal tract filter function.[92] Any bifurcation or split within the vocal tract can generate the presence of anti-formants (zeros). Lateral consonants like /l/ and variations of /r/ consonants also produce anti-formants caused by the position of the tongue that creates a bifurcation in the air stream within the vocal tract[93]. This suggests a need to represent nasal and lateral consonants in a system that approximate zeros as well as poles for the analysis and re-synthesis of speech.

A recursive digital filter can be used to design the effect of poles and zero combined and to explore how the effect of anti-resonances can be represented. Recursive filters allow separate control over the numerator and denominator in the equation of their transfer function[94]. Poles (or peaks) can be controlled by operating on the denominator while zeros can be controlled by operating on the numerator. Recursive filters have an infinite impulse response (IIR). There is an approach to digital filter design that involves computing the frequency response of a filter by choosing the position of poles and zeros on the z-plane and finding the filter's difference equation.[95] Therefore, one can control the magnitude response of a filter by mathematically manipulating the filter coefficients of the numerator and denominator. Thus, it is worth exploring how an arrangement with both numerator and denominator can be adapted for musical applications as it gives control over zeros as well as poles. For instance, it could be a way to shape the timbre in sound synthesis. Interestingly, Roads reports of the use of Prony's method as a spectrum analysis technique that models the input signal as a series of damped sinusoids that have been employed in the analysis stage of the CHANT synthesizer and for the re-synthesis of percussion sounds.[96] Section 2.4 also provides examples of synthesized

[92] Gunnar Fant, *Speech Acoustics and Phonetics.* (Dordrecht; London: Kluwer Academic, 2004), 151-152.

[93] Kent and Read, A*coustic analysis of speech*, 181.

[94] Lynn, Paul A. and Fuerst. Wolfgang. *Introductory digital signal processing with computer applications. (*Chichester; New York: John Wiley, c1998), 168.

[95] Lynn and Fuerst. *Introductory digital signal processing with computer applications*, 168-169.

[96] Roads, *The computer music tutorial*, 597-598.

vowel-like percussion sounds in music with CHANT. This prompts the exploration of new tools featuring Prony's method as a technique to improve and expand the sonic palette of speech-inspired timbre manipulations. This is a process similar to autoregressive methods such as LPC; the resulting coefficients are used in combination with Fourier Analysis to produce a re-synthesis of the signal. This suggests the need to explore the extension of Prony's method as a method to approximate the spectral envelope of input sounds and to explore how it is different from LPC analysis for musical applications such as those previously outlined.

Parks and Burrus explain how a formulation initially derived by Prony in 1790, to study the elastic properties of gases produces linear equations, can be applied to design IIR filters defined by the equation:[97]

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_M z^{-N}} \tag{1}$$

(1) Filter coefficients, the numerator represents the zeros (notches) and the denominator represent the poles (peaks)

The presence of a numerator and denominator in the transfer function of the filter suggests that Prony's method can model the effect of anti-resonances as well as resonances. The numerator *b* represents a series of coefficients that model the zeros while the denominator *a* models the effect of poles. By factorising the numerator and denominator of the polynomials of the transfer functions it is possible to obtain the pole-zero description of the filter and by specifying the poles and zeros of a recursive filter (like IIR filters) it is possible to estimate and plot its frequency response characteristics.[98]

The factorisation can be represented in the following equation:

[97] T.W. Parks and C.S. Burrus, *Digital Filter Design* (New York: John Wiley and Sons,1987),226.

[98] Lynn and Fuerst, *Introductory digital signal processing with computer applications*, 167-168.

$$H(z) = \frac{K(z-z_1)(z-z_2)(z-z_3)\cdots}{(z-p_1)(z-p_2)(z-p_3)\cdots}$$

(2)

(2) Pole-zero description by factorizing numerator and denominator.

The z transform and the z-plane provide a useful way to describe the properties of the filter and to monitor the stability of the digital filter considered.

Z describes the time shift factor and this is defined by:

$$z^{-1}= e^{-j2\pi f\tau_s}$$

(3)

The expression (3) is used to describe a time shift on a signal of frequency 2πf by the delay time (fixed) $\tau_{s.}$ The value τ_s represents the period of one sample, considering the value z^{-1} represents the delay of one sample, z^{-2} the delay of two samples, z^{-3} a delay (time shift) of three samples and so on.

When considering the following transfer function of a recursive filter (IIR):

$$H(z) = \frac{b_0+b_1z^{-1}+\cdots+b_Mz^{-M}}{1+a_0+az^{-1}+\cdots+a_Nz^{-N}} = \frac{B(z)}{A(z)}$$

(4)

Z represents the delay elements, the coefficients *b* represent the gain of the different time shift of the feed-forward signal (zeros) while the coefficients *a* represent the gain values of delayed elements of the feed-back of the time shifted signal. Kirk and Hunt provide a tutorial on the properties of z, the unit circle and how they can be used to represent the behaviour of a digital filter.[99]

It is difficult to edit the filter coefficients derived with LPC analysis as they don't provide an intuitive way to perform the desired changes to the spectral envelope estimated, also the all-pole filter can become unstable as a result of the editing

[99] Ross Kirk and Andy Hunt*, Digital Sound Processing for Music and Multimedia* (Oxford, England; Boston: Focal Press, 1999), 149-150-151

process. In a recursive filter if the gain of any feedback path is greater than one then the output signal will exponentially grow making the filter unstable. This prompts the exploration of a different way to manipulate the coefficients.

The roots of the vocal tract polynomials of different segments of speech can be computed and displayed from coefficients of LPC analysis of speech.[100] This suggests the need to perform an operation on the roots of the polynomials to control the position of formants and anti-formants approximated with Prony's method. The use of the z-plane (described in more detail in chapter 6) during the editing of the filter coefficients helps to monitor the stability of the filter during the modification of its coefficients.[101] The idea is to design a filter that is able to model the interaction between formants and anti-formants but the position of complex conjugate pairs of poles and zeros on the z-plane will be extracted from recorded speech. More detail on how complex conjugate pairs have been used in this thesis for the transformation of sound is provided in Chapter 6.

A frame-based system with overlapping windows can be implemented to design a time-varying filter from the interpolation points representing the movements of the poles and zeros.[102] However, the manipulation of filter coefficients provided by Prony's method can offer the possibility to perform interpolations between the spectral envelope of speech sounds over time.

Wishart describes the use of consonants as /l/, /m/, /n/, /ng/ to perform transitions from one formant state to another.[103] This suggests extracting such formant states with Prony and computing interpolations between them. A system that can extract zeros from speech will also allow the capture and change over time of the anti-formant states to design a morphing system consisting of a pole-zero filter varying in time. Dodge and Jerse explain how the coefficients of the all-pole filter in LPC synthesis can be edited to alter the spectrum of the resulting sounds.[104] A similar

[100] Gopi, *Digital speech processing using Matlab* (New Delhi: Springer,2014), 104-105.
[101] Dodge and Jerse, *Computer music: synthesis, composition, and performance,* 234.
[102] Thomas P. Barnwell III and Kambiz Nayebi and Craig H. Richardson *Speech coding: A computer laboratory textbook.* (New York; Chichester: Wiley, 1996), 90-91-92.
[103] Wishart, *On sonic art,* 278
[104] Dodge and Jerse, *Computer music: synthesis, composition, and performance*, 234

approach can be used to edit the coefficients of the Prony analysis. It is possible to employ a method to design recursive filters by using complex conjugate pairs of poles and zeros on the z-plane[105]. This approach can be explored to design transformation of sound over time. This process can be achieved by extracting the position of poles and zeros of two specific formant states. Next, one can perform interpolation between the roots of poles and zeros to manipulate the filter over time. The result is a seamless transition between the two formant structures. Therefore, we can use the resulting time-varying filter to gradually change timbre in music. The synthesizer proposed by Klatt uses five formants, the resulting five pole filter is represented with five conjugate pairs of poles.[106] Adding pairs of zeros on the numerator allows the introduction of anti-formants for the synthesis of nasals. The position of poles and zeros can be extracted with Prony's method. The development of the device proposed in this chapter, that makes extensive use of Prony's method, can be used to explore the research questions; this is achieved by studying how the colour of speech sounds and extraction of formant/anti-formant states provides control over sound, and what types of sound can be designed in the studio by the manipulation of recorded sources. The approximation of three formants is sufficient in speech synthesis for the perception and study of vowel sounds; however, as discussed in section 2.1, higher formants are important to determine the idiosyncratic timbre of an individual's voice. More formants allow one to transfer more details about the timbre of speech and this feature could be desirable to explore the colour of speech in music. Furthermore, the audibility of zeros increases the more of them there are or when groups of them are moving at the same time.[107] This suggests the design of a morphing system that approximates five formants and three zeros; however, the configuration based on a IIR filter transfer function allows an increase of the order of analysis and the number of poles/zeros that the system is able to manipulate.

Triangular or Hanning windows are widely used as a shape of the frame's window for LPC analysis of speech[108]. One can employ Hanning or triangular as a frame's window

[105] Lynn and Fuerst. *Introductory digital signal processing with computer applications*, 168

[106] Klatt "Software for a Cascade/Parallel Formant Synthesizer*,"* 986.

[107]Perry R. Cook, "Formant Peaks and Spectral Valleys,"136-137.

[108]Barnwell III and Nayebi and Richardson, *Speech coding: A computer laboratory textbook*, 96.

to capture the starting and ending spectral envelope of the transformations and as windows for frame-by-frame re-synthesis. For example, when performing analysis and re-synthesis through overlapping frames one might consider the use of a smooth window shape, such as Hanning, which in general provides smooth results. However, there are cases in which over-lapping frames are not used, for instance, the analysis of a single frame or the concatenation of frames during the analysis or re-synthesis. There exists a state-save method for signals processed in a frame-by-frame basis, the method consists in saving the current state of the delay elements to be used in the next frame[109]. This method can be used to explore how saving the state of the current frame (or interpolation point) for the next one can provide smooth manipulation of the filter over time for the synthesis of sound. This suggests the need to explore the use of concatenated rectangular windows to investigate the difference between interpolations performed with cross-fading static filters and the update of a filter state with no overlapping windows.

An interpolation tool that allows one to control timbre is of interest for a wide range of applications from music to sound design and scoring for film and video games. There is evidence of experiments conducted to explore the relationship between the features of timbre and the emotional state that it can evoke in the listener.[110] An example of practical application of the interpolation between spectral features of source and target sounds can be found in the production of emotion-driven processing of sounds. This involves the creation of automations in the sound design of video game soundtracks by applying spectro-temporal changes to the features of a sound over time from the current emotional state to the target one[111]. One can vary the formant structures of sounds over time as a way to control tension in music or any other form of audio for entertainment in order to create interest and induce different emotional responses.

[109] Wai C. Chu, *Speech coding algorithms: foundation and evolution of standardized coders*. (Hoboken, N.J.: Wiley, 2003), 50.

[110] Duncan, Williams. "Toward Emotionally-Congruent Dynamic Soundtrack Generation." *Journal of the Audio Engineering Society* 64, no. 9 (2016): 655-657.

[111] Duncan, Williams “Emotion in Speech, Singing, and Sound Effects.” In *Emotion in Video Game Soundtracking. ed.* Duncan Williams, Newton Lee,17-26, International Series on Computer Entertainment and Media Technology. Springer, Cham ,2018

Matlab is a scientific computing environment that has a wide range of applications and features and is particularly useful in musical signal processing as it allows the user to perform audio analysis and processing with an extensive range of existing functions.[112] It is a powerful tool that can be used to design algorithms that deal with the manipulation of sounds and to build customised audio processors through computer code. A function to compute linear prediction coefficients is available in Matlab and can be used for comparisons with coefficients computed by Prony. The literature provides examples of Matlab code to perform cross-synthesis by using a frame-by-frame LPC method.[113] This suggests the need to investigate how a similar frame-by-frame arrangement can be used to implement a Prony based cross-synthesizer in order to explore what features this method can add to the estimation of time-varying filters.

[112] *Matlab*, version. 2018a (The MathWorks, Inc., 2018), computer software.

[113] D.Arfib, F. Keiler, Zölzer, U. “Source-Filter Processing” in *DAFX: Digital Audio Effects.* Ed. Udo Zölzer. (Chichester: Wiley, 2002): 317-318.

Chapter 3

3.1 Introduction

This chapter outlines the differences between LPC and Prony's method as techniques for spectrum analysis. Parametric modeling is explored in this chapter to extract spectral envelopes from the analysis of recorded speech and to convert the resulting magnitude response to filter coefficients for time-domain design of digital filters. Comparisons between the audio analysis of nasal consonant sounds between the all-pole LPC and Prony's pole-zero method are documented in order to develop a tool able to extract both resonances and anti-resonances from speech.

The difference between the two methods is documented in the following sections in order to study how the type, the phase and the length of the signal used as input sequence affects the resulting frequency response of the filter. The chapter will also provide an overview of current technology for cross-synthesis to investigate how Prony's method can help combine audio transformation and filter design techniques to design a novel tool to perform morphing.

The term parametric modeling includes two words that need defining: parametric and modeling. When we create a model of sound, we make some assumptions about the components of sound production or impose a structure in terms of how the sound is produced or perceived. With parametric modeling we impose an assumption about the way in which sound is produced (or perceived) in a particular model onto the way we describe its parameters. That model will have particular parameters; in the case of LPC the parameters are the excitation signal (sometimes also referred as the error signal) and then there are the coefficients of a filter. The model assumes that we have an excitation which is feeding a resonator, in particular LPC attempts to find the best parameters for the resonator. It is possible then to choose the order of analysis (e.g. order of 5,10,25) depending on how intuitive or simple the desired approximation. By increasing the order of the analysis LPC becomes better and better in approximating the resonator to the extent that an order too high might approximate the individual harmonics of the sound analysed. However usually it is preferable to have a more intuitive approximation of the spectral shape derived and to avoid too high values of order, and instead use an excitation signal that contains a broad spectrum with a wide range of partials. There is a trade-off between model accuracy and model simplicity: the higher the order of analysis the more accurate the model but, at the same time, the model becomes less intuitive. Having decided upon the model and how to represent the model as an excitation passing through a resonator, the next step is to find the parameters of that model; the idea is to set the parameters of that model to be such that they minimize the difference between the signal that is being analysed and the signal that can be synthesized by that model. The Levinson-Durbin algorithm is used in LPC to perform this task. The algorithm iteratively adjusts the parameters of the model until the output sound of the model is as close as possible to the input sound of the model.

The difference between parametric modelling and Fourier analysis is that Fourier is a transform, it doesn't assume a model of sound production. It determines how much energy there is and at what phase the underlining sinusoid is at, in fixed frequency bands while LPC estimates the parameter of a source filter model. LPC tells us where the peaks in the spectrum are, the DFT tells us how much energy there is in a fixed grid. LPC is an attempt to find the parameters of the underlining model which gives the fixed grid.

3.2 Time-domain filter design with LPC

LPC allows the design of resonant filters that approximate the spectrum from an input sequence of samples and it provides a way to perform spectrum analysis.

LPC analysis differs from the Fourier method of analysis as it estimates the overall spectral envelope rather than computing the energy of the individual frequencies of the spectrum analysed.[114] This is because LPC approximates the spectrum as the result of an excitation (source) to a resonator (filter).

The advantage of LPC and other autoregressive methods over the Fourier analysis method is they offer the ability to provide spectral estimation from a small amount

[114] Roads, Curtis. *The computer music tutorial*, 185.

of data and to provide an intuitive model of the resonant structures of the spectrum analysed.

In an all-pole model, linear prediction is represented as a given output signal S_n a combination of past values (samples in this case) and an unknown input u_n multiplied by a gain factor G as shown in equation (5):[115]

$$s_n = -\sum_{k=1}^{p} a_k\, s_{n-k} + Gu_n \tag{5}$$

(5) All-pole model linear prediction.

The subscript n represents the value of the current sample value and p represents the order of the filter. The value u_n is the excitation, sometimes called the error signal as it describes what the filter has not been able to describe. When attempting to determine the parameters of the filter, the system is trying to reduce the error signal (i.e. the difference between the signal and the filter's prediction of it).

The aim of solving the mathematical problem expressed in this equation is to calculate the predictor coefficients a_k and the gain G from the given signal $S_{n.}$

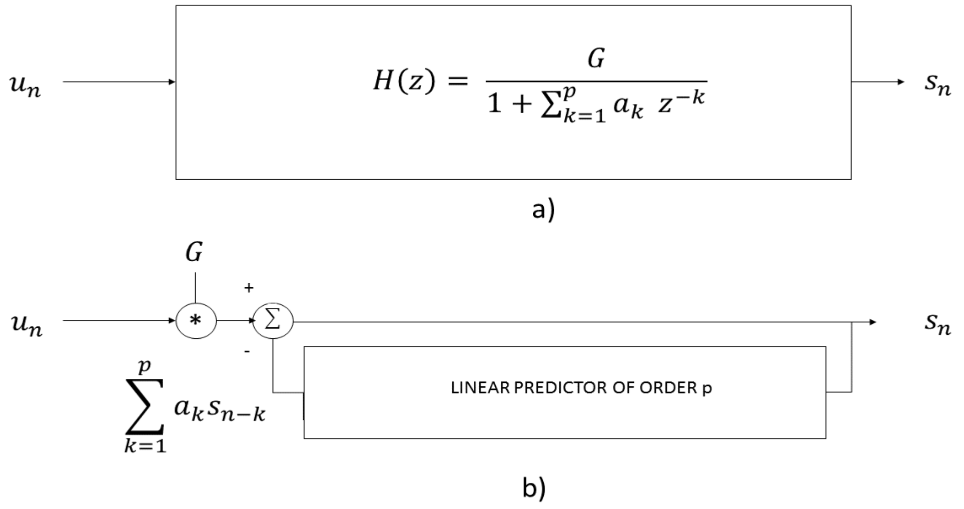

u_n
$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k\ z^{-k}}$
s_n
a)
G
u_n
*
+
Σ
-
s_n
$\sum_{k=1}^{p} a_k s_{n-k}$
LINEAR PREDICTOR OF ORDER p
b)

Figure 14. all-pole model expressed in a) frequency domain and b) time domain after Makhoul.[116]

[115] John Makhoul, "Linear Prediction: A Tutorial Review," Proceeding of the IEEE, Vol. 63, no. 4 (1975): 562-563.

[116] Makhoul, "Linear Prediction: A Tutorial Review," 563.

Figure 14 shows how equation (5) can be expressed in the frequency and time domain. The representation in the time domain suggests a relation between the order of prediction *p* and signal length. The Levinson-Durbin algorithm can be used to solve the linear prediction problem and compute the filter coefficients of the all-pole filter.

Auto-correlation means the multiplication of the signal by itself at different time shifts, where we multiply the signal by itself, and the sum of the multiplication of all the samples is a high value shows that the signal is well correlated with itself at a specific time shift. It is a way to analyse signals to determine at what time shifts it is well correlated with itself, at those particular time shifts there is a resonance at the frequency which that time shift represents. If we consider the analysis of the samples of a known output signal that has autocorrelation values *R[l]* for *l = 0, 1, . . ., M* with *M* being the highest order of the predictor. Levinson-Durbin is a recursive-iterative process that finds the solution to the equation at order 0 and repeats the process to find a solution for order one, and so on, until the *(M-1)*th in order to find a solution for the *M*th order.[117] The sequence of values included in the correlation matrix of the signal R[*l*] is a Toeplitz matrix, that means that all the diagonal elements of the matrix are equal. The blocks representing the correlation of lower-order analysis of the sequence of values are included in the *M*th order correlation matrix. This technique is used to compute the filter coefficients of the all-pole filter from a given sequence of samples; the values of the samples of the signals are arranged in a similar way as shown in Figure 14b. This suggests a maximum order of analysis, equal to the length of the signal analysed. The order is the number of poles approximated; the maximum value possible for the order is defined by the number of samples of the input signal minus one.

The coefficients computed and the order of prediction in LPC affect are meaningful for sound synthesis as they define the number of formants approximated. An all-pole system can be represented by the following formula:

[117] Chu, *Speech coding algorithms: Foundation and evolution of standardized coders*, 107-108.

$$y(n) = b_0x(n) - a_1y(n-1) - a_2y(n-2) - \cdots - a_Ny(n-N) \tag{6}$$

The formula in (6) defines a system that has order *N,* the order of the filter is defined by the number of delays. The higher the number of delays, the higher the order of the filter will be; a_N represents the coefficients of the filter that define its characteristics while the coefficient b_0 scales the amplitude of magnitude response of the filter. [118]

3.3 Time-domain filter design with Prony's method

This section gives a basic and simple overview of how filter coefficients are calculated using Prony's method, although this is not essential to understanding the processing system and creative applications of that system which appear later in Chapter 6.

Prony's method is a collection of approaches that can be applied to the spectral analysis of sound. Similarly, to LPC, Prony's method estimates a set of coefficients derived from the past samples of a known input signal. Instead of deriving the filter coefficients, Prony's method derives information about the phase, frequency and damping factor (rate of decay) of a set of sinusoids. This feature of Prony analysis has been applied to sound analysis and re-synthesis of percussion sounds and in the spectrum analysis stages of the CHANT system.[119] Prony's method provides a precise method to design time-varying sinusoidal models, for instance an approach to using Prony's method in sound synthesis has been used to estimate the parameters of additive synthesis of percussive sounds, for instance for the re-synthesis of a piano note.[120] However, Prony's method can be used in parametric modelling to design digital filters, providing parameters to control the filters rather than estimating the sinusoidal components to control additive synthesis. Prony's method can be applied to design IIR filters by using a matrix description; with this method it is possible to

[118] Dodge and Jerse, Computer music: synthesis, composition, and performance, 210-211. Charles Dodge and Thomas A. Jerse Computer music: synthesis, composition, and performance. 2nd ed. (Boston; Schirmer. 1997), 210-211

[119] Roads, The computer music tutorial, 597-598.

[120] Jean LaRoche, "A new analysis/synthesis system of musical signals using Prony's method-application to heavily damped percussive sounds," *International Conference on Acoustics, Speech, and Signal Processing,* Glasgow, UK, vol.3, (1989): 2053-2056.

compute the filter coefficients of both the numerator and denominator of the following transfer function:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_M z^{-N}} \tag{7}$$

In (7) the denominator represents the poles (feed-back filter coefficients) and the numerator represents the zeros (feed-forward filter coefficients) in the frequency response of the filter. More zeros (valleys) will be displayed in the spectral envelope, the higher the order value M of the numerator; on the other hand, the higher the order value N of the denominator the more poles (peaks) will be displayed. This approach suggests the use of Prony's method as a possible option to design a pole-zero model for the estimation of the vocal tract filter. As consonant sounds display anti-formants, due to the feed-forward propagation of sound in the vocal tract, a pole-zero model provides a solution to a more accurate estimation of nasal consonant resonant structures. An all-zero model can also be referred to as a moving average (MA) model; an all-pole system as an autoregressive (AR) model and a pole-zero system as autoregressive moving average (ARMA) model.[121] This suggests the use of Prony's method provides in order to employ an ARMA model for the control of subtractive synthesis.

Poles and zeroes can be studied through visual representation on the Z-plane. The Z-plane consists of a unit circle and axis from -1 to 1 for a real and an imaginary part. The poles are represented on the unit circle by an 'x' while the zeroes are represented as 'o'. Poles and zeros can be mirrored in the imaginary part in order to convert the coordinates of their position on the plane-to-filter coefficients and are called a conjugate pair. A conjugate pair of poles can be used to model the effect of one formant. In the same way a conjugate pair of zeros can be used to model an anti-formant. Combining more conjugate pairs of poles and zeros allows us to model more

[121] Makhoul. “Linear Prediction: A Tutorial Review,” 562.

formants and anti-formants. This is a crucial as it allows the expansion of the all-pole model of LPC by including anti-resonances in the frequency response. Furthermore, it suggests the possibility of editing the coefficients of the numerator and denominator separately. This is a meaningful feature for the design and visualisation of sound transformation over time by moving the position of the poles and zeros on the Z-plane.

The equation (7) can also be written as convolution in the Z transform domain[122]:

$$B(z) = H(z)\,A(z) \tag{8}$$

This allows one to represent the convolution as a matrix multiplication where the values of a known input sequence are arranged in a symmetrical way according to the diagonal elements (Toeplitz matrix). However, the matrices are divided in partitions to perform calculation between separate sections of the matrices, in order to compute the filter coefficients.

The filter coefficients a_N and b_M can be computed by performing multiplications between different partitions of the Toeplitz matrix containing the sequence of samples. A key element of this approach is that it derives the denominator coefficients first and then it derives the numerator coefficients *b* from the denominator's coefficients ***a*****.** This means that the zeros are derived from the poles and a partition of the matrix. A detailed continuation of this explanation is given in Appendix VII.

3.4 Analysis of nasal sounds: Prony's method vs. LPC.

This section aims to compare the differences in spectral envelopes computed by LPC and Prony using the same order of analysis. In LPC analysis and synthesis the recorded sound that is analysed is divided into shorter segments of audio called frames.

[122] T.W. Parks and C.S. Burrus, *Digital Filter Design* (New York: John Wiley and Sons,1987),226.

A frame is multiplied by a window function like the Hanning or Triangular window. The process is repeated for all segments of the sound and are concatenated in a way that the next frame cross-fades with the previous one, to model the variation over time of the vocal tract. The length of the frame, the shape of the window and the order of analysis all contribute to a successful approximation of the vocal tract filter.

If the order of analysis is too high there is the danger that the harmonics of the signal might be approximated as resonances, usually the harmonics should be part of the excitation. If the order of analysis is too low, then the description of the vocal tract is not detailed enough with a low number of resonances and in general with a broader bandwidth. In typical implementations, the variation over time is created by cross-fading fixed filters.

The order of analysis affects the number of poles approximated and usually two poles are used to model the effect of one formant.[123] For instance, a denominator order of six models the effect of the three formants that are usually approximated for the perception of vowel sounds. As shown in (5) LPC analysis is an all-pole model, in a similar way, Prony's method approximates the poles but also anti-formants are represented by a pair of zeros.

A way to compare the spectral envelope estimated by LPC and with Prony's method is to perform the analysis on the same sequence of samples and then compare the magnitude response of the filter approximated. As nasal consonants include both formants and anti-formants, the sequence of samples of a recorded nasal sound should provide a meaningful starting point to investigate the approximation of zeros.

The signal should be a segment of a recorded consonant that includes enough waveform cycles to provide as accurate a representation of the spectrum as possible.

In Figures 15 and 16 an example of analysis is shown of the same segment of speech from an /m/ consonant. The analysis is performed in Matlab by using the *lpc* and *prony* functions that allow one to set the order of analysis. The function *prony* provides control over the order of zeros and poles separately, the coefficients of the zeros are then provided as numerator and the poles as denominator (as in (7)).

[123] Barnwell III, Nayebi and Richardson, *Speech coding: A computer laboratory textbook,* 94.

The order of analysis chosen is set to approximate five formants (poles) and, in the case of *prony,* also three anti-formants (zeros). The resulting plot shows different shapes of the valleys within the spectral envelope, also a different overall maximum magnitude value.

The difference between the spectral envelope computed with LPC and the spectral envelope computed with Prony's method can be summarized as follow:

- Prony designs a sharper frequency response that successfully includes zeros
- The zeros seem to affect the bandwidth of the poles interacting with them. It seems to truncate the effect of the pole around 500 Hz

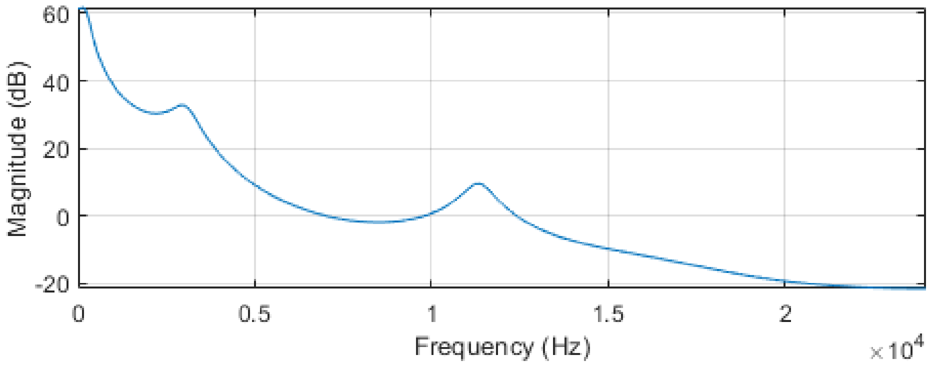


Figure 15. LPC analysis consonant /m/ for order 10 (five conjugate poles).

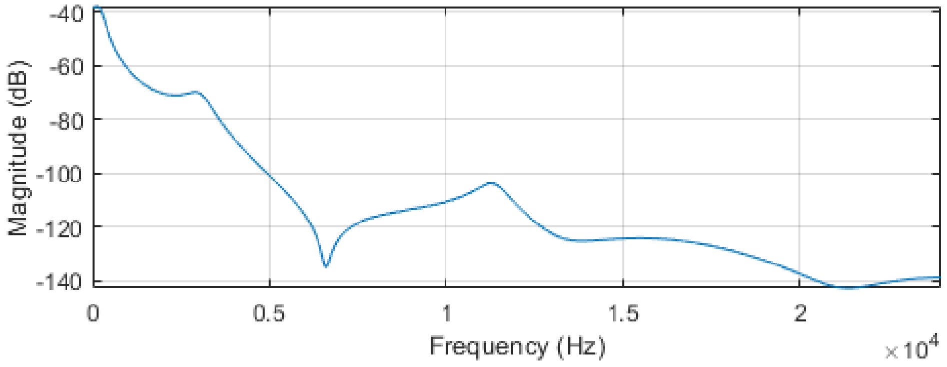


Figure 16. Prony's analysis consonant /m/ order 10 (five conjugate poles, three conjugate zeros.

- LPC does not approximate the effect of a very prominent zero between 600 Hz and 700 Hz. This frequency band in many cases overlaps with the first formant, suggesting possible attenuations of poles for nasalized vowels.
- LPC also does not represent the zero around 2000 Hz.

This suggests that Prony's method can be used in time-domain parametric modelling techniques to approximate the filter coefficients from the analysis of speech, and to derive the roots of the polynomials from the coefficients approximated, in order to provide control over their position on the z-plane. This process allows one to specify zeros as well as poles from recorded sound, while also generating creative possibilities to perform manipulation over time of the centre-frequency and bandwidth of formants and anti-formants.

The idea is to design a filter that is able to model the interaction between formants and anti-formants, but the position of complex conjugate pairs of poles and zeros on the z-plane will be extracted from recorded speech.

Chapter 6 provides details on the effect of the movement of a complex conjugate pair of poles on the magnitude response of a filter. The effect of moving the poles on the z-plane can be summarised in this way:

- Moving the poles closer to the origin of the circle widens the bandwidth of the filter. Moving the poles closer to the circle narrows the band of the filter.
- Rotations on the position of the poles affects the position of the centre-frequency of the filter.

In relation to formants: moving poles closer to the circle produces a narrower peak, while rotating the poles moves the position of formants. For example, by moving a pole anti-clockwise along the circle its movements will be mirrored by its conjugate pole, the result is a formant with frequency that increases in frequency over time. The mirroring of the conjugate pair is not an automatic process and calculation needs to be performed to enable the use of this feature for creative transformations of sound.

Different vowels, for example, can be represented by the position of different pole conjugate pairs on the z-plane. It is possible to explore the interpolation between vowels by performing interpolation on the position of poles on the z-plane to gradually match a different formant pattern.

Audio analysis of consonants is the first step to exploring the differences in features between Prony's method and LPC; the idea is to compare the analysis of a nasal consonant /m/ as it produces anti-resonances. LPC is still largely used in current tools for cross-synthesis. For instance, TRAX includes voice and sound processing tools to shape, change, transfer and shift formants in order to perform transformation of timbre, or to combine two different sources in one sound.[124] It also provides control over the length of the frame, number of overlaps and the order of the LPC analysis and synthesis. Another IRCAM program, Audiosculpt, uses LPC and employs the true envelope LPC method (TE-LPC).[125] The true envelope LPC analysis provides better results for the analysis of higher pitch signals with fewer harmonics[126]. Although a tool for phonetics, Praat allows LPC analysis of recorded sound to control another sound with the spectral envelope obtained. Praat is explored as a creative tool in Music in this thesis. Other examples of tools which use LPC within their audio processing functions include Csound and Supercollider.[127]

This suggests that LPC analysis is still relevant to the design of hybrid sounds through the superimposition of a spectral envelope estimated through audio analysis.

The approach to Prony's method in order to control the features of audio filters by 'sampling' the parameters from recorded speech suggests it can be used to go beyond the all-pole model of LPC and design new techniques for the manipulation of spectral envelopes derived from audio analysis.

[124] Flux, “Ircam Trax v3,” accessed September 14, 2020, https://www.flux.audio/project/ircam-trax-v3/.

[125] Niels Bogaards, “Analysis-Assisted Sound Processing with Audiosculpt,”8th International Conference on Digital Audio Effects DAFX-05 (Spain: Madrid, 2005): 269-272.

[126] F. Villavicencio, and A. Robel and X. Rodet, "Improving LPC Spectral Envelope Extraction of Voiced Speech by True-Envelope Estimation," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings* (Toulouse,2006): I-868 - I-872.

127 “Linear Predictive Coding (LPC) Resynthesis”, The Canonical Csound Reference Manual, Csound, Accessed September 14, 2020. https://csound.com/docs/manual/SpectralLpcresyn.html; “LPCAnalyzer” in Classes, SuperCollider 3.11.1, Accessed September 14, 2020. https://doc.sccode.org/Classes/LPCAnalyzer.html.

This thesis explores the use of tools based on Prony to enable more intuitive, flexible and better models of speech and speech-like signals for music signal creation and processing.

Chapter 4

4.1 Introduction

This chapter explores a collection of techniques used to transfer the resonant structures of vowel and diphthong sounds to synthetic sound sources. In particular, it investigates the features of formants and how they affect the perception of vowels as a single entity.

The aim is to investigate how the voice can be used as a controlling mechanism by providing instructions to audio processors: for instance, details about the qualities of the voice can be approximated through the audio analysis of recorded speech and used to control digital filters. The techniques described in this chapter are aimed at finding new ways to obtain and transfer the features of voice and to manipulate those features so as to alter the perception of human voice. Technology is used to explore the set resonances that contribute to the identity of vowels in order to separate the individual formants. Cascade and parallel formant synthesizers have been used in phonetics and linguistics to study how sound is produced and perceived, thereby providing control over individual features of the vocal tract modelled with digital filters. The use of sound synthesis provides a degree of precision that is not possible with acoustic speech. For instance, isolating and hearing the three formants of a vowel sound as three separate resonances. An implementation of both a parallel and cascade synthesizer is implemented in Csound to explore how control over the individual or groups of formants can affect the perception of speech. The idea is to manipulate formants to produce sounds from a model of the vocal tract that are not perceived as speech. This technique gives the user control over how much the synthesized sounds are recognisable as vowels.

Praat is used to perform LPC analysis on a collection of recorded diphthongs. Diphthong sounds display variation over time of the formant frequencies, as one vowel is morphed into the next one, making them ideal to explore different methods to transfer time-varying vowel-like features to synthetic sounds. Pulse train and white noise are usually used in combination with LPC for the re-synthesis of speech. As previously described in Chapter 2, LPC analysis can be used in cross-synthesis techniques to transfer the spectral envelope of speech to another sound.

A single glottal pulse can be used to define the characteristics of a vowel and represent its resonant features.[128] These features can be used to capture the formants of a vowel by using an impulse as the excitation of a filter whose coefficients are derived from LPC analysis of recorded speech. Vocal fry is a type of voice source that makes the individual glottal pulse more audible and can be synthesized with a very low frequency pulse train (e.g. less than 1 Hz). Curtis Roads describes a sound synthesis technique called Pulsar synthesis that involves the use of a database of pulse train sounds convolved with a database of sampled sound sources.[129] This suggests combining the synthesis of a collection of pulse trains that mimic vocal fry (creaky voice) with convolution with other sound sources. The technique differs from pulsar synthesis because it uses vocal fry pulses exciting a filter derived from recorded speech instead of a pulsar generator. LPC analysis of a collection of diphthong sounds can be used to filter the vocal fry pulse train to introduce movement in the resonant structures; which are then transferred through convolution with another sampled or synthesized sound.

A new approach to cross-synthesis that combines the synthesis of vocal fry voice sources, LPC analysis and convolution is explored in the composition *Vocal Fry* (2015). This approach is used to study different methods for the transfer of the timbre of vowels to other sound sources. The software Praat provides a way to explore how technology designed for phonetics can be employed in music to design hybrid sounds from recorded speech.

4.2 Tools for synthesis of vowel qualities.

As discussed in Chapter 2 the first three formants affect the perception of vowels the most. Each vowel sound is associated with a specific pattern of articulator adjustments which also produce a specific combination of formant frequencies[130].

To start exploring how the features of vowels can be altered or combined with another sound, subtractive synthesis through the models proposed by Klatt and techniques using LPC analysis in combination with convolution are used. The idea is

[128] Kent and Read, *Acoustic analysis of Speech*, 105.

[129] Roads, Curtis, "Sound Composition with Pulsars,"J. Audio Eng.Soc. Volume 49, No.3 (2001): 139.

[130] Sundberg, *The science of the singing voice*, 23.

to use the estimation of formant frequencies changing in time to extract multiple impulse responses of the vocal tract from a recorded vowel, to be used in convolution with another sound. The process is inspired by the one used to create artificial reverberation through convolution but, instead of using a single pulse, this technique involves a sequence of pulses that excite the LPC analysis of diphthongs. Each pulse excites the formant pattern of a different articulation state of the vocal tract. The resulting sequence of pulses (each exciting a different vowel) is then convolved with a recorded sound of stereo white noise (e.g. 22 seconds). Different amplitude envelopes can be applied to the stereo white noise to achieve different effects. For instance, one could use a Hanning window or random amplitude modulation. This results in a synthetic texture evolving over time.

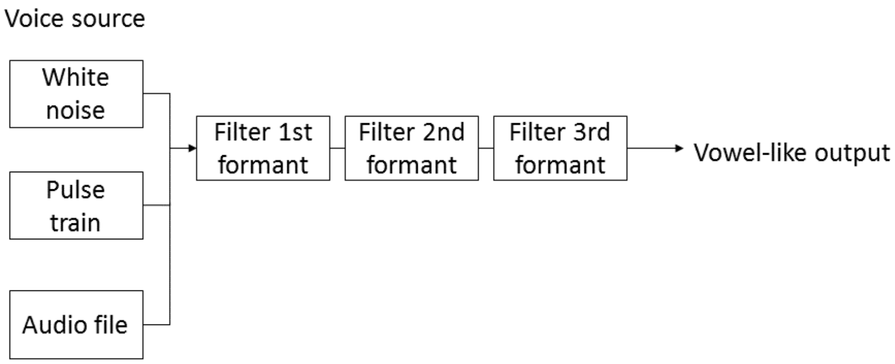


Figure 17. Example cascade formant synthesizer for vowel synthesis. Version adapted for CSound instrument.

A model able to approximate the effect of the first three formants can be used to explore the qualities of vowel sounds. Figure 17 shows a model designed as a cascade formant synthesizer which provides control over the voice source and the filters. This model has been implemented in Csound by connecting a white noise generator and a pulse train generator to a bank of three bandpass filters in cascade; it is based on the cascade model designed by Klatt.[131] Refer to Example 1.1 to hear an example of synthesis of vowel with the cascade model.

The user of the synthesizer has control over the type of voice source used, the formant frequencies and the bandwidth of formants. The voice source is assigned to different Csound instruments; one for voiced source, another for unvoiced sources or a third that allows one to read audio files to filter them with formant frequencies. Selecting the number of instruments to operate allows the user to select the desired voice source (See Appendix I for Csound instrument). The formant frequencies and bandwidths are assigned to variables so the user can change the input parameters of the synthesis.

Below is an example of the implementation of cascade formant synthesizer in Csound, text following ';' represent a comment.

```
kenv linseg 0.0001, p3*0.5,1, p3*0.5,0.0001

; amplitude envelope

kmod randi 2,6
;randomized variation in frequency voice source

kvar randi 400,5
;randomized modulation in amplitude.

asig buzz iamp+kvar,ifreq+kmod,inharm,ifn ;phonation
voice source

afilt butterlp asig,ifreq*2
; spectral slope higher partials

a1   butterbp afilt,ifc,iband
a2   butterbp a1,ifc2,iband2
a3   butterbp a2,ifc3,iband3
a4   balance a3,asig

outs(a4*kenv)*igainsx,(a4*kenv)*igaindx
endin
```

The code above can be explained in relation to Figure 17 as follows:

[131] Klatt, "Software for a Cascade/Parallel Formant Synthesizer," 975.

- The signal 'asig' is a voiced source generated by the oscillator opcode function buzz. This opcode was chosen because it produces a series of sinusoidal harmonics that can be used to emulate the voice source idealized harmonic series with a 12dB per octave roll-off[132]. The signal is filtered by a lowpass filter whose output 'afilt' approximates the -12dB per octave roll-off of the partials exhibited by a vocal source. Speakers do not produce completely static voice sources, so movements are introduced by randomized variations in amplitude and frequency by the random LFOs 'kvar' and 'kmod' to mimic the natural variation in speech emission that is not in control of the speaker.
- 'a1' represents the first formant frequency and is produced by filtering the voice source with a bandpass filter.
- The output of the first formant filter is fed to 'a2', a bandpass filter that represent the second formant.
- The output of the second formant filter is fed to 'a3', a bandpass filter that represent the third formant.
- The effect of the filters might reduce the gain of the output signal in unpredictable ways. 'a4' helps to match the root mean square (RMS) power of the signal 'a3' to the RMS power of the oscillator signal. This helps to produce more similar gain levels after the filter bank while synthesizing different vowels.
- In the out section it is possible to control the panning of the signal by setting values between 0 and 1. 1 for only the left speaker, 0 for only the right speaker.

By interacting with this model, it is possible to produce a variety of vowel sounds by matching the variables of the synthesizer with the values included in vowel formant charts. A transcript of the code can be found in Appendix I.

This Csound code has been adapted to synthesize different voice sources like whispering (using white noise) and sub-harmonic sounds, a specific technique that allows the user to let the false vocal folds (or ventricular vocal folds) oscillate an

[132] Sundberg, *The science of the singing voice*, 65.

octave lower than the vibration of the vocal folds (or half the frequency of the fundamental).[133] By adding a pulse train with pitch an octave lower before the low-pass filter of the voice source fundamental it is possible to approximate the vibrations of the false vocal folds. This idea was inspired also by octaver audio processors which follow the pitch of an input signal an octave lower than the original. Another way to experiment with sub-harmonics is to produce two notes with the same vowel parameters but different pitches (one of the two an octave lower). Using different randomization on the lower of the two (usually slower) can produce a similar effect. An example of the sounds that this model can produce will be demonstrated in the item *Vocal Fry*. During the production of vowels, the three main formant frequencies affect the voice source simultaneously, as they are part of the vocal tract magnitude response. It is possible to attempt to hear the effect of the individual formant frequencies using whispered sounds. By whispering a long sustained /a/ vowel and gradually changing the sound to an /o/ it is possible to hear differences in the resonances of the vocal tract that are made more obvious by the unvoiced voice source.[134] However, this technique does not provide control over the individual formant frequencies independently in order to hear them separated from one another. In this case, the use of filters can provide a way to select only the desired formant frequencies and to isolate them. This idea can be adapted and implemented in a parallel formant synthesizer to isolate the peaks and explore the perception of formants by filtering white noise.

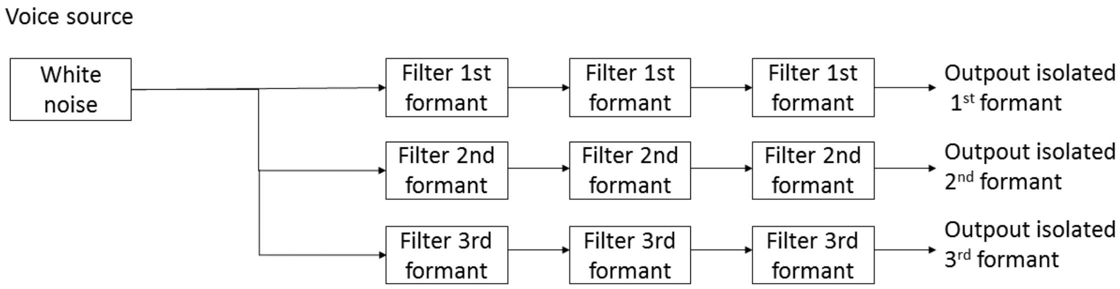


Figure 18. Using a parallel formant synthesizer allows to separate the output of each formant filter.

[133] Wishart, *On sonic art*, 265.
[134] Ladefoged, *Vowels and consonants: an introduction to the sounds of languages*, 34.

The model in Figure 18 allows the user to separate the output of each bandpass formant filter, the output of each formant can then be panned, processed separately and played back at different times. To hear a vowel with isolated formants panned in the stereo image refer to the audio Example 1.2. This technique allows the movement of the formant frequencies independently which enables the exploration of vowel perception in *Vocal Fry.* This idea can be simplified in Csound by filtering white noise with a single bandpass filter. It is possible to isolate and hear the formants of the same vowel by using as a parameter only one of the three formants or creating formant 'arpeggios' by writing a series of notes that start at different times. The Csound instrument below is an example of a bandpass filter used to isolate a single formant frequency:

```
a1   butterbp asig,ifc,iband          ; Formant selected

a2   butterbp a1,ifc,iband            ; signal is filtered again
with same formant

a3   butterbp a2,ifc,iband            ; two times in cascade.
```

The output of the filter that represents the formant filter 'a1' is filtered twice to increase the resonance of the filtering process and highlight the selected formant. The panning parameter can be explored to distribute the formant frequencies of the same vowel in the stereo image to alter the perception of speech. When heard, isolated formants seem to lose the immediate connection with speech; on the other hand, as soon as they overlap, filtered white noise seems to become a whispered vowel. A version of the code for this instrument can be found in Appendix II. Diphthongs are an example of the vocal tract changing shape over time producing the effect of a filter that changes its frequency and bandwidth parameters over time.

The software Praat is used in phonetics to perform several types of audio analysis and speech re-synthesis as it has useful features to synthesize several types of voice sources, to build formant filters with time-varying contours and to perform spectral and LPC analysis on recordings of speech. The peculiar interface of Praat combines functions for sound synthesis of voice sources, formant synthesis, generation of signals to study speech, audio analysis, pitch transpositions, audio editing and convolution. It allows one to open several files and functions at the same time and to process them separately or combine some of them. It performs like a workbench for sounds providing a high degree of flexibility to fine tune the parameters of different sounds before combining them through tools for cross-synthesis that Praat includes. LPC analysis and convolution can be used very creatively to extract and transfer features from speech.

The analysis interface in Praat allows the user to see the amplitude of the file analysed, the spectrogram and the formant trajectories. Figure 19 shows the analysis of a Cantonese diphthong /e/ to /i/ from a female speaker. The choice of Cantonese vowels comes from an interest in capturing the sound itself of the language in order explore its formant estimation in sound synthesis. At the top of the graph it is possible to see the amplitude of the signal and at the bottom, the spectrogram and formant analysis. The variations over time of the formant frequencies are highlighted by red lines. It is this kind of variation in time that LPC analysis can approximate as a filter; we could describe a similar process as the red lines representing the variation over time of the frequency of each bandpass filter. LPC analysis can approximate variations over time only if the analysed signal is windowed and therefore multiple snapshot are taken, usually every 30 ms.

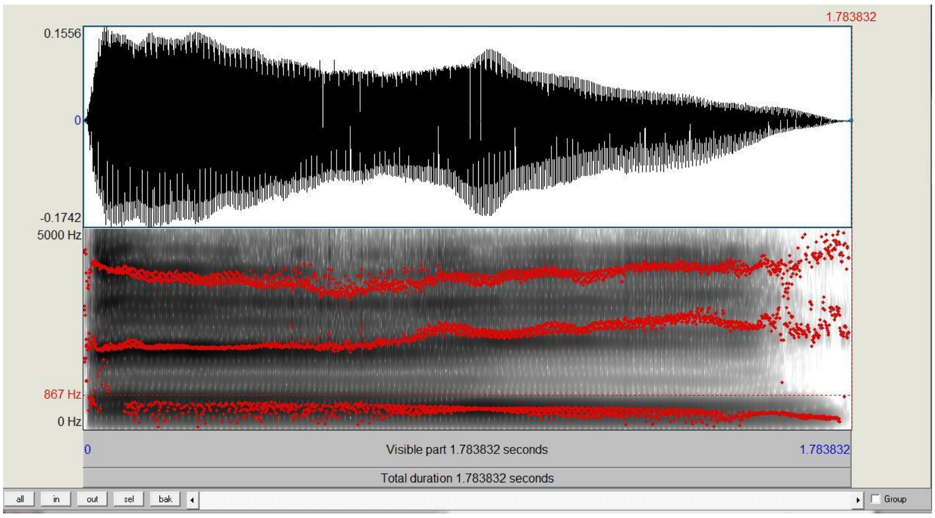
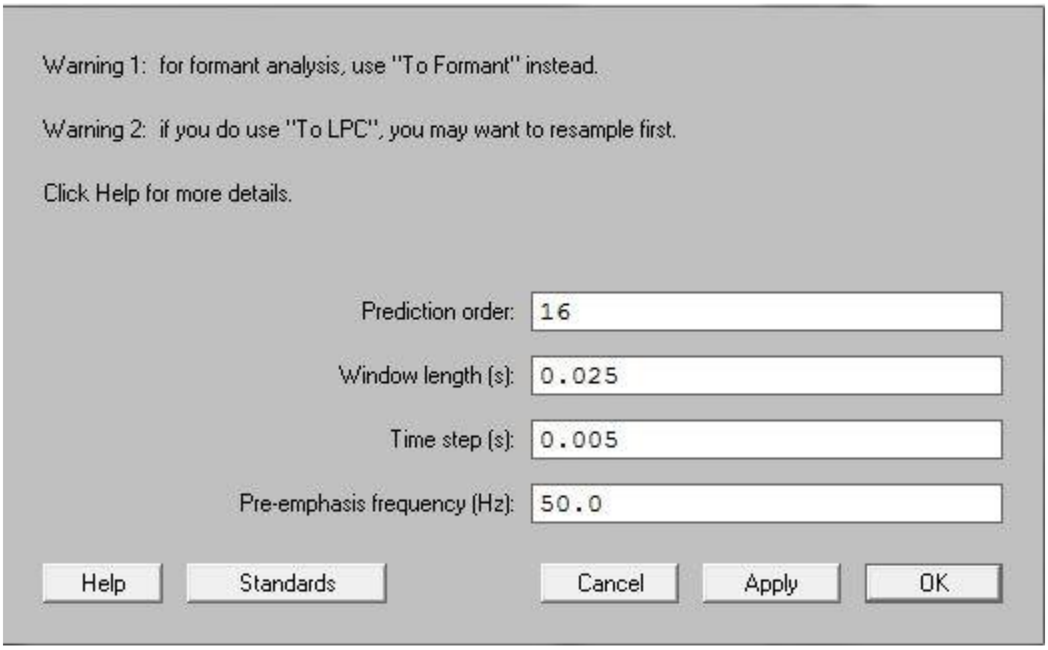


Figure 19. Example of Praat analysis window. The red lines are the formant trajectories of the dipthong /ei/. (Refer Example 1.3).

The output of the analysis is the filter coefficients as a function of time represented by frames with constant time sampling. The coefficients appear as an LPC object in the menu and, by selecting the name of the analysis object and the object of a voice source, it is possible to perform re-synthesis of the audio file of analysed speech. Within Praat, pulse trains and white noise can be synthesized to be filtered by LPC coefficients, but the system also allows the use of sound files (e.g. recordings of instruments). This provides a powerful cross-synthesis tool that can be used to explore how to transfer the qualities of vowels on different sounds.



Warning 1: for formant analysis, use "To Formant" instead.

Warning 2: if you do use "To LPC", you may want to resample first.

Click Help for more details.

Prediction order: 16

Window length (s): 0.025

Time step (s): 0.005

Pre-emphasis frequency (Hz): 50.0

Figure 20. the LPC analysis interface in Praat.

As discussed in Chapter 3 the order of prediction determines the number of poles in the filter and therefore the detail approximated in the spectral envelope. The higher the order the less intuitive the model, an order too low might underrepresent the features of the vocal tract.

It is also possible to convert the coefficients to polynomials by using the function 'Extract: To Polynomials (slice)' and then extract the roots of the coefficients. This suggests the possibility to explore if manipulation of the coefficients can be performed by operating on the polynomials to provide further transformation to the user. This is dealt with in Chapter 6.

A single glottal pulse can be used to define a vowel sound and provide information on its formant frequencies. It is possible to mimic this process through the use of speech synthesis by performing the LPC analysis on a vowel sound and by exciting the resulting filter with an impulse. It would be possible to consider the resulting signal as an impulse response of the vocal tract while producing that vowel. The Bernoulli effect causes the glottal pulses and one type of voice source that makes this obvious is the vocal fry register. Vocal fry is the type of voice source that has a very low frequency of phonation; this register of voice allows one to perceive each individual voice pulse.[135] Praat enables the user to synthesize a voice source very similar to the vocal fry register by using a function called New>Tiers>Create *Poisson point process*; this function generates pulses at randomized intervals and the average density of pulses per second can be set by the user. For example, a density between 1.1 and 12 allows one to synthesize vocal fry sources in which the individual glottal pulses are perceivable as individual temporal events.

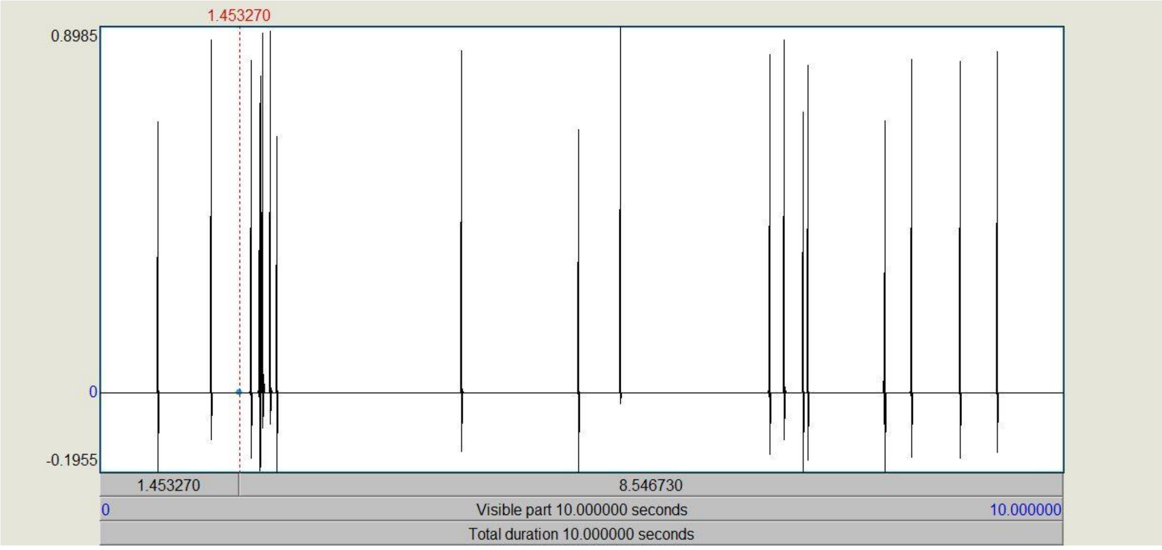

1.453270
0.8985
0
-0.1955
1.453270
8.546730
0
Visible part 10.000000 seconds
10.000000
Total duration 10.000000 seconds

Figure 21. A vocal fry source synthesized in Praat. (Refer Example 1.4)

Figure 21 shows an example of a pulse train that Praat can synthesize by using the 'Poisson point process' in the 'Tiers' menu. The duration of the source presented is

[135] Sundberg, *The science of the singing voice*, 50.

10 seconds and the very low frequency of the pulses is shown by silence between them that makes the single glottal pulses very obvious. The vocal fry sources generated in Praat can be filtered by the analysis of diphthongs to extract the impulse responses of the different instants in a collection of recorded diphthongs. The resulting signal is a series of filtered pulses and each pulse carries the qualities of the vowel sound analysed with the LPC. This suggests an option to explore how it would be possible to use this technique to transfer the formant pattern derived to another sound. One way to do this is through the use of convolution with another sound. Unvoiced sources can be used to explore and isolate the formant frequencies of vowels, so white noise can be convolved with the vowel impulse to explore what sounds this process can generate. By synthesizing a vocal fry source and filtering it with the LPC analysis of diphthongs it is possible to combine the qualities of other sounds with the variations of the diphthongs' magnitude response. Each glottal pulse of the fry source generates the formant pattern of the diphthong at that specific time in the formant trajectories of the LPC filter. The resulting collection of 'impulse responses' convolved with white noise to synthesize long musical textures varying over time.

The word convolution refers to the mathematical operation that allows a wide range of audio processing techniques. The output of a filter, for instance, is the result of the convolution between an input signal and the impulse response of the filter. However, convolution can be used to transfer the characteristics of an acoustic space (e.g. a concert hall) or different equipment for audio processing by convolving an input signal with another recorded sound used as impulse response.[136] The process of combining a recorded sound with an impulse response of a room is widely used in music production to apply reverberation to a recorded sound through a convolution processor. An example of this application is: by recording a percussive sound (e.g a balloon popping) in a room it is possible to capture its reverberation features such as length of decay, resonant features, damping of high frequencies over time. By convolving the resulting impulse response with any recorded instrument, we will

[136] Roads, *The computer music tutorial*, 429-430

hear that instrument playing in the space where the impulse response was captured. An application of this technique for cross-synthesis is the convolution of an input signal with an impulse response drawn from another sound. For example, convolving a note played by a saxophone with the sound of cymbals to create a saxophone/cymbals hybrid sound. The resulting sound blends the characteristic of both sounds but the original sources might not be recognisable. This technique can be used to create complex variation over time in the spectral content of the input sound. For example, the convolution of a sequence of grains or percussive sounds with a long note might create the effect of a filter that changes over time.[137] This suggests the option to use clouds of vocal fry convolved with a sustained stereo white noise sound. The idea is to combine LPC analysis and convolution with noise-based sound sources in a process that transfers the resonant structures of vowels as in cross-synthesis with LPC. The resulting process aims to create textures or clouds of resonances that vary over time. Vocal fry sources synthesized with LPC provide the variation over time of timbre, while convolution with white noise smooths the percussive nature of pulses and time-stretch and blurs the features captured by the excitation of LPC analysis with random glottal pulses.

Convolution provides an effect similar to reverberation or a coloured filter and allows one to make discrete sounds continuous.[138] In this idea of coloured filter convolution, with impulses exciting the LPC filter varying over time, it is possible to control the colour of the filter with the voice; the change in filter colour is controlled by what vowels are performed in the recorded file analysed. The use of stereo white noise for convolution makes it possible to smooth the rhythmic patterns of the vocal fry sound sources to create a smooth, long synthetic texture that is affected by the formants of vowels. The use of stereo white noise, which have two uncorrelated channels, affects the stereo focus of the resulting sound which is diffuse and wide. Refer to Example 1.5 to hear the order in which processing has been applied to the sound sources to achieve this effect. The vocal tract has a type of randomized rhythmic pattern during the production of vocal fry; this suggests that convolution allows one to blur the

[137] Ibid., 430.

[138] Moore, *Sonic Art: An introduction to Electroacoustic Music Composition*, 98.

rhythmic patterns of vocal fry and at the same time blur the formant pattern of different vowels.

4.3 *Vocal Fry*

The following paragraph explains the aims and techniques employed to realise the composition *Vocal Fry* (2015) (see digital copy of accompanying audio tracks, Track 1).

The aim of the piece is to explore the synthesis of vowels using models of the vocal tract to alter the perception of vowels through speech synthesis, and to explore the use of LPC analysis of diphthongs and to use the magnitude response of the vocal tract for convolution to create hybrid sounds in electronic music. *Vocal Fry* explores how vowels emerge from resonant structures and how resonant structures can be manipulated to create non-vowel sounds from the features of vowels.

The item was designed as a way to understand and explore the palette of sounds that the models presented can produce, in particular to study the perception of the first three formants in vowel sounds.

Three types of sounds were used for the realization of 'Vocal Fry' and were the first elements of the item to be defined.

The sounds employed can be described as follows:

- Impulses with LPC: Percussive sounds obtained through exciting the filter of an LPC analysis of a recording of diphthongs performed by a female voice speaker. The pulse trains used to excite the LPC analysis have a very low density (1.1 HZ), the silence between the pulses is computed by using the *Poisson point process* in Praat. The sound was designed with the idea of emulating the individual glottal pulses audible during the emission of vocal fry voice sources. The sound generated by this process is also be used for convolution. The aim of the technique is to extract the formant patterns for each pulse and each pulse of the sequence exciting the corresponding vowel. The convolution is performed with stereo white noise with different amplitude envelope shapes to create long textures of synthetic sounds. The white noise sounds designed for the convolution were synthesized in Csound, emulating the random variation in breath intensity during whispering in speech. The convolution is applied in Praat to combine the two sound sources.

- Vowel Deconstruction: White noise is filtered by a bandpass filter; this instrument is intended as a reduction of a parallel speech synthesizer for whispered voice. The idea is to play each of the three first formants of vowels at different times. For example, it allows a delay to the second formant of a vowel so the first and third formants disappear before the second starts playing.
- A cascade vowel synthesizer filtering a pulse train: this emulates the sound production of voiced vowels. An LFO of random numbers modulates the fundamental frequency of the voice source to approximate the variation in frequency in the vocal folds during sustained phonation.

A graphic score was realised to guide the production of the piece in the studio, along with a chart with instructions to follow to control the parameters of the three instruments and to organise the sounds generated in Csound and Praat.

A graphical sketch can provide an intuitive way to represent the organisation of the musical material in a piece over time. One can design their customised method to outline the dynamics, sound synthesis parameters and reverberation. The use of graphical sketches is one method to map the musical intentions and visualize them all at once, providing an overview of the evolution of the music material over time. The symbols included in the sketch of *Vocal Fry* were inspired by the scores, symbols and notation used by Karlheinz Stockhausen and Brian Eno. The scores of the works by Stockhausen such as *Studie I, Studie II* and *Telemusik* feature rectangular and triangular shapes.[139] This graphical notation uses intuitive geometrical shapes to represent the evolution of different parameters of sound over time. An example of graphical notation applied to music production is the one that Eno employed to

[139] Krzyzaniak, Mike "Stockhausen's Studies I and II" accessed Aug 14,2020. https://michaelkrzyzaniak.com/Research/Stockhausen_Studie_II/; Stockhausen, Karlheinz. *Telemusik: Nr.20*. Music Online: Classical Scores Library, Volume I (Wien: Universal Edition, 1969): 12-15.

represent the structure of the pieces in *Ambient 1 Music for Airports.*[140] The resulting scores, even provided on the CD cover, encapsulate in one chart the ever-evolving character of the pieces created by overlapping tape loops with different durations. This approach is also useful to plan the audio processing and to note down the details of the studio setup. One example of this approach is the graphical score in Eno's *Discreet Music*[141]*.* Figure 19 shows the graphic sketch in *Vocal Fry* is a drawing of an x-y chart in which the x axis represents time in minutes while the y axis represents the dynamics (from **pp** to **ff**). The line indicating the time-varying dynamic contour set also the amount of tension (see Figure 22).

Underneath the timeline (x axis) is provided an expansion of the chart listing the number 1),2),3) which represents the type of sounds (pulses, deconstructed vowels, vowel synthesis). Each type of sound represents an algorithm that was designed in Csound (combined with Praat for pulses). The length of the rectangles in relation to the timeline on the x-axis above them defines how long a type of instrument can be used. On the bottom pages are instructions about the use of convolution.

[140] Eno, Brian. *Ambient 1 Music for Airports*. Astralwerks. 1978. CD.

[141] Eno, Brian. *Discreet Music*. Astralwerks. 1975. CD.

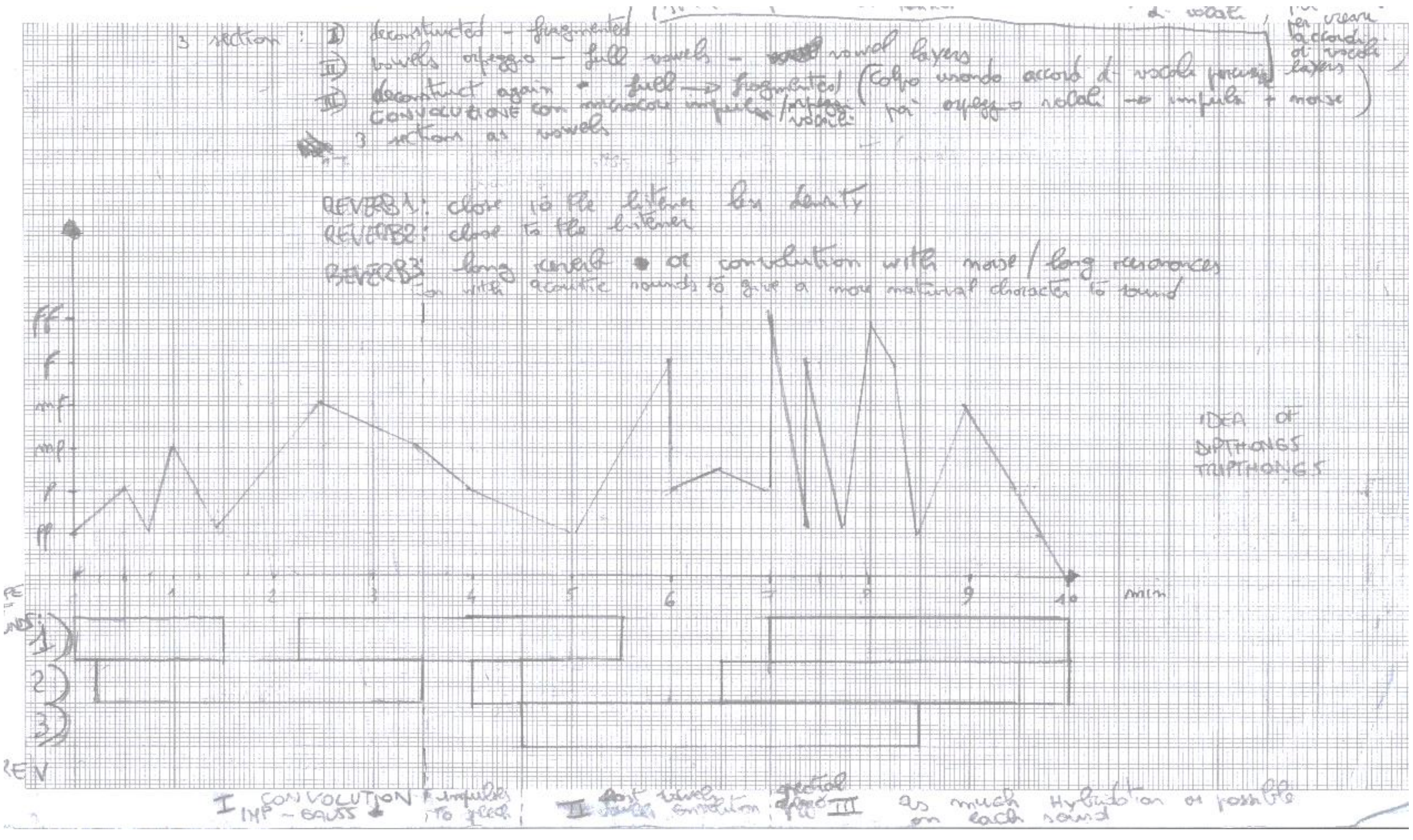

3 section : I) deconstructed – fragmented
II) vowels arpeggio – full vowels – vowel layers
III) deconstruct again • full → fragmented (Corpo usando accord di vocali presenti / poi arpeggio vocali → impulse + noise)
CONVOLUZIONE con microcosmi impulsi / arpeggio vocali
3 sections as vowels
per creare la nuvola di vocali layers
di vocali
REVERB 1: close to the listener by density
REVERB 2: close to the listener
REVERB 3: long reverb • or convolution with noise / long resonances on with acoustic sounds to give a more natural character to sound
ff
f
mf
mp
p
pp
1
2
3
4
5
6
7
8
9
10
min
FE
SOUNDS
1
2
3
REV
IDEA OF DIPTHONGS TRIPTHONGS
I CONVOLUTION IMP – GAUSS • impulses to freq.
II fast vowels / vowels evolution
spectral freeze III as much Hybridation as possible on each sound

Figure 22. Initial sketch of 'Vocal Fry'.

The score was realised at the very beginning of the piece, once the instruments and tools for sound synthesis were chosen. The idea was to imagine the complete structure of the piece on a time over tension axis, starting with quiet and sparse sound events. Whispers are quiet, vocal fry is slightly louder, phonation is increasingly louder as it mimics singing that is louder than whispers. Overall, the idea was to introduce sudden peaks in loudness with very rich spectral envelopes generated by the stratification of pitch transposition of sounds. The item is divided into three sections and its structure is inspired by the emission of the voice.

Figure 23 outlines the main elements of the piece.

Section 1	Section 2	Section3
Breath/whisper+ vocal fry	Vowels	Breath/whisper+ Vowels+ LPC pulses
Isolated formants + LPC pulses	Klatt Cascade synth + Lpc Pulses	All 3 types
0 min – 3.30 min	3.30 min – 7 min	7 min – 10 min

Figure 23. The structure of *Vocal Fry*.

The score does not define the specific duration of each sound object but sets a framework to operate in the studio and rules to follow during the coding of the synthesized element in Csound and Praat. The resulting events are then organised and mixed in a digital audio workstation (Nuendo) to follow as much as possible the overall contour of tension and the number of Csound instruments that are active and playing back, according to the graphic score. This score was used in combination with a set of instructions to organise the design of the sounds and guide the creative process. The sketch in Figure 22 is aimed at the author to organize the synthesis of the material produced in the recording studio and provides an overview of the full structure to assist the arrangement of the final item.

Keane describes a way to control the parameters available in tape music to develop musical ideas and inspired the system developed for *Vocal Fry* to control the amount of tension according to the score's contour.[142] For instance, Figure 24 shows a list of specific actions to perform over the element of the piece while following the tension contour on the graph.

- Tension	+ Tension
Longer duration of events	Shorter duration of events
Narrow pitch intervals	Wide pitch intervals
To MID vowels	To Front vowels
Pp and ***mp*** dynamics	***Mf*** , ***f*** and ***ff*** dynamics
+ reverberation	- reverberation
- density of events	+ density of events
- Convolution stereo white noise and hybrid sounds	-+Convolution stereo white noise and hybrid sounds

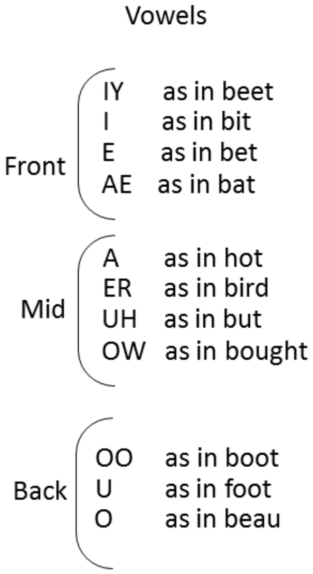
Figure 24. Instruction for *Vocal Fry.*

The synthesizers employed in *Vocal Fry* need as input the formant frequencies of the charts shown in Dodge and Jerse.[143] Figure 25 outlines the classification of vowels in back, mid and front, it only features the vowel sounds that will be generated by the formant synthesizer. The figure provides an intuitive way to group vowel sounds that can be useful in order to design sequences of vowel sounds and transitions by moving from one group to the other. A change in parameters from the group back to front is

[142] Keane, David, *Tape Music Composition.* (London: Oxford University Press, 1980), 26-27-28-29.

[143] Dodge and Jerse, *Computer music: Synthesis, composition, and performance,*223-224.

intended to create more tension than a change from a back vowel to a mid-vowel. The grouped chart of vowels back, mid and front allows one to explore parameters that we are not always able to perceive. For example, front vowels have in general higher 3rd formant frequencies–and this feature can be used to introduce tension when sequencing a series of vowels. Figure 26 includes a list of the formant frequencies of the different vowels described in the back, mid and front groups

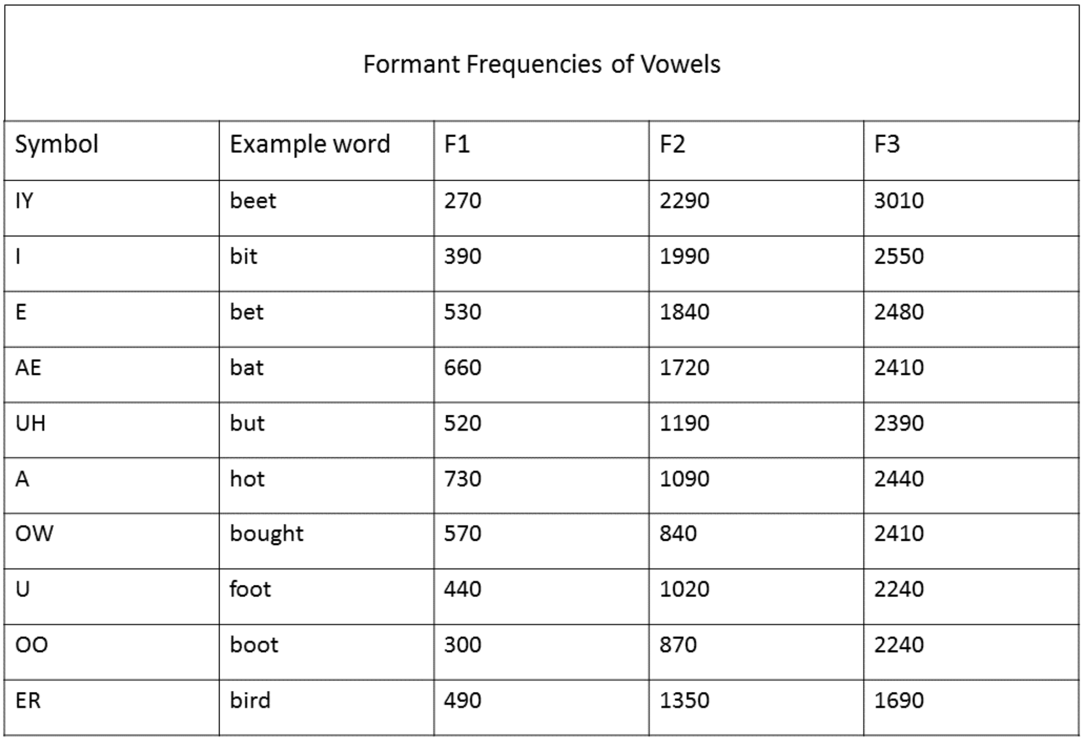


Vowels

Front	IY	as in beet
	I	as in bit
	E	as in bet
	AE	as in bat
Mid	A	as in hot
	ER	as in bird
	UH	as in but
	OW	as in bought
Back	OO	as in boot
	U	as in foot
	O	as in beau

Figure 25. Particular of back, mid and front vowel groups from American English. After Dodge and Jerse.[144]

[144] Dodge and Jerse, Computer music: Synthesis, composition, and performance, 223.



Formant Frequencies of Vowels

Symbol	Example word	F1	F2	F3
IY	beet	270	2290	3010
I	bit	390	1990	2550
E	bet	530	1840	2480
AE	bat	660	1720	2410
UH	but	520	1190	2390
A	hot	730	1090	2440
OW	bought	570	840	2410
U	foot	440	1020	2240
OO	boot	300	870	2240
ER	bird	490	1350	1690

Figure 26. Charts of the Vowels' formants used for the Klatt synthesizer and to isolate a single formant resonance. After Dodge and Jerse.[145]

The formant frequencies shown in Figure 26 were used as input to a cascade Klatt synthesizer to synthesize vowels and of a parallel formant synthesizer to isolate the individual resonances. This is intended particularly for the 'deconstructed vowel' type of sounds, to play each formant frequency at different times or delaying one or two of these instead of appearing all at once, to explore the perception and isolation of formant resonances as described in section 4.1. Playing the formant of the same vowels at different times can be used to mix one vowel with the next vowel. This allows for the mixing of the formants of different vowels to create tension or smoother transitions by sustaining the formants in common or highlighting the closest ones. The opposite would be moving to a vowel with no formant in common to create tension. Also, it enables the user to explore panning as each formant can be moved around the stereo image instead of being fused in a single vowel sound entity. The duration, dynamics, amplitude envelope, panning, pitch and density are controlled in Csound while convolution is performed in Praat. Also pitch

[145] Ibid., 224.

transpositions inspired by tape music are used on LPC pulses to create tension through variations in pitch. The feature of this type of transposition is that transposition in pitch also affects the duration of the sound processed; this allows one to layer sounds that have different timing, revealing new random variations and blending of sounds. This also causes transpositions in formant frequencies when transposition in pitch is applied to the hybrid sounds as a result of the convolution process. The pitch transpositions also cause accellerando and rallentando effects, along with lowering or rising of the spectral content of the sound. This also provides a means to experiment with layering techniques in multi-track mixing to explore the use of vowels (in this mixing technique) for sound design applications. Artificial reverberation has been applied to the sounds produced in Csound and Praat, the approach included the creation of groups for the audio effects bus in Nuendo. The reverberation is used to give the impression of distance from the listener: close (dry sound predominant around 80%, short reverb around 0.8 seconds); it aims to mimic a small room with the source close to the listener and is meant to provide a more natural tail to the percussive fry pulses. Medium distance (dry sound predominant around 60%, 1.6 reverb time) mimics a medium size room that allows the source to be positioned further away from the listener. Large distance (equal 50% dry/wet signal, 7 seconds reverb time) is inspired by the impulse response of a large church. This arrangement of parameters reveals that with the distance from the source it is not only the wet signal that is higher in the mix but also the size of the room increases, so creating variations of the reverberation on different sections of the piece. Furthermore, reverberation is a manifestation of resonance that can be explored to assist the transformation from fry to phonation/subharmonics, as it blurs the rhythmic patterns of strata of pulses in a subtle way to emerge from textures generated from convolution.

Several stereo tracks are routed to a processor for artificial reverberation to produce the effect of each distance by sending them to one of the three groups. This allows layering of several sounds as well as control over the type of reverberation of each layer. This approach was inspired by layering techniques used in sound design of Foley sound sources and techniques used to mix kick drums in dance music.

The main goal of the piece was to design a transition from unvoiced sources (whispering) to vocal fry, and from vocal fry to phonation in vowel synthesis. Also, the concept is to hide the perception of speech-like sounds while at the same time using models meant to synthesize speech. This was achieved by starting with isolating the formant of the different vowels and exploring different combinations and transitions by using slow amplitude attack envelopes on the unvoiced source. Individual LPC pulses have been sampled in Csound and played back with different speed and panning parameters to emulate the glottal pulses of the vocal fry register. Convolution of LPC pulses with white noise is then introduced to perform the transitions between isolated formants and the sound of vowel textures. The LPC pulses gradually increase in density to mimic the acceleration of the vocal fold vibrations in transitioning from fry to voiced source. The synthesis of vowels is then introduced to complete the effect of this transformation. The voiced sounds between minutes 5.00 and 6.00 include the synthesis of subharmonics and are intended as a way to synthesize throat singing by applying the techniques described by Whishart, via the design of synthesizers and audio processing.[146] The imitation of the acoustic features of the vocal folds helped control this transformation. The voice is not always perceived as such, but its features control the musical material. The stratification of texture, generated with LPC pulses and through convolution, allowed exploration of the resonant nature of vowels and produced a type of sound that would be associated with cymbals or other metallic resonant objects struck with drumsticks. This is particularly noticeable in the section after 6 minutes into the piece, where percussive sounds and long textures with a wide range of frequencies have been generated only by using the techniques explained in 4.1 and 4.2 in combination with pitch transpositions. The imitation of the acoustic features of the vocal folds helped control this transformation. Csound has been used to create layers of fry pulses from different vowels and transposed at different speeds. This creates a single percussive sound that includes the manipulated features of several vowels. When convolved with long samples of stereo white noise metallic resonances and non-voice features

[146] Wishart, *On sonic art*, 265.

emerge from the layer of glottal pulses and are time stretched through the use of convolution.

Vocal Fry helped to provide an understanding of how to hide or reveal the perception of speech and what type of control the user can achieve with formant synthesizers; also, it shows how vowels can be combined with convolution to design sounds that are not perceivable as a human voice. The use of Csound and Praat to layer the resonances of speech extracted through the synthesis of vocal fry sources allowed an exploration of convolution in Praat to design a smooth transition from from whispering, to increasingly denser strata of vocal fry, to clouds that fuse and blur layers of formant frequencies from the sequences of synthesized glottal pulses. The use of convolution in combination with vocal fry synthesis is the starting point for the design of an audio processor that is able to combine many of these techniques at once and provide the user with the ability to synthesize textures from speech. This use of vocal fry pulses can be similar to the techniques employed by Roads in composing with pulsar synthesis. The idea of vowel deconstruction prompted the exploration of the properties of formant variation over time, altering the perception of sound sources. The concepts explored in this chapter were the foundations of an approach to the extraction and elaboration of the sound colour of speech, starting with vowels.

The techniques discovered and designed during the production of *Vocal Fry* prompted an exploration and implementation of fry textures, in combination with Prony's method. This prompts the design of a tool that summarises the techniques described in this chapter and makes them readily available to the user. The techniques used in *Vocal Fry* highlighted a collection of features necessary to implement the different processing techniques in one algorithm. For example, Appendix V provides a tool that includes the following features:

- Ability to upload any audio file for audio analysis
- Includes a frame-by-frame analysis and re-synthesis algorithm in order to represent variation in formant frequencies over time.
- Ability to upload any source of the filter. For example, vocal fry sources.

- Offers the option to perform convolution with stereo white noise.
- Saves the processed signal to an audio file.

The instrument is provided in Appendix V in form of Matlab code. It allows us to explore the synthesis of speech-driven textures based on Prony’s method and convolution.

Chapter 5

5.1 From consonants to vowels: vocal tract feedback

This chapter describes an approach to sound synthesis which transfers the resonant structures of vowel sounds to the recordings of unvoiced speech sounds. The idea is to use the possibilities offered by computer models of the vocal tract to perform transformation on speech sounds that would not be possible for the vocalist to produce acoustically.

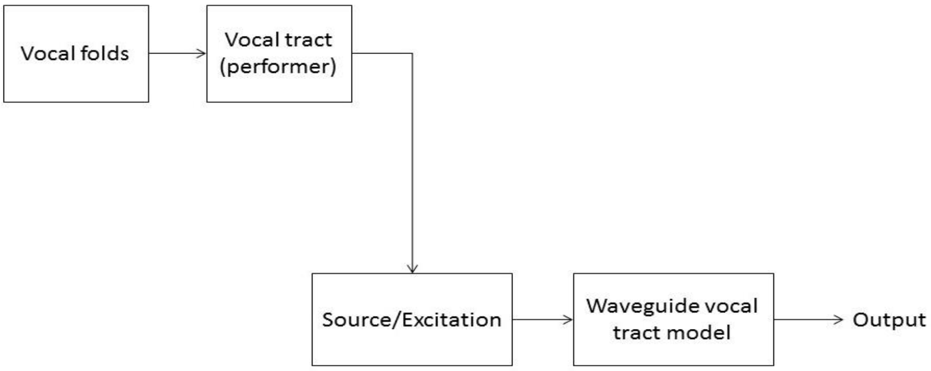


Figure 27. Idea of the output of vocal tract feeding back as voice source.

The specific goal is to study the use of digital waveguides to build a vocal tract model to approximate and transfer the qualities of vowels to recorded non-vowel speech sounds. This affects the performance of the speaker, who is not singing or delivering intelligible speech but generating the source material for sound synthesis. The parameters of the vocal tract model can be explored to control the position of resonances to create effects inspired by chorusing audio processors. The aim is to alter the perception of recorded consonant sounds through sound synthesis and explore the possibility of a hybrid vocal tract model which employs an acoustic recorded source and a computer model of the filter. It is a way to apply a feed-back process to the source-filter model of speech production.

The use of speech technologies such as digital waveguides will be explored to study the perception of unvoiced consonants such as /t/, /s/, /f/ in order to morph them into vowel sounds. The approach shown in Figure 27 can be implemented by recording speech sounds and using the resulting audio files as voice source of a digital waveguide model of the vocal tract that is able to transfer the formant frequencies of vowels to the recorded speech. In this way it is possible to explore the perception of non-vowel speech sounds and develop techniques to control physical models of the vocal tract to alter the perception of speech. Furthermore, audio editing techniques can be used as a creative tool to shape recorded sounds in order to mimic the features of voice source, whether this is voiced, unvoiced, mixed or vocal fry, and to design different effects and transitions in a musical context. The choice of manipulating the input source through audio editing is a result of experimentation with the resulting vocal tract model. It is something that happened alongside the creation of a vocal tract model to widen its sonic palette. The sound of recorded fricative consonants used as input of a vocal tract model might cause a combination of the resonance of the acoustic vocal tract with resonances of the physical model. However, this is an intentional choice of the author with the aim of exploring the perception and manipulation of speech sounds, recorded and performed, to mimic electronic sources used in speech synthesis. The resulting collection of sounds display a rich spectrum, even after the emission from the physical vocal tract, and are suitable as sources to excite the computer-modelled vocal tract. For example, the synthetic voice sources imitated through performance are a single glottal pulse and white noise.

The goal is to examine how computer models of the vocal tract enable the design of vowel-like sounds from consonants, and to investigate how technology provides an expansion to the performer's vocal tract by allowing the use of extreme parameters that go beyond the constraints of the human body. The objective is to explore technology as an extension of the performer’s vocal tract.

5.2 The filter

As explained in Chapter Two, a vowel sound can be defined and associated with a particular set of formant frequencies. One way to model vowel resonance is to use a tube resonator of length *l* with a uniform cross-section and one end closed.[147] The closed end consists of a vibrating membrane that simulates the action of the vocal

[147] Kent and Read, A*coustic analysis of speech*, 19.

folds. The resulting apparatus is able to produce a set of resonances that have a uniform spacing and are dependent on the length *l;* the longer the acoustic tube the lower the frequency of formants. To produce a variety of vowel sounds a two-tube resonator can be used; this employs the variation in the cross-sectional area between the tubes to approximate the behaviour of the oral cavities during the production of different formant patterns.

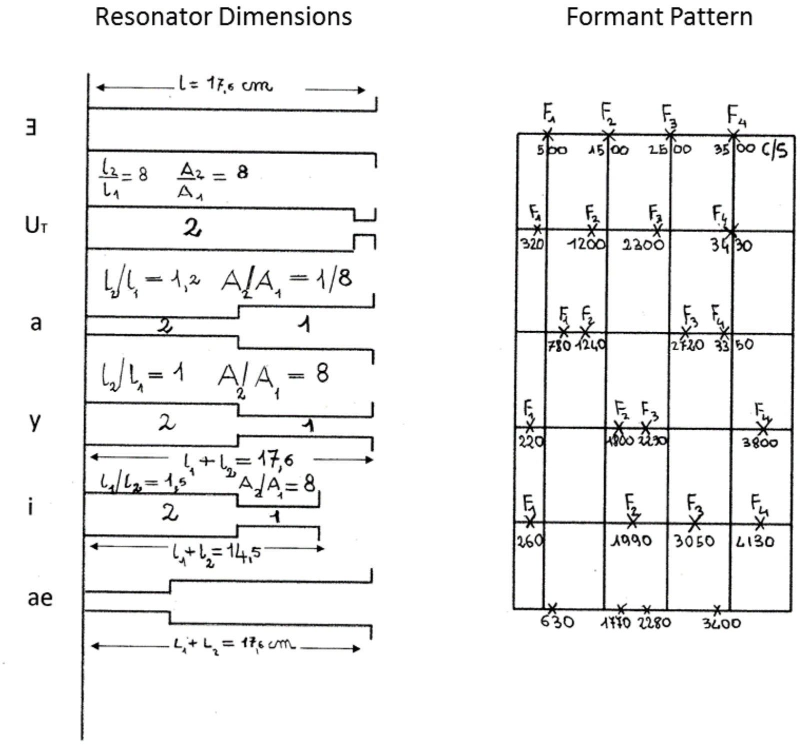


Figure 28. Twin-tube resonator provides formant patterns similar to vowel sounds. After Fant.[148]

Fant provides details about a resonator consisting of two tubes of two different lengths and areas.[149] By varying the ratio between their length and areas according to the values shown in Figure 28 it is possible to produce the formant pattern of different vowels.

[148] Fant, *Acoustic theory of speech production: With calculations based on X-ray studies of Russian articulations*, 66

[149] Ibid., 66

The way sound propagates though a tube resonator can be approximated using waveguide models. A waveguide provides a structure for the propagation of waves and is used in musical applications to model the body of musical instruments. An example of the application of a waveguide is to model how the cylindrical body of an instrument makes the sound travel from one end of the tube to the other[150]. The time that the sound takes to travel through the resonator can be modelled with the use of a delay line. The duration of the delay time is given by the length of the tube and by the speed of sound. A one-dimensional model of a tube can be defined as a bi-directional delay line that contains soundwaves travelling forward and backwards. A way to simulate the acoustic features of the vocal tract in the time-domain is through a one-dimensional model (1D) consisting of a concatenation of cylindrical tubes; the differences in cross sectional areas between the tubes cause a mismatch in impedance.[151]

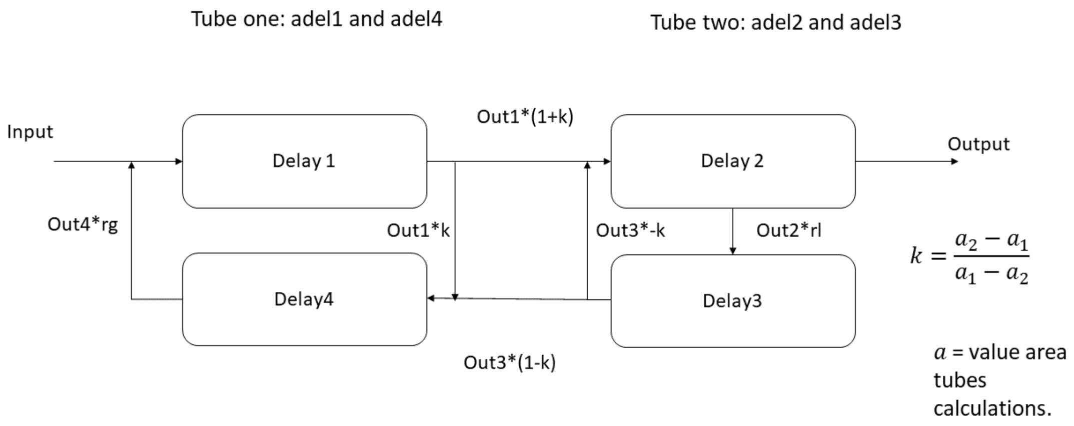
A waveguide model can be built to approximate the production of a formant pattern of a two-tube resonator by connecting two waveguide tubes in cascade.

Figure 29 shows how delay lines can be used to model the behaviour of two tube resonators connected in cascade to approximate the resonant structures of a two-tube vocal tract resonator.

The difference in cross-sectional area between the two tubes causes some of the sound waves to be reflected back to the input of the model. Delay 1 and Delay 4 define the length of the first tube (on the left side is the voice source or input) and Delay 2 and Delay 3 define the length of the second resonator (on the right side the lips).

[150] Dodge and Jerse, *Computer music: synthesis, composition, and performance,* 280.

[151] J. Mullen, D.M. Howard and D.T. Murphy, "Waveguide Physical Modeling of Vocal Tract Acoustics: Flexible Formant Bandwidth Control from Increased Model Dimensionality," IEEE Transactions on Audio, Speech and Language Processing, Vol. 14, no.3 (2006): 966-967.


Tube one: adel1 and adel4
Tube two: adel2 and adel3
Input
Out1*(1+k)
Delay 1
Delay 2
Output
Out4*rg
Out1*k
Out3*-k
Out2*rl
$k = \frac{a_2 - a_1}{a_1 - a_2}$
Delay4
Delay3
Out3*(1-k)
a = value area tubes calculations.

The length of the tube in centimetres is converted into delay time in samples inside the model to achieve the vowel formant patterns.

Figure 29. Adapted from Dodge and Jerse:[152] Scheme of a digital simulation of the interconnection of two waveguides.

The model is based on the schematics and calculation provided by Dodge and Jerse and is designed to be controlled by using the ratio values in length and area shown in Figure 28.[153] Increasing the time of the delay will result in a longer tube (and the opposite) affecting the length *l1* or *l2* while the scattering coefficients *k* is calculated from the difference in area of the two tubes. Delay 1 and Delay 2 conduct the forward travelling waves while Delay 3 and Delay 4 conduct the backward travelling waves. The scattering coefficients model the changes in cross-sectional area between the tubes and affect how the signal travelling through the waveguide model is reflected back to the input.

As in this model, the areas of the two tubes is known as the reflection coefficient *k* and can be calculated by adapting the following formula:[154]

$$k = \frac{A_2 - A_1}{A_2 - A_1}$$

(9)

[152] Dodge and Jerse, *Computer music: Synthesis, composition, and performance*, 280-281.
[153] Ibid., 281.
[154] J. Mullen, D.M. Howard and D.T. Murphy, *Waveguide Physical Modeling of Vocal Tract Acoustics: Flexible Formant Bandwidth Control from Increased Model Dimensionality,* IEEE Transactions on Audio, Speech and Language Processing, Vol. 14, no.3 (2006): 967.

The difference in area between the tubes affects the amount of sound reflected or absorbed by the vocal tract; it also models the effect of the difference in area in the physiology of the acoustic vocal tract.

The coefficient r_g models the reflection of sounds from the glottis and r_l models reflections of sounds from the lips. The model is represented by the following equations where P_1^+ represents the pressure of the wave travelling forward while P_1^- represents the backward travelling wave:

$$P_1^- = kP_1^+ + (1-k)P_2^- + P_1^- r_g$$

$$P_2^+ = (1+k)P_1^+ - kP_2^- + P_2^+ r_l$$

(10)

The schematics are now ready to be implemented in the computer by using digital delay lines.

There exist examples of delay line structures implemented in Matlab that have been re-organised and adapted to include in the model the use of four delay lines and a scattering junction.[155] The following code shows how the junction between two tubes has been implemented, the text following the symbol ‘%’ are comments:

```
adel1=[imp(n)+rg*(out4(n));adel1(1:N-1)];
%Delay 1    Tube1 forward  waves

adel2=[(out(n)*(1+k))+(out3(n)*-k);adel2(1:N2-1)];

%Delay 2    Tube2 forward waves

adel3=[out2(n)*rl;adel3(1:N2-1)];
%Delay 3    Tube2 backward waves

adel4=[(out3(n)*(1-k))+(out(n)*k);adel4(1:N-1)];

%Delay 4    Tube 1 backward waves
```

[155] P. Dutilleux and U.Zölzer, “Delays” in *DAFX: Digital Audio Effects.* Ed. Udo Zölzer, (Chichester: Wiley, 2002):64.

This extract shows how the schematics discussed above can be implemented as feedback variables within *adel1*, *adel2*, *adel3* and *adel4*. By using the ratio shown in Figure 28, it is possible to calculate the length and area values that can be used as input of the waveguide to synthesize vowel sounds.

Vowel	Length tube 1 in cm	Length tube 2 in cm	Area tube 1 In cm^2	Area tube 2 In cm^2
U_T	15.64	1.96	8	1
a	9.6	8	0.25	1
y	8.8	8.8	8	1
i	8.7	5.8	8	1
ae	4.4	13.2	0.25	1

Figure 30. Ratios of lengths and area to produce vowel formant patterns after Fant.[156] (Refer to Example 2.1).

The following formula can be used to calculate the frequency of the first formant from the length of the tube[157]:

$$F_1 = \frac{c}{4l}$$

(11)

F_1 is the first formant, c is the speed of sound (around 350 m/s) and *l* is the length of the tube. This formula can be implemented within the waveguide model to convert the length of the tubes to delay times for the sound synthesis. A collection of vowels can be synthesized by using the parameters shown in Figure 30 in the two-tube vocal tract model. When these parameters are used the two-tube model produces formant

[156] Fant, Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations, 66.

[157] Kent and Read, Acoustic analysis of speech, 19.

patterns that are close to formant patterns of a collection of vowels calculated by Fant (as shown in Figure 28).[158]

A feature of a waveguide vocal tract model is that it can be easily controlled by its physical descriptors, and the features of the filters are affected by the shape of the model. An example of sound synthesis based on physical descriptors can be found in the SPASM system that allows control of the shape of the vocal tract to change the features of the filter.[159] This suggests the possibility of exploring shapes and physical descriptors that are pushed beyond the accurate physiology of vocal tract, and to study how this can be used to perform transformations over recorded consonants. See Appendix III for a full code transcription with details about implementation of formulas and adaptation to control the four delay lines in Matlab.

5.3 The source

The waveguide model needs a sound source as input to produce the desired formant patterns and to implement the vocal tract feedback approach discussed in Figure 27. There are speech sounds that don't involve the vibration of the vocal folds in their production, but once recorded these can be manipulated through audio editing to mimic the vibration of the vocal folds. White noise sources can be imitated by stressing the whispered (or unvoiced) sound of consonants, in particular stop consonants like *t* and fricatives like *f* and *s* sounds. In fricatives noise is produced, as a result of a narrow constriction within the vocal tract, that generates turbulence.[160] The resulting sound has a wide range of frequency content that is ideal to be subject to filtering techniques. Using recordings of fricatives, (like long and sustained /f/ and /s/ sounds,) as input for the waveguide vocal tract model can transfer the resonant structures of vowels to consonants producing a sound close to whispered vowels. A recording of these sounds is available with Example 2.2.

[158] Fant, Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations, 66.

[159] Cook, Perry R.,” SPASM, a Real-Time Vocal Tract Physical Model Controller; and Singer, the Companion Software Synthesis System”, Computer Music Journal, Vol. 17, no. 1 (1993): 30-31.

[160] Kent and Read, *Acoustic analysis of speech*, 160.

Stops (or stop consonants) can be used to produce pulsed sounds by creating impulses with the glottis or the tongue and can sometimes involve the action of lungs to produce a variety of short sounds.[161] By producing very short bursts of noise and stopping it with the tongue, as when stressing the *t* while whispering the word '*ten*', it is possible to mimic an individual glottal pulse. Refer to Example 2.3.

Sounds can be artificially prolonged to introduce continuity by cutting segments of a recorded sound and overlapping them by using a technique named brassage.[162] This is performed through audio editing of a section of an audio sample and allows, for instance, time-warping of an audio sample by dividing the sound in segments and re-arranging these in time. A similar process can be applied to achieve continuity from short bursts of noise in consonants. Through audio editing it is possible to repeat the recorded sound of an unvoiced *t* several times per second (frequency) achieving a pitched sound similar to the vibration of the vocal folds. Brassage allows one to build a periodic signal from a single percussive sound, in a similar way that many repetitions per second of a glottal pulse produce a pitched voice source. Figure 31 shows an example of a pulse train created with this technique from recorded /t/ sounds.

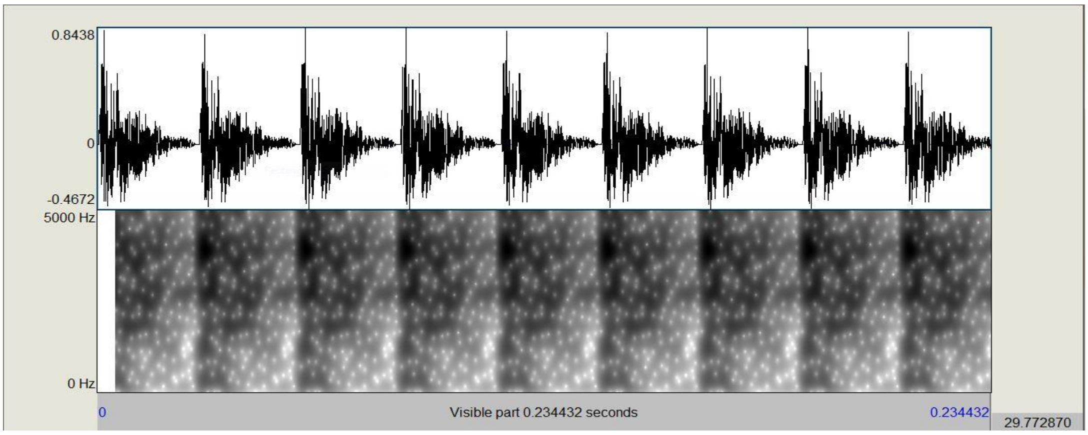


Figure 31. Example of recorded t sounds repeated several times per second to achieve a steady pulse train sound resembling phonation. (Refer Example 2.4).

[161] Wishart, *On sonic art*, 276.

[162] Wishart, *Audible Design*, 53.

The result of this application of brassage techniques to audio editing can be used to control the continuity of the voice source as input of a waveguide vocal tract. The clicks and pulses produced by a performer are now used as 'vocal folds' of another vocal tract model implementing the vocal tract feedback approach. The delay lines that are part of the model have themselves a feedback parameter that can be controlled and, by increasing the feedback coefficient, it is possible to increase the number of times the voice source passes through the waveguide model, thereby exaggerating the resonant structures of vowels even further.

Audio editing can be used to produce stuttering rhythmic effects, for example by slicing a sustained sound in segments and introducing silence between the segments. By altering the distance between the different slices of sounds it is possible to control the frequency of the effect or to blur the perception of the original sound.[163] This suggests that the effect produced by increasing and reducing the distance between percussive /t/ sounds can blur the perception of consonants; it prompts the study of how a single sound can become vocal fry and then pulse train through repetition, providing control over audio editing. Using this technique, vocal fry voice sources can be created by repeating the event at a lower frequency and by introducing irregular silence between the individual repetitions of *t* to mimic Praat's *Poisson point process*.

5.4 Fricatives

This section describes the creative process involved in the creation of the item *Fricatives* (2016) (see Digital copy of accompanying Audio Tracks, Track 2). The item is an original composition realised to explore the possibilities of the two-tube waveguide vocal tract model described in section 5.2. The aim of the piece is to explore how the resonant structures of vowels can be transferred to recordings of consonants and, in particular, how the perception of fricative and vowel sounds can be altered by controlling the parameters of the vocal tract model. *Fricatives*

[163] Jem Godfrey. "Creative Sound Design for Music", Sound on Sound, 2011, accessed Sept 14, 2018, https://www.soundonsound.com/techniques/creative-sound-design-music

investigates how knowledge of the physiology of the human voice can inform brassage and audio editing techniques to enable the use of recorded speech sources as excitation of physical models of the vocal tract (as an alternative for synthetic voice sources). The goal is to combine brassage and physical modelling in a single process, to alter the perception of consonants and design transitions that gradually move from consonant-like sounds to vowel-like sounds.

The synthesis of the different sounds of the item were entirely realised in Matlab by using the two-tube vocal tract waveguide model. The ratios of tube length and areas shown in Figure 28 and Figure 30 were used to control the synthesis of vowel like sounds. However, the length of the tubes and size of area were modified freely by using extreme parameter settings. The average length of the human vocal tract is around 17.5 cm; the flexibility of waveguide models allows us to model vocal tracts with extreme parameters that are able to go further than the constraints of the human body. For example, in some sounds a vocal tract length of more than 60cm has been used.

Feedback factors have also been used to push the resonance exaggeration of formant structure to the maximum allowed before the threshold of audio clipping. This is achieved by setting, for example, the glottis reflection coefficient (r_g) and the lips reflection coefficients (r_l) to values around 0.999.

The author performed and recorded the sound sources in the studio with the mindset of collecting noise sources, pulses and clicks from speech. The process involved the recording of speech sounds in mono, close to a single microphone. The choice of material was affected by the result of several attempts and tests: once recorded the sounds needed to be tested and filtered with the two-tube vocal tract. Initially all consonant sounds were recorded both voiced and unvoiced. Unvoiced sources provided a richer and unpitched sound source; this was in order to hide the quality of speech with more flexibility and to transform them into vowels through the use of waveguides and editing techniques. The source sounds used for the creation of the piece is a collection of recorded long unvoiced /f/ and unvoiced /s/ sounds with different dynamics. Fricative sounds are used to mimic whispered vowels or to create long sounds with highly resonant qualities. Long /s/ and /f/ sounds have been recorded with crescendo and diminuendo dynamics to introduce interest and explore how variation in the intensity of fricatives interact with the vocal tract model to explore the interaction of recorded sound with waveguide models.

Unvoiced /t/ sounds have been recorded to produce sounds that have as short an attack as possible to exaggerate the noise-like features of this particular consonant in whispered speech. This sound was recorded with the idea of building a collection of /t/ samples with different dynamics, used to design vocal fry and voiced sources through audio editing and brassage techniques. The recorded /t/ sounds mimic a glottal pulse that can be duplicated and edited in a collage of sounds that repeat several times per second. The recording of a collection of short unvoiced /t/ sounds are used to build vocal fry and voiced sources by using brassage techniques.

The structure of the item is organised in three sections and revolves around the transformations and transition techniques used in the second section that is halfway through the duration of the entire item which is a total of five minutes.

Figure 32 describes the elements in each part of the different sections of the item. Three main ideas have been used to design the transition and structure of the piece:

- To alter the perception of fricative sounds. These are not recognisable as recordings of speech; they are gradually morphed into vowels and then revealed as consonants.
- To alter the perception of stop consonant's *t* sounds, controlling their density to mimic vocal fry and then become a steady pitch.
- The idea of several elements morphing over time to create a single transition. Three changes in parameters appear: from rhythm to pitch, from glottal pulses to fry and then phonation, from consonants to vowels.

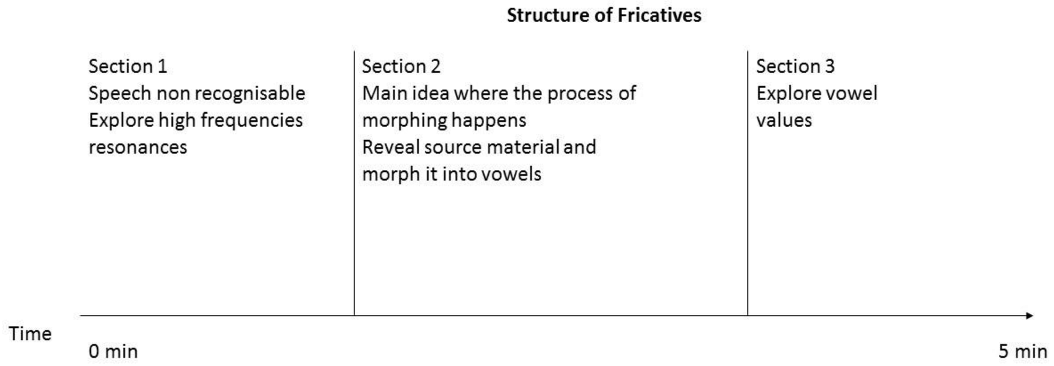

Structure of Fricatives
Section 1
Speech non recognisable
Explore high frequencies resonances
Section 2
Main idea where the process of morphing happens
Reveal source material and morph it into vowels
Section 3
Explore vowel values
Time
0 min
5 min

Figure 32. The different sections of *Fricatives*.

The structure described above has been used as a framework to generate the material by interacting and experimenting with the parameters of the model; this enabled the exploration of the workflow. This workflow is influenced by the way the Matlab code performs and renders the synthesis of the different sound objects. The process was largely inspired by multi-track drum sample layering techniques.[164] These techniques are used in music production to blend the features of different drum sounds and to provide control over the frequency content to generate a cohesive single new tone from the mix of different layers of sound. This suggests the application of layering techniques, generating sounds with different features by filtering recorded sound with extreme parameter values of the two-tube model, such as the lips and glottis reflection coefficients and length of the vocal tract. This allows the user to dramatically alter the qualities of speech sounds and create interest in the overall arrangement of the piece. The resulting textural elements blur the perception of unvoiced consonants and by gradually decreasing the parameter values to match vowel production, the resonant structures of vowels emerge from the transformation of non-vowel sources.

[164] Eddie Bazil. “Layers of Complexity”, Sound on Sound, 2012, accessed Sept 14, 2018, https://www.soundonsound.com/techniques/layers-complexity.

The idea is to render longer mono sounds synthesized from Matlab and then import them in Nuendo to perform audio editing (if required), controlling the panning in the stereo image and adding reverberation.

To better blend the variety of sounds synthesized from the model through brassage, a patch in a Nuendo session has been designed to maintain (since the beginning of the creative process) more cohesion; this is provided by a division into three groups. This technique expands the approach used for the mixing and layering process described in Chapter 4 for the item *Vocal Fry*:

- Close: sounds that are close to the listener. Artificial reverberation is used with a very short time (0.1sec) and an 80% dry signal, 20% wet with a pre-delay setting of 11ms.
- Mid: sounds that appear to be slightly farther away from the listener. 0.8 sec reverberation time. 60% dry signal, 40% wet with a longer pre-delay setting of 27ms.
- Far: sounds that appear very far away from the listener. 7 sec reverberation time with 50% dry and 50% wet signal.

The choice of these parameters aims to exaggerate the effect of distance from the sound source by increasing the size of the room as well as the amount of wet signal in the mix.

Each group (close, mid, far) with the following panning arrangements:

- Centre tracks (fixed)
- Halfway right track (fixed)
- Halfway left track (fixed)
- Automated movements right
- Automated movements left

For each group of panning at least one exact copy was made to provide a means to perform precise overlapping editing techniques. The tracks are routed through sends to the three reverberation groups.

Once a sound is synthesized and rendered from Matlab the resulting mono sound can be imported in any of the panning position tracks of any of the distance groups.

This arrangement allows one to reduce the number of tracks needed to organise the material, and the number of artificial reverberation units employed, providing more cohesion in the depth of the mixed elements. The use of routing to feed a collection of tracks to three different artificial reverberation processors provides more flexibility; this enables a more exaggerated effect of reverberation in certain sections than the use of a single device for all tracks of the piece. At the same time, organising groups of sounds treated with three different types of reverberation provides a way to blend groups of sound sources. The parameters of the reverberation were designed to give the impression of three different distances from the sound source. This allows control over time on the amount of reverberation on sound sources.

Figure 33 shows how the source-filter model has been used to plan the transformation of the source separately from the transformations over the filter (vocal tract lengths and area of the two tubes). The figure shows the development of the musical material over time. It also presents how the filter and the source are manipulated independently. The common goal is to design a single transition from consonant to vowel during the arrangement. The x axis represents the time and total duration in minutes of *Fricatives* in order to display its division in three sections.

In Section 1 the musical material is not recognisable as speech, the filter consists of layers of sounds generated with extreme vocal tract length parameters and higher lips and glottis feedback coefficients; the source consists of a collage of /f/ and /s/ sounds. The goal here is to hide the presence of the human voice, the effect is to create noise and metallic resonance and use unnatural shapes of the vocal tract beyond the constraints of the physical body of a performer. It focuses on the use of sustained sounds that explore the design of textures of resonances from a collection of long fricatives.

In section 2 the physical model gradually shrinks in size to match the average vocal tract length of 17.6 cm; in this way the resonant features of vowels gradually emerge from the frequency response of the vocal tract. Unprocessed collages of /t/ sounds reveal the original speech sound source and filtering is gradually applied to the collages as the density of sound increases. This assists the transition from vocal fry source to voiced source and at the same time aids the transition from consonant to vowel. In section 3 a collection of whispered vowels is synthesized by exploring the parameters as in Figure 25 that model the vocal tract's shape of different vowel sounds. This is achieved by using a collage of long /s/ fricative sources. *Fricatives* ends with the playback of the unprocessed /s/ fricative source. Refer to Example 2.5.

An IIR comb filter is a type of filter that displays a uniform spacing between the peaks of its frequency response[165]. Similar to a waveguide model, an IIR comb filter feeds back part of its output, the amount of feedback also affects the level of accentuation of the resonant qualities of the filter. Moore explains how a comb filter can be used to morph rough into smooth, as the resonance increases in sounds with a wide range of frequency content; he describes comb filters as colouring filters.[166] Given the nature of the waveguide algorithm based on the FIR comb filter and used in the vocal tract model, it is possible to explore the latter as an expanded comb filter. Such a filter can colour an input sound source with the timbre of vowels. An example of this technique can be heard in the Section 3 of *Fricatives* at 4'05''. Also, when using the same diameter for each of the two tubes, the model produces uniformly spaced formant patterns that are identical to a comb filter. When r_g and r_l provides a high reflection feedback coefficient it is possible to explore the high-frequency content of the fricatives used as voice source; this is achieved by dramatically reducing the area of one of the two tubes while using a stretched vocal tract length such as 50cm or even 70cm. (Refer to Example 2.6). The resonance provided by the feedback element in the schematics provides more continuity to the random character of fricative sounds; by overlaying several sounds with such resonant features it is also possible to increase or decrease tension within the arrangement of the musical material.

[165] Roads, *The computer music tutorial*, 417.

[166] Moore, *Sonic Art: An introduction to Electroacoustic Music Composition*, 97.

Layering can be used to create a chorusing effect by applying slightly different tube lengths to filter the same sound. This can be seen as the individual differences in vocal tract length and size between two singers singing the same note.

More layers can be used to create changes in dynamics and introduce interest by overlaying several rendered notes. Each rendered sound has a difference in its parameters leading to an effect similar to chorusing between the formants. This technique is the main element of the Section 1 of *Fricatives* and provides a way to experiment with formants to mimic existing audio effects. The technique shown in Figure 34 has been used to explore this concept and hide the perception of fricative sounds as recorded from human speech.

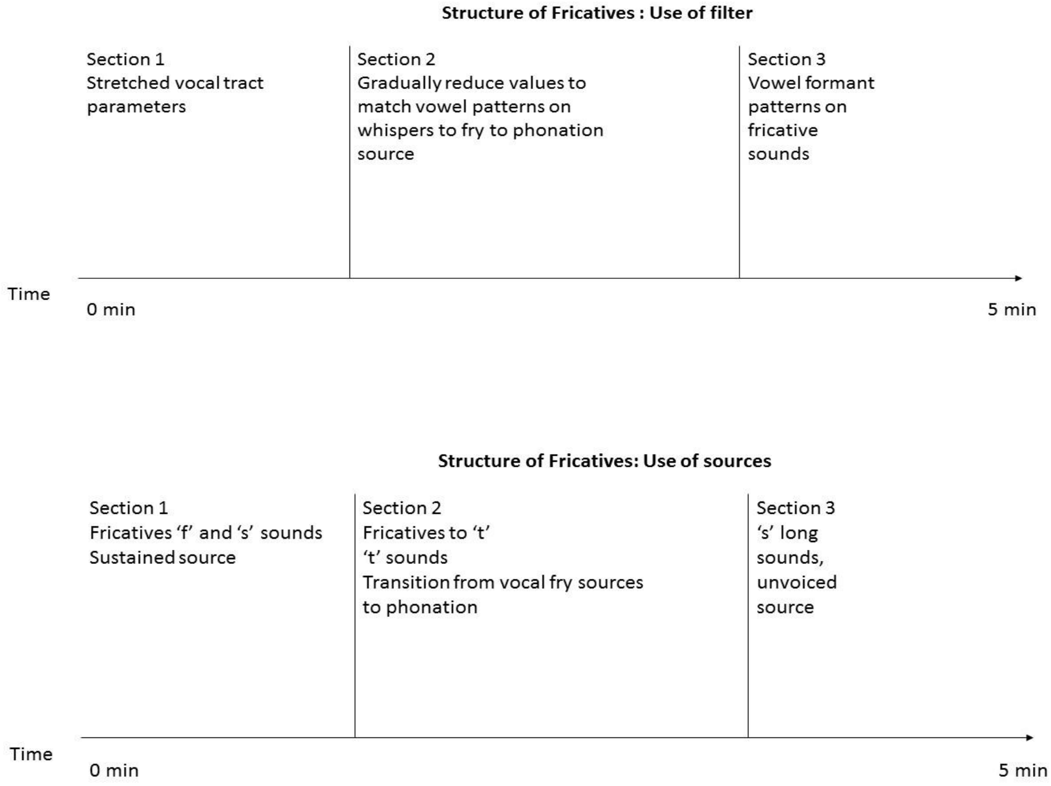
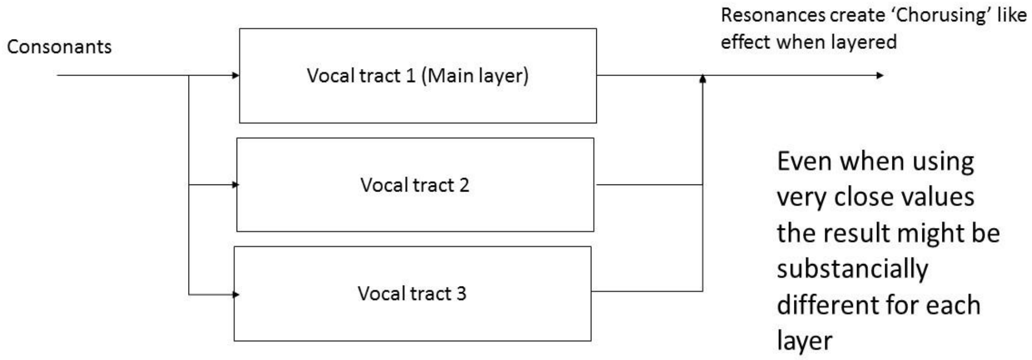

Structure of Fricatives : Use of filter
Section 1
Stretched vocal tract parameters
Section 2
Gradually reduce values to match vowel patterns on whispers to fry to phonation source
Section 3
Vowel formant patterns on fricative sounds
Time
0 min
5 min
Structure of Fricatives: Use of sources
Section 1
Fricatives 'f' and 's' sounds
Sustained source
Section 2
Fricatives to 't'
't' sounds
Transition from vocal fry sources to phonation
Section 3
's' long sounds, unvoiced source
Time
0 min
5 min

Figure 33. shows the use of the source-filter model to design the transformation of recorded consonants over time in the item *Fricatives*.



Resonances create 'Chorusing' like effect when layered

Even when using very close values the result might be substancially different for each layer

Each layer uses a different audio rendering of the model which has slightly different parameters and amplitude envelope from the main layer.
They are mixed in multi-track to produce the 'chorusing' effect and dissonant sounds to create tension.

Figure 34. Waveguide 'chorusing' technique through different layers of formant resonances. (Refer to Example 2.7).

By reducing the value of r_g and r_l, over time the original recorded fricative sound becomes more apparent. The reduction over time of the length of the tube to match the calculations by Fant can be used to gradually transform consonants into vowels.[167] This has been used in particular on the recorded /t/ sounds that are first presented unprocessed in Section 2, they then excite a waveguide model with a long vocal tract length. Specifically, this seems to confer the glottal pulses a character that resembles metallic percussion instruments.

Section 2 focuses on the transition from vocal fry to phonation, and from consonant to vowel at the same time, by gradually shortening the length of the tubes until they match the formant patterns of vowel sounds.

By using smaller vocal tract lengths (8cm) or narrowing (or widening) one of the tubes to the maximum before either the model produces clipping or imperceptible differences it is possible to generate very rich spectra by filtering long */s/* fricative

[167] Fant, *Acoustic theory of speech production: With calculations based on X-ray studies of Russian articulations*, 66.

sounds (see Section 3 of the item). Exploring and stretching the value of the vowel parameters produces the greatest variety of sounds of an artificial vocal tract. However, using these as a target points for parameters allows one to control the perception of the audio material processed through the waveguide model. Vocal Fry sources can be used to excite 60cm long vocal tracts. This allows for an expansion of the uses of the convolution vocal fry synthesis technique described in Chapter 2. Reverberation-like effects can be achieved by convolving the resulting Fry pulses with stereo white noise to transfer the qualities of formants produced by the tubes. This technique has been used in Section 3 to provide a background musical texture; the sound of whispered vowels is synthesized from recorded */s/* sounds. The aim of this section was to explore all the vowel formant patterns derived from the calculations in Figure 30 and to gradually reduce the feedback until the fricative sounds appear unprocessed for the first time, at the very end of the piece. The duration of the notes and individual sounds were guided by the overall planned structure, but this was dramatically affected by the features of the sounds synthesized in Matlab and by the source material that was edited to perform brassage until the different types of voice sources were approximated. Layering of different sounds has been used to support the transition from consonant to vowel and to add tension in order to gradually reveal vowel sounds as a result of an increase in density of sounds. This process is similar to the increase in density from fry to voiced sounds.

Fricatives shows how a model of the vocal tract can not only transfer the qualities of vowel formant patterns to consonants, but also how the flexibility of a computer model in dealing with a variety of combination of settings (that are not meant to synthesize vowels) provides a means of using creatively formant frequencies. Controlling the amount of feedback reduces or enhances the peaks in the magnitude response. A value between 0.9 and 0.999 of r_l and r_g confers a metallic character to the sound which exaggerates the resonances of the model and introduces ringing that smooths percussive sources and differences in dynamics. The resulting vocal tract model used for the techniques used in *Fricatives* is provided in Appendix III.

Chapter 6

6.1 Prony's method: exploring anti-formants

This chapter includes a collection of techniques to explore the use of zeros in combination with poles to design new tools for sound interpolations. In particular, it explores the development of techniques to capture the qualities of nasal consonant sounds, to control transformation over time of another recorded or synthetic source.

The techniques described in this chapter aim to provide a more accurate representation of nasal consonants; this is achieved by exploring the approximation of zeros as well as poles to transfer the resonant structures of both consonants and vowels to other sound sources. The goal is to gather a collection of spectral envelopes from recorded speech and investigate different techniques to perform interpolation between the timbre of speech. This interpolation is used to control digital filters in order, for example, to gradually perform transformations from consonants to vowels (and for the opposite purpose) and to control the features of sound interpolation with recorded speech.

A technique using Prony's method, to estimate both formants and anti-formants, has been implemented in order to design new tools to manipulate data derived from speech, in order to morph the timbre of a sound source over time. This process resulted in the design of a novel morphing system that uses audio analysis of speech to manipulate the features of sound over time, by operating over an intuitive representation of formants and anti-formants as a parametric model. This allows us to explore variation in sound colour by performing gradual changes in the formant patterns collected from speech. Slawson defines different dimensions of sound colour according to the position of the first two formants in a collection of vowels.[168] There exists the potential to explore transformations in sound colour over time, by using a filter that approximates the effect of the position of five formants and three anti-formants. The use of anti-formants enables the exploration of the timbre of consonants (nasals, laterals) that usually produce zeros in the magnitude response

[168] Slawson, *Sound Color*, 55.

of the vocal tract. This also allows us to apply the extraction of bandwidths and frequency of formants and anti-formants derived from recorded speech in the context of Slawson's dimension of sound colour. Slawson further explores the theory of colour of vowels by considering variations over time of the first two formants, specifying formant transitions in acoustic features as a two beginning position of formant frequencies, two ending position of formant frequencies and the duration of the change over time.[169] This suggests the potential to explore beginning and ending spectral envelopes, consisting of anti-resonances as well as resonances, changing over the duration of a sound source. Furthermore, it permits the exploration of the interaction between a group of peaks (poles) and a group of valleys (zeros) for audio interpolation techniques. As discussed in chapter 3, Prony's method can be adapted to parametric modelling techniques to provide an intuitive approximation of the features of the vocal tract from the analysis of a segment of recorded speech that includes valleys as well as peaks in the spectral envelope; this suggests the possibility of exploring its application to capture the beginning and ending spectral envelopes of the interpolations described above.

This chapter describes an approach to performing interpolation between different spectral envelopes extracted from recorded speech. The design of digital filters will be explored to create sound transformations by manipulating data extracted from speech. The aim is to transfer the qualities of both vowels and nasal consonants using Prony's method to approximate the position of anti-formants as well as formants from the audio analysis of speech. The use of speech coding will be explored to perform interpolation between different formant and anti-formant patterns, by manipulating the filter coefficients extracted with Prony's method, in order to design smooth sound transformations over time. The technique allows the capture from recorded speech of the qualities of nasal sounds as well as vowels by using a pole-zero arrangement.

An approach to speech processing will be explored in this chapter to design audio morphing techniques for the synthesis of hybrid sounds. The ability to separate the

[169] Wayne Slawson, "Sound-Color Dynamics", *Perspectives of New Music*, Vol. 25, no. 1-2, 25th Anniversary Issue (1987): 166-167.

source from the filter in speech processing means it is possible to perform audio morphing only to the filter, thereby generating hybrid sounds that will transfer the qualities of different target sounds to another source. Existing techniques for speech processing and digital filter design will be combined to investigate how these technologies can provide meaningful transformations of hybrid sounds in music production and for the creation of sound effects.

The interpolation between different consonant and vowel sounds is explored in the original composition *Whispers* (2018); this demonstrates the use of a morphing system that employs Prony's method to capture the qualities of speech sounds. This chapter also includes an overview on how Prony's method has been adapted to expand the techniques discussed in Chapter 4. Figure 32 outlines the approach to using Prony's method to examine the possibilities of analysis-based sound synthesis to create sounds that evolve over time. The ability to control the amount of time for the interpolation to take place allows the study of the perception of vowels and consonants and exploration of the manipulation of filters based on recorded sounds. The resulting morphing system is used to produce transformations inspired by Slawson's theory of sound colour, followed by an exploration of how this technique can be used to perform interpolations over the dimensions of sound colour.[170] The ability to capture the anti-formant states of consonants is demonstrated and tested during the production of *Whispers* to design interpolations between consonants and sound colour of vowels. The control over the time in which formant changes occur allows us to explore how the perception of diphthongs and consonant transition can be altered. Examples of music production techniques are discussed, describing some of the possibilities of interpolations between formant states that the morphing system brings to sound design. This method should be better at capturing the features of sounds that are characterised by anti-resonance, such as nasal vowels, as well as resonance.

[170] Slawson, *Sound Color*, 55.

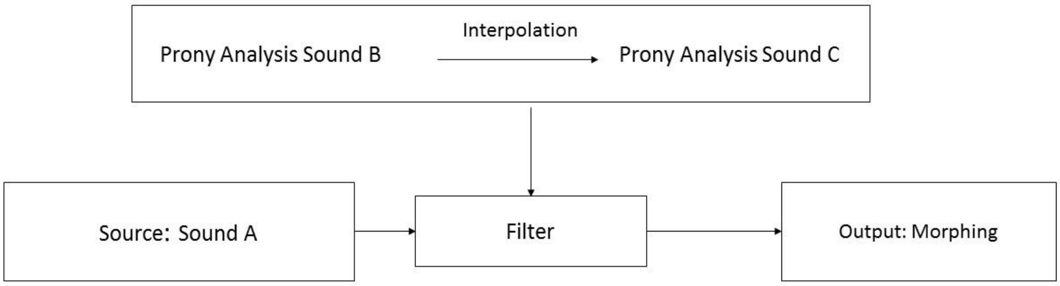


Figure 35. Approach to sound interpolation by using Prony's method to control the filter.

Figure 35 shows the idea used to design a new tool to perform audio interpolations over time. Sound A is the input sound source to be filtered (or voice source). Sound B (starting sound colour) provides the starting spectral envelope of the pole-zero filter that will be shaped over time into the spectral envelope of Sound C (target sound colour). The duration of the interpolation is the same as that of the audio file used as Sound A.

6.2 Poles and zeros in consonant sounds

In Chapter 2 an overview was provided of how the vocal tract can be represented as a source-filter model; during the production of nasal and lateral consonants certain regions of frequencies are attenuated as result of the action of bifurcation with the nasal cavities or position of the tongue, as discussed.

This chapter investigates an approach to digital filter design that involves computing the frequency response of a filter by choosing the position of poles and zeros on the z-plane and finding the filter's difference equation to control the features of another sound.

The idea is to design a filter that can model the interaction between formants *and* anti-formants. The position of complex conjugate pairs of poles and zeros on the z-plane will be extracted from recorded speech. Chapter 2 provides more details about the z-transform. It also presents the use of complex conjugate pairs of poles and zeroes in relation to the magnitude response of a filter.

Figure 36 shows the effect of the movement of a complex conjugate pair of poles on the magnitude response of a filter. The effect of moving the poles on the z-plane can be summarised in this way:

- Movement of the poles closer to the origin of the circle widen the bandwidth of the filter. Moving the poles closer to the circle narrows the band of the filter.
- Rotations of the position of the poles affect the position of the centre-frequency of the filter.

The position of poles and zeros on the unit circle influences the magnitude response of a filter. This can be used as a means to control timbre. This approach allows us to plot spectral envelopes of sounds as groups of poles and zeros on the unit circle. One can design timbre transformations by drawing the trajectories of poles and zeros and moving them over time. The coordinates can be translated as the coefficients of filter displaying both resonances and notches moving in frequency. This process allows us to use the coordinates of poles and zeroes as a time-varying parameter for sound synthesis. This prompts the exploration of timbre through mathematical manipulation of the coordinates of poles and zeros. This results in the creation of a tool for the interpolation of the position of poles and zeros on the unit circle. Therefore, it is possible to explore what the position of poles and zeros on the plot means in terms of sound and how it could be of interest for music applications. More specifically, one can investigate how interpolation of filter coefficients can result in interpolations of timbre. In relation to formants, moving poles closer to the circle produce a narrower peak while rotating the poles moves the position of formants in frequency. For example, by moving a pole anti-clockwise along the circle its movements are mirrored by its conjugate pole, the result is a formant that increases in frequency over time. Different vowels, for example, can be represented by the position of different pole conjugate pairs on the z-plane. It is possible to explore the interpolation between vowels by performing interpolation on the position of poles on the z-plane to gradually match a different formant pattern.

The same concept applies to conjugate pairs of zeros, except their effect on the magnitude response is to produce valleys in the magnitude response instead of peaks, so reducing certain frequency regions that can be compared to the anti-formants in nasal consonants. By combining poles and zeros and changing their positions on the z-plane it is possible to compute interpolations between the spectral envelope of consonants and vowels.

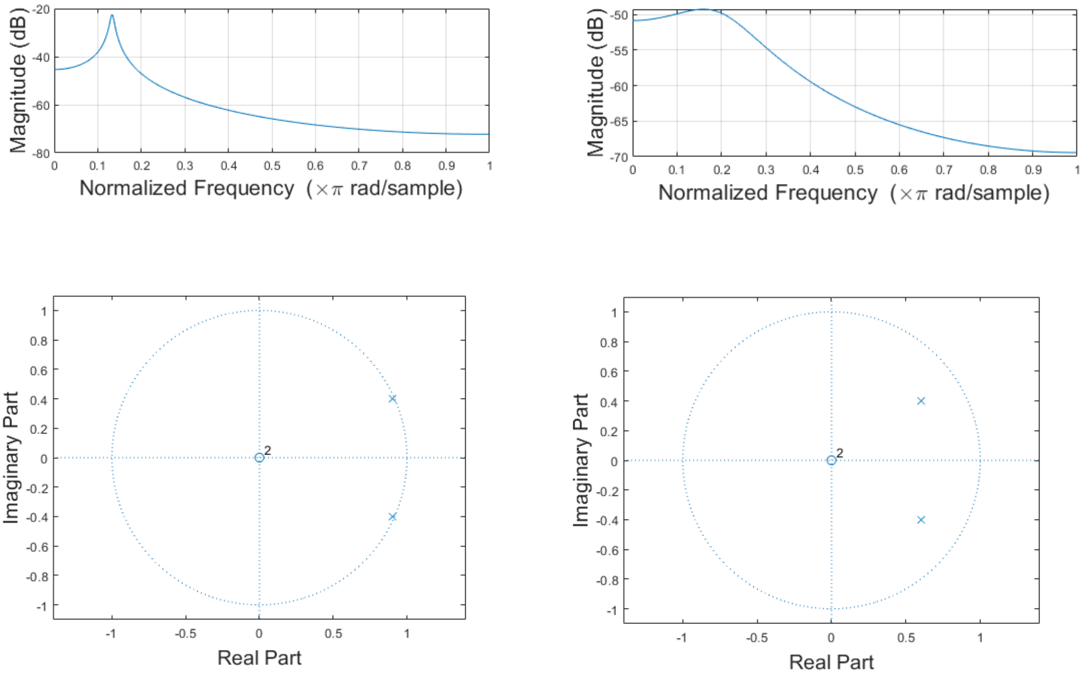


Figure 36. Effect of pole's movement on the magnitude response of the filter. On the left the starting position, on the right the target position shows a larger bandwidth.

The visual representation in Figure 36 and Figure 37 is a useful tool to predict sound transformations over time. This can be achieved by providing an animation of the conjugate pairs moving within the circle. The position of the pairs gives an indication of the bandwidth and frequency and stability of the filter. One can use it as a visual reference to study sound transformations on each frame. This is meaningful when used to monitor the features of a filter derived with Prony's method from recorded speech.

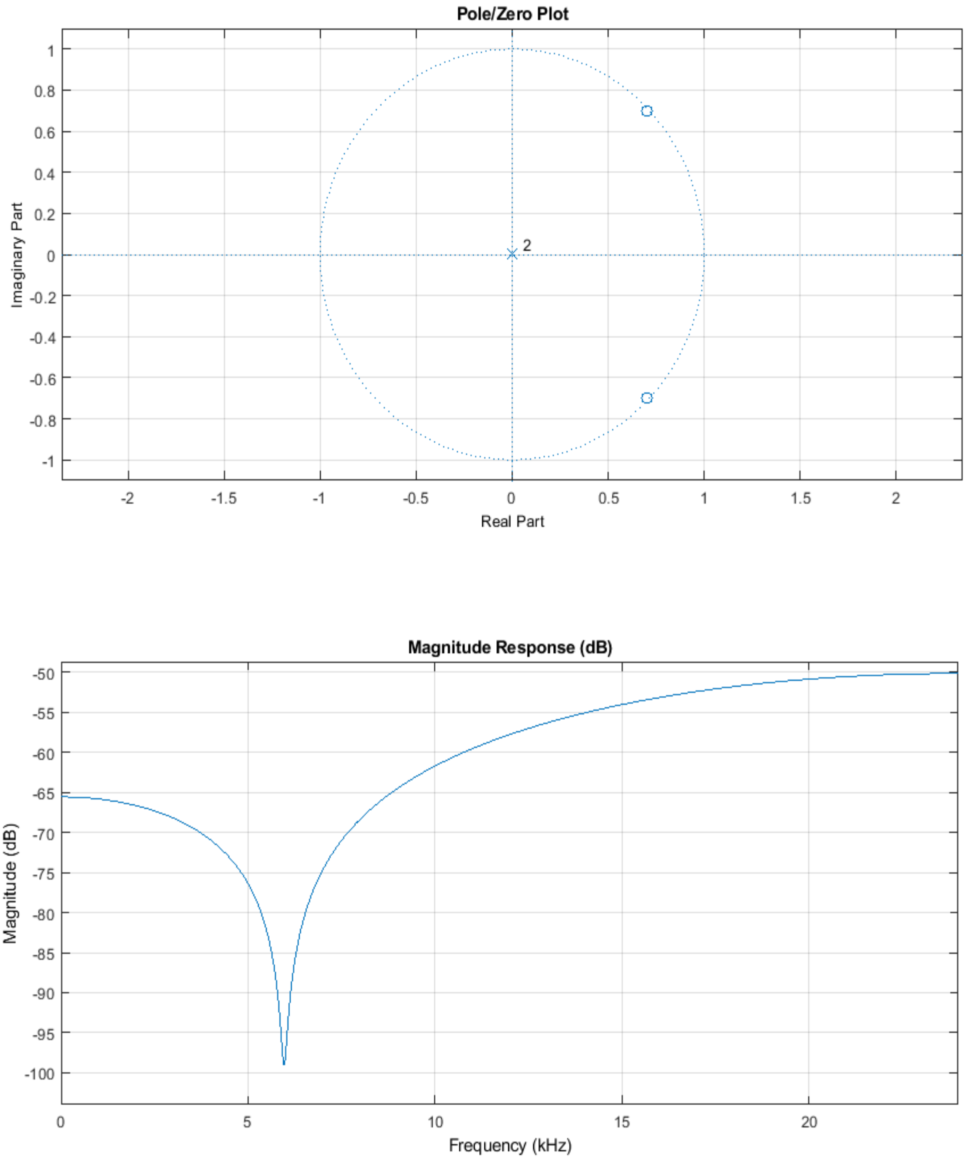


Figure 37. Effect of zero conjugate pairs on the magnitude response of the filter.

Chapter 3 provided an overview of how the filter coefficients of an IIR filter can be computed from an input sequence of samples of a signal, resulting in the equation:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_M z^{-N}} \tag{12}$$

The factorisation of both the numerator and denominator of the polynomials of the transfer functions enable us to obtain the pole-zero description of the filter and, by specifying the poles and zeros of a recursive filter (IIR filters), it is possible to estimate and plot its frequency response characteristics. The factorisation can be represented in the following equation:

$$H(z) = \frac{K(z - z_1)(z - z_2)(z - z_3)\cdots}{(z - p_1)(z - p_2)(z - p_3)\cdots} \tag{13}$$

(13) Pole-zero description by factorising numerator and denominator.

Prony's method can be used in time-domain parametric modelling techniques to approximate the filter coefficients from the analysis of speech (or any other signal), to derive the roots of the polynomials from the coefficients approximated in order to provide control over their position on the z-plane. This process allows one to specify poles and zeros from recorded sound, while generating creative possibilities to perform manipulation over time of the centre-frequency and bandwidth of formants and anti-formants. Audio analysis of consonants is the first step to exploring the difference in features between Prony's method and LPC (by comparing the analysis of a nasal consonant /m/ as it produces anti-resonances).

Figure 38 and Figure 39 show an example of an analysis of the same segment of speech from an /m/ consonant. The analysis is performed in Matlab by using the *lpc* and *prony* functions that allow to set the order of analysis. The function *prony* provides control over the order of zeros and poles separately, the coefficients of the zeros are then provided as numerator and the poles as denominator (as in Equation (6.1). The order of analysis chosen is set to approximate five formants (poles) and, in the case of *prony,* also three anti-formants (zeros). The resulting plot shows different shapes of the valleys within the spectral envelope, also a different overall maximum magnitude value.

Chapter 3 provided an overview of how this approach to Prony’s method works and how it is different from LPC analysis.

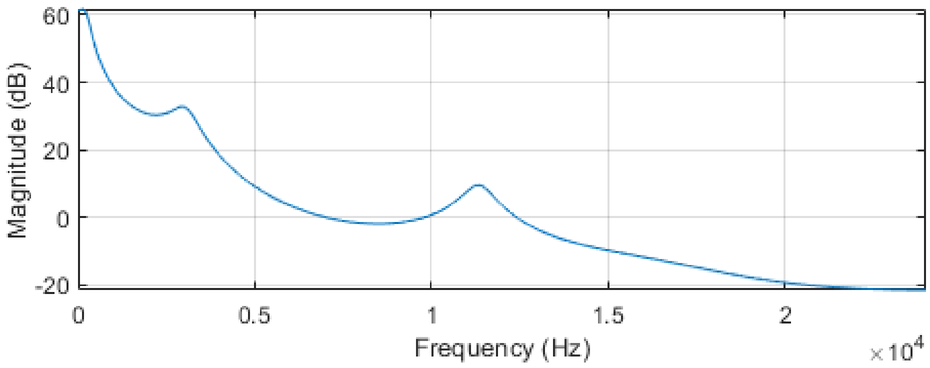


Figure 38. LPC analysis consonant /m/ for order 10 (five conjugate poles).

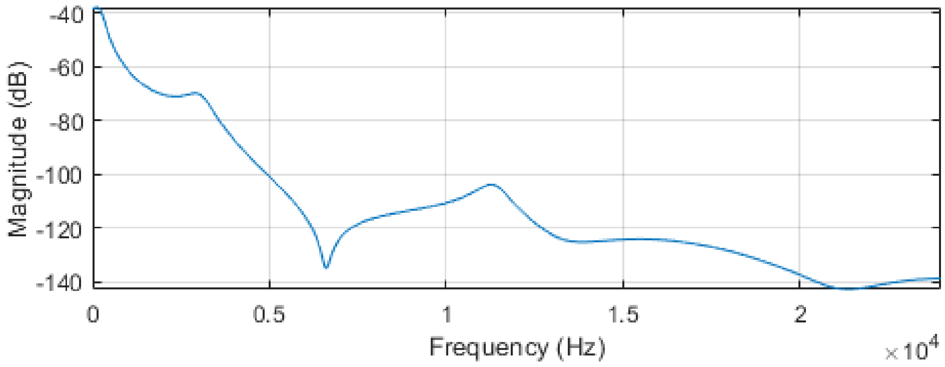


Figure 39. Prony's analysis consonant /m/ order 10 (five conjugate poles, three conjugate zeros).

6.3 Capture and interpolation of filter coefficients

This section outlines an approach to filter manipulation used to create changes over time in the resonant structures of spectral envelopes approximated with Prony's method.

The roots of the poles and zeros on the z-plane can be computed from the coefficients approximated by Prony by factorising the polynomials of the numerator to convert the filter coefficients into poles and zeros coordinates. Changes in the real part move the pole or zero horizontally while changes in the imaginary part produce vertical movements. The function *roots* in Matlab has been explored to extract the complex numbers representing the coordinates of the conjugate pairs of poles and zeros on the Z-plane. It is possible to perfom morphing between resonant structures of different speech sounds by finding the roots of the poles and zeros approximated with *prony* and storing the coordinates. A set of coordinates is assigned as the starting position of each pole and zero separately on the z-plane while the other set is the target position. By deciding how many interpolation points to use, it is possible to control how long or how smooth the interpolation will be. Each point of the interpolation is then converted to time-domain filter coefficients. The use of a frame-based system with overlapping windows can be implemented to design a time-varying filter from the interpolation points, representing the movements of the poles and zeros.[171] The idea is to match the filter coefficients derived from the current interpolation points with the output sequences of frame of the synthesis part. The use of the state-save method for signals processed in a frame-by-frame basis can be implemented, the method consists in saving the current state of the delay elements to be stored and used as initial state of the filter for the next frame.[172] This method can be used to explore how saving the state of the current frame (or interpolation point) for the next one can provide smooth manipulation of the filter over time for the synthesis of sound.

6.4 *Whispers*: the idea

This section discusses how the item *Whispers* (2018) (see Digital copy of accompanying Audio tracks, Track 3) explored the perception of vowels and nasal consonants through the interpolation of different spectral envelopes derived from recorded speech sounds.

Sound synthesis techniques are designed in order to explore audio morphing based on the Prony analysis of speech sounds to estimate the starting and target state of the spectral envelope. As a way of implementing the techniques outlined by the ideas discussed in this chapter, a new morphing system has been designed in Matlab for the interpolation of formant and anti-formant structures derived from recorded speech. Consonants like /m/, /n/ and /l/ are used in speech during transitions and can be used to perform transitions from a formant state to another.[173] (Refer to

[171] Barnwell III, Nayebi and Richardson *Speech coding: A computer laboratory textbook*, 91.

[172] Chu, *Speech coding algorithms: Foundation and evolution of standardized coders*, 50.

[173] Wishart, *On sonic art*, 278.

Example 3.1). The idea is to explore how analysis of speech with Prony's method can be combined with techniques to perform interpolation and inbetweening of two sounds in order to design transitions in the formant state of the filter.[174] The aim is to capture the starting and ending point from recorded speech. The voice of the performer will control the parameters of the starting and ending point of the interpolation; a morphing system would then compute the intermediate anti-formants and formants states. The bandwidths of anti-resonances and resonances are also derived from recorded speech.

In a source-filter model the sound colour is associated with the filter rather than the source; keeping the formant frequencies and bandwidth of the filter constant allows the sound colour to remain constant, as in Slawson.[175] Figure 40 shows the dimensions of sound colour by Slawson that are named openness, acuteness, laxness and smallness.[176] Each dimension of sound colour is characterised by the position of the first and second formant of the filter; by increasing or decreasing the position of formants it is possible to change the dimension of sound colour. The position of poles and zeros on the unit circle affects the resonant structures of the filter. One can manipulate the sound colour by changing the coordinates of poles and zeros. When the coordinates remain invariant, the sound colour stays the same even when the input source of the filter is varied. Changing the position of poles and zeros over time allows us to explore interpolation in sound colour, also when keeping the input source constant.

[174] Wishart, *Audible Design*, 96-99.
[175] Slawson, *Sound Color*, 23-43.
[176] Ibid.,55.

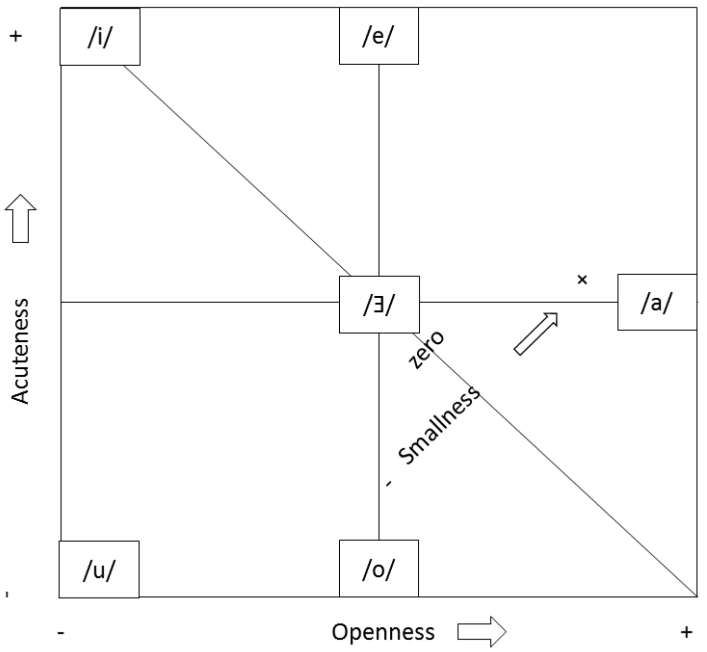


Figure 40. Sound colour map sound colour related to the formants of vowels. After Slawson.[177]

This theory suggests an option to explore the synthesis of gradual transformation over time of sound colour, by setting a starting colour and a target colour with the aim of exploring the interpolation of qualities of different speech sounds.

6.5 A morphing system for sound interpolations

The design of an interpolation in sound colour requires three sound sources:

- Starting colour: Spectral envelope sound A
- Target colour: Spectral envelope sound B
- Source input: Sound C, will be filtered with the resulting time-varying spectral envelope.

As described in 6.1 and 6.2 the roots of the filter coefficients derived with Prony's method can be manipulated to change the coordinates of the poles and zeros on the z-plane. A morphing system has been designed in MatLab to compute the interpolation points of the intermediate formant states and to translate such

[177] Slawson, *Sound Color,* 59.

transformation as a frame-based, time-varying filter. Appendix IV includes the transcription of the Matlab code with additional notes on how the interpolation has been implemented.

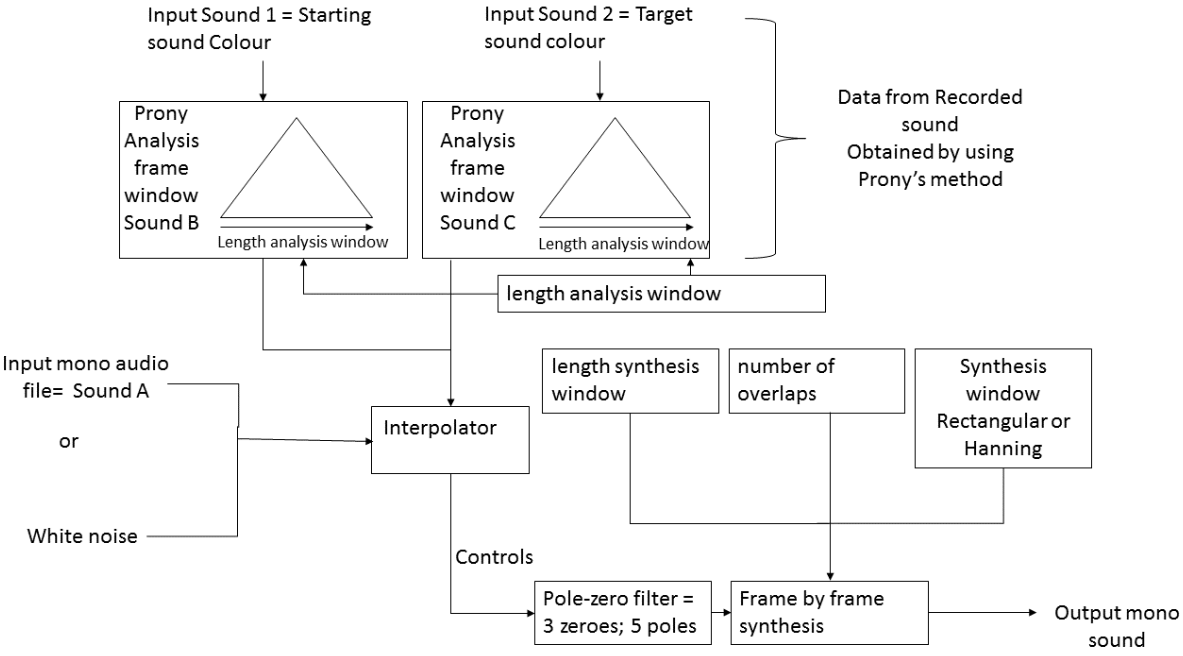

Input Sound 1 = Starting sound Colour
Input Sound 2 = Target sound colour
Prony Analysis frame window Sound B
Length analysis window
Prony Analysis frame window Sound C
Length analysis window
Data from Recorded sound Obtained by using Prony's method
length analysis window
Input mono audio file= Sound A
or
White noise
Interpolator
length synthesis window
number of overlaps
Synthesis window Rectangular or Hanning
Controls
Pole-zero filter = 3 zeroes; 5 poles
Frame by frame synthesis
Output mono sound

Figure 41. Schematics of an interpolation system to perform morphing from audio analysis of recorded sound.

Figure 41 shows the schematics of an interpolation that employs Prony's method to extract data from recorded sound in order to control the interpolation of the position of poles and zeros on the z-plane to create transformation of sound over time.

The morphing system is implemented in Matlab and available both as full code transcript and as a graphical user interface (GUI) and standalone computer application named *PZeroSynth: Audio interpolation system.* See Accompanying material for the installation of PZeroSynth and to Appendix IV of the user guide of the software; or refer to Appendix IV for a transcript of the code.

The morphing system has four main parts:

- Analysis: converts speech sounds to filter coefficients
- Roots extraction: converts filter coefficients to pole/zero coordinates

- Interpolation: performs linear interpolation between pole/zero coordinates and converts interpolation points to filter coefficients
- Synthesis: computation of the output on a frame-by-frame basis

In the analysis part, two separate Prony analysis processes are carried out to store the filter coefficients of the starting sound A and the target sound B separately.

The Prony analysis is performed by positioning a marker at the time where the desired sound colour is placed in the audio recording. The morphing system is related to the formant synthesizer as in the Klatt formant synthesizer as it approximates the effect of five formants[178]. However, a substantial difference is that the transfer function of the filter approximates the effect of three anti-formants for the synthesis of nasal sounds; further, both the formant frequencies and the bandwidth of the filter is extracted from recorded speech (or any recorded source). The interpolation part derives from the roots of the poles and zeros of both spectral envelopes and uses linear interpolation (linspace in Matlab) to compute a vector representing the interpolation points of the coordinates of sound A morphing into the coordinate of Sound B. The duration of the interpolation is set to take place in the same amount of time as the duration of the source Sound C.

The aim of timbre interpolation is to gradually morph a source sound into the target sound. One can manipulate the amplitude and frequency data from different frames of DFT analysis in order to design interpolations between similar sounds.[179] This process can be achieved by using audio processing tools to perform analysis and re-synthesis on both the starting and destination sounds. Consequently, one can manipulate the frequency and magnitude of the individual partials of the original source spectrum to morph it into the target spectrum.[180] An intuitive way to create interpolations in timbre is to generate intermediate shapes between the spectral envelopes of the original and target sounds.[181] One method to change the shape of a

[178] Klatt, *"Software for a Cascade/Parallel Formant Synthesizer,"* 986.

[179] Trevor Wishart, "From Sound Morphing to the Synthesis of Starlight. Musical Experiences with the Phase Vocoder over 25 Years." Musica, Tecnologia = Music, Technology 7 (2013): 65-66.

[180] Sethares, William A., Milne, Andrew J., Tiedje, Stefan, Prechtl, Anthony, and Plamondon, Jame s. "Spectral Tools for Dynamic Tonality and Audio Morphing." *Computer Music Journal* 33, no. 2 (2009): 74-75.

[181] Caetano, Marcelo, and Xavier Rodet. "Sound Morphing by Feature Interpolation." 2011 IEEE

spectral envelope is to employ filters that sculpt the frequency content of an input source. This can be achieved by controlling the variation in radius and angle of poles and zeros independently in order to compute the coefficients and update the filter's parameters.[182] Therefore, one can combine this approach to manipulate filters with a parametric analysis techniques, for example Prony's method. The result of such analysis can represent the spectral envelope of the original and target sounds as two separate sets of filter coefficients. The idea of a filter that changes over time is particularly meaningful in relation to the vocal tract and speech synthesis techniques. This approach prompts the exploration of filter manipulation based on the analysis of recorded speech sounds in order to design timbre interpolations.

The synthesis converts the interpolation vector from roots to time-domain filter coefficients that can be accessed to compute an output sequence of frames. Sound C is divided into frames where each frame represents an interpolation point of the filter's magnitude response. A single segment (windowed frame) of the sound A and B are fed to the Prony function. The synthesis part uses overlapping windows in an arrangement similar to the method used for autocorrelation in LPC analysis and re-synthesis. The system provides a choice of using Hanning overlapping windows or rectangular windows controlled by a modified frame-by-frame synthesis that avoids overlapping frames.

The user can control the starting point and length of the window of analysis of the segment that Prony uses to determine the starting and target spectral envelope. The system does not perform spectral whitening of the sound source that is defined to be filtered.

The algorithm performs two independent audio analyses of two separate sounds that are defined as input by the user. A triangular window has been used in this chapter to capture the spectral envelopes of recorded sounds with the morphing system; this was chosen because it is better suited to the material used to create the piece *Whispers,* which explores the capabilities of this method to perform sound

International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, 163.

[182] Ding, Yinong, and David Rossum. "Filter Morphing of Parametric Equalizers and Shelving Filters for Audio Signal Processing." Journal of the Audio Engineering Society 43, no. 10 (1995): 822-823.

interpolations. The user can define the starting point and the length of the triangular windows of analysis that will be used to perform audio analysis with Prony's method:

```
[num1,den1]= prony(vowel1.*wmark2,6,10);

rootszeros1 = roots(num1);
rootspoles1 = roots(den1);

[num2,den2]= prony(vowel2.*wmark2,6,10);

rootszeros2 = roots(num2);
rootspoles2 = roots(den2);
```

The code above shows how the *prony* function is used to produce a separate set of coefficients for numerator and denominator from the analysis of two frames of two separate vowels sounds multiplied by a triangular window (*wmark2*).

The functions num1 and den1 represent the numerator and denominator of coefficients of the starting sound colour for the interpolation; the functions num2 and den2 represent the numerator and denominator coefficients of the target sound colour. The functions rootspoles1, rootspoles2 represent the coordinates of the poles' conjugate pairs from Sound A and Sound B. The functions rootszeroes1 and rootszeroes2 represent the coordinates of the conjugate pairs of zeros.

This process provides two separate spectral envelope estimates, however the information about formants and anti-formants can be represented as complex numbers to manipulate the position of poles and zeroes in the Z-plane in order to create intermediate spectral envelopes between the starting and target analysis data. In order to compute the interpolations between the coordinates of poles and zeroes, the resulting filter coefficients of the numerator and denominator are processed separately; this allows for separate extraction of the roots of the polynomials for poles and zeros. The order of the analysis of the morphing system is set to 6th order for numerator and 10th order for denominator. This approximates the position of three conjugate pairs of zeroes and five conjugate pairs of poles resulting in the estimation of three anti-formants and five formants.

```
values_zeros1= rootszeros1(find(imag(rootszeros1)>=0);
rz1 = real(values_zeros1(1,:));
rz2 = real(values_zeros1(2,:));
rz3 = real(values_zeros1(3,:));


iz1 = imag(values_zeros1(1,:));
iz2 = imag(values_zeros1(2,:));
iz3 = imag(values_zeros1(3,:));
```

The code above shows the section of the algorithm that extracts and separates the real and imaginary part of each zero estimated from the numerator (rootszeroes1) coefficients of Sound A (starting sound) with Prony's method. The variable rz1, rz2, rz3 represent the value of real parts of the three conjugate pairs of zeros of the starting position; iz1, iz2, iz3 represent the imaginary part of the three conjugate pairs. The same process is repeated for the numerator (rootszeroes2) to store the resulting six values separately.

The roots are extracted from the roots of the denominator of the starting sound A (rootspoles1):

```
values_poles1= rootspoles1(find(imag(rootspoles1)>=0));

rp1 = real(values_poles1(1,:));
rp2 = real(values_poles1(2,:));
rp3 = real(values_poles1(3,:));
rp4 = real(values_poles1(4,:));
rp5 = real(values_poles1(5,:));
```

```
ip1 = imag(values_poles1(1,:));
ip2 = imag(values_poles1(2,:));
ip3 = imag(values_poles1(3,:));
ip4 = imag(values_poles1(4,:));
ip5 = imag(values_poles1(5,:));
```

The values rp1,rp2,rp3,rp4,rp5 represent the values of the real part of the poles being ip1,ip2,ip3,ip4,ip5 the imaginary values. In a similar way to the zeros, the process is repeated with the resulting ten values from the denominator of the Prony analysis of the target Sound B (rootspoles2). The values of the real and imaginary parts are then used within a linear interpolation function (linspace) in order to design a gradual change in the position of each pole and its conjugate pair. The computation of the interpolation of the position of the poles is shown in the code below:

```
%p1
realpart1 = linspace(rp1,rp1b,ninterp);
imapart1 = linspace(ip1*i,ip1b*i,ninterp);
%p2
realpart2 = linspace(rp2,rp2b,ninterp);
imapart2 = linspace(ip2*i,ip2b*i,ninterp);
%p3
realpart3 = linspace(rp3,rp3b,ninterp);
imapart3 = linspace(ip3*i,ip3b*i,ninterp);
%p4
realpart4 = linspace(rp4,rp4b,ninterp);
imapart4 = linspace(ip4*i,ip4b*i,ninterp);
%p5
realpart5 = linspace(rp5,rp5b,ninterp);
imapart5 = linspace(ip5*i,ip5b*i,ninterp);
```

The variables rp1,rp2,rp3,rp4,rp5,ip1,ip2,ip3,ip4,ip5 represent the real parts and imaginary parts of the poles of the starting sound A that are interpolated over a number of interpolation points (ninterp) until they match the position of the target coordinates of the sound B called rp1b, rp2b, rp3b, rp4b, rp5b, ip1b, ip2b, ip3b, ip4b, ip5b.

The result is five poles moving simultaneously in ninterp numbers of point of interpolation, from the start position to the target position. The number of interpolation points has a close connection with the number of frames of frame-by-frame synthesis algorithms, as will be described in the following sections.

The identical process is performed in parallel for the coordinates of the three zeros.

The interpolation points are then matched again into conjugate pairs, and then converted to time-domain filter coefficients by manually solving the factorization of polynomials for both the poles and zeros. This process is required to process the data relating to the roots of both poles and zeros in order to represent anti-formants and formants and to change them over time. The code below shows how the coordinates of the interpolation are mirrored in conjugate pairs in order to compute the filter coefficients of one pole a0, a1, a2:

```
%p1 conj
pole1 = realpart1+imapart1;
conju1= realpart1-imapart1;

a1= -(pole1+conju1);
a2= (pole1.*conju1);

a0= ones(size(a2));
```

Equation (14) suggests that the numerator and denominator of the filter can be represented as multiplication of the poles and zeros. This feature can be used to develop a strategy to compute the filter coefficients (of the filters) by multiplication of the coefficients of the individual poles.

$$H(z) = \frac{K(z-z_1)(z-z_2)(z-z_3)\cdots}{(z-p_1)(z-p_2)(z-p_3)\cdots}$$

(14)

The code shows how the imaginary part of the first pole (imapart1) is mirrored in a conjugate pair by inverting the sign of the imaginary part. For example, the conjugate of the pole (0.7+0.2j) will be (0.7-0.2j). To find the filter coefficients of the conjugate pair (0.7±0.2j) it is possible to adapt (14) as:

$$H(z) = \big(z - (0.7 + 0.2j)\big) + (z - (0.7 - 0.2j))$$

(15)

The filter coefficients of each 5 conjugate pairs can then be calculated with this method and multiplied between them, to obtain the filter coefficients of the denominator. An identical process can be applied to mirror the conjugate pairs of the 3 zeros and compute the filter coefficients of the numerator by multiplying the coefficients of each conjugate pair of zeros.

The code below shows an example of how the coefficients of the denominator have been implemented in Matlab by multiplying the coefficients of five conjugate pairs in order to convert the coordinates of the poles to time-domain filter coefficients.

```
%P1*P2

%calculates coefficients first two poles
a1p12 = (a0.*a12)+(a1.*a02);


a2p12 = (a0.*a22)+(a1.*a12)+(a2.*a02);


a3p12 = (a1.*a22)+(a2.*a12);


a4p12 = (a2.*a22);


a0p12 =ones(size(a2p12));
```

```
%P1*P2*P3
%adding third pole
a1p123 = (a0p12.*a13)+(a1p12.*a03);

a2p123 = (a0p12.*a23)+(a1p12.*a13)+(a2p12.*a03);

a3p123 = (a1p12.*a23)+(a2p12.*a13)+(a3p12.*a03);

a4p123 = (a4p12.*a03)+(a3p12.*a13)+(a2p12.*a23);

a5p123 = (a4p12.*a13)+(a3p12.*a23);

a6p123 = (a4p12.*a23);

a0p123 =ones(size(a1p123));

%P1*P2*P3*P4
%adding fourth pole
a1p1234  = (a1p123.*a04)+(a0p123.*a14);

a2p1234  = (a0p123.*a24)+(a1p123.*a14)+(a2p123.*a04);

a3p1234  = (a1p123.*a24)+(a2p123.*a14)+(a3p123.*a04);

a4p1234 =  (a2p123.*a24)+(a3p123.*a14)+(a4p123.*a04);

a5p1234  = (a3p123.*a24)+(a4p123.*a14)+(a5p123.*a04);

a6p1234  = (a4p123.*a24)+(a5p123.*a14)+(a6p123.*a04);

a7p1234  = (a5p123.*a24)+(a6p123.*a14);

a8p1234  = (a6p123.*a24);
```

```
a0p1234  =ones(size(a2p1234));


%P1*P2*P3*P4*P5

a1ptot   = (a0p1234.*a15)+(a1p1234.*a05);

a2ptot   =
(a0p1234.*a25)+(a1p1234.*a15)+(a2p1234.*a05);

a3ptot   =
(a1p1234.*a25)+(a2p1234.*a15)+(a3p1234.*a05);

a4ptot   =
(a2p1234.*a25)+(a3p1234.*a15)+(a4p1234.*a05);

a5ptot   =
(a3p1234.*a25)+(a4p1234.*a15)+(a5p1234.*a05);

a6ptot   =
(a4p1234.*a25)+(a5p1234.*a15)+(a6p1234.*a05);

a7ptot   =
(a5p1234.*a25)+(a6p1234.*a15)+(a7p1234.*a05);

a8ptot   =
(a6p1234.*a25)+(a7p1234.*a15)+(a8p1234.*a05);

a9ptot   = (a7p1234.*a25)+(a8p1234.*a15);

a10ptot  = (a8p1234.*a25);

a0ptot   = ones(size(a2ptot));



coeffap12 = [a0p12;a1p12;a2p12;a3p12;a4p12]';

coeffap123 =
[a0p123;a1p123;a2p123;a3p123;a4p123;a5p123;a6p123]';

coeffap1234 =
[a0p1234;a1p1234;a2p1234;a3p1234;a4p1234;a5p1234;a6p123
4;...
    a7p1234;a8p1234]';


coeffaptot =
[a0ptot;a1ptot;a2ptot;a3ptot;a4ptot;a5ptot;a6ptot;a7pto
t;...
    a8ptot;a9ptot;a10ptot]';
```

The use of this strategy to deal with the computation of filter coefficients by multiplying groups of coefficients, allows the order to be easily increased in future implementations; thereby increasing and eventually doubling the number of formants and anti-formants to order 20 and 12.

Once the coefficients of the poles are calculated they are stored in a matrix and accessed by a frame-by-frame synthesis algorithm implemented in Matlab as follows:

```
for j=1:nframes

    k = (framelen-hopsize)*j;



[frame,zf] =
filter(coeffbtot(j,:)',coeffaptot(j,:)',sig(k:k+hop-
1),zi(j,:));

 zi(j+1,:)= zf;


monitor(j,:)= k;


stablec(j,:)=
isstable(coeffbtot(j,:)',coeffaptot(j,:)');


    out(k:k+hop-1)=out(k:k+hop-1)+(frame.*wrect);


end
```

The frame-by-frame synthesis employs the save-state method to store the state of the filter of the current interpolation point for the next frame; this is in order to reduce blurring of the quality of sound. Appendix IV provides a transcript of the complete Matlab code used to build the morphing system based on Prony's method and save-state-method rectangular window.

There exists a Matlab implementation of a frame based cross-synthesis technique with LPC, the code is an example of common implementations of LPC based audio processors that would allow the control of a filter from recorded speech.[183] A similar

[183] D.Arfib, F. Keiler, Zölzer,U. "Source-Filter Processing" in *DAFX: Digital Audio Effects.* Ed. Udo Zölzer.

arrangement has been designed to automatically compute the number of frames and interpolation points according to the length of the source filtered, length and number of overlaps of the frames of the synthesis.

Figure 42 shows how the save-state method has been implemented in Matlab by using rectangular windows and storing the final state of the filter zf as the initial state for the next frame. This allow us to perform smooth interpolation of sound by updating the initial stated of each frame by using a higher number of overlapping windows. The more overlaps, the more often the initial state of the filter will be updated. But no actual overlap is happening. This approach has been designed to be compatible with the automatic segmentation to frames with Hanning windows, in order to explore interpolation with both shapes controlled by the same interpolation algorithm.

The use of rectangular windows with save-state methods provides the user with the potential to avoid cross-fading static filters during the design of time-varying transformation of sound. By using shorter frames with rectangular windows is possible to model the effect of a higher overlap factor but removing the blurring effect of many, for instance, Hanning windows cross-fades.

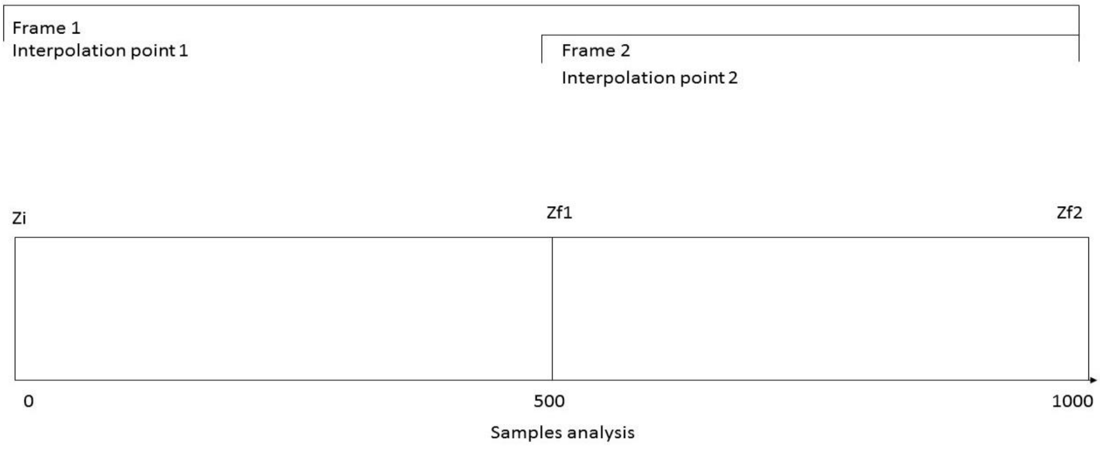


Figure 42. Save-state method implemented by storing the final state of the filter and by using the value as initial state of the filter at the start of the next frame.

(Chichester: Wiley, 2002): 317-318.

Due to the way Prony's method computes the filter coefficients of the numerator and denominator, the phase of the starting point of analysis affects the result of the transformation creating dramatic changes in amplitude of the resulting sound. By fine-tuning the position of the analysis frame on the input sound and its length it is possible to design smoother transitions from sound A to sound B. The results vary, depending on the sound chosen as starting and target formant state. Once coloured with a filter, the sound source can be coloured further by being used as input of another filter system in a process called by Slawson successive filtering[184]. This indicates the option to explore how the output of different interpolations in sound colour can be concatenated in order to design a hybrid sound that possesses overlaid changes in spectral envelope.

Linear interpolations between the position of poles and zeros creates variation over time in the frequency and bandwidth of formants and anti-formants, as poles and zeros interact with each other or move closer to the origin in the z-plane.

The morphing system includes an animated z-plane to study the movement of the conjugate pairs. The Prony analysis in the morphing system uses an order of 6 for the numerator and 10 for denominator. This is translated as five pairs of conjugate poles and three pairs of conjugate zeros. The user has control over the duration of the interpolation, the longer the interpolation the more interpolation points will be computed. The maximum duration of the interpolation is limited by the duration of the sound source that will be filtered. The synthesis part splits the source in frames and filters each frame with the position of poles and zeros of the corresponding interpolation point. Using a higher number of overlapping windows will automatically generate more interpolation points.

The morphing system offers the ability to move the poles and zeros to rotate within the unit circle. This is achieved by converting the starting and target cartesian coordinates of poles and zeros to polar ones by using the Matlab function *cart2pol*. Linear interpolation is then performed polar coordinates so the intermediate position of poles and zeros can be computed. Finally, the coordinates are converted

[184] Slawson, *Sound Color,* 85.

back to cartesian by using the Matlab function *pol2cart* after the interpolation. The process is outlined in this example of code:

```
%Cartesian coordinates converted to polar coordinates

[thetap1,rhop1]=cart2pol(rp1,ip1);%starting coordinates

           [thetap1b,rhop1b]=cart2pol(rp1b,ip1b);%target
coordinates

%interpolation polar coordinates

interpthetap1  = linspace(thetap1,thetap1b,ninterp);
interprhop1    = linspace(rhop1,rhop1b,ninterp);

%polar coordinates converted to cartesian coordinates

[realpart1,imapart1]=pol2cart(interpthetap1,interprhop1);
imapart1 = imapart1*i;
```

This is to provide more options and flexibility to work with different audio files when designing timbre interpolations. The poles and zeros have, in general, a narrower bandwidth when they perform rotations because they tend to stay closer to the unit circle compared to movements in straight line. This can result in an increase in resonance due to the increased sharpness of poles. When zeros move closer to the origin, their bandwidth widens and might result in drops in amplitude during the interpolation. The rotation movement provides a way to keep the zeros closer to the circle and thus maintaining a narrower bandwidth throughout the interpolation, which might be desired.

PZeroSynth offers the option to perform interpolation both with Prony's method and LPC. The analysis extracts five formants that can then be manipulated by the morphing system with the same techniques of interpolation discussed earlier. This is easily achieved through the controls included in the GUI. As a result, the user can easily switch between LPC and Prony (i.e. switch between an all-pole model and a pole-zero one).

6.6 Creation of *Whispers*.

The piece *Whispers* (see digital copy of accompanying audio tracks, Track 3) explores how changes over time in sound colour can be transferred to three types of sound sources. The aim of the piece is to explore the parameters of the morphing system to study how it can synthesize smooth interpolations.

The main focus of the piece is to synthesize a smooth interpolation between the nasal sound /m/ and a collection of vowel sounds.

The piece also explores how the qualities of both vowels and nasal sounds can be transferred to long textural sounds by using a Prony based vocoder to update and expand the LPC fry synthesis techniques described in chapter 4.

The type of sounds used within *Whispers* includes:

- Vocal Fry: high density and low density vocal fry sources have been synthesized in Praat (see Chapter 4).
- White noise or unvoiced voice source.
- Subharmonics singing synthesis. A 30 second pulse train has been synthesized in Csound with random modulation in frequency to mimic the variation in the phonation that occurs in the vocal folds. (Refer to Example 3.2).

Each source has been filtered in order to create interpolation based on the sound colour dimension chart in Figure 40. A collection of diphthongs and consonant sounds has been recorded from two male speakers in order to perform interpolation between the analysis of both different speech sounds and the voice of a different performer. The use of the morphing system allowed a performance of very slow interpolations of about 30-45 seconds. When filtering vocal fry sources with very slow changes the perception of speech and particularly vowels is altered. This is because the random excitation of the morphing filter highlights certain intermediate points of the interpolation instead of producing a steady, smooth transformation. By shortening or prolonging the time of interpolation on the subharmonic singing source it is possible to explore the perception of higher resonances of the voice source. The way the movement of poles and zeros affects the transfer function introduces smooth changes over time in the high frequency content of the voice source. Subharmonic voice sources have been used in particular to morph between the consonant /m/ and a collection of vowel colours.

A noise source has been used to implement successive filtering techniques concatenating the output of two interpolations; one from /m/ to /a/ and one to /o/ to /m/. For an example of interpolation from /o/ to /a/ refer to Example 3.3; refer to Example 3.4 to listen to successive filtering results. The idea here was to overlay two interpolations that move to opposite directions in vowel to consonant transitions. The choice of the timing was defined by experimenting with the duration of the interpolations. The production of the piece was focused on studying the behaviour of the morphing system in relation to the analysis of different speech sounds and varying parameters as length of analysis, position of marker, overlap factor in order to find which parameters are essential to be available to for a user in a GUI to adapt to different situations and create more stable (or unstable) transformations. For the nature of the morphing system, the overall musical material focuses on continuously changing sounds as the formant state is never held on a single colour. It is possible to synthesize static vowels and nasal sounds by using the same sound as both start and target colour.

Chapters 4 and 5 discussed how an approach to convolution between synthetic LPC fry pulses and white noise, used to transfer the qualities of vowel sounds to long synthetic textures, and how this synthesis technique can be considered a variation of Pulsar synthesis. Prony's method has been incorporated within Fry synthesis, to transfer the qualities of nasal sounds to synthetic textures. Appendix V includes a transcript of the Matlab code used to build a cross-synthesis system that includes Prony's method in the analysis part of the fry synthesis.

Recordings of dipthongs that match interpolation in openness (e.g uu to aa) and acuteness (e.g uu to ii) as well as /m/ and /l/ consonants are used to create long textures in the first section of the piece.

The algorithm designed to synthesize the fry textures (see Figure 43) works in the following steps:

- It allows the user to define the mono audio file used to approximate the filter (Sound B), another mono audio file designed to be used as source of the filter (Sound A), and a stereo file (Sound C) to be used for convolution with the output signal of the filter (hybrid sound A and B). The algorithm does not perform spectral whitening on the signal used as source.
- It allows the user to set the order of poles and zeroes approximated by the audio analysis.
- It performs a frame-by-frame audio analysis of sound B
- It filters sound A with the results of the analysis of sound B, using a frame-by-frame arrangement to model variation over time of the resonant and anti-resonant properties of sound B.
- It allows the user to render the resulting hybrid sound between sounds A and B.
- It convolves the hybrid output of the cross-synthesis process with another stereo audio sound source C. In this chapter, stereo white noise has been used.
- It renders the result of the convolution.

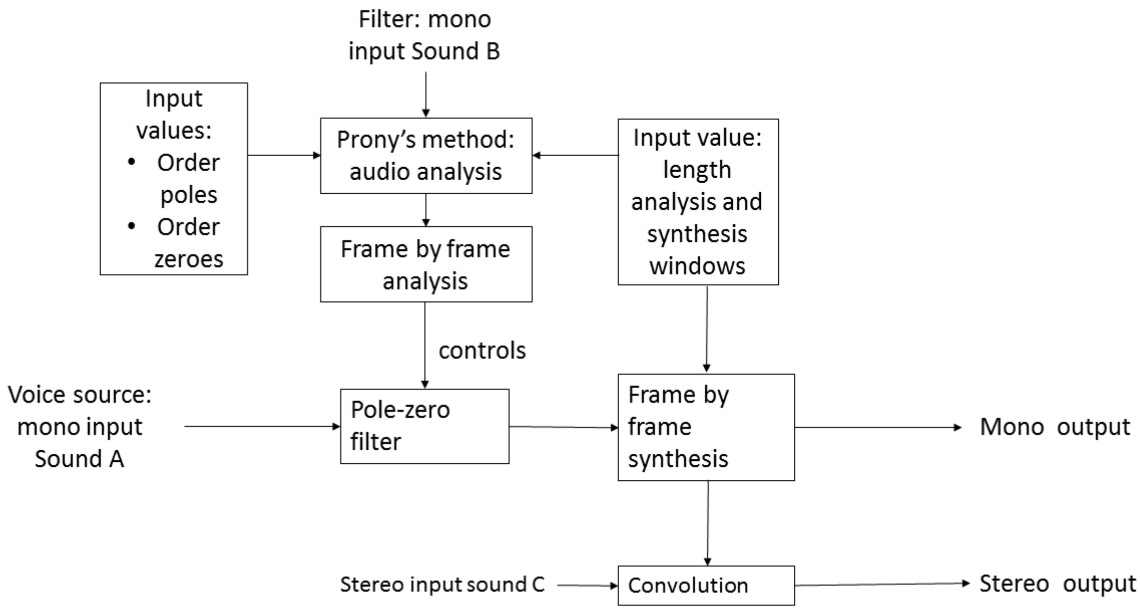


Figure 43. Schematics of a vocal fry convolution texture synthesizer based on Prony's method.

This approach to Prony's method allows one to expand the techniques developed with LPC in Chapter 4 by and provides a way to estimate the zeros as wells as poles from recorded speech (or any recorded sound) for the synthesis vocal fry-like. This enables the user to create vocal fry sounds from recordings of nasal consonants and to use their qualities, as well as the qualities of vowels, in fry texture synthesis. (Refer to Example 3.5).

The use of a frame-by-frame analysis with Prony's method might cause attenuations in the overall magnitude of certain frames. Chapter 3 provided an overview of how Prony's method computes the filter coefficients of both the numerator and denominator of the transfer function. The jumps can be related to the phase of the sound analysed for the way the coefficients are computed.

To reduce the difference in magnitude of the frames affected by the attenuation, two Prony analyses are run in parallel:

- One analysis uses a hanning window, the resulting coefficients of the numerator are used to compute the energy of the frame. By using the following code where *en* is the energy

- The other analysis uses a square rooted hanning window to provide a smoother result in the frame by frame synthesis of the filter.

The values of the energy computed from the coefficients of the numerator are then used in a normalisation algorithm within the frame-by-frame iteration:

```
frame =
filter(coeffb(j,:),coeffa(j,:),source(k:k+framelen-
1)).*w; % filters a frame of the input source
%sound with numerator and denominator coefficients and
multiply the frame by a square rooted hanning window


framenormen =
(frame./(sqrt(en)+constant)).*w2;%performs
%normalization from numerator coefficients


out(k:k+framelen-1)=out(k:k+framelen-1)+framenormen;
```

In the code above the energy en is the value of the energy of the frame of the analysis derived from the numerator coefficients (zeros), the variable constant is provided to the user as control to improve the results of the normalisation and is usually set to 0.001. A higher or lower order of analysis might be required to decrease or increase this value to obtain the desired result.

The sounds in *Whispers* have been arranged according to the following order:

In section 1 (0-1'44'') the Prony based tool for fry synthesis has been used to produce overlapping layers of synthetic textures, transferring the qualities of fry of nasals and dipthongs to stereo white noise to create diffuse and evolving background sounds. In this interpretation of the vocal fry texture synthesizer the qualities of both vowels and nasals are transferred to synthetic sounds through the convolution with stereo white noise sounds. The new tool aims to expand the technique enabling the introduction of anti-resonances in this approach. The morphing system has been used to filter synthetic vocal fry sources; the slow changes in the filter over a random rhythmic source as vocal fry aims to make less perceptible the speech-like features of the filter computed by the interpolation tool. In section 2 (1'44'-2'5'') a sustained sound source of white noise is filtered in succession with interpolations in sound colour performed with the analysis of nasals/laterals morphing into vowels and vowels morphing into other vowels. The idea was to blur the qualities of voice by overlaying the effect of the time-varying resonant structures of different interpolations between vowels and consonants as they emerge from (or disappear into) the fry synthetic textures.

In section 3 (2'5''-3'14'') the perception of the features of voice are fully revealed by filtering sustained subharmonic sources to produce a sound similar to throat singing dipthongs and transforming nasals and lateral consonants to vowels (and the opposite). Vocal fry sources with higher density and filtered with faster sound colour interpolations are combined with the synthetic textures and synthetic singing voice interpolations. Successive filtering techniques applied to white noise are used to blend the rhythmic features of fry sources with the background textures.

As in Chapter 4, pitch transpositions that cause time stretching (slower playback speed) can be used on the Fry texture synthesizer to produce strata of evolving resonances derived from the analysis of speech. In this case the author explored audio analysis of speech through Prony's method, but any recorded sound source can be explored with the new tools provided in this chapter.

Figure 44 is a sketch of the structure of *Whispers* in which different types of sounds have been organised to alter the perception of the human voice. It is an interpretation of the techniques used to manipulate the source and the filter during the arc of the piece, in order to explore how the qualities of speechlike sounds can be altered and revealed through the manipulation of the parameters of the new tools developed.

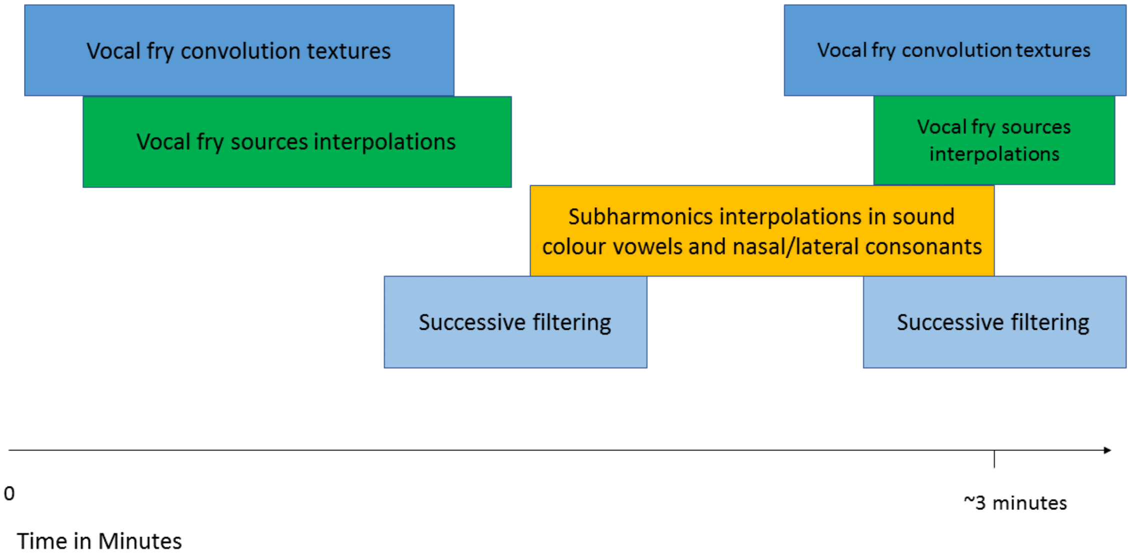

Vocal fry convolution textures
Vocal fry sources interpolations
Subharmonics interpolations in sound colour vowels and nasal/lateral consonants
Successive filtering
Vocal fry convolution textures
Vocal fry sources interpolations
Successive filtering
0
~3 minutes
Time in Minutes

Figure 44. Sketch of the overall structure of *Whispers* to guide the creative process and interaction with the material of the item.

Chapter 7

This chapter outlines how the output of the techniques developed in the previous chapters answer the research questions described in Chapter 1.

7.1 How can digital signal processing techniques enable the creation of tools to better represent the qualities of the voice in order to extend, improve or generate new opportunities for timbre control?

Chapter 4 explores how vowels emerge from resonance and how they are perceived as a single entity by dissecting their resonant structure through the use of a formant synthesizer in parallel and in cascade. The piece *Vocal Fry* provides an example of a collection of techniques that use vocal fry voice source synthesis for the excitation of a time-varying resonator that is approximated from recorded speech with LPC analysis. The combination of vocal fry synthesis and convolution with stereo white noise sources investigates the design of long, synthesized textures of sound to create a stratification of resonances that are not perceived as speech, even though these are modelled using knowledge of the human voice. The order in which audio processing techniques are performed on synthetic sounds and the type of vocal fry sources synthesized from recorded voice enable one to hide the features of speech. The qualities of the voice are transferred to introduce variations in the formant structure of the vocal fry sources. The percussive features of vocal fry are then smoothed and time-stretched through convolution. Further stratification of these resonant structures, through multiple pitch transposition of the resulting texture, produce sounds that do not resemble the original speech sounds nor are perceived as voice. The aim is to consider the human voice as an input to control audio processing devices which enable the design of sounds that have little resemblance to the original vowels sounds they were derived from.

Chapter 5 explores a collection of techniques that provide examples of the use of physical models of the vocal tract: *Fricatives* investigates how a two-tube waveguide resonator can enable the use of extreme physical descriptors that go beyond the physical constraints of the human body. A waveguide vocal tract model provides a simple and intuitive way to model the qualities of speech. However, it is an extreme manipulation of physical descriptors; at the same time, the 1D model provides a more stable filter suitable to explore exaggerated parameters. Therefore, the techniques described provide a way to control the perception of vowels through intuitive changes of parameters (e.g. tube lengths, feedback coefficients). Furthermore, Chapter 5 explores the use of recorded speech as excitation of a computer model of the vocal tract. For instance, fricative sounds and unvoiced stop consonants are fed as input excitation of the vocal tract model to investigate the idea of technology as an extension of the performer's body. Vowel sounds gradually emerge from consonant sounds, and Chapter 6 explores how the spectral envelope of both nasal consonants and vowel sounds can be captured to control filters in order to perform interpolations in sound colour. The resulting permutations in sound colour of speech are then transferred onto another sound, fed as input of the time-varying filter.

This tool demonstrates the importance for certain sounds of anti-resonance offering greater control from speech.

7.2 In what new ways can the mathematical manipulation of poles and zeros provide novel techniques to create meaningful timbre transformations?

In Chapter 6 an approach to Prony's method allows for the inclusion of zeros as well as poles in the filter approximated from recorded speech.

The chapter investigates the analysis of speech sounds that display, at least partly, zeros (valleys, dips) in their magnitude response and looks at how these can be used to control transformations of sound over time. A collection of techniques is described to provide examples of filter coefficients manipulation captured from recorded speech and brought to the recording studio. This resulted in a new tool that approximates zeros from audio analysis and gradually morphs the shapes of the spectral envelope of nasal consonants into vowels. This would not be possible with LPC as it doesn't model zeroes.

The morphing system called ZeroSynth provides an interface to collect a starting and target spectral envelope from two recorded sounds. Examples of interpolations between filters approximated from nasal and vowel sounds can be heard in *Whispers.* The piece explores the perception of interpolations between vowel colour and nasal/lateral consonant colours inspired by Slawson's theory of sound colour.

The perception of speech is explored by time-stretching interpolations between vowels and nasal/lateral consonants filtering vocal fry sources. Furthermore, overlays of sound colours changing over time are explored by means of successive filtering (see chapter 6). An example of the application of this technique is used to filter (in cascade) a white noise source: The output of a sound colour interpolation is filtered again with a different interpolation in sound colour. Reiterating this process two or three times blurs the speech-like features of the resulting sound.

Furthermore, this approach to Prony's method (representing the vocal tract as a time-varying resonator as well as resonator) has been applied to the techniques explored in chapter 4. *Whispers* provides an example of the use of a new tool that allows the excitation of a pole-zero filter with vocal fry sources and allows the convolution with stereo white noise sources (or any input stereo file). This allows for the inclusion of the anti-resonant features of nasal sounds (zeroes) within the sound design process of evolving textures of sound. Which, again, is not possible with LPC.

7.3 To what extent can speech processing be used creatively to emulate any existing audio processing techniques?

In Chapter 4, knowledge of the human voice is used to develop techniques that have elements in common with pulsar synthesis combined with convolution.

Furthermore, the Fry texture synthesis technique often produces an effect similar to reverberation, with a very long decay parameter that blurs the position in the stereo image, similar to diffuse reverberation through convolution with white noise samples.

In Chapter 5, *Fricatives* provides an example of a sound similar to chorusing by mixing layers of vocal tract models. For instance, let us consider the main layer as a sustained pattern of resonances produced by the excitation of a particular set of parameters of the vocal tract model with a sustained fricative sound. It is possible to create variation similar to modulation effects in two steps:

1. Layering in multi-track two copies of a similar sound synthesized with slightly different parameters (e.g. shorter model length)

2. Writing two different volume automations on the level parameter for each of the copies synthesized. The aim of this is to cross-fade variations in the resonant structures to mimic variation in the delay time of the chorusing device.

   Also, chorus is usually feed-forward, this is achieved from high feedback in a vocal tract model. The idea was inspired by a choir: different singers with different vocal tract size often sing the same note. This chorusing effect is achieved by combining that of different vocal tract models, different in size.

   In chapter 6 the morphing system employs groups of zeros (notches) moving over time, this technique shares similarities with feed-forward flanger devices that produce an ensemble of notches moving their position over time in the magnitude response. However, in this system the notches are not constrained to be harmonically related and are extracted from the analysis of recorded sounds.

   Chapter 6 also includes an extended version of the Fry texture synthesizer described in chapter 4. It is inspired by convolution in combination with pulsar synthesis and it provides the design of evolving textures with convolution with white noise sources. As described earlier, this produces an effect similar to artificial reverberation, with a very long decay with diffuse qualities that blur the position of sound within the stereo image. The feature approximated from recorded voice controls that audible effect while variation in speech and type of speech sounds affect blending of resonant structures that are time-stretched and smoothed by convolution with white noise. The type of performance affects the speed of these changes in formant structure. For instance, using recordings of shorter dipthongs in combination with a higher density vocal fry, allows the features of the recorded voice and the synthetic voice source to interact to create movement in the resulting anti-resonant and resonant structures.

7.4 What is the role of the vocalist (or speaker) when the required performance is designed for sound analysis and audio processing?

In *Vocal fry* (Chapter 4) diphthongs are recorded in advance to control the formant structure of the evolving textures.

In Chapter 5 the performer employs knowledge of speech synthesis techniques to produce a collection of recorded fricative sounds and stop consonants to be used as voice source of a physical model. In *Fricatives*, the vocalist exaggerates the performance of unvoiced speech sounds to hide the phonation qualities of speech. Furthermore, the performer applied variation in dynamics (crescendo, diminuendo) to the sustained unvoiced sources. The aim was to introduce variations and interest to a noise source originated by a human performer, processed by a computer model that expands both the possibilities and the vocal tract of the performer. The equivalent in sound synthesis would be to apply an amplitude envelope to create variation in loudness to a white noise source. Stop consonants have been recorded with different dynamics (piano, mezzo piano, as loud as possible) to introduce variations in the collages of sounds used to perform the transformation from consonant to vowel.

In chapter 6, the performers create a collection (palette) of vowel and nasalised sounds to control the interpolation of the morphing system, the performance provides the initial and the goal parameters. The morphing system computes intermediate states of the recorded performance. Furthermore, the vowel sounds from Brazilian Portuguese (vocalist 1) are morphed into Italian nasal consonants (vocalist 2).

The performers affect the starting and target frequencies and bandwidth of both anti-resonances (zeros) and resonances (poles) that are captured and processed by the morphing system PZeroSynth.

The outcome of the research presented throughout the previous chapters has demonstrated new ways of shaping the human voice, and how it can be used to control studio-based processes. This has been achieved by drawing on a range of ideas and tools from the fields of phonetics to telecommunications. The work includes a study on vowel resonance which uses LPC and formant synthesizers to explore different ways to isolate, extract and blend formants. Furthermore, how processing techniques can be employed to transfer the resonant structures of vowels to other sound sources has been explored. Recorded speech sounds have been used as an input to a vocal tract computer model that offers an intuitive way to both blend technology with human performance and to go beyond the physiological constraints of the human vocal tract. A pole-zero system based on Prony’s method has been designed to extract more accurate data from nasal sounds. It offers a new approach for performing smooth timbre manipulations, combining mathematical manipulation of the filters derived from recorded speech with non-overlapping windows in the frame-by-frame synthesis engine. The result is a collection of new voice processing techniques and tools that open a range of new possibilities for imaginative and expressive sound transformations. Some of these new options have been demonstrated through the production of musical examples that showcase different approaches to harness the potential of the human voice as a platform for creative sound design. These examples examine the limitations of the physiology of the vocal tract and LPC. In particular, they creatively combine knowledge of the human voice to find ways to go beyond such limitations. This is achieved in the form of new tools that allow for the exploration of sonic possibilities beyond those limits.

This research and experimentation have led to the development of a new tool that allows extended vocal techniques to be interpreted into extended processing techniques to create new and meaningful sounds, which previously would have been unobtainable. The knowledge of the human voice enables us to design and combine different techniques to control timbre in music by using recorded performance to shape the sound transformations. The resonant qualities of the voice are indeed important to design such transformations. However, so are anti-resonances: the techniques discussed in previous chapters and demonstrated in the pieces and in the tools that accompany this thesis show what can be achieved in expressive terms when we deal with anti-resonances as well as resonances.

Appendix

Appendix I: Example Cascade Synthesizer

```
<CsoundSynthesizer>
<CsOptions>
</CsOptions>
<CsInstruments>

sr = 44100
kr = 4410
ksmps = 10
nchnls = 2

instr 1

iamp  = p4
ifreq = p5
inharm= p6
ifn   = p7
ifc   = p8
ifc2  = p9
ifc3  = p10
iband = p11
iband2= p12
iband3= p13
igainsx= p14
igaindx= 1-igainsx

kenv expseg 0.0001,p3*0.05,1,p3*0.9,1,p3*0.05,0.0001
kmod expseg 0.0001,p3/2,10,p3/2,120
kvar randi 0.5,0.1
```

```
asig buzz iamp*kenv,ifreq+(kmod+kvar),inharm,ifn
afilt butterlp asig,ifreq

a1   butterbp afilt,ifc,iband
a2   butterbp a1,ifc2,iband2
a3   butterbp a2,ifc3,iband3
a4   balance a3,asig

outs  a4*igainsx,a4*igaindx
endin

</CsInstruments>

<CsScore>
f1    0     16384 10    1
;..............................................................
........................................
;p1   p2    p3    p4    p5    p6    p7    p8    p9    p10   p11
;     p12   p13   p14

;i1   0     70    4000  7     70    1     570   840   2410  30 //
      100     250      0.7
```

Appendix II: Example Isolated formant: Csound instrument

```
<CsoundSynthesizer>

<CsOptions>
-W  -oe:/comp1november/notes/Note4c.wav
</CsOptions>
<CsInstruments>

sr = 44100
kr = 4410
ksmps = 10
nchnls = 2



instr 3

iamp      = p4 ;amp noise
ifc       = p5 ;freq c formant
iband     = p6 ;bw
ifreqenv  = p7 ;freq envelope
ifnenv = p8 ;envelope window
igainsx= p9 ;panning
igaindx= 1-igainsx

kenv oscili 1,ifreqenv/p3,ifnenv
asig rand iamp*kenv

a1   butterbp asig,ifc,iband
a2   butterbp a1,ifc,iband
a3   butterbp a2,ifc,iband
```

```
a4   balance a3, asig
outs a4*igainsx, a4*igaindx
endin
</CsInstruments>
<CsScore>
f1   0    8192 10   1; sinusoid
f2   0    8192 20      2; Hanning
;p1  p2   p3   p4   p5   p6   p7   p8   p9   p10  p11
;.......................................................
..................................................
;p1  p2   p3   p4   p5   p6   p7   p8   p9   p10  p11
     p12  p13  p14
;i3  0    5    1000 390  105  1    2    0.4
</CsScore>
</CsoundSynthesizer>
```

Appendix III: Two-tube waveguide vocal tract: Matlab code

```
Fs=48000;

extension= zeros(Fs*5,1);% define seconds silence to add
imp=audioread('my_input_sound_name.wav');
imp = [imp;extension];% add senconds of silence to allow
fdbk to end

l1=28.8; %length tube1
l2=18.8; %.8ength tube2
a1=8;    %area tube 1
a2=1;    %area tube 2

if a1<a2;
    imp=imp.*0.3;
else imp=imp.*1;
end

k=((a2-a1)/(a1+a2)); %k set to 0 in the formula

c =35000;    %speed of sound
f1=c/(4*l1); %convert length to frequency
f2=c/(4*l2); %length to frequency
tau1=1/f1;    %frequency to delay in time
tau2=1/f2;    %frequency to delay in time
N =fix((Fs*tau1)/4);%delay tube 1 conversion to 2 delay
lines in samples
N2=fix((Fs*tau2)/4);%delay tube 2 conversion to 2 delays
lines samples


rl=0.78;   %reflection lips
rg=-0.72 ; %reflection glottis negative introduces
%negative phase pulses and halves the fundamental
frequency


%initializations
out=imp;
out2=imp;
out3=imp;
out4=(imp);
adel1=zeros(length(imp),1);
adel2=zeros(length(out),1);
adel3=zeros(length(out2),1);
adel4=zeros(length(out3),1);
%Tube model
for n=1:length(imp);

    out(n)=adel1(N);
```

```
        out2(n)=adel2(N2);
        out3(n)=adel3(N2);
        out4(n)=adel4(N);
        adel1=[imp(n)+rg*(out4(n));adel1(1:N-1)];
        adel2=[(out(n)*(1+k))+(out3(n)*-k);adel2(1:N2-1)];
        adel3=[out2(n)*rl;adel3(1:N2-1)];
        adel4=[(out3(n)*(1-k))+(out(n)*k);adel4(1:N-1)];

end;
%plotting and analysis section
plot(out2);
diff(find(out2~=0));
freqz(out2,1,2^17,Fs)

audiowrite('render_my_output_file_name.wav',out2,Fs);
```

Appendix IV: Examples Morphing System: Matlab code

```
%%%%%%%% Input sounds%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
Fs = 48000;

%%%% Sounds for analysis%%%%%%%%%
speech = audioread('sound_b_name_file.wav');

speech2 = audioread('sound_c_name_file.wav');

marker  = floor(0.61*Fs); %change values to move marker
marker2 = floor(51.88*Fs);%and capture a different
spectral envelope

endmarker = 2000; %lenght in samples window of analysis
vowel1 = speech(marker:marker+endmarker-1);

vowel2 = speech2(marker2:marker2+endmarker-1);


sig = audioread('sound_a_name_file.wav');
sig = sig(1:floor(Fs*9.8));

att  = linspace(0,1,(Fs/1000)*2);

dec  = linspace(1,0,(Fs/1000)*2);
sus  = ones(length(sig)-length(att)-length(dec),1)';
env2 = [ att sus dec]';

sig  = sig.*env2;

framelen = floor(Fs/20); %e.g 2400 % lenght in sample
window synthesis

hop      = floor(framelen/4);
hopsize  = floor(framelen-(framelen/4));

ninterp = floor(length(sig)/floor(framelen-hopsize));

nframes  = ninterp;



added    = framelen;

w        = hanning(framelen);
```

```
wmark    = hanning(endmarker);
wmark2   = triang(endmarker);
wrect    = rectwin(hop);


%initializations
sig = [zeros(added,1);sig;zeros(added,1)];

out = zeros(length(sig),1);


zi = zeros(nframes,10);


%%%%%%%%%% Prony analysis section
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%


[num1,den1]= prony(vowel1.*wmark2,6,10);

rootszeros1 = roots(num1);
rootspoles1 = roots(den1);

[num2,den2]= prony(vowel2.*wmark2,6,10);

rootszeros2 = roots(num2);
rootspoles2 = roots(den2);



%zero_extraction coordinates sound1


values_zeros1 = rootszeros1(find(imag(rootszeros1)>=0);
rz1 = real(values_zeros1(1,:));
rz2 = real(values_zeros1(2,:));
rz3 = real(values_zeros1(3,:));


iz1 = imag(values_zeros1(1,:));
iz2 = imag(values_zeros1(2,:));
iz3 = imag(values_zeros1(3,:));

%pole_extraction coordinates sound1

values_poles1 = rootspoles1(find(imag(rootspoles1)>=0));

rp1 = real(values_poles1(1,:));
```

```
rp2 = real(values_poles1(2,:));
rp3 = real(values_poles1(3,:));
rp4 = real(values_poles1(4,:));
rp5 = real(values_poles1(5,:));

ip1 = imag(values_poles1(1,:));
ip2 = imag(values_poles1(2,:));
ip3 = imag(values_poles1(3,:));
ip4 = imag(values_poles1(4,:));
ip5 = imag(values_poles1(5,:));

%pole_extraction coordinates sound2

values_poles2 = rootspoles2(find(imag(rootspoles2)>=0));

rp1b = real(values_poles2(1,:));
rp2b = real(values_poles2(2,:));
rp3b = real(values_poles2(3,:));
rp4b = real(values_poles2(4,:));
rp5b = real(values_poles2(5,:));

ip1b = imag(values_poles2(1,:));
ip2b = imag(values_poles2(2,:));
ip3b = imag(values_poles2(3,:));
ip4b = imag(values_poles2(4,:));
ip5b = imag(values_poles2(5,:));

%zero_extraction coordinates sound2

values_zeros2 = rootszeros2(find(imag(rootszeros2)>=0));
rz1b = real(values_zeros2(1,:));
rz2b = real(values_zeros2(2,:));
rz3b = real(values_zeros2(3,:));

iz1b = imag(values_zeros2(1,:));
iz2b = imag(values_zeros2(2,:));
iz3b = imag(values_zeros2(3,:));

%Zeros  input values
section....................................
realpartz1 = linspace(rz1,rz1b,ninterp);
imapartz1 = linspace(iz1*i,iz1b*i,ninterp);

realpartz2 = linspace(rz2,rz2b,ninterp);
imapartz2 = linspace(iz2*i,iz2b*i,ninterp);

realpartz3 = linspace(rz3,rz3b,ninterp);
imapartz3 = linspace(iz3*i,iz3b*i,ninterp);
%.....................................................
........
```

```
%try add linspace to 'real' variable
zero1 = realpartz1+imapartz1;
conjuz1=  realpartz1-imapartz1;

b1= -(zero1+conjuz1);
b2= (zero1.*conjuz1);

b0= ones(size(b2));



zero2 = realpartz2+imapartz2;
conjuz2=  realpartz2-imapartz2;

b12= -(zero2+conjuz2);

b22= (zero2.*conjuz2);

b02= ones(size(b22));

coeffb2 =[b02;b12;b22]';


zero3 = realpartz3+imapartz3;
conjuz3=  realpartz3-imapartz3;

b13= -(zero3+conjuz3);

b23= (zero3.*conjuz3);

b03= ones(size(b23));

coeffb3 =[b03;b13;b23]';

b1z12 = (b0.*b12)+(b1.*b02);

b2z12 = (b0.*b22)+(b1.*b12)+(b2.*b02);

b3z12 = (b1.*b22)+(b2.*b12);

b4z12 = (b2.*b22);

b0z12 =ones(size(b2z12));

b1tot = (b0z12.*b13)+(b1z12.*b03);

b2tot = (b0z12.*b23)+(b1z12.*b13)+(b2z12.*b03);

b3tot = (b1z12.*b23)+(b2z12.*b13)+(b3z12.*b03);
```

```
b4tot = (b4z12.*b03)+(b3z12.*b13)+(b2z12.*b23);

b5tot = (b4z12.*b13)+(b3z12.*b23);

b6tot = (b4z12.*b23);

b0tot =ones(size(b2tot));
coeffbz12 = [b0z12;b1z12;b2z12;b3z12;b4z12]';
coeffbtot =
[b0tot;b1tot;b2tot;b3tot;b4tot;b5tot;b6tot]';
%Pole  input values
section.....................................
%p1
realpart1 = linspace(rp1,rp1b,ninterp);
imapart1 = linspace(ip1*i,ip1b*i,ninterp);
%p2
realpart2 = linspace(rp2,rp2b,ninterp);
imapart2 = linspace(ip2*i,ip2b*i,ninterp);
%p3
realpart3 = linspace(rp3,rp3b,ninterp);
imapart3 = linspace(ip3*i,ip3b*i,ninterp);
%p4
realpart4 = linspace(rp4,rp4b,ninterp);
imapart4 = linspace(ip4*i,ip4b*i,ninterp);
%p5
realpart5 = linspace(rp5,rp5b,ninterp);
imapart5 = linspace(ip5*i,ip5b*i,ninterp);
%.......................................................
.........
%try add linspace to 'real' variable
%p1 conj
pole1 = realpart1+imapart1;
conju1=  realpart1-imapart1;

a1= -(pole1+conju1);
a2= (pole1.*conju1);

a0= ones(size(a2));

%p2 conj

pole2 = realpart2+imapart2;
conju2=  realpart2-imapart2;

a12= -(pole2+conju2);

a22= (pole2.*conju2);

a02= ones(size(a22));
```

```
coeffa2 =[a02;a12;a22]';


%p3 conj
pole3 = realpart3+imapart3;
conju3=  realpart3-imapart3;

a13= -(pole3+conju3);

a23= (pole3.*conju3);

a03= ones(size(a23));

coeffa3 =[a03;a13;a23]';

%p4 conj
pole4 = realpart4+imapart4;
conju4=  realpart4-imapart4;

a14= -(pole4+conju4);

a24= (pole4.*conju4);

a04= ones(size(a24));

coeffa4 =[a04;a14;a24]';


%p4 conj
pole5 = realpart5+imapart5;
conju5=  realpart5-imapart5;

a15= -(pole5+conju5);

a25= (pole5.*conju5);

a05= ones(size(a25));

coeffa5 =[a05;a15;a25]';


%----------------------------------------
%P1*P2
%calculates coefficients first two poles
a1p12 = (a0.*a12)+(a1.*a02);

a2p12 = (a0.*a22)+(a1.*a12)+(a2.*a02);

a3p12 = (a1.*a22)+(a2.*a12);
```

```
a4p12 = (a2.*a22);

a0p12 =ones(size(a2p12));
%P1*P2*P3
%adding third pole
a1p123 = (a0p12.*a13)+(a1p12.*a03);

a2p123 = (a0p12.*a23)+(a1p12.*a13)+(a2p12.*a03);

a3p123 = (a1p12.*a23)+(a2p12.*a13)+(a3p12.*a03);

a4p123 = (a4p12.*a03)+(a3p12.*a13)+(a2p12.*a23);

a5p123 = (a4p12.*a13)+(a3p12.*a23);

a6p123 = (a4p12.*a23);

a0p123 =ones(size(a1p123));

%P1*P2*P3*P4
%adding fourth pole
a1p1234  = (a1p123.*a04)+(a0p123.*a14);

a2p1234  = (a0p123.*a24)+(a1p123.*a14)+(a2p123.*a04);

a3p1234  = (a1p123.*a24)+(a2p123.*a14)+(a3p123.*a04);

a4p1234 = (a2p123.*a24)+(a3p123.*a14)+(a4p123.*a04);

a5p1234  = (a3p123.*a24)+(a4p123.*a14)+(a5p123.*a04);

a6p1234  = (a4p123.*a24)+(a5p123.*a14)+(a6p123.*a04);

a7p1234  = (a5p123.*a24)+(a6p123.*a14);

a8p1234  = (a6p123.*a24);

a0p1234  =ones(size(a2p1234));

%P1*P2*P3*P4*P5
a1ptot   = (a0p1234.*a15)+(a1p1234.*a05);
a2ptot   = (a0p1234.*a25)+(a1p1234.*a15)+(a2p1234.*a05);
a3ptot   = (a1p1234.*a25)+(a2p1234.*a15)+(a3p1234.*a05);
a4ptot   = (a2p1234.*a25)+(a3p1234.*a15)+(a4p1234.*a05);
a5ptot   = (a3p1234.*a25)+(a4p1234.*a15)+(a5p1234.*a05);
a6ptot   = (a4p1234.*a25)+(a5p1234.*a15)+(a6p1234.*a05);
a7ptot   = (a5p1234.*a25)+(a6p1234.*a15)+(a7p1234.*a05);
a8ptot   = (a6p1234.*a25)+(a7p1234.*a15)+(a8p1234.*a05);
a9ptot   = (a7p1234.*a25)+(a8p1234.*a15);
a10ptot  = (a8p1234.*a25);
```

```
a0ptot   = ones(size(a2ptot));


coeffap12 = [a0p12;a1p12;a2p12;a3p12;a4p12]';
coeffap123 =
[a0p123;a1p123;a2p123;a3p123;a4p123;a5p123;a6p123]';
coeffap1234 =
[a0p1234;a1p1234;a2p1234;a3p1234;a4p1234;a5p1234;a6p1234
;...
    a7p1234;a8p1234]';

coeffaptot =
[a0ptot;a1ptot;a2ptot;a3ptot;a4ptot;a5ptot;a6ptot;a7ptot
;...
    a8ptot;a9ptot;a10ptot]';


  for n = 1:ninterp
 zplane(coeffbtot(n,:),coeffaptot(n,:));
 pause(0.00012)   %speed repetition e.g if 1 computes a

  end


% frame Rectangular window, synthesis
for j=1:nframes
    k = (framelen-hopsize)*j;


[frame,zf] =
filter(coeffbtot(j,:)',coeffaptot(j,:)',sig(k:k+hop-
1),zi(j,:));
 zi(j+1,:)= zf;

monitor(j,:)= k;

stablec(j,:)=
isstable(coeffbtot(j,:)',coeffaptot(j,:)');

    out(k:k+hop-1)=out(k:k+hop-1)+(frame.*wrect);

end
gainnoclip = 0.9;

norm_out = out/max(abs(out)).*gainnoclip;
plot(norm_out)
```

Appendix V: Example Fry Texture synthesis: Matlab code

```
clear
Fs = 48000;



sig = audioread('input_file_name_prony_analysis.wav');

%Select segment for analysis (or beginning end file)
starts = 0; %change value to select start time analysis
sound
ends   = 5;  %change value to select end time (>starts)
startl = floor(Fs*starts);
endl    = floor(Fs*ends);
sig = sig(startl:endl);


frystart = floor(Fs*11.7);
fryend = floor(Fs*14.6);

noise =
audioread('input_excitation_filter.wav');%Excitation
signal
% for instance,vocalfry source
noise = noise(frystart:fryend);


%framing analysis and filtering




framelen = 2400; %lenght frame analysis and synthesis

hopsize   = floor(framelen-(framelen/2));
%the bigger hopsize the more overlaps

added     = framelen;

w         = sqrt(hanning(framelen));
w2         = hanning(framelen);


nframes   = floor(length(sig)/floor(framelen-hopsize));
%initializations
sig = [zeros(added,1);sig;zeros(added,1)];
noise = [zeros(added,1);noise;zeros(added,1)];
out      = zeros(length(sig),1);
outlpc   = zeros(length(sig),1);
```

```
orderzeros = 100;
orderpoles = 140;

orderen = 100;

constant = 0.0001;%0.000000000000001;



tic
 for j=1:nframes
     k= (framelen-hopsize)*j;
 %
[num,den]= prony((sig(k:k+framelen-
1).*w),orderzeros,orderpoles);
[numen,denen]= prony((sig(k:k+framelen-
1).*w2),orderen,orderen);


coeffb(j,:) = num;
coeffben(j,:) = numen;

en = sum(coeffben(j,:).^2);

enmonitor(j,:) = en;
coeffa(j,:) = den;


stablec(j,:)= isstable(coeffb(j,:),coeffa(j,:));


frame =
filter(coeffb(j,:),coeffa(j,:),noise(k:k+framelen-
1)).*w;

framenormen = (frame./(sqrt(en)+constant)).*w2;

out(k:k+framelen-1)=out(k:k+framelen-1)+framenormen;


 end
 toc

gainnoclip = 0.9;

norm_out = out/max(abs(out)).*gainnoclip; %normalization


sigconv =
audioread('insert_stereo_file_name_convolution.wav');
```

```
%the output of the analysis/synthesis is convolved with
sigconv

chn1 = sigconv(:,1);
chn2 = sigconv(:,2);
lconv = conv(chn1,norm_out);
rconv = conv(chn2,norm_out);
lconv = lconv/max(abs(lconv)).*gainnoclip;
rconv = rconv/max(abs(rconv)).*gainnoclip;

outconv = [lconv,rconv];
```

Appendix VI: PZeroSynth: Audio Interpolation System User guide

Step 1: Installation of PZeroSynth as standalone application (Windows only)

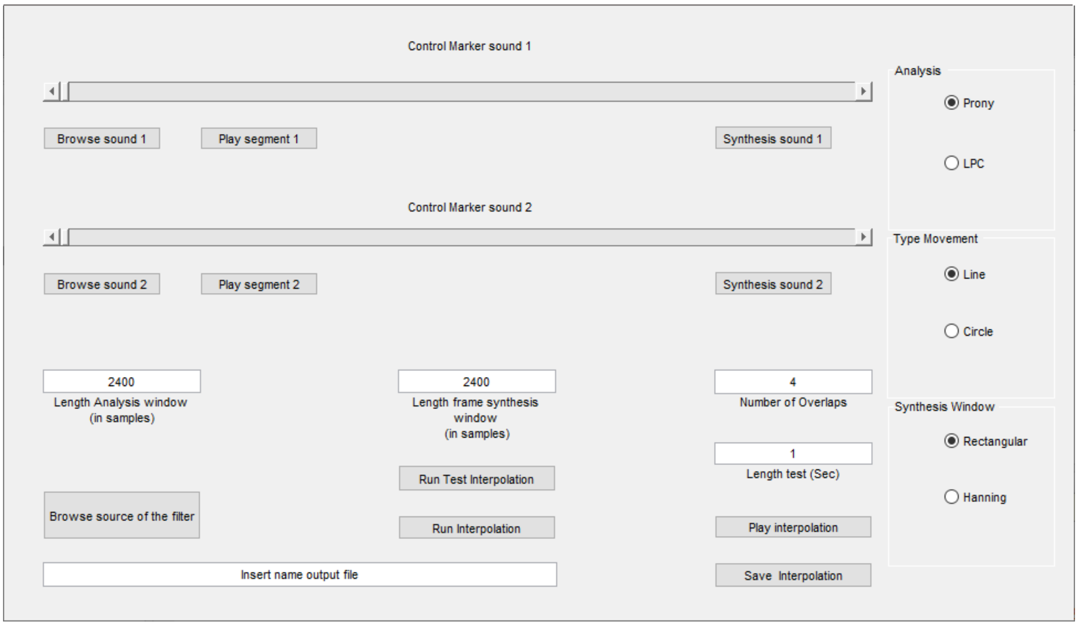
- To install PZeroSynth as standalone click on the folder ‘for_redistribution’ within the folder named ‘PZeroSynth’.
- Run the installer ‘PZeroSynth_Installer_web.exe’, please notice that connection to the internet is required to allow the installer to complete the installation of all the supporting software required for the application to operate correctly.
- Select the preferred path folder for installation of zero synth and MATLAB Runtime
- Follow the instructions on screen to complete the installation.

Installation of PZeroSynth as standalone application (Mac only)

- To install PZeroSynth as standalone click on the folder ‘for_redistribution’ within the folder named ‘PZeroSynth’.
- Run the installer ‘PZeroSynth_Installer_web’ application, please notice that connection to the internet is required to allow the installer to complete the installation of all the supporting software required for the application to operate correctly.
- Select the preferred path folder for installation of zero synth and MATLAB Runtime
- Follow the instructions on screen to complete the installation.
- To find PZeroSynth, click on ‘Application’ in ‘Finder’.

Step 2: Start PZeroSynth

To start the application, open the folder in which PZeroSynth have been installed on your computer (e.g. C:\Program Files\PZeroSynth) and click on ‘PZeroSynth.exe.
You should now be able to see the following interface:



Analysis: ● Prony ○ LPC

Type Movement: ● Line ○ Circle

Synthesis Window: ● Rectangular ○ Hanning

Length Analysis window (in samples): 2400

Length frame synthesis window (in samples): 2400

Number of Overlaps: 4

Length test (Sec): 1

Insert name output file

Operate PZeroSynth

PZeroSynth performs interpolation between the spectral envelope derived from 'Sound 1' and 'Sound 2'. 'Sound 1' represents the starting point of the interpolation while 'Sound 2' is the target point. This results in an audio filter morphing over time that the system uses to filter 'Source of the filter'. The user can specify what sounds 'Sound1', 'Sound2' will be and 'Source of the filter' by using the buttons on the interface to browse a collection of sounds.

Step 3: Upload sounds on PZeroSynth

Plese note: All audio files uploaded in PZeroSynth should have the **same sampling rate** and be in **mono** for the system to operate. To upload the sounds to PZeroSynth click on 'Browse sound1','Browse Sound 2' and 'Browse source of the filter'.

Controls on PZeroSynth:

- **Browse Sound 1:** Upload a sound samples to derive the starting filter of the interpolation. Click to browse the sounds previously uploaded in the 'application' folder (see Step 3).
- **Control Marker Sound 1:** Set the starting point of the audio analysis to take a snapshot of Sound 1 to extract the filter at the time selected. It will display the Spectral envelope that is used as the starting point of the interpolation.

Move toward the left to select a time closer to the beginning of the file uploaded, move to the right to select a time closer to the end of the file.

- **Play segment 1:** Allow to listen where the starting point of the analysis is on the audio file uploaded as Sound 1
- **Synthesis sound 1:** Allow to listen to the timbre of the filter 2 seconds of white noise with spectral envelope estimated by the analysis of 'Sound 1'.

- **Browse Sound 2:** Upload a sound samples to derive the target filter of the interpolation. Click to browse and upload sounds sources to PZeroSynth.

- **Control Marker Sound 2:** Set the starting point of the audio analysis to take a snapshot of Sound 2 to extract the filter at the time selected. It will display the Spectral envelope that is used as the end point of the interpolation. Move toward the left to select a time closer to the beginning of the file uploaded, move to the right to select a time closer to the end of the file.
- **Play segment 2:** Allow to listen where the starting point of the analysis is on the audio file uploaded as 'Sound 2'

- **Synthesis sound 2:** Allow to listen to the timbre of the filter 2 seconds of white noise with spectral envelope estimated by the analysis of 'Sound 1'. It automatically same the result in the 'application' folder as
- **Length Analysis Window:** Allows to specify the length in samples of the window of analysis for both 'Sound 1' and 'Sound 2'. As default 2400 samples. Shorter of longer windows length might provide different results in the interpolations that can be adapted to taste. **Please note:** after changing the Length of the analysis window the system needs to update the internal parameters: moving both Control Marker Sound 1 and Control Marker Sound 2 solves the resulting error.
- **Length Frame Synthesis Window:** Allows to specify the length in samples of the window of synthesis. As default 2400 samples. The shorter the window

the more interpolation points PZeroSynth will compute therefore it might increase the time of computation but provide more details.

- **Number of overlaps:** Number of overlapping windows for the synthesis of the interpolation. 4 as default. Using a value 1 provides no overlaps.
- **Analysis:** Allows to choose between Prony's method (Prony) or LPC to capture the spectral envelope of Sound1 and Sound 2. Prony provides a pole-zero model while LPC provides an all-pole model.
- **Type of movement:** Allows to select between Line and Circle. When Line is selected the poles and zeros move in straight line from the starting point to the target position. When Circle is selected the poles and zeros rotate around the origin to reach the target position.
- **Synthesis Window:** Allows to choose between Rectangular or Hanning windows for the synthesis of the interpolation. **When Rectangular is selected there are no overlaps. Increasing the number of overlaps with the Rectangular window is equivalent to using shorter frames.**

- **Browse Source of the filter:** Upload a sound sample to be fed as input of the interpolation filter. Click to browse the sounds previously uploaded in the 'application' folder.
- **Run test interpolation:** Allow to test the interpolation on a white noise source to fine tune the other parameters. 1 sec as default. This feature is useful to repeat several tests in a shorter amount of time and fine-tune the interpolation by changing the parameters. It automatically saves a file named 'interpolation_white_noise.wav' within the 'application' folder with the other sounds copied by the user (see Step 3).
- **Length test:** Allows to set the duration of the test interpolation. For instance, allow to time-stretch a test interpolation
- **Run interpolation:** Filter the sound uploaded as 'Source of the filter'. The duration of the interpolation will be the same as the duration of the sound file uploaded with the 'Browse source of the filter' button.

- **Play Interpolation:** Plays the interpolation Once the interpolation is complete. Pressing on this button might cause errors if no interpolation has been run yet.
- **Insert name output file:** Allows to specify the name of the output sound of the interpolation. To save the file add .wav in the end (e.g.my_interpolation.wav').
- **Save Interpolation:** Runs the interpolation and saves the output interpolation with the name specified by the user. Runs the interpolation again and open a menu to choose where to save the file on the computer.

Step 4: Example Interpolation

1. Open 'dipthongs1.wav','m_consonant.wav' and 'pulse_train.wav' in the 'application' folder
2. Click 'Browse sound 1' and upload 'dipthongs1.wav'
3. Click 'Browse sound 2' and upload 'm_consonant.wav
4. Click 'Browse source of the filter' and 'upload'.

5. Move 'Control Marker Sound 1' and 'Control Marker Sound 2' to set the starting and ending point of the interpolation

6. Use the features provided to design your interpolation.

7. If at any time any error happens after changing the parameters make sure you move again 'Control Marker Sound 1' and 'Control Marker Sound 2' and it should update the interpolation

8. If point 7 doesn't solve the problem consider repeating points 1,2,3.

9. **Warning:** Pressing any button without uploading the files with the Browse button will cause an error.

10. Make sure the output file is properly named before pressing ‘Save interpolation’ (see ‘Save Interpolation’ function)

This installation has been tested and installed on several Windows and Mac computers, if you experience any problem please visit www.pizzimusic.com to get in touch or email info@pizzimusic.com.

Step 4: Run PZeroSynth as a MATLAB GUI.

1. Click ‘Open’ from the HOME menu in MATLAB
2. Open the folder named ‘PZeroSynth_gui’ and click on ‘morphtwosegments.fig’
3. Set ‘PZeroSynth_gui’ as current folder
4. Alternatively, click on the ‘Browse for folder’ and select ‘PZeroSynth_gui’. You should now be able to see ‘morphtwosegments.fig’ in the ‘Current folder’ space and to click on the file.
5. The control and features of the GUI version for MATLAB are identical to the standalone application. Please refer the previous section for an overview of PZeroSynth’s features.

Appendix VII: Study on phase for frame-by-frame analysis with Prony's method.

The following partitions are implemented to uncouple the calculations over the numerator and denominator coefficients:[185] In Figure 45, **b** represents the vector *M*+1 that represents the numerator coefficients (zeros). The vector a* includes values from a_0 to the *Nth* coefficient for a_0= 1. The partition h_1 includes the values contained in the last *K-M* terms of the impulse response; H_1 is a (M+1) by (N+1) partition of the matrix containing the signal *h* (as shown *in* Fig (3.3)). The last partition H_2 consists of the elements (*K-M*)-by-*N.*[186]

The equations included in Figure 45 show how solutions to the equation a) can provide a way to compute to calculate the denominator coefficients. As h_1 is known in the equation b) it is possible to solve the equation in order to find the coefficients **a.** The equation c) shows how the coefficients **b** can be calculated by multiplying the coefficients **a** with the partition H_1.

a) $$\mathbf{0} = \boldsymbol{h}_1 + H_2\boldsymbol{a}^*$$

b) $$\boldsymbol{h}_1 = -H_2\boldsymbol{a}^*,$$

c) $$\mathbf{b} = H_1\boldsymbol{a},$$

Figure 45. Equations for computation of numerator coefficients. After Parks and Burrus.[187]

This technique shares similarities with the method of singular value decomposition.

It is a particularly iterative approach to trying to derive a set of values that best solves an equation.

[185] Parks and Burrus, *Digital Filter Design*, 226.
[186] Parks and Burrus, *Digital Filter Design*,226-227.
[187] Parks and Burrus, *Digital Filter Design*, 227.

A test signal can be used to demonstrate the ways in which the samples of an audio signal are affected by this method of matrix multiplication. For instance, considering the signal h, consisting of 10 values for h= 0,0,0,0,0,1,1,1,1,1; the resulting signal consists of a series of five samples of value 0 followed by five samples of value 1. This allows one to study where the ones are used in the matrix, giving an insight into how the phase or silence in the signal affects the result of the partition in the matrices.

This approach to Prony's method through the use of matrices is available in Matlab through the function *prony* and has that have been used to perform a series of tests on the signal h in order to study the features of the matrix partitions and explore their features to find the implications for sound synthesis.

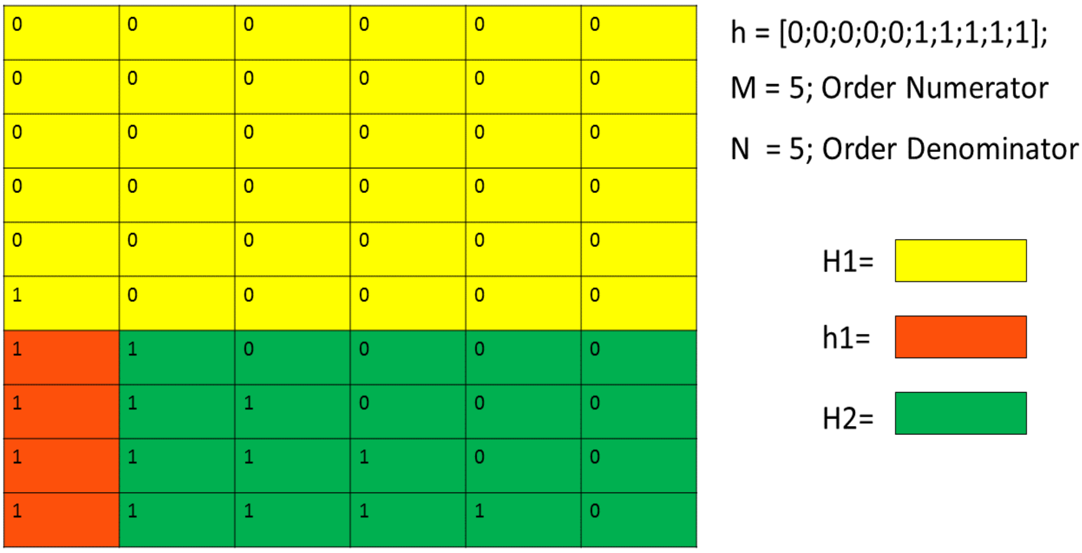
The algorithm derives the coefficients of the zeros (anti-resonances) from the coefficients of the poles (resonances) in combination with a portion of the samples of the signal (sound) analysed:

1. The samples part of signal *h* are arranged and in a Toeplitz matrix.
2. The resulting matrix is divided in smaller regions, each of these regions contain part of the input sequence of samples and part of the mirrored versions of the sequence.
3. The order of analysis of the numerator M and denominator N affect the size of these regions of the signal analysed: the higher the order of analysis the larger in size the regions (H1, h1, H2) of the matrix will be.
4. The coefficients of the denominator *a* (poles, resonances) are computed first by dividing dividend the matrix region -H2 by the matrix region h1
5. Finally, the coefficients of the numerator *b* (zeroes, anti-resonances) are computed by the multiplication of the coefficient of the denominator *a* multiplied with the samples of the signal included in the matrix H1.

Figure 46 shows how Prony's method has been used to analyse the signal h with an order of analysis of the numerator (zeros) M = 5, and an order analysis of coefficients of denominator (poles) N = 5. The bigger matrix H contains all the other partitions and represents the matrix in the centre of Figure 46. The values of the signal are displayed vertically on the left side, the rest of the values are equal vertically suggesting equal diagonal elements. The colours show how the matrix is partitioned.

For instance, H1 is a (M+1) by (N+1) matrix. For M=5, and N=5 it results in a 6-by-6 matrix. K can be defined as the length of the signal h-1 and allows to compute the other partitions, in this case K=9 as the signal h has a length of 10 samples. The partition h1 is the last K-M values of the signal so it appears in the bottom-left part of the matrix in a 4 x 1 matrix. The partition H2 is a (K-M)-by-N matrix so it is a 4 x 5 matrix.

Having seen how the matrix is organised, we now move on to consider a signal with an increasing series of numbers that can be used in order to investigate how the signal h is distributed within the matrix H. One can use a test signal consisting of ten samples with values increasing from 1 to 10 in order to detect the introduction of any zeros at any stage of the calculations. For example, the sequence *h* = 1,2,3,…,10 will reveal how the elements of the signal are mirrored within the different partitions of the matrix in order to find out what parts of the sequence are included in which partition of the matrix. This is relevant as it represents how a sequence of samples from an audio frame are mirrored. Furthermore, it appears that the method used to distribute the element of the sequence within the matrix introduces zeros.



0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0
1	1	1	1	0	0
1	1	1	1	1	0

h = [0;0;0;0;0;1;1;1;1;1];

M = 5; Order Numerator

N = 5; Order Denominator

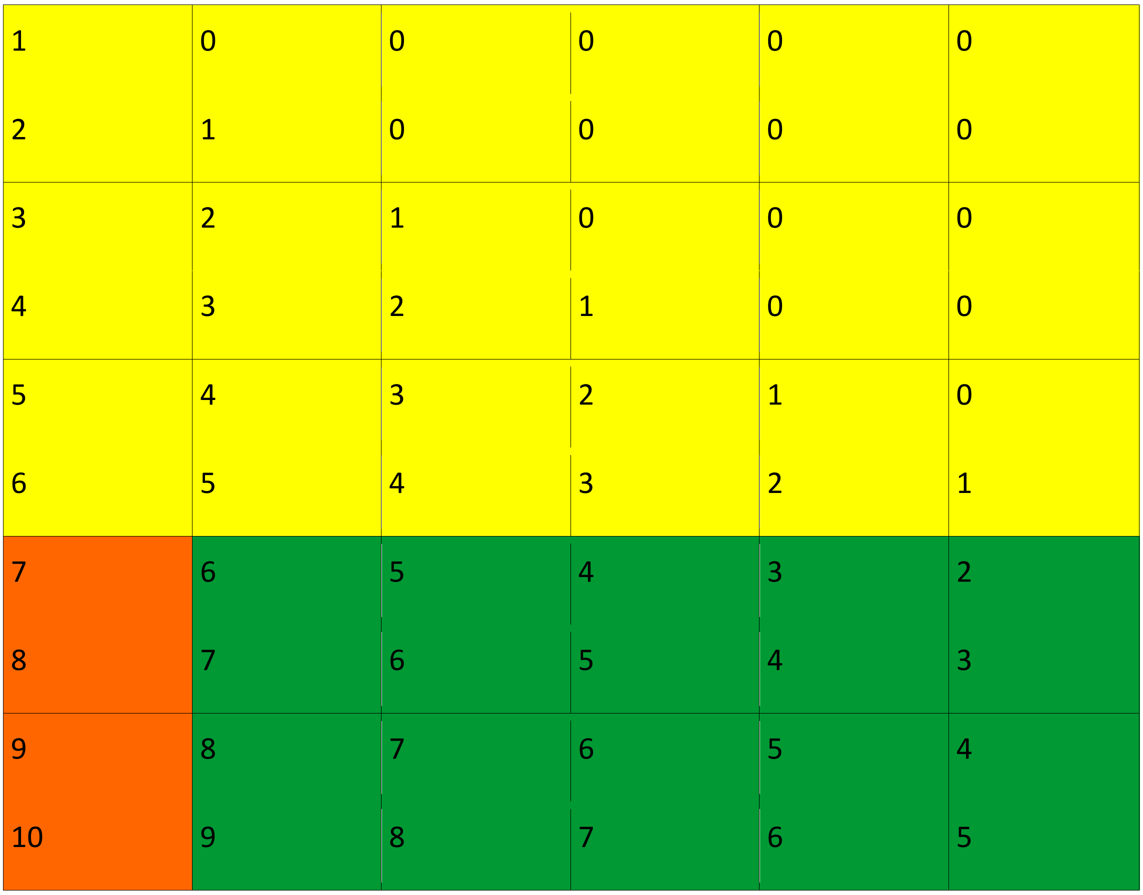
H1= 🟨

h1= 🟥

H2= 🟩

Figure 46. Partitioning of matrix H in the smaller matrices H1, h1, H2 to compute coefficients with Prony's method.

Figure 46 shows the results of the analysis of h= 1,2,3,4,5,6,7,8,9,10 that allows to better display how the elements of the signal are arranged within the matrix H and how the elements appear mirrored vertically, horizontally and diagonally. The order of analysis is for M=5, N=5; in this case the value K=9. The result of the partition produces zeros in the top-right corner of the matrix. This shows that even if none of the elements of the signal *h* are 0, a series of zeros appears to be introduced by this method. The sequence of samples is reversed in the different rows and columns of the matrix. One might need to consider the introduction of zeros when the sequence of samples of an audio signal includes silence. This feature could affect the analysis of frames that include portions of silence. In audio signals silence is represented by zeros, therefore it is important to bear this in mind during the analysis of sounds. This is in order to avoid–dividing zeros by zeros when computing the coefficients, in particular during the re-synthesis of signals on a frame-by-frame basis. To solve the equations in b) in Figure 45 it is necessary to introduce negative signs, and this might cause the phase to create variations in results of the computation of the same signal at different phases. This feature is tested and explored in this thesis to describe how the characteristic of this approach to Prony's method affects the design of sound synthesis tools.



1	0	0	0	0	0
2	1	0	0	0	0
3	2	1	0	0	0
4	3	2	1	0	0
5	4	3	2	1	0
6	5	4	3	2	1
7	6	5	4	3	2
8	7	6	5	4	3
9	8	7	6	5	4
10	9	8	7	6	5

Figure 47. Partitioning of matrix H for h= [1;2;3;4;5;6;7;8;9;10].

The longer the signal the more rows the matrix H will have, suggesting a longer time of computation in the case of longer frames of analysis. Similarly, to LPC a higher order of analysis will increase the detail. Prony's method allows to set the order of the numerator and denominator independently to provide control over the amount of details of the resonances and anti-resonances.

References

Arfib,D. and Keiler F. and Zölzer, U. ”Source-Filter Processing” in *DAFX: Digital Audio Effects.* Edited by Udo Zölzer,299-372, Chichester: Wiley, 2002.

Barnwell III, Thomas P. and Nayebi, Kambiz and Richardson Craig H. *Speech coding : Aa computer laboratory textbook.* New York ; Chichester : Wiley, 1996.

Bazil, Eddie. “Layers of Complexity”, Sound on Sound, 2012, accessed Sept 14, 2018, https://www.soundonsound.com/techniques/layers-complexity.

Bennet, Gerald and Rodet, Xavier “Synthesis of the singing Voice.” In *Current Directions in Computer Music Research*, edited by Max V. Mathews and John R. Pierce, 19-44, Cambridge, Massachusetts: MIT Press, 1989.

Bogaards, Niels. “Analysis-Assisted Sound Processing with Audiosculpt,”8th International Conference on Digital Audio Effects DAFX-05 (Spain: Madrid, 2005):269-272.

Burrus, Charles S. and Parks Thomas W. “Time Domain Design of Recursive Digital Filters” IEEE Transactions on Audio and Electroacoustics, 18, no.2 (1970): 137-141.

Caetano, Marcelo, and Xavier Rodet. "Sound Morphing by Feature Interpolation." 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, 161-64.

Campion, Thomas. “Observations in the Art of English Poesie.” The Work of Thomas Campion, edited by Walter R. Davis, 287-318, London: Faber and Faber, 1967.

Cann, Richard. "An Analysis / Synthesis Tutorial, Part 2." *Computer Music Journal* 3, no. 4 (1979): 9-13.

Chu, Wai C. *Speech coding algorithms: Foundation and evolution of standardized coders*. Hoboken, N.J.: Wiley, 2003.

Cody, Joshua and Paul Lansky. "An Interview with Paul Lansky." *Computer Music Journal* 20, no. 1 (1996): 19-24.

Cook, Perry R. "Singing Voice Synthesis: History, Current Work, and Future Directions." *Computer Music Journal* 20, no. 3 (1996): 38-46.

Cook, Perry R. “Formant Peaks and Spectral Valleys” in *Music, cognition, and computerized sound: An introduction to psychoacoustics.* Edited by Perry R. Cook, 129-138, Cambridge, Mass : MIT Press, 1999.

Cook, Perry R. *Real Sound synthesis for interactive applications*. Natick, MA: AK Peters, 2002.

Cook, Perry R. *SPASM, a Real-Time Vocal Tract Physical Model Controller; and Singer, the Companion Software Synthesis System.* Computer Music Journal, Vol. 17, no. 1 (1993): 30-44.

Cook, Vivian J. "IPA Transcription of English Phonemes." In *The English Writing System*, 215. Routledge, 2014.

Ding, Yinong, and David Rossum. "Filter Morphing of Parametric Equalizers and Shelving Filters for Audio Signal Processing." Journal of the Audio Engineering Society 43, no. 10 (1995): 821-26.

Dodge, Charles and Jerse, Thomas A. *Computer music: Synthesis, composition, and performance.* 2nd ed.Boston; Schirmer. 1997.

Dodge, Charles. ”On Speech Songs.” In *Current Directions in Computer Music Research*, edited by Max V. Mathews and John R. Pierce, 9-17. Cambridge, Massachusetts: MIT Press, 1989.

Dutilleux, P. and Zölzer U. “Delays” in *DAFX: Digital Audio Effects.* Edited by Udo Zölzer, 63-74, Chichester: Wiley, 2002.

Eno, Brian. *Ambient 1 Music for Airports*. Astralwerks. 1978. CD.

Eno, Brian. *Discreet Music*. Astralwerks. 1975. CD.

Fant, Gunnar. *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations.* Paris; The Hague: Mouton, 1970.

Fant, Gunnar. *Speech Acoustics and Phonetics*. Text, Speech, and Language Technology; v. 24. Dordrecht; London: Kluwer Academic, 2004.

Felder, David and Karlheinz Stockhausen. "An Interview with Karlheinz Stockhausen." *Perspectives of New Music* 16, no. 1 (1977): 85-101.

Flux. “Ircam Trax v3.” Accessed September 14, 2020, https://www.flux.audio/project/ircam-trax-v3/.

Godfrey, Jem. “Creative Sound Design for Music”, Sound on Sound, 2011, accessed Sept 14, 2018, https://www.soundonsound.com/techniques/ creative-sound-design-music.

Gopi, E.S. *Digital speech processing using Matlab.* New Delhi: Springer,2014.

Howard, David M. and Murphy, Damian. *Voice science, acoustics and recording.* San Diego; Oxford: Plural Publishing, 2008.

Hugill, Andrew. *The Digital Musician.* New York and London: Routledge, 2008.

Jaffe, David A. "Spectrum Analysis Tutorial, Part 1: The Discrete Fourier Transform." *Computer Music Journal* 11, no. 2 (1987): 9-24.

Jones, David Evan. "Speech Extrapolated." *Perspectives of New Music* 28, no. 1 (1990): 112-142.

Keane, David *Tape Music Composition.* London: Oxford University Press, 1980.

Kent, Ray D. and Read, Charles. *Acoustic analysis of Speech.* Albany, NY: Singular Press, 2002**.**

Kirk, Ross and Hunt, Andy. *Digital Sound Processing for Music and Multimedia.* Oxford, England; Boston: Focal Press, 1999.

Klatt, D. H. *Software for a Cascade/Parallel Formant Synthesizer.* J. Acoust. Soc. Am. 67 (1980): 971-995.

Krzyzaniak, Mike “Stockhausen's Studies I and II” accessed Aug 14, 2020. https://michaelkrzyzaniak.com/Research/Stockhausen_Studie_II/.

Ladefoged, Peter. *Vowels and consonants: An introduction to the sounds of languages.* Malden, MA; Oxford: Blackwell, 2005.

Lansky, Paul and Jeffrey Perry. "The Inner Voices of Simple Things: A Conversation with Paul Lansky." *Perspectives of New Music* 34, no. 2 (1996): 40-60.

Lansky, Paul. "Notjustmoreidlechatter." *Electro Acoustic Music 1.* Neuma Records, 1990. CD.

Lansky, Paul. “Compositional Applications of Linear Predictive Coding” in *Current directions in computer music research*. Edited by Max V. Mathews and J. Pierce, 5-8. Cambridge, Massachusetts: MIT Press, 1991.

Lansky, Paul. *Fantasies and Tableaux.* NWCR 683, 2007. CD.

LaRoche, Jean. "A new analysis/synthesis system of musical signals using Prony's method-application to heavily damped percussive sounds," *International Conference on Acoustics, Speech, and Signal Processing,* Glasgow, UK, vol.3(1989) 2053-2056.

Lynn, Paul A. and Fuerst. Wolfgang. *Introductory digital signal processing with computer applications.* Chichester; New York: John Wiley, c1998

Makhoul, John. “Linear Prediction: A Tutorial Review” Proceeding of the IEEE, Vol. 63, no. 4 (1975): 561 – 580.

Matlab. version. 2018a. The MathWorks,Inc., 2018. Computer software.

McAdams, Stephen. “Perspectives on the Contribuition of Timbre to Musical Structure.” *Computer Music Journal* 23, no. 3 (1999): 85-102.

Moore, Adrian. *Sonic Art: An introduction to Electroacoustic Music Composition.* New York: Routledge, 2016

Moorer, James A. "The Use of Linear Prediction of Speech in Computer Music Applications." *Journal of the Audio Engineering Society* 27.3 (1979): 134-40.

Mullen, J. and Howard D.M. And Murphy, D.T. *Waveguide Physical Modeling of Vocal Tract Acoustics: Flexible Formant Bandwidth Control from Increased Model Dimensionality,* IEEE Transactions on Audio, Speech and Language Processing, Vol. 14, no.3 (2006): 964-971.

Parks, T.W. Burrus, C.S. *Digital Filter Design.* New York: John Wiley and Sons,1987.

Risset, Jean-Claude and Wessel, David L. "Exploration of timbre by Analysis and Synthesis". In *Pychology of Music.*2nd ed. Edited by Diana Deutsh, San Diego: Academic Press,1999.

Risset, Jean-Claude. "Examples of the Musical Use of Digital Audio Effects." *Journal of New Music Research* 31, no. 2 (2002): 93-97.

Roads, C. and Lansky Paul. "Interview with Paul Lansky", *Computer Music Journal, Vol*. 7, no.3 (1983): 16-24.

Roads, Curtis. *Sound Composition with Pulsars,* J. Audio Eng. Soc. Volume 49, No.3 (2001): 134-147

Roads, Curtis. *The computer music tutorial*. Cambridge, Mass.; London: MIT Press, 1996.

Rodet, X. and Depalle, P. "Synthesis by rule: LPC diphones and calculation of formant trajectories," *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tampa, FL, USA, 1985, 736-739.

Sethares, William, Milne, Andrew, Tiedje, Stefan, Prechtl, Anthony, and Plamondon, James. "Musical Applications and Audio Signal Processing: Spectral Tools for Dynamic Tonality and Audio Morphing." Computer Music Journal 33, no. 2 (2009): 71-84.

Slawson, A. Wayne. "The musical control of Sound Color.", *Canadian University Music Review/Revue de musique des universities canadiennes*, no. 3 (1982): 67-79.

Slawson, Wayne "Sound-Color Dynamics", *Perspectives of New Music*, Vol. 25, no. 1-2, 25th Anniversary Issue ,156-181, 1987.

Slawson, Wayne. "A Speech-Oriented Synthesizer of Computer Music." *Journal of Music Theory* 13, no. 1 (1969): 94-127.

Slawson, Wayne. *Sound Color*. Berkeley: University of California Press, 1985.

Stockhausen, Karlheinz. *Telemusik: Nr.20*. Music Online: Classical Scores Library, Volume I. Wien: Universal Edition, 1969.

Stylianou, Y. “Voice Transformation” in *Springer Handbook of speech processing. Edited by* Benesty, Sohndi, Huang, Berlin: Springer.2007.

Sundberg, Johan. *The science of the singing voice.* Dekalb, Illinois: Nothern Illinois University Press, 1987.

SuperCollider 3.11.1.“LPCAnalyzer” in Classes, SuperCollider 3.11.1, Accessed September 14, 2020. https://doc.sccode.org/Classes/LPCAnalyzer.html.

The Canonical Csound Reference Manual. “Linear Predictive Coding (LPC) Resynthesis”, The Canonical Csound Reference Manual, Csound, Accessed September 14, 2020. https://csound.com/docs/manual/SpectralLpcresyn.html.

Toole, E. Floyd and Sean E. Olive. “The Modification of Timbre by Resonances: Perception and Measurement.” *J. Audio Eng. Soc.* 36, no. 3 (1988): 122-142.

Villavicencio, F. and Robel A. and Rodet X. "Improving Lpc Spectral Envelope Extraction of Voiced Speech By True-Envelope Estimation," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Toulouse (2006), I -868-I-872.

Williams Duncan “Emotion in Speech, Singing, and Sound Effects.” In *Emotion in Video Game Soundtracking.* Edited by Duncan Williams, Newton Lee, 17-26, International Series on Computer Entertainment and Media Technology. Springer, Cham ,2018

Williams, Duncan. "Toward Emotionally-Congruent Dynamic Soundtrack Generation." *Journal of the Audio Engineering Society* 64, no. 9 (2016): 654-63.

Wishart, Trevor. "From Sound Morphing to the Synthesis of Starlight. Musical Experiences with the Phase Vocoder over 25 Years." Musica, Tecnologia = Music, Technology 7 (2013): 65-69,119-120.

Wishart, Trevor. *Audible Design.* Orpheus and Phantomime, 1994.

Wishart, Trevor. *On sonic art*. Abingdon: Routledge Taylor & Francis group,1996.