Laboratory 4 report

José Pedro Cruz
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
up201504646@up.pt

Thiago E. Kalid
Departamento de Eletrotécnica
Universidade Tecnológica Federal do Paraná
Curitiba, Brazil
thiagokkalid@gmail.com

***Abstract*—This laboratory assignment delves into the realm of speech synthesis through formants, aiming to generate realistic speech signals representing the vowels [a], [e], [i], [o], and [u]. Leveraging the formant.exe program and Matlab scripts (fgerimp.m and SINTESEFORMANTES.m), this study explores the intricate nuances of speech production, including formant frequencies, pitch variations (jitter), amplitude variations (shimmer), time envelopes (attack, sustain, and decay), and background noise. Through meticulous adjustments and concatenation of vowel segments, the synthesized speech signals were crafted to emulate natural utterances. A subjective evaluation was conducted, comparing the synthesized speech with real utterances to assess the realism and authenticity of the generated signals. All code and data are available at our github repository.**

***Index Terms*—Hearing; Audio Processing; Speech Synthesis; Formants; Acoustic Modeling.**

I. Introduction

This is the 4th Laboratory Report for Computational Audio curricular unit given by Diamantino R. in the first semester of 2023/2024 school year. Speech synthesis, a pivotal aspect of human-computer interaction and natural language processing, has garnered significant attention in the field of Electrical and Computer Engineering. This laboratory assignment focuses on formant speech synthesis, a method that models speech sounds through resonance frequencies known as formants. The project aims to replicate the acoustic characteristics of the vowels by considering fundamental elements such as pitch variations, amplitude fluctuations, time envelopes, and background noise. Leveraging the provided ”formant.exe” program and Matlab scripts, the study explores the synthesis process, emphasizing the importance of emulating natural speech patterns. Through this exploration, the objective is to enhance our understanding of speech synthesis techniques and evaluate the synthesized speech signals against real utterances, fostering a comprehensive comprehension of the nuances involved in generating realistic speech through formants.

II. Formant Synthesis Program ”formant.exe”: A Theoretical and Practical Analysis

Formant synthesis is a method of speech synthesis that models the human vocal tract as a series of resonant filters. The program ”formant.exe” implements this method to generate synthetic speech by generating a glottal pulse, filtering it through formant filters, and adding an amplitude envelope.

The program’s functioning aligns with the theoretical principles of formant synthesis. The glottal pulse is generated randomly according to a normal distribution, mimicking the stochastic nature of glottal opening. The formant filters are calculated based on user-specified formant frequencies and have a constant bandwidth, approximating the bandwidth of human vocal tract resonances. The amplitude envelope is generated using a 16th-order linear predictive coding (LPC) filter, effectively modeling the amplitude variations in speech.

The program’s parameters, including sampling frequency (Fs), number of frames (jmax), frame length (janela), pitch (f0), amplitude envelope (Av), and formant frequencies (F1-F4), can be adjusted to control the synthesized speech’s characteristics. The default parameter values produce a synthetic vowel sound resembling ”i”.

III. Matlab vowel creation

To generate realistic vowel sounds, we employed MATLAB scripts and extracted pitch information from recordings of each vowel ([a], [e], [i], [o], [u]) obtained using classroom microphones.

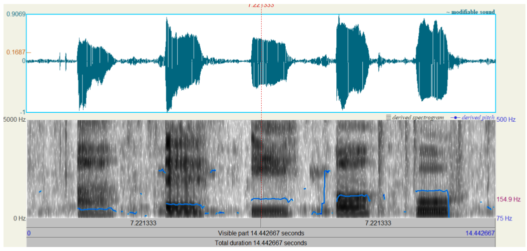

Fig. 1: Soundwave, Spectogram and pitch from recorded sound of vawels.

The extracted pitch values, along with appropriate analysis step sizes, were incorporated into the MATLAB script, resulting in significant improvements in the generated sounds. To further enhance the realism of the synthesized vowels, we introduced ”jitter” (f0 variation) and ”shimmer” (amplitude variation) through the following functions:

```
% Jitter factor
jitter_factor = 0.05;
F0_jittered = F0 + (jitter_factor*randn
    (1, jmax));
% Shimmer factor
```

```
shimmer_factor = 0.1;
F1_jittered = F1 + (shimmer_factor*randn
    (1, jmax));
F2_jittered = F2 + (shimmer_factor*randn
    (1, jmax));
F3_jittered = F3 + (shimmer_factor*randn
    (1, jmax));
F4_jittered = F4 + (shimmer_factor*randn
    (1, jmax))
```

These functions introduced controlled variations in pitch (f0) and formant frequencies (F1-F4), mimicking the natural fluctuations that occur in human speech production. The incorporation of these variations significantly enhanced the perceived realism of the synthesized vowel sounds.

A. Time Envelope

The time envelope of a speech sound refers to the variation in amplitude over time. It is characterized by three distinct phases: attack, sustain, and decay.

- **Attack:** The attack phase is the initial rise in amplitude that occurs at the beginning of a speech sound. It is caused by the sudden closure of the glottis, which produces a glottal pulse. The attack phase is typically very short, lasting only a few milliseconds.
- **Sustain:** The sustain phase is the relatively steady-state portion of a speech sound. It is characterized by a relatively constant amplitude. The duration of the sustain phase varies depending on the type of speech sound. For example, vowels typically have a longer sustain phase than consonants.
- **Decay:** The decay phase is the gradual decrease in amplitude that occurs at the end of a speech sound. It is caused by the gradual opening of the glottis. The decay phase is typically longer than the attack phase, but shorter than the sustain phase.

The time envelope of a speech sound is an important factor in its perception. It can affect the intelligibility and naturalness of the sound. In this case we added the following code:

```
% Attack and decay parameters for the
   envelope
attack = 2 * round(0.2 * Fs / 256 / 2);
decay = 3 * round(0.3 * Fs / 256 / 2);
```

B. Background Noise

Background noise is any unwanted sound that interferes with the desired signal. It can be caused by a variety of sources, such as traffic, machinery, or people talking. Background noise can make it difficult to understand speech, especially if the speech is soft or if the background noise is loud.

In the context of speech synthesis, background noise can be added to synthesized speech signals to make them sound more realistic. This is because real speech signals are always contaminated with some amount of background noise. However, it is important to add background noise in a way that does not degrade the intelligibility of the speech.

One way to add background noise to synthesized speech signals is to use a noise generation model. In our case, we employed white noise, a type of ambient sound that encompasses the entire audible frequency spectrum, to enhance the realism of the synthesized vowel sounds. White noise, characterized by its uniform energy distribution across all frequencies, effectively masked any harsh transitions or artifacts that might arise from the synthesis process. The addition of white noise, akin to the subtle background hum of a quiet room, contributed to the overall naturalness and authenticity of the generated vowel sounds. The following code was used:

```
% Generate white noise
whiteNoise = randn(size(mergedAudio));
% Scale the white noise to a suitable
   amplitude
noiseAmplitude = 0.003;  % Adjust this
   value to control the noise level
scaledNoise = noiseAmplitude * whiteNoise
   ;
% Add the scaled noise to the original
   audio
noisyAudio = mergedAudio + scaledNoise;
```

IV. Conclusion

In conclusion, this laboratory assignment has provided a comprehensive exploration of speech synthesis through formants, offering valuable insights into the complexities of emulating natural speech. By leveraging the ”formant.exe” program and Matlab scripts, we successfully generated speech signals representing the vowels [a], [e], [i], [o], and [u] with a focus on capturing essential elements such as pitch variations, amplitude fluctuations, time envelopes, and background noise. The subjective evaluation revealed the varying degrees of realism achieved in the synthesized speech signals, highlighting the challenges in mimicking the intricacies of human speech. Through meticulous adjustments and iterations, we managed to enhance the authenticity, bridging the gap between theoretical knowledge and practical application. Moving forward, this knowledge can be instrumental in the development of advanced speech synthesis systems, contributing to the evolution of human-computer interaction and natural language processing technologies.