ECSE 523: Speech Communication

Final Project

Farid Rener: 260171831

A simple Klatt formant synthesizer: Project Report

Background

Speech Synthesis

Today, speech synthesizers provide highly intelligible, almost natural speech from input text. There are many ways of doing this: concatenative methods, where phonemes are stored and recovered from a large database of sounds; articulatory methods, which try to model the physics of the vocal tract; and formant synthesis, which employs digital resonators to synthesize the spectral and temporal shape of the desired speech.

Concatenative methods require large memory databases to store phrases or phonemes. One modern example of this is the AT&T Natural Voices software [1] which has several hours of high quality phonemes or half-phonemes [2] to produce natural sounding speech. To produce high-quality speech, this requires complex combinatorial mathematics, as there are billions of combinations possible.

Articulatory methods are less successful than other forms of speech synthesis [3]. Articulatory synthesis models the movement of the vocal tract and the articulators (tongue, jaw, lips), which creates a simulation of the air flow through this system. Articulatory methods have been around for centuries in the form of mechanical “talking heads,” which could synthesize simple vowel sounds [4]. However, digitally, this is the most complex to implement, especially as measuring the articulators for each phoneme is extremely challenging [3].

Instead of modelling the physical production of the human voice, formant synthesis aims to model the vocal tract as a series of digital resonators. Formant synthesizers employ two sources: a periodic impulse train, or a noise source. This is often simpler to implement digitally. This project implemented

a simple formant synthesizer, based on Klatt [5].

Klatt Formant Synthesis

Speech formant synthesis is a form of additive synthesis that takes either a periodic impulse train or a noise source as input. It is based on acoustic theory of speech production, where sound sources build up in the lungs, and are expelled through the lips. Modelling the vocal tract as a linear system, each sound source can be added separately, as the resonating system is similar to an organ pipe [5].

The Klatt synthesizer contains both cascade and parallel configurations, as shown in figure 1. The parallel configuration has all of the formant resonators connected in parallel, with each resonator having its own amplitude control. The cascade synthesizer, on the other hand, has each resonator connected in series [5]. The parallel bank requires calculating the amplitude of each formant, and hence contains an extra multiplication for each resonance [3]. The cascade connection allows the relative amplitudes of the formant peaks to come out 'just right', as this is a more accurate model of the physical vocal tract. However, the cascade cannot adequately model fricatives or plosives [5].

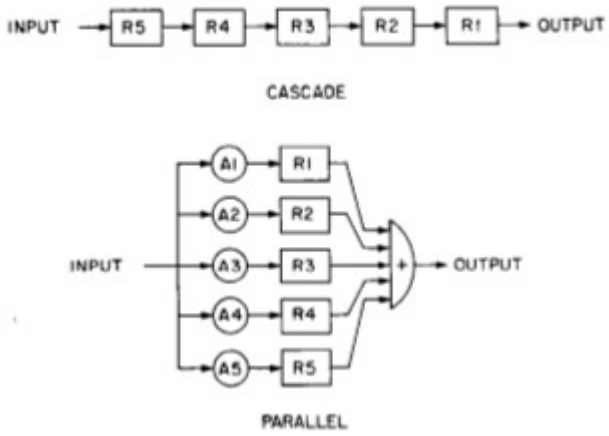

Figure 1: Cascade and Parallel structure of formant resonators [5].

When the outputs are added in the parallel, unwanted spectral zeros may occur, so alternating the signs of each amplifier is required, accounting for the 180 degree phase shift of the vocal tract for each formant [3]. The parallel structure is unsuited to create some vowel sounds.

Sources:

Two sources are required for the Klatt synthesizer: a noise source and a periodic impulse train. The impulse train amplitude ranges from 0dB for unvoiced phonemes to 60dB in a strong vowel. The frequency of the impulse is the fundamental frequency f_0 of the speech, i.e. a pulse is generated every $1/f_0$ seconds. The absence of spectral zeros affects the perception of naturalness [5]. Quasi-sinusoidal voicing is also created by using a low-pass filter on the impulse train. This is used especially in breathy voices.

The noise source is 50% amplitude modulated at a frequency f_0. The noise source is used for both frication noise and aspiration noise. In aspiration, a cascade resonator bank is better suited, as aspiration noise is created in the larynx [3], which is well modelled in cascade [5].

Resonators & Formant Frequencies:

The resonators essentially act as bandpass filters, introducing a peak in the frequency domain at the centre frequency of the resonator. Each formant has different ranges, and is affected differently depending on the type of phoneme. Formants also vary differently depending on the speaker, however, other sonic aspects allow listeners to understand different phonemes (e.g. f_0, upper formants, formant bandwidths, etc. [3]). Klatt observed the ranges of the formant frequencies: F1 from 180-750Hz, F2 from 600-2300Hz, F3 from 1300 to 3100Hz for male voices [5]. Above this, formant frequencies are not required for intelligibility. In the Klatt synthesizer, formant bandwidths can be varied. This increases or decreases the intensity of formant energy concentration [5].

Implementation:

I implemented a simple version of the Klatt formant synthesizer in Puredata (Pd). Pd is an open-source real-time graphical programming environment for interactive computer music [6]. Puredata works with 'patches' which are somewhat like functions in other programming languages. An example of a patch is in figure 3, which shows the pulse generator.

At the start of this project I had the ambitious idea to create a whisper synthesizer, following the studies undertaken by Jovicic [7],[8]. However, after starting to implement the Klatt synthesizer, I realized that this would be much past the scope of this project. I only managed to implement whispered vowels, following the method in [7].

Time constraints and the scope of the project meant that I didn't implement actual speech, instead I only managed to obtain the sounds for vowels: /a/, /e/, /i/, /o/, /u/, sonorants: /w/, /y/, /r/, /l/, fricatives: /f/, /v/, /θ/, /TH/ (as in Then), /s/ and /z/, to varying degrees of success. I only created a static synthesis method so none of the parameters change in time, which would be required to string the phonemes together in any semblance of natural speech.

When the program is running, it allows the user to type the desired phoneme (e.g. 'a' for /a/, 'y' for /y/ etc.) for it to be synthesized. It uses a similar structure to that proposed by Klatt in figure 2 [5]. The main differences is that it doesn't implement the nasal poles/zeros (RNP, RNZ) or the radiation characteristic.

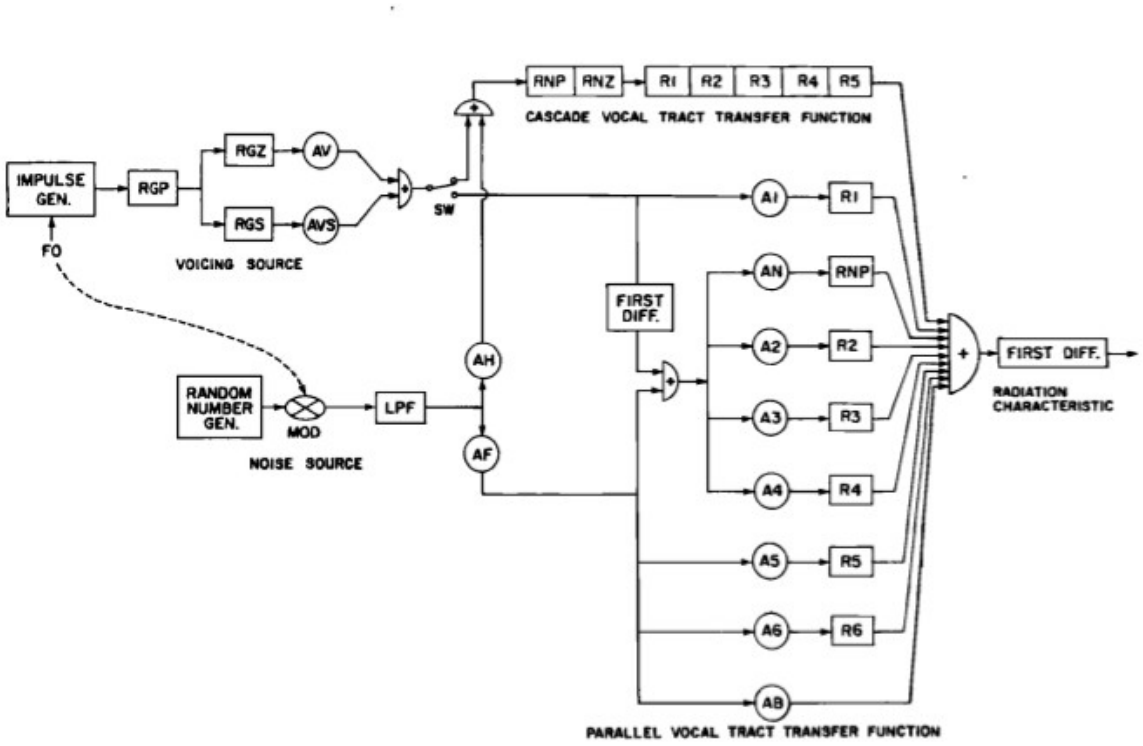

Figure 2: Klatt formant synthesizer [5]

Pulse Generator

For the pulse generator, I used a modified version of the F03.pulse.spectrum.pd help patch provided with Pd, shown in figure 3. This takes an input of the fundamental frequency, f_0, and passes it to the

phasor~ object, which essentially creates a sawtooth wave at that frequency. This is then multiplied by the bandwidth and clipped between -0.5 and 0.5. This feeds cos~ which outputs a sinusoid at that frequency. As can be seen in the figure, this outputs pulses, spaced by 1/f0.

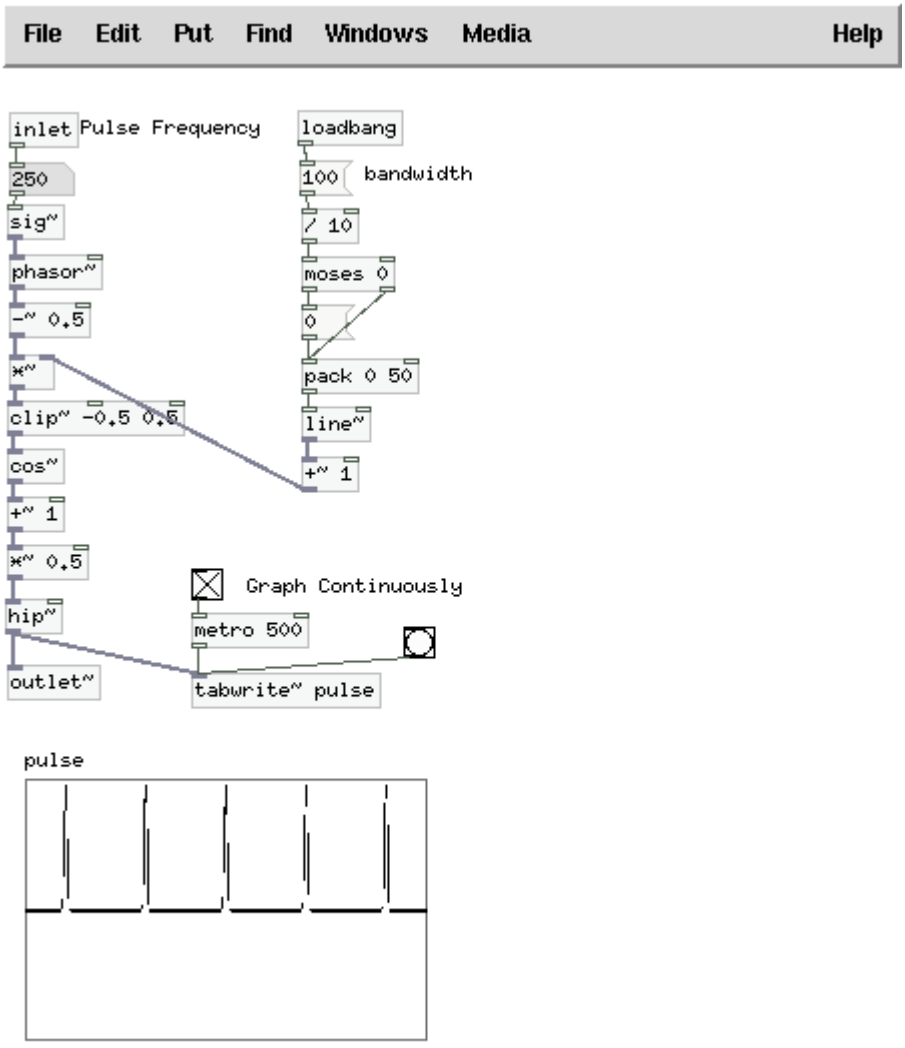

Figure 3: Pulse Generator

Cascade Filterbank:

The cascade filterbank shown in figure 4, implements four bandpass filters (bp~ objects), which take as input the centre frequency (r1, r2, r3, r4) and Q value (q1, q2, q3, q4). Since Klatt specifies bandwidths for each filter instead of Q value, bandwidths are converted to Q using the formula

$$Q = \frac{f_c}{(f_2 - f_1)} = \frac{f_c}{bandwidth} \quad .$$

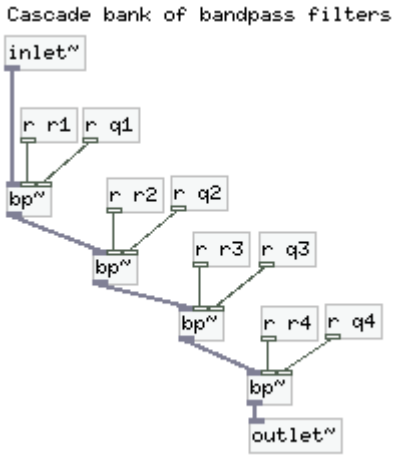

Figure 4: Cascade filterbank

Parallel Filterbank:

Similarly to the cascade filterbank, the parallel filterbank takes centre frequency values and Q values. Since each resonator has its own amplifier, these are also shown in figure 5. Since Klatt uses dB values for amplitudes and Pd uses rms values, I used the *dbtorms* object which converts from dB to rms. I also alternated the signs of every other resonator to account for the 180 degree phase shift which occurs in the vocal tract, as discussed above.

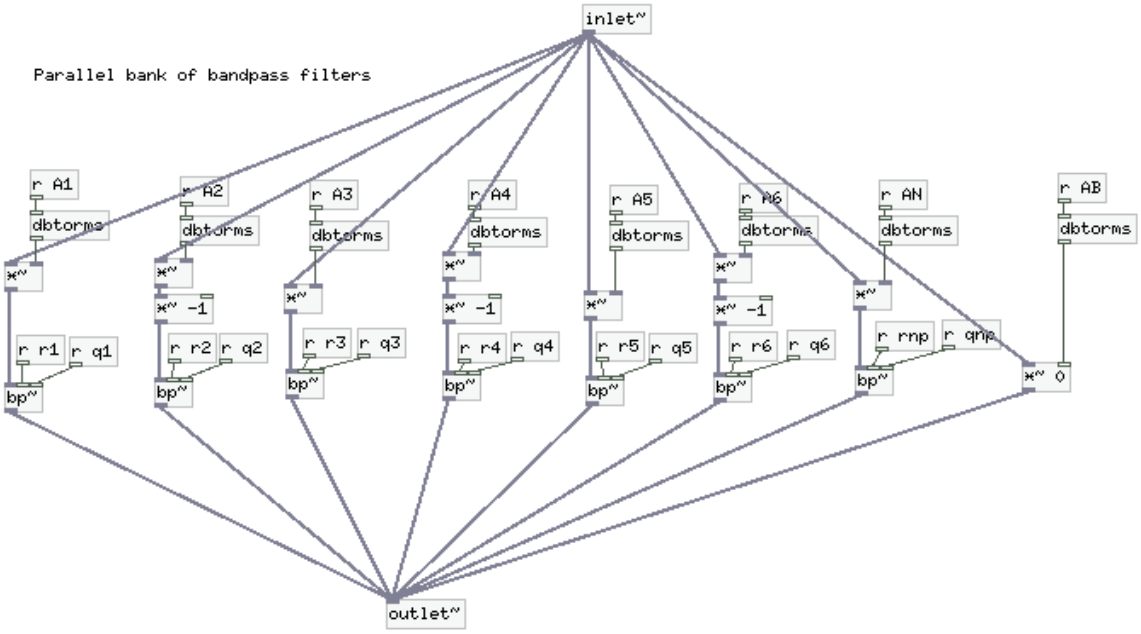

Figure 5: Parallel Filterbank

Amplitude Envelope:

The amplitude envelope is a simple attack-decay-sustain-release, using different values for the vowels, sonorants, fricatives and for the plosives. For the vowels, sonorants and fricatives, there is a longer sustain, whereas for the plosives a shorter more brutal attack is used to try and simulate the nature of the way these phonemes are pronounced.

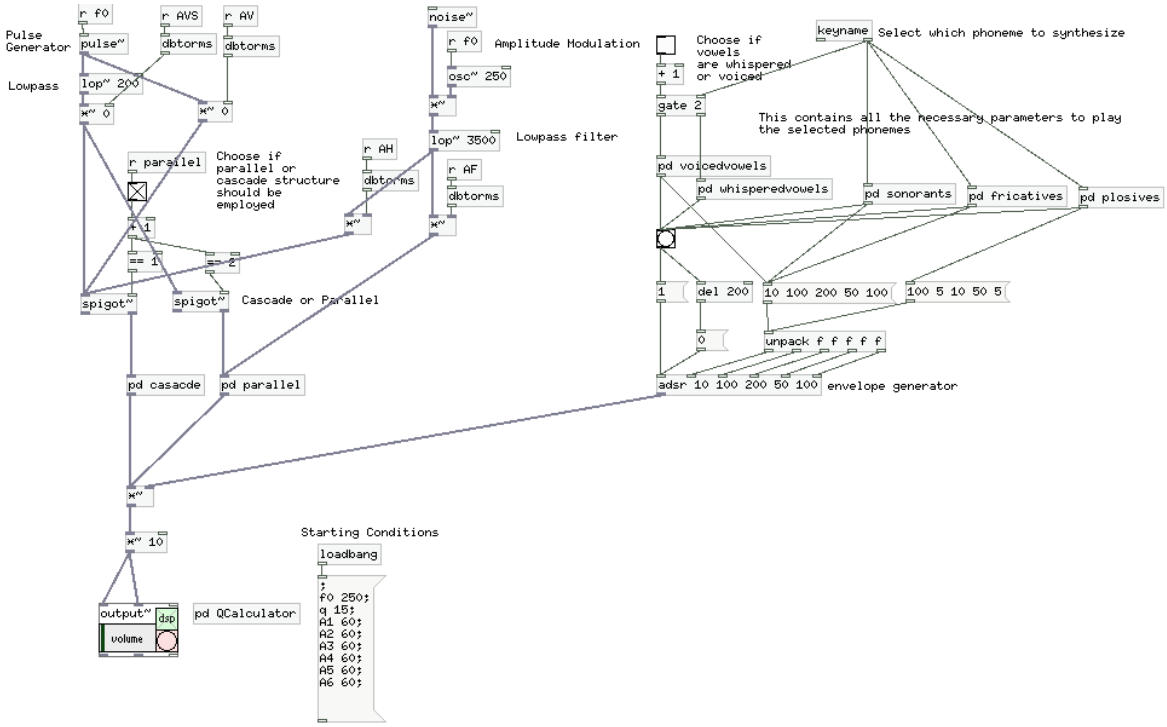

Figure 6: Entire implementation

Vowels:

In this program, there is the option to either have whispered or voiced vowels by clicking the toggle box, shown in figure 6. The vowels were the most convincing of the synthesized speech sounds. All synthesized sounds are provided in wave format along with this document.

Vowel sounds use the cascade filterbank in figure 4, and thus only require the centre frequencies of the resonators to be set. These parameters were taken from Jovicic [7] for both voiced and whispered vowels. The vowels only use the pulse input as vowels are naturally only made with the vibration of the vocal cords. The whispered vowels, however, use some shaped noise, as this contributes to the breathy characteristic of the whisper. As found by Jovicic, most of the formant frequencies for the vowels increase when they are whispered, especially those of the first and second formant. It was also found that formant bandwidths were increased. However, due to the nature of the cascasde implementation, it

wasn't possible to change the formant bandwidths.

The voiced vowels patch is shown in figure 7.

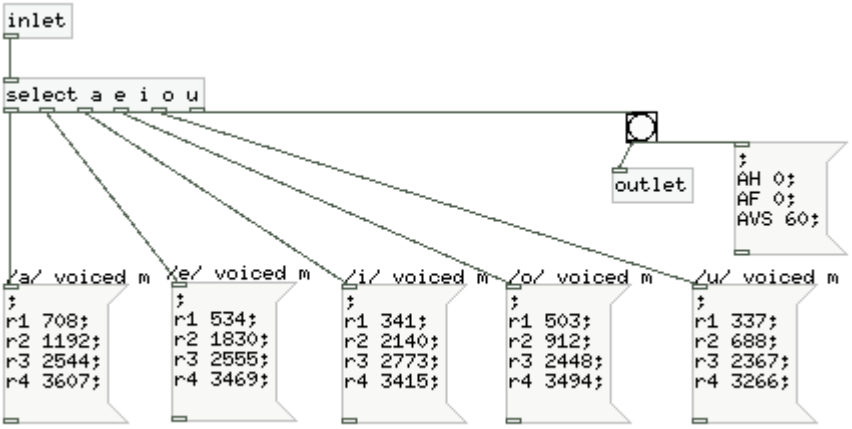

Figure 7: Voiced Vowels

Consonants:

Sonorants:

The sonorants /w/, /y/, /r/, /l/ were synthesized using the parameters in Klatt [5].
These use the parallel part of the synthesizer as this allows the varying of the bandwidths of each formant. The sonorants use the quasi-sinusoidal input, with AVS set to 60dB and all the other inputs (AV, AH, AF) set to zero. As these are pretty close to the vowel sounds, they were quite convincing from this implementation.

Fricatives:

These were only somewhat convincing in this program. The fricatives also use the parallel portion of the synthesizer, however, the voiced fricatives use both the impulse train and the noise as inputs. The voiced fricatives have AF as 50dB, AV as 47dB and AVS as 47dB. Where AF is the frication amplitude, AV is the voicing amplitude and AVS as the sinusoidal voicing amplitude.

Plosives:

The plosives use a different amplitude envelope mainly because a large portion of how these are perceived is through their explosive nature. However, these are not that convincing in this program. I think that were I to implement a frame-by-frame paradigm that would allow changing each parameter

every frame, this would be less of a problem.

Conclusions / Evaluation:

I am quite happy with how the vowels and the sonorants have turned out, as they do not sound as robotic as I was expecting them to. However, those consonant phonemes which rely on a noise source are less convincing, perhaps to do with amplitude envelopes, etc.

I found that this was a lot more challenging than I had at first expected, especially when I started trying to make simple words by simply stringing together the phonemes. This made me realize how important the transitions were. If this program is to be anywhere more than a pedagogical tool (for me), it will require a lot more work. I might carry on this project in the summer, since as of yet there isn't a good speech synthesizer written for Pd.

References:

[1] *AT&T Labs Natural Voices Text-to-Speech Demo* (Online). Available: http://www2.research.att.com/~ttsweb/tts/demo.php [Apr. 12 2011]

[2] *The Mathematics of ... Artificial Speech | Math | Discover Mag* (Online). Available: http://discovermagazine.com/2003/jan/featmath/ [Apr. 12 2011]

[3] D. O'Shaughnessy. *Speech Communication: Human and Machine (2nd ed.)*. Universities Press, India, 2001.

[4] *Talking Heads: Simulacra, The Early History of Talking Machines* (Online). Available: http://www.haskins.yale.edu/featured/heads/simulacra.html [Apr. 12 2011]

[5] D. Klatt. Software for a cascade/parallel formant synthesizer. *Journal of the Acoustical Society of America*, 67, pp. 971-995, 1980,

[6] *Puredata* (Online) Available: www.puredata.info [Apr.12 2011]

[7] S. T. Jovicic. Formant Feature Differences Between Whispered And Voiced Sustained Vowels. *Acta Acustica united with Acustica*, 84, p. 739-743(5), 1998

[8] S. T. Jovicic, Z. Saric. Acoustic analysis of Consonants in Whispered Speech. *Journal of Voice*, 22, p. 263-274(3), 2008