Ed Davies 13006608

APDI Assignment 1 Report

Contents list

Introduction

The synthesiser plugin is a formant based speech synthesiser. This was chosen over other speech synthesis methods such as wavetable and concatenative synthesis as, although it sounds rather robotic and could not easily be mistaken for human speech; it has the benefit of being intelligible at high frequencies, which is necessary when it is used in a musical context.

Synthesiser - exciter

The basic form of this vocal synthesiser is an exciter-resonator model that attempts to recreate the human vocal tract. In the vocal tract, the excitation of air being forced from the lungs through the trachea and the glottis (vocal cords) has resonances imparted on it by the pharynx, mouth and oral cavity. The shape of these resonant chambers alters the position of formants (peaks in the frequency spectrum) and the frequencies of these formants, particularly the lower three, allow us to identify the different phonemes. The vocal cords produce a repeating waveform with a large number of harmonics that can be roughly modelled by a pulse train. This needs to have a flat spectrum so the band-pass filters can predictably and accurately shape the spectral envelope to model certain sounds. This was achieved using the band-limited pulse generator model from page 162 of Charles Dodge's "Computer Music" (Dodge and Jerse, 1985). This signal generator produces harmonics of equal amplitude from the fundamental up to the Nyquist frequency using only two sine wave oscillators (figs 1 & 2), significantly reducing the CPU load compared to other methods such as additive synthesis using a cosine wave for each harmonic. In order to further reduce CPU load, the harmonics are limited to Nyquist limit / 2 as the signal is low-pass filtered with a cutoff at 7kHz to remove unnecessary high frequencies that are not generated by the human vocal tract.

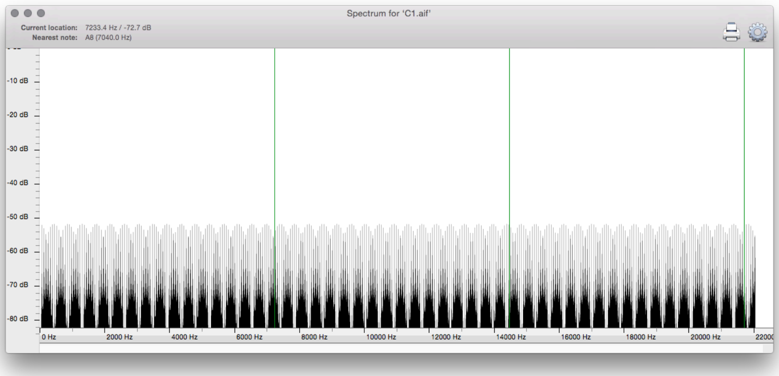

Figure 1: pulse at note C1

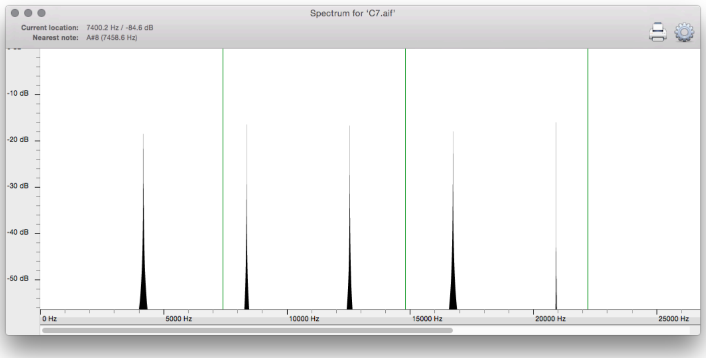

Figure 2: pulse at note C7

The final code for this signal generator model went through a number of iterations. First, the spectrum was generated using up to 50 individual cosine waves at integer multiples of the fundamental with equal amplitudes relative to the number of harmonics, *N* (amplitude / N) which was very inefficient. Next, the model from "Computer Music" was used with 2 STK sine objects but this led to a considerable amount of aliasing due to rounding errors as the objects cast to floating-point numbers. Therefore the final model uses the math.h "sin" function with double variables for maximum accuracy, removing the aliasing errors that were arising from the sine oscillators being out of phase. This is encapsulated in the "BLPulseTrain" class.

Synthesiser – resonator

The excitation signal is low-pass filtered with a cutoff at 7kHz before being passed through a parallel bank of 5 band-pass filters to impart formant frequency peaks corresponding to those of 10 different phonemes. The first 3 formant filters are variable as they are most affected by altering the shape of the vocal tract (Roads, 1996) and the final 2 are fixed at 3.5kHz and 4.5kHz, as they are not vowel-dependent. The amplitudes of each successive formant decreases and the Q-factor increases so the lower formants have more of an effect on the overall sound character. The centre frequencies of the 3 variable formants are based on averaged frequencies of 10 vowel sounds from a wide range of speakers (Rabiner, 1978) and an example of a real phoneme (an "ir" sound) compared the output of the synthesiser's "ir" can be seen in figs 3 and 4. The centre frequencies are set in the while loop of the process function so can be varied between the different vowel sounds during the performance. This is controlled using mod 5 and has been designed so the phonemes change significantly throughout the performance, sometimes during held notes to give a more convincing vocal sound (e.g. moving from "aa" to "ee") and sometimes in breaks between sections so parts of the piece have a different character. The movement from one centre frequency to the next is not linear and is mapped using the exponential function x^0.5 where x is the distance between two phonemes (between 0 and 1). This means that most of the change occurs in the lower range which is similar to a trained singer's transition between phonemes (Sundberg, 1977). Setting the filter cuttoffs and getting the output from them in encapsulated in the "VocalFormants" class.

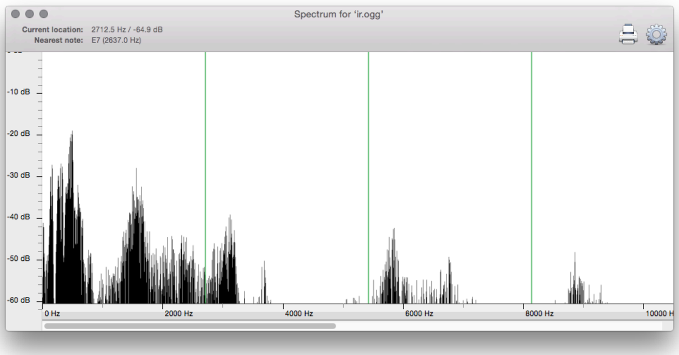

Figure 3: recorded "ir" sound

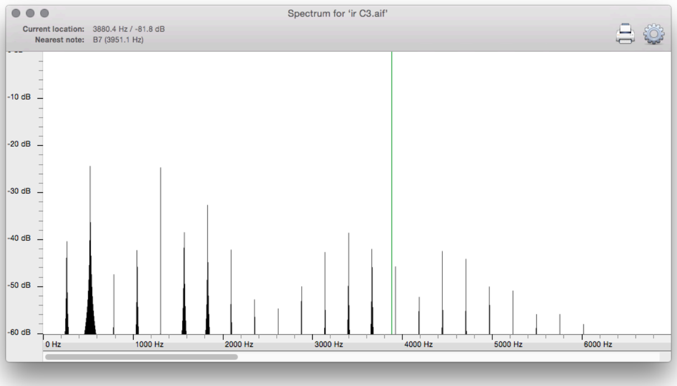

Figure 4: synthesised "ir" sound

<u>Modulation</u>

In order to give a more realistic vocal sound and sound less robotic, the output needs to have a dynamic sound character. This has been achieved with a vibrato effect that varies between 5-6Hz depending on the value of mod 1 to give a subtle character change for longer held notes. The start time of the vibrato effect varies between 0 and 600ms for each note, as this is similar to a human singing voice where vibrato does not start instantaneously. This is achieved using the "randomNum" function, which returns a random value in that range that is used to determine the start time.

Mod 2 is used to set the release time of the note and varies between 0.05 and 1.05s depending on the density of notes. This gives a pad-like texture for chords and a punchier, lead texture for single notes and runs.

<u>Conclusion</u>

This synthesiser creates an interesting and intelligible set of vowel sounds using a relatively computationally efficient method to generate them. However, the sound character is somewhat static compared to a real human voice and sounds quite robotic because of this so could be improved by including sounds other than voiced vowels. This could include fricatives such as "ff" and "z" generated by filtering a noise source or complex combinations of phonemes to create words or phrases. Other methods than formant synthesis could be used to generate these such as linear predictive coding or formant tracking, which are based on more sophisticated analysis and re-creation of speech rather than an exciter-resonator model.

References

Dodge, C., 1997. *Computer Music: Synthesis, Composition, and Performance*. 2 Edition. Cengage Learning.

Rabiner, L., 1978. *Digital Processing of Speech Signals*. US Edition. Prentice Hall.

Roads, C., 1996. *The Computer Music Tutorial (Technology*). 0 Edition. The MIT Press

Sundberg, J., 1977. The Acoustics of the Singing Voice. *Scientific American*, 236(3), 82-91.