DESC9115
Digital Audio Systems

2011

Cross Synthesis using Linear Predictive Coding

Assessment 2

Michael Tortely
311175074

Monday 6th June 2011

Venture Capitalists, I present to you an idea, a tool, a revolution in the world of Digital Audio Synthesis. Since the 80's, there have been simple tools in which engineers and musicians alike could take two sounds and intertwine them in unique and wonderful ways. Technology has evolved since then, and so has the digital industry with many new additions to the synthesis realm. But with each new toy on the market, we seem to throw away our old tools to make way for the upcoming market.

I believe that we have reached a point in technology and processing power, to successful rehash our formal favourites, combining the old and new in a fresh and different way. I believe that my product will deliver exactly this.

I present to you the future of Vocoders. My vocoder performs the cross synthesis between its assigned inputs using latest Linear Prediction Coding (LPC) techniques. While LPC has been around for years, my product utilises the processing ability of current systems to provide not only a quick and efficient means of cross synthesis, but does so in such a way, which allows the user to define and manipulate how the cross synthesis is performed better then ever.

The problem with the current market is that everything is the same more or less.
As an electronic musician, I know from experience that essentially every vocoder and/or cross synthesis always have the same buttons and functions. Wave choice and volume, with rare inclusion of other DSP tools such as reverb, vibrato, etc to enhance the sound. This is where my product really lifts the standards.
Included in the prototype code provided, I have chosen to include parameters that focus on controlling the LPC analysis to allow total control how your sounds are shaped for the cross synthesis.

Before I go on, I should discuss how the cross synthesis is processed in my model. LPC is best classified as a parametric, spectral, source-filter model, in which the short-time spectrum is decomposed into separate parts. It is called the Linear Prediction model as it works on a linear estimation system, which estimates an all-pole filter that matches the content of the sound.

The analysis starts with the assumption that a buzzer, or some sort of noise excitation device at the end of a tube produces the signal. This buzz is characterized by its intensity and frequency. This is called the excitation information. The code then analyzes the speech signal by estimating the resonances, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining excitation signal. The remaining signal is called the residual signal.

This closely resembles our vocal path as humans as LPC is fundamentally a speech analysis tool (hence why I have chosen to use it within a Vocoder, although it is not limited to just speech).

The idea behind the analysis here is to gather enough information to estimate basic signal parameters such as pitch, gain, and spectra, which will allow use to recreate the signal later. This information is stored as a set of numbers called "Prediction Coefficients" which are estimated by a set frame, every set frame length (length pre-determined).

We then must set how close we want the original and the analysis and synthesised signal to be using a values known as the prediction order and block length.
The prediction order determines the accuracy. Higher orders yield cleaner, more generally more accurate signals, however the downside is that it requires higher computing effort.
Block length determines the length of each section we are going to analysis at a single time. Again, Higher lengths yield cleaner, more generally more accurate signals, however the downside is that it requires higher computing effort.

A sound is re-synthesized by taking 2 signals as input and running them through the LPC analysis. The first signal has a FIR filter applied to it such that becomes an excitation signal. The second signal is run through the process and just the residual signal is kept.

Essentially what we are doing here is using the residual signal produced to spectrally shape the excitation signal of the first input to create the new signal.

Now that you understand how it works, I shall continue with how my product is worth your time and funding. You see, I have discovered a means of coding which has allowed me to set the parameters to the 4 essentials for the LPC aspect of the product. They are:

- Block Length
- Frame Size
- Excitation LPC Order
- Residual LPC Order

These unique parameters have allowed me in my experimentation period to truly spectral shape sounds in an easy and controllable manner. A diagram attached to the end shows the signal flow. (The dotted lines show where the parameters are inserted).

To further prove to you why my idea is worth investing in, I have personally designed a Graphic User Interface, which allows you to assess the performance yourself (Picture of this attached to end of presentation).

Within this GUI I have given you the ability to change the parameters on sliders, listen to the original sounds and the Cross Synthesised sounds and even look at the amplitude values on a graph to show how you are shaping your sounds.

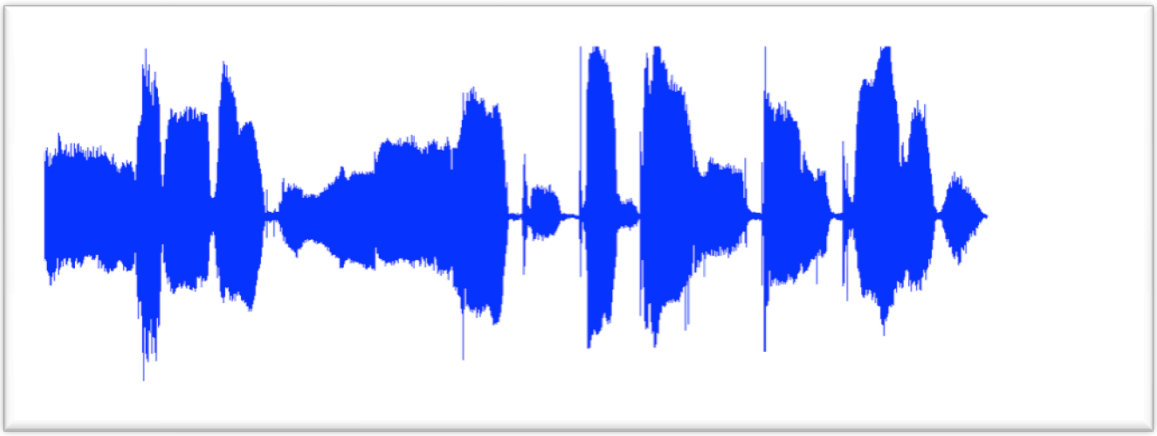
This Code and GUI are programmed well enough so that they run efficiently and smoothly every time it is run. It requires minimal CPU effort, and generally has an average processing time of 2 -10 seconds. As clearly seen in the pictures presented below, the resulting signals resemble the inputs, meaning that the code is doing exactly what it supposed to do.

With more funding, I would hope to make this process even quicker and more efficient such that it could be processed in real time with minimal latency. Then, and only then, will the Digital Audio Synthesis be truly under our control.

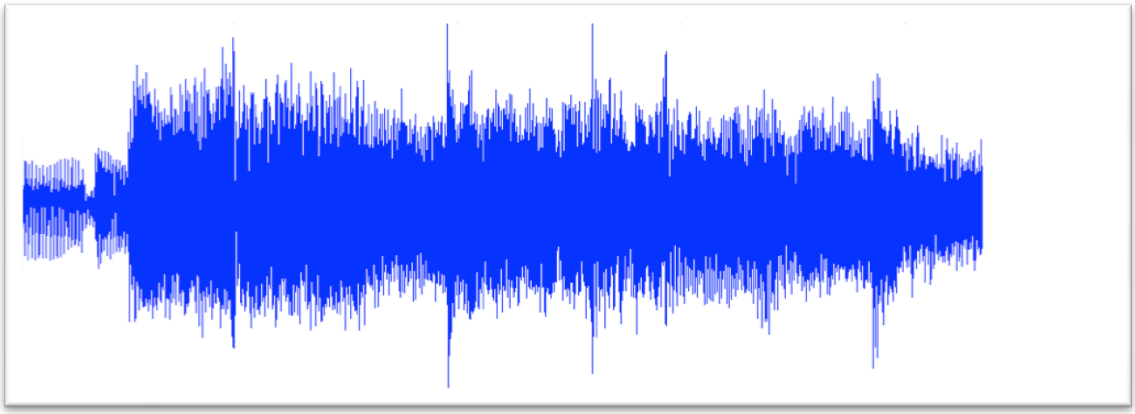
Thank you.

Case 1

Signal 1 – (Girl Singing)



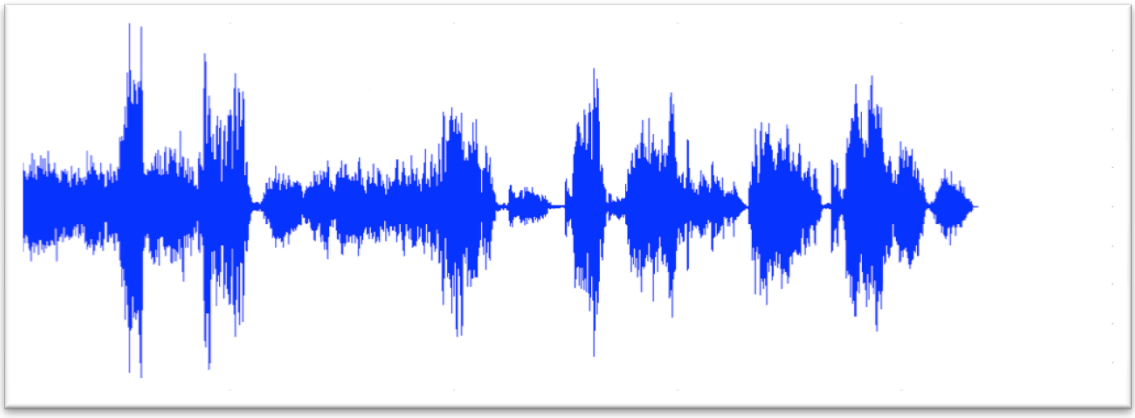
Signal 2 – (Guitar Riff)



LPC Cross Synthesis with;

Block Length = 1000 (100ms), Frame size =500 (50ms)

Excitation LPC order = 20, Residual LPC order = 20



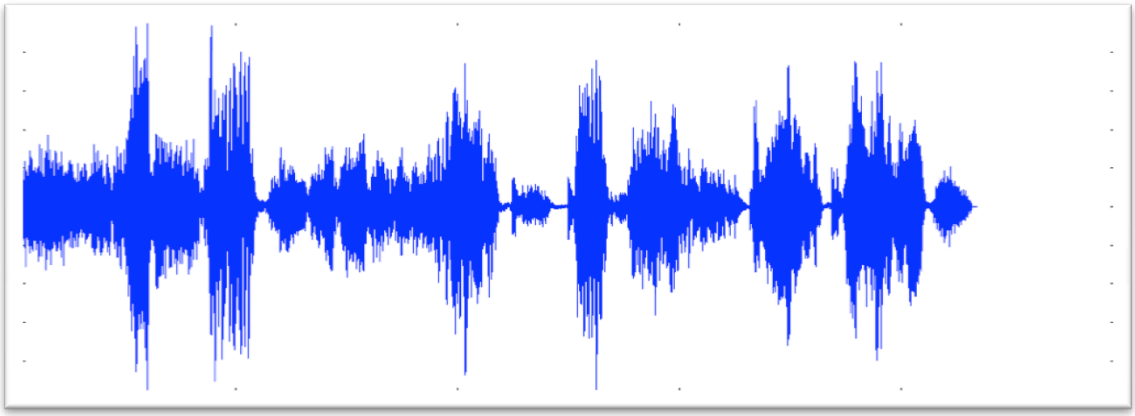
Case 2:

Using the same 2 inputs (Sound 1 & 2 from previous case),

LPC Cross Synthesis with;

Block Length = 1000 (100ms), Frame size =500 (50ms)

Excitation LPC order = 200, Residual LPC order = 100



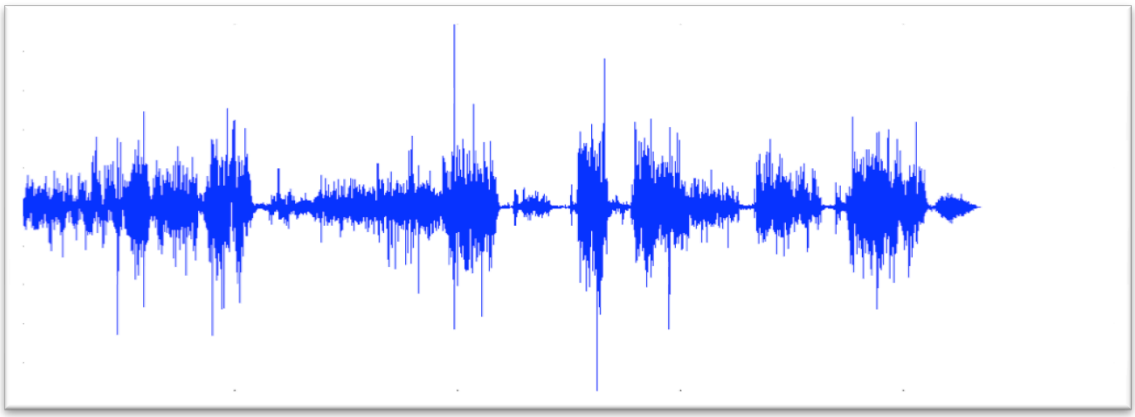
Case 3:

Using the same 2 inputs (Sound 1 & 2 from previous case),

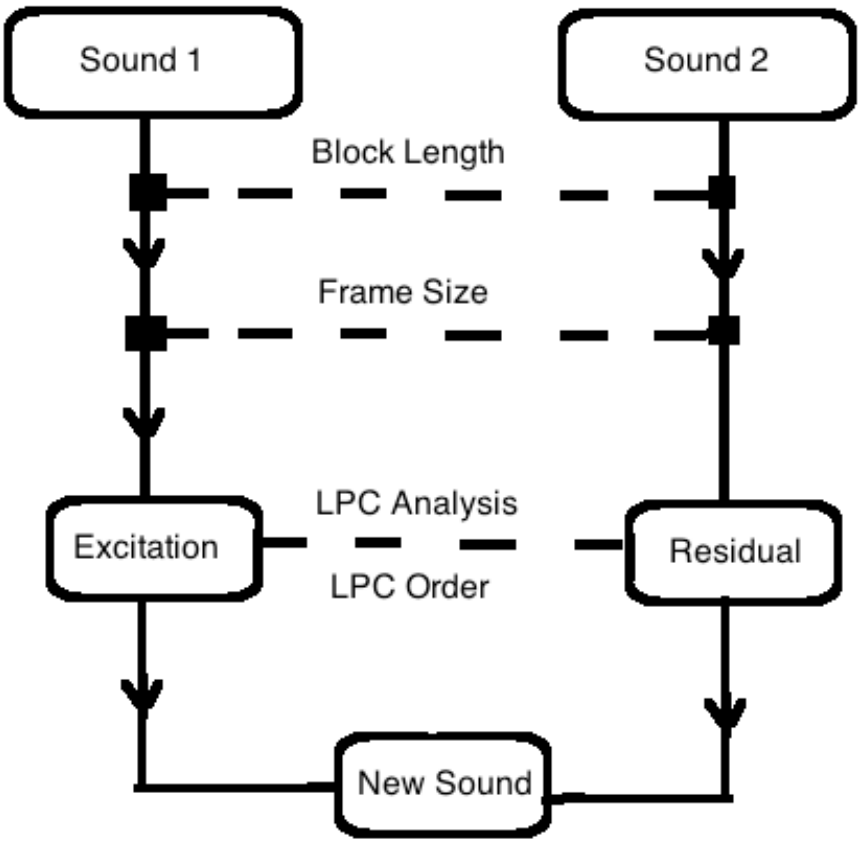
LPC Cross Synthesis with;

Block Length = 100 (10ms), Frame size =10 (1ms)

Excitation LPC order = 20, Residual LPC order = 10



Signal Flow:



Vocoder GUI: