

# Speech Synthesis

**By:**

Robert Ahlfinger  
Brenton Cheeseman  
Patrick Doody



# Speech Synthesis

**By:**

Robert Ahlfinger  
Brenton Cheeseman  
Patrick Doody

**Online:**

< <http://cnx.org/content/col10253/1.7/> >

**C O N N E X I O N S**

Rice University, Houston, Texas

This selection and arrangement of content as a collection is copyrighted by Robert Ahlfinger, Brenton Cheeseman, Patrick Doody. It is licensed under the Creative Commons Attribution 2.0 license (<http://creativecommons.org/licenses/by/2.0/>).  
Collection structure revised: December 18, 2004  
PDF generated: February 4, 2011  
For copyright and attribution information for the modules contained in this collection, see p. 27.

## Table of Contents

<b>1 Perfect Pitch: Using Software to Alter Your Voice</b> .....	1
<b>2 The Pitch Correction Algorithm: An Overview</b> .....	3
<b>3 Harmonic Detection</b> .....	7
<b>4 Reconstructing A DFT With A Pitch Shift</b> .....	11
<b>5 Voice Randomization</b> .....	13
<b>6 Length Changer</b> .....	17
<b>7 Speach Synthesis Summation</b> .....	21
<b>8 The Wyld Stallyn Team</b> .....	23
<b>Bibliography</b> .....	25
<b>Index</b> .....	26
<b>Attributions</b> .....	27



# Chapter 1

## Perfect Pitch: Using Software to Alter Your Voice<sup>1</sup>

### 1.1 Introduction on Speech Synthesis

#### 1.1.1 Overview

The ability to tweak or manipulate a person's voice has always been useful. Before the advent of digital signal processing, however, this task was extremely hard. In those days, the most sophisticated manipulations could be found in rock and roll synthesizers that used analog devices to distort the noise, producing a pseudo random feel to the voice. Other naïve approaches could be taken to alter somebody's voice, such as changing the playback speed of the clip or modulating the signal. However, these techniques just resulted in making the person speak with a lower pitch that sounded slurred or a higher pitch that resembled Alvin the Chipmunk.

#### 1.1.2 Goals

The goal of this project was to develop a more sophisticated set of voice manipulation tools using digital signal processing by developing software in Matlab. The first and most complicated tool raises or lowers the pitch of a recorded voice without changing the length of the sound or otherwise changing the characteristics of the voice. The second changes the length of the clip without altering the pitch of the voice. The resulting voices from both of these tools should sound as if the original clip had been recorded anew while instructing the person doing the talking to speak more slowly, more quickly, or with a higher or lower pitch. Finally, the third tool randomizes the voice in order to mask the identity of the speaker yet preserve her ability to communicate.

#### 1.1.3 Applications and Examples

As you can imagine, there are several potential applications for our new software. An out of tune singer can go back after a recording and tweak his or her voice to match precisely the correct tone regardless of whether the problem persists for the duration of the song or a fraction of a second. If a newscaster's segment goes over or under the preferred time allotment by a few seconds, his or her speech may be reduced or extended by exactly the necessary amount.

---

<sup>1</sup>This content is available online at <http://cnx.org/content/m12553/1.6/>.

**Pitch Shifted and Randomized Speech Examples**

Unaltered voice	Original <sup>2</sup>
Pitch Shifted Voice	Up <sup>3</sup> Down <sup>4</sup>
Randomized Voice	Random <sup>5</sup>

**Table 1.1****Length Changer Speech Examples**

Unaltered voice	Original <sup>6</sup>
Length Changed Voice	Slower <sup>7</sup> Faster <sup>8</sup>

**Table 1.2**<sup>2</sup><http://cnx.org/content/m12553/latest/fudge.wav><sup>3</sup><http://cnx.org/content/m12553/latest/fudge100.wav><sup>4</sup><http://cnx.org/content/m12553/latest/fudge-50.wav><sup>5</sup><http://cnx.org/content/m12553/latest/randfudge.wav><sup>6</sup><http://cnx.org/content/m12553/latest/male.wav><sup>7</sup><http://cnx.org/content/m12553/latest/male2.wav><sup>8</sup><http://cnx.org/content/m12553/latest/male3.wav>

## Chapter 2

# The Pitch Correction Algorithm: An Overview<sup>1</sup>

### 2.1 Time-Domain vs. Frequency-Domain

Clearly, the goal of this algorithm is to take an input voice signal, change the pitch of the voice, and output the otherwise unaltered signal. In order to do so, the first step is to decide whether to analyze and manipulate these signals in the time domain or the frequency domain. Because our algorithm is primarily concerned with quickly identifying and shifting individual frequencies, we worked solely in the realm of the frequency domain. Of course, there are effective ways to deal with this problem without the frequency domain. However, as will later become obvious, there are some very useful techniques we developed that are not possible in the time domain. With pitch correction it seems that Parseval has made a mistake; there is simply more power in the spectrum.

### 2.2 Basic System Model

Now that we have decided how to look at our signals, we need to develop a general layout and strategy for how it will work.

#### 2.2.1 General Process Summary

First, the signal is “Matricized,” a term we coined to describe our particular algorithm to break up the signal and convert it into the Fourier Domain. Basically, the signal comes in as a long string of sampled values that together represent the whole sound. We, in turn, convert this vector of samples into a matrix for which each column represents the spectrum of one slice, or chunk, of the signal. Although any chunk size could be used, we found the best performance with chunk sizes of 512 samples, which represents about .02 seconds of sound for the 22 kHz sampling rate used on our signals. Next, we take the Discrete Fourier Transform for each of the chunks, showing us the frequencies present at every given moment during the speech. These DFTs are then collected into a matrix with 512 rows and as many columns as there are .02 second long chunks in the voice. With a given chunk, our Harmonic Detection algorithm has the extremely difficult task of accurately and consistently identifying the first harmonic of the voice. With that information in hand, the program reconstructs a new DFT representation for the current chunk by first sliding the first harmonic down the spectrum by the desired shift in pitch, and then following up with all of the other harmonics, shifting each one by an incremental multiple of the first shift. After all of the chunks have been processed and put into a

---

<sup>1</sup>This content is available online at <<http://cnx.org/content/m12539/1.5/>>.

matrix, this new matrix is “Dematricized” in order to convert the information back into the time domain as a new string of digital samples that represent the freshly manipulated voice.

## 2.3 Detailed System Model: Step-by-Step

The pitch synthesizer relies on several algorithms to properly alter the pitch of a person’s voice without mutilating its clarity.

### 2.3.1 Matricize

First, the signal is “matricized,” a term we coined to represent the task of transforming the string of speech samples into a matrix whose columns each represent the spectrum of an overlapping rectangular window, or chunk, of the signal. Each portion of the voice is contained twice in this information since exactly one half of each chunk is overlapped and contained within an adjacent chunk. Next, each column of the matrix is processed separately, meaning we attempt to change the characteristics of the voice one piece at a time and do so redundantly.

### 2.3.2 Harmonic Detection

Now that we have isolated the spectrum of a chunk of our signal, we use a harmonic detector to find the first harmonic of the voice at that particular point in time. This task is harder than it first appears and its level of accuracy makes the single biggest contribution to the functionality and accuracy of the pitch synthesizer as a whole. Voiced vowel noises are the only parts of speech that contain pitch, so they need to be processed differently than the rest of the signal. However, since there are many periods of noise as well as voiced (s and z sounds) and unvoiced (like f and t) fricatives alongside these important voiced vowel noises, the harmonic detector must wade through each chunk and first determine whether or not it is dealing with a voiced vowel noise. If so, it computes the index of the first harmonic of the sample by taking the DFT of the first half of the magnitude of the DFT of the original signal chunk. The resulting spectrum will have a very large DC component which represents the grab bag of frequencies present in the original signal, as well as repeating peaks corresponding to the only periodic aspect of the original DFT – the signal’s harmonics. Therefore, the harmonic detector compares the DC amplitude with the next biggest peak, determining simultaneously whether or not this chunk is likely to be a voiced vowel noise and if so the frequency of its first harmonic.

### 2.3.3 Frequency Shift

With this information in hand, our program determines how far each and every frequency must be shifted. Since you interpret the pitch of a voice as the frequency of its first harmonic, the first harmonic is shifted by exactly the desired result. In turn, the frequency of every harmonic is a multiple of the first, so the second harmonic must be shifted twice as far as the first, the third is shifted three times as far, and so on. In fact, we use the index of the first harmonic to determine how much each and every frequency in the original chunk will shift to build up the first half of the DFT for our new, processed chunk. We are trying to alter the pitch without affecting the length of the sound, so this stretched out DFT must be cut off at half the length of the original DFT, at which point we have the completed version of the front half of the new DFT. To complete the second half, we rely on the DFT’s symmetry properties, noting that our original and final sound signals are both purely real. Therefore, the real portion of the DFT is mirrored about the middle, and the imaginary portion is mirrored and flipped. Finally we have completely processed the given window of the original signal.

### 2.3.4 Reconstruction

To reconstruct the original signal, these processed DFT’s each become a column of another matrix which is then “dematricized” by taking the inverse FFT of each spectrum and placing them side by side into a new

signal that has the same length as the original. The only difference of course being that the voice in the signal has become as high or as low as the desired shift.

### Pitch Shifted Speech Examples

Unaltered voice	Original <sup>2</sup>
Pitch Shifted Voice	Up <sup>3</sup> Down <sup>4</sup>

**Table 2.1**

---

<sup>2</sup><http://cnx.org/content/m12539/latest/fudge.wav>

<sup>3</sup><http://cnx.org/content/m12539/latest/fudge100.wav>

<sup>4</sup><http://cnx.org/content/m12539/latest/fudge-50.wav>



## Chapter 3

# Harmonic Detection<sup>1</sup>

### 3.1 The Biggest Obstacle

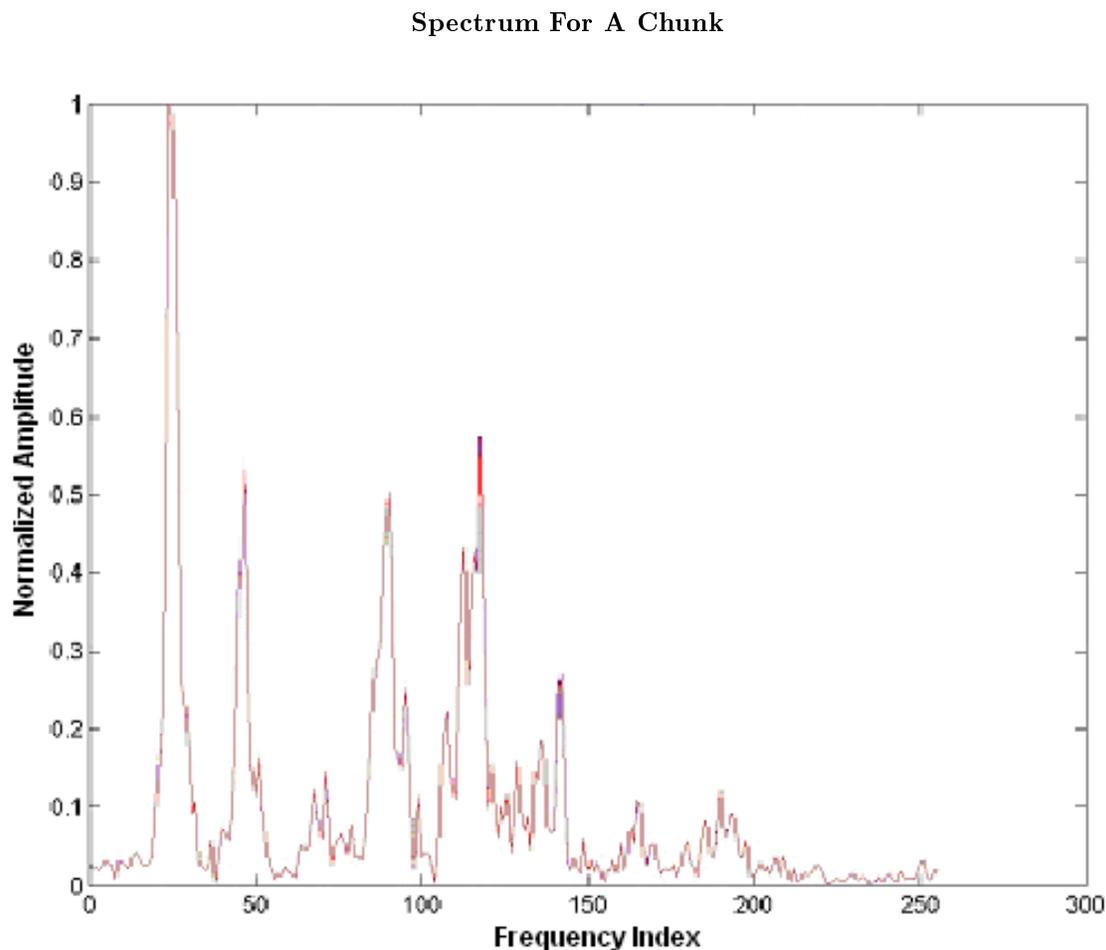
There is no question about it. For this algorithm to work correctly, the obstacle that is simultaneously most critical and most prone to error is accurately and consistently detecting the first harmonic in a chunk of speech. For instance, if the software incorrectly thinks the person speaks with a very deep voice in a particular chunk, the resulting frequency shift to the actual first harmonic will be enormous. The ratio of the correct index to the approximated index of the first harmonic is equal to the ratio of the actual shift in pitch and the desired shift in pitch after the voice manipulation is complete.

### 3.2 A Brief Overview of Harmonics and Speech

Why does middle C sound different from a piano, a trumpet, or an opera singer? After all, they all have the same pitch. The difference rests not in the base frequency that is being played per se, but rather in the sound's harmonics. Whenever an instrument (or a voice) makes a sound, the pitch you hear is called the first harmonic, it is the lowest and usually the strongest frequency emitted. However, this is not the only noise that is produced. There are also waves produced at all the higher octaves on the same note. The sound produced exactly one octave higher than the first harmonic is the second harmonic, the next octave up is the third harmonic, and so on. Looking at the Fourier Domain, it is important to remember that each octave, and therefore each harmonic, is exactly twice the frequency of the one below it. The relative strength or weakness of each individual harmonic gives each instrument a unique sound. In the case of speech, our vocal cords determine the pitch and produce the harmonics while our mouths individually dampen each harmonic in a set pattern to make a particular vowel. Consonants, unlike vowels, do not have a pitch nor do they have harmonics. A person's articulation of an 's' or 'z' sound, for instance, does not change depending on whether or not he has just been kicked in the groin.

---

<sup>1</sup>This content is available online at <http://cnx.org/content/m12555/1.5/>.



**Figure 3.1:** DFT of one 512 sample chunk of a speech signal.

### 3.3 Multitasking

Because consonants (along with periods of silence or noise) do not have pitch, our harmonic detection algorithm has the double duty of determining if a vowel noise is being produced in the first place, and if so, the location of the first harmonic as well. If a ‘k’ sound is mistaken for a vowel, for instance, the pitch synthesizer would attempt to shift its frequencies up the spectrum, resulting in a nasty high frequency noise that would not be mistaken for a ‘k’.

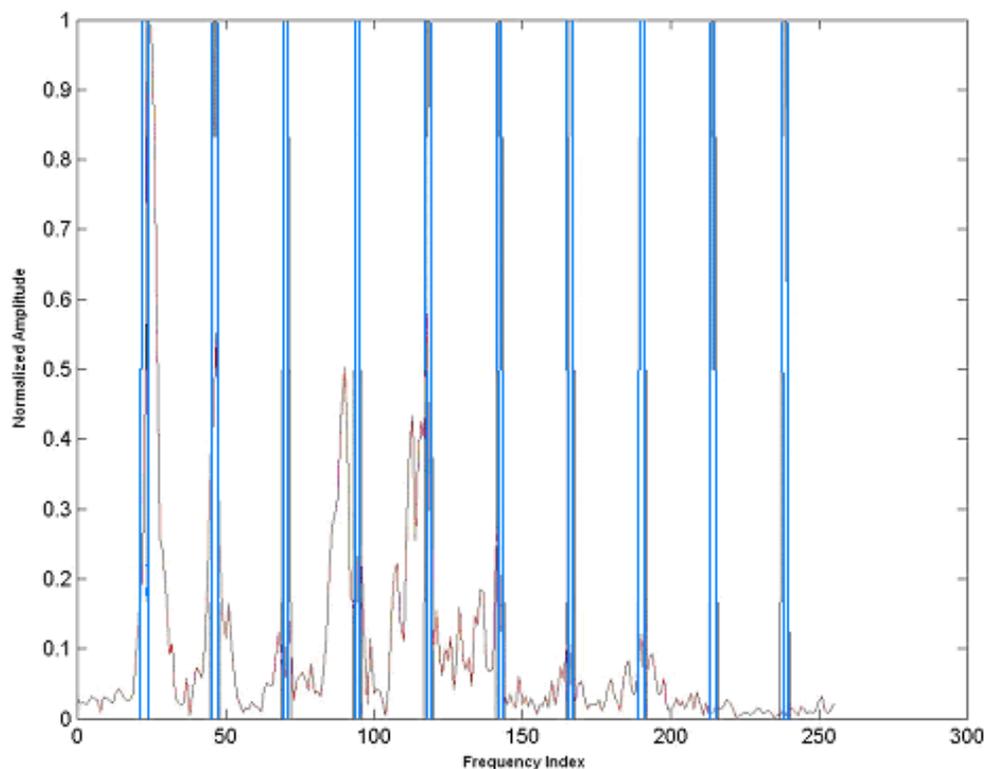
### 3.4 A Naïve Approach

Before hitting gold, we developed several techniques to do this job that all fell short of satisfaction. One such technique was to construct a zero padded vector equal to the length of the DFT that had ones only at multiples of an integer that was a candidate for being the location of the first harmonic. After taking a dot

product of these two vectors, we would try again for a different candidate index. The thought was that the largest resulting dot product would correspond to the correct placement of harmonics since they lined up with the largest values in the spectrum. However, if the harmonics do not appear at exact multiples of the candidate integer, this technique is worthless. Too much noise ruins its effectiveness as well.

---

### DFT and an Example Comparison Vector



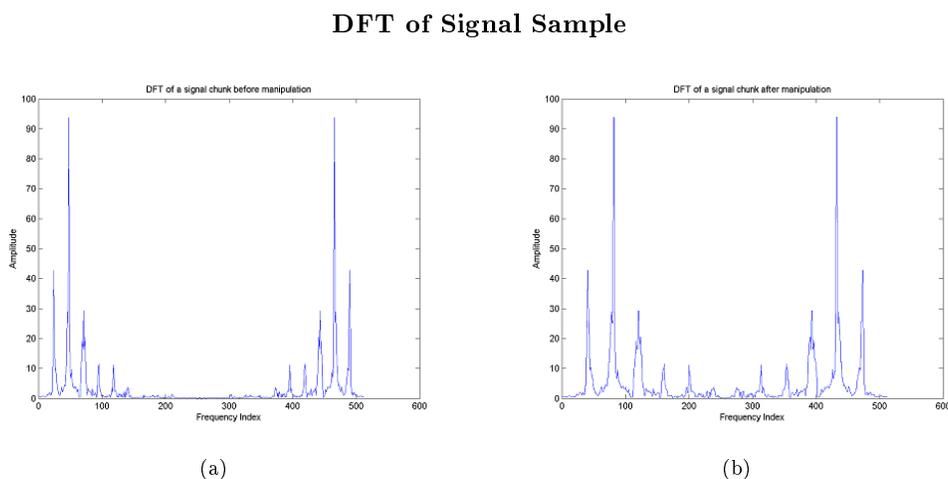
**Figure 3.2:** DFT of one 512 sample chunk of a speech signal overlapped with a comparison vector of about 24Hz, illustrating the general location of the harmonics.

---

To get rid of the first problem, we started using vectors that had a window of three ones around integer multiples of the candidate to allow some wiggle room for the actual location of the higher harmonics. Finally, we tried taking the logarithm of the values in the spectrum with the hope that the borders of the harmonics would stick up much farther than adjacent frequencies. If this held true to a greater extent than any other random locations in the spectrum, we could isolate the harmonics with the right type of high pass filter. In the end, we discovered each of these techniques were pretty good at finding harmonics in a certain kind of spectrum and failed miserably in other conditions. We needed something that worked all the time.

### 3.5 Hitting the Jackpot

The algorithm that works far and away better than any others we tested relies on the principle that the DFT of a chunk, like the time domain version of the chunk itself, has non-periodic and periodic aspects. In the first half of the DFT, the only repetition comes from the evenly spaced peaks of the harmonics. Everything else, whether noise or spectrum elements resulting from a consonant, is not periodic. Therefore, we take the first half of the magnitude of our DFT as a new signal to look at. Naturally, to analyze it we take the DFT of this vector, and look at the magnitude of the result. So now we have the tongue twisting magnitude of the DFT of the magnitude of the first half of the DFT of the original signal chunk. The DFT of the DFT!



**Figure 3.3:** (a) Discrete Fourier Transform showing the spectra for one 512 sample chunk of the speech signal before manipulation by the Pitch Synthesizer. (b) Discrete Fourier Transform of the original DFT spectra for one 512 sample chunk of the speech signal after manipulation by the Pitch Synthesizer.

---

The new spectrum invariably contains a very large DC value and a lot of power on the low end of the spectrum resulting from the necessarily positive average value of a magnitude plot (remember we used the magnitude of the original DFT) along with non-periodic elements from noise or consonants. But for  $n$  greater than two or three, this new DFT goes straight to zero and stays there until it hits the only periodic element of the original DFT—the harmonics. By ignoring the first couple of values on our new spectrum, we very accurately find the first harmonic by taking the first frequency with a magnitude that is on par with the large DC value. If no such frequencies exist, we can safely assume that the chunk does not contain a vowel and does not need manipulation. This new sneaky trick (taking the DFT of the DFT) is very precise and extremely consistent, especially in the presence of noise. In fact, had we discovered this earlier, there is probably another whole project in developing this particular tool in much greater depth. It could be used to automatically detect different types of human sounds, such as separate voiced and unvoiced fricative sounds as well as the tried and true vowels. Another use would be to compute the signal to noise ratio without having access to the original signal and figuring out whether the signal chunk should even be considered worthy of processing because of the prevalence of noise.

## Chapter 4

# Reconstructing A DFT With A Pitch Shift<sup>1</sup>

### 4.1 Reconstructing the First Half of the DFT

With the first harmonic in hand (if, of course, it exists) the program is ready to manipulate the signal chunk by building a new DFT from scratch but based upon the original. The pitch you hear is the position of the fundamental frequency – the first harmonic. So the new DFT must take the frequencies at and around the original first harmonic and copy them, without alteration, to a spot further down the spectrum. Further, in fact, by exactly the desired pitch shift. The frequency of the second harmonic needs to be twice as large as the first so that the new voice sounds like it came from a real person, so the second harmonic and its neighboring frequencies are shifted twice as far down the spectrum as the first group. This is repeated with every harmonic in a similar way until half of the new DFT is full.

### 4.2 Reconstructing the Second Half of the DFT

To reduce computational complexity, there is no need to perform this same task starting at the end of the original DFT working our way towards the middle. We know that the resulting time-domain signal we produce must be comprised of real numbers since people are going to actually listen to it, so we can exploit the symmetry properties of the DFT. That is, the DFT of a real-valued signal follows rule that the real part of the samples in the first half is a mirror image of the real part of the samples in the second half. Similarly, the imaginary part of the samples in the second half is a flipped (negative) mirror image of the imaginary part of the samples in the second half. This line simple for loop constructs the entire second half of the new DFT without any further analysis or computation. Here is an example of the magnitude of the spectrum for a chunk of signal before and after pitch manipulation. Notice how the harmonics are not merely shifted over, but spread out as well.

---

<sup>1</sup>This content is available online at <http://cnx.org/content/m12552/1.3/>.



## Chapter 5

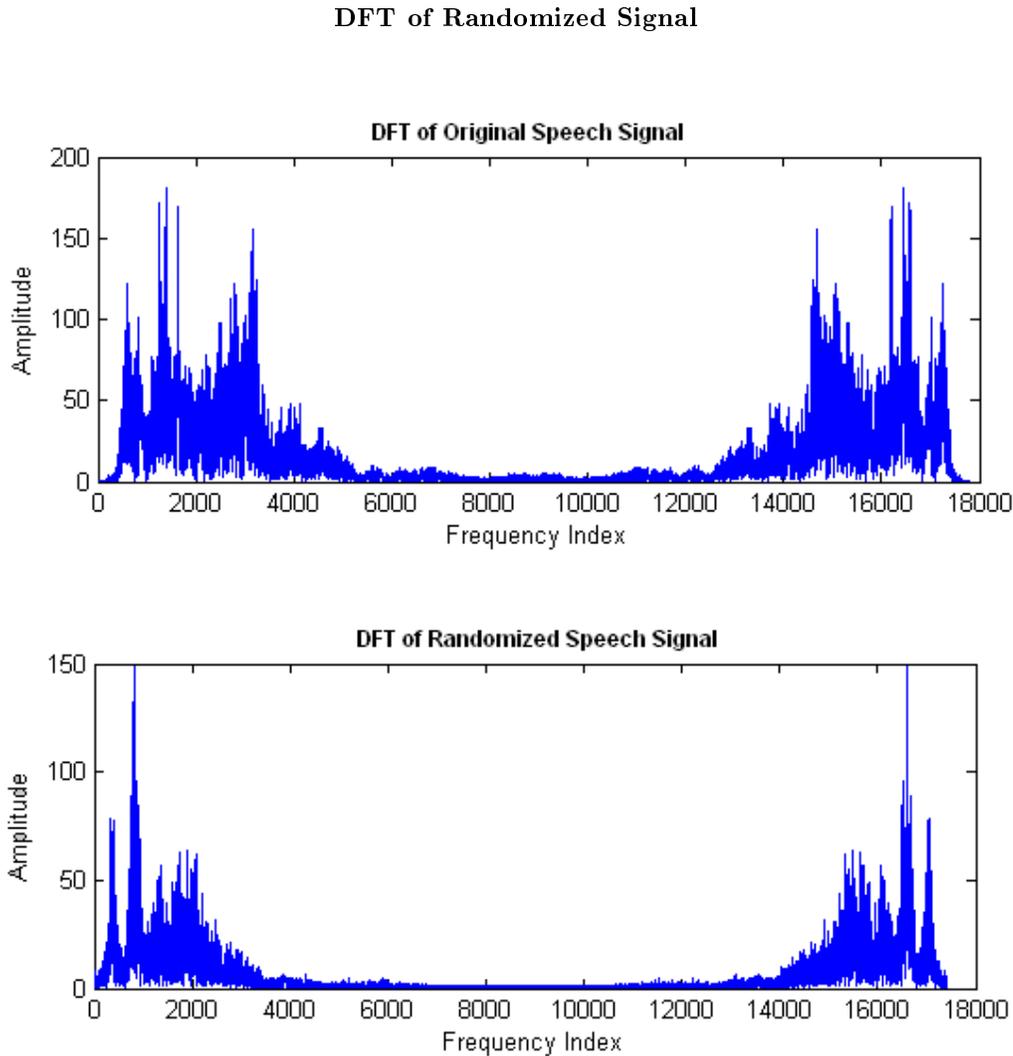
# Voice Randomization<sup>1</sup>

### 5.1 Initial Approach

When two people speak with the same pitch, there is still no mistaking one for the other; the uniqueness of a voice goes beyond its tone. The placement of harmonics, then, clearly does not make a voice distinguishable since two people with identical pitch have harmonics at exactly the same locations. Rather, the ability to identify a voice comes from the relative height of each harmonic to the next, just like the heights of each harmonic on a clarinet and a guitar make these instruments sound different even as they play the same note.

---

<sup>1</sup>This content is available online at <http://cnx.org/content/m12556/1.4/>.



**Figure 5.1:** DFT of one 512 sample chunk of a speech signal after it has had its pitches randomly altered.

---

With this in mind, our first algorithm tackled the problem by first using the harmonic detection described earlier to pinpoint the location of each harmonic. Using this information, the height of each harmonic was randomly lowered or raised by a slight amount. Usually, though, the resulting voice sounded just like the original with some noise added in on top of it. After fooling around with this concept for some time to no avail, we reached the conclusion that the idea is solid, but that to make up a new voice requires much more finesse than simply making the magnitude of each harmonic higher or lower. Without perfectly adapting the phases and making sure that the envelope of the magnitudes is a shape that can be comprehended by a human ear as real speech, the only result is linearly adding a new signal to our old one. The DFT of the new signal is equal to the additions we made to the harmonics of the voice.

## 5.2 Simplification of Process Using the Speech Synthesizer

The second attempt at a voice randomizer directly utilizes our pitch shifting algorithm and works much better. First, the signal is matricized just like before. But instead of processing each chunk in the same way, our algorithm asks the pitch shifter to shift each chunk separately, specifying a different and random shift every time. The result is a voice with a pitch that changes wildly and extremely quickly, making it impossible to tell who it is with your raw hearing. The main drawback with this technique is that there is no true security or identity masking. The NSA could easily break the signal into the same 512 sample long chunks and analyze them individually along with a normal sample of the voice to determine a potential match. However, for certain purposes this randomizer performs superbly.

### Randomized Speech Examples

Unaltered voice	Original <sup>2</sup>
Randomized Voice	Random <sup>3</sup>

Table 5.1

---

<sup>2</sup><http://cnx.org/content/m12556/latest/fudge.wav>

<sup>3</sup><http://cnx.org/content/m12556/latest/randfudge.wav>



## Chapter 6

# Length Changer<sup>1</sup>

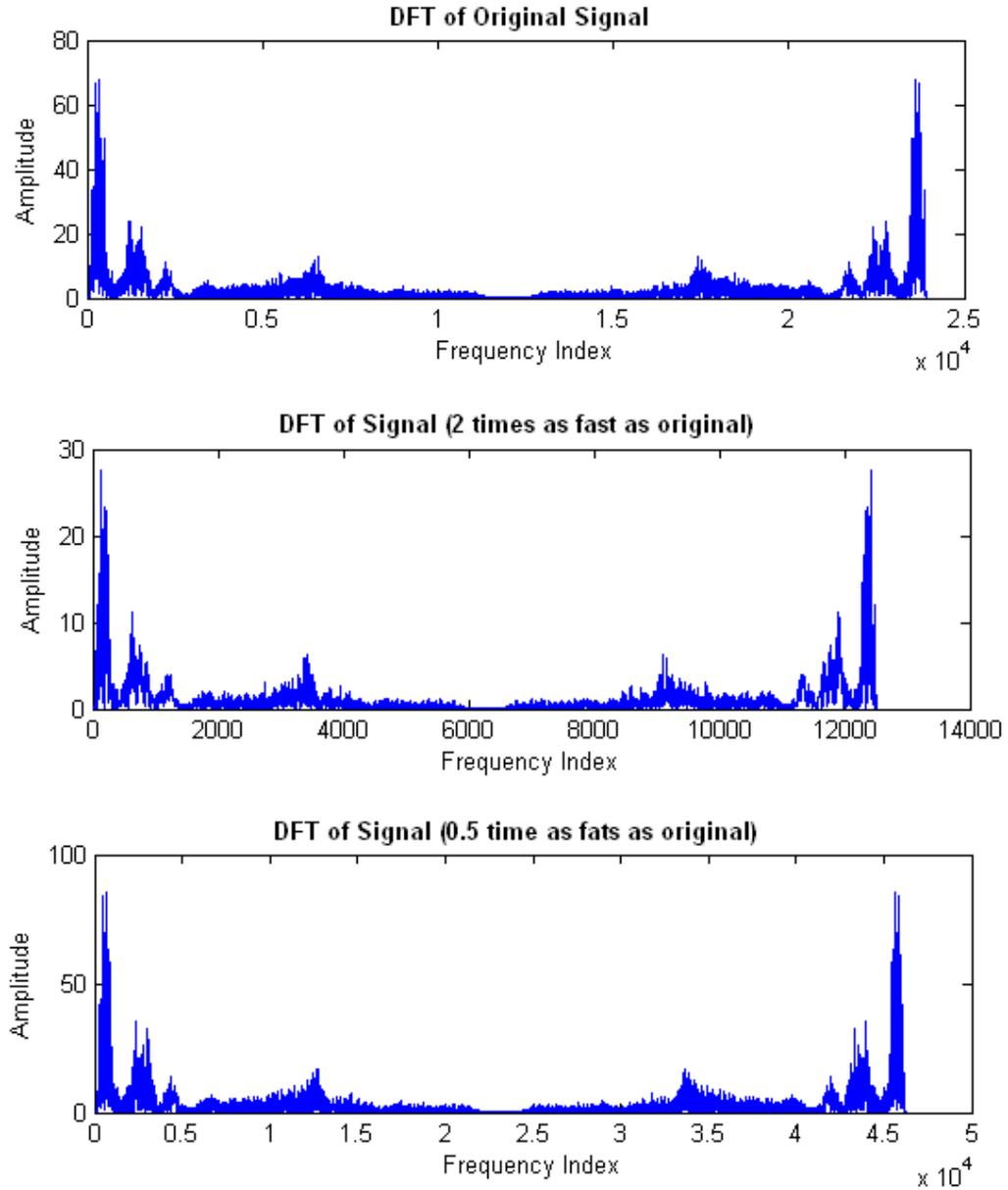
### 6.1 Simple Concept, Difficult Implementation

The third and final voice manipulation tool we developed changes the length of the signal without altering its pitch or clarity, and the basic strategy to do so is extremely simple. After breaking the signal into chunks by matricizing, some of the chunks are either trashed or repeated in order to compress or extend the length of the signal. Since nobody can perceive a voice's changing within the span of .02 seconds or less, this repetition never creates an audibly repeated noise. It can only create an audibly lengthened or shortened noise. Playing this sound back, though, sounds incredibly choppy, as if you were listening to the sound version of strobe lights. But if concatenating or removing signal windows in and of itself does not create the desired result, what could the problem be?

---

<sup>1</sup>This content is available online at <http://cnx.org/content/m12554/1.7/>.

---

**DFT of Length Changer Signals****Figure 6.1**

---

## 6.2 Phase Makes All The Difference

Upon closer inspection, it is obvious that the phase of the complex sinusoids at the beginning of a chunk is often very different than the phase at the end of the same chunk or of a previous chunk. After slapping two windows together, this sharp phase difference becomes very clear, producing our unacceptably choppy sound. To correct this, the length changing algorithm makes another run past each window after the new signal has been constructed, this time taking care to compute the phase at the end of the previous chunk  $\theta_o$ , the old  $\theta$ , and the phase the beginning of the next chunk  $\theta_n$ , the new  $\theta$ . Next, every value of the next chunk's DFT gets multiplied by  $e^{i(\theta_o - \theta_n)}$ . As a result, the phase at the beginning of the next chunk equals the phase at the end of the previous one, and the phase will transition smoothly between all other points in time. This process is repeated for each and every chunk, resulting in the complete removal of the stutters.

### Length Changer Speech Examples

Unaltered voice	Original <sup>2</sup>
Length Changed Voice	Slower <sup>3</sup> Faster <sup>4</sup>

Table 6.1

---

<sup>2</sup><http://cnx.org/content/m12554/latest/male.wav>

<sup>3</sup><http://cnx.org/content/m12554/latest/male2.wav>

<sup>4</sup><http://cnx.org/content/m12554/latest/male3.wav>



## Chapter 7

# Speech Synthesis Summation<sup>1</sup>

### 7.1 Concluding Thoughts on Speech Synthesis

As you can tell from the sound clips in Chapter 1, all three algorithms work very well. The potential applications for these tools, especially in the recording industry, are boundless. With more reliance upon digital signal processing and less reliance upon recording a segment two or three or more times, clips may be produced much more quickly, for less money, and with better quality.

The shift synthesizer took by far the most time and effort to produce, mostly due to the difficulty of implementing a high quality harmonic detection algorithm. The randomizer ended up directly utilizing the pitch software, which provided even more incentive for us to improve our harmonic detection. Finally, the length changing program involved much more phase analysis than we had previously expected.

These triumphs and failures all provided great experience in dealing with some important issues in digital signal processing in general and analysis of human voice signals in particular. Over the course of developing the project, we dealt extensively with the characteristics of harmonics, the spectra of voiced and unvoiced fricatives, phase matching, and techniques for analysis and reconstruction of DFT vectors. All in all, the project was a huge success.

### 7.2 Acknowledgements

**Special thanks to Richard Baraniuk and MATLAB 6.5 for their generous assistance on the project.**

---

<sup>1</sup>This content is available online at <<http://cnx.org/content/m12551/1.3/>>.



## Chapter 8

# The Wyld Stallyn Team<sup>1</sup>

### 8.1 Wyld Stallyns

The Wyld Stallyn Team consisted of the following people, each contributing evenly to the work load for this project. The work for the project included: Developing and researching appropriate methods, writing code, testing and debugging code, creating the poster and writing the connexions report.

---

**Robert Ahlfinger**



**Figure 8.1:** Junior at Lovett College. email: ahlfing@rice.edu

---

---

<sup>1</sup>This content is available online at <<http://cnx.org/content/m12550/1.3/>>.

**Brenton Cheeseman****Figure 8.2:** Junior at Hanszen College. email: [bcheese@rice.edu](mailto:bcheese@rice.edu)

---

**Patrick Doody****Figure 8.3:** Junior at Hanszen College. email: [pdoody@rice.edu](mailto:pdoody@rice.edu)

---

# Bibliography

- [1] Stefan Karneback. "expanded examinations of a low frequency modulation feature for speech discrimination". Technical report, Centre for Speech Technology, KTH and Flextronics International. <[http://www.speech.kth.se/~stefank/Expand\\_inv\\_lf.pdf](http://www.speech.kth.se/~stefank/Expand_inv_lf.pdf)>, 2004.
- [2] Keith R Kluender. 8220;effects of contrast between onsets of speech and other complex spectra8221;. Technical report, Psychology Department at University of Wisconsin. <<http://www.physiology.wisc.edu/pubs/JAS02225.pdf>>, 2004.
- [3] Meng Wang. 8220;an algorithm for v/uv/s segmentation of speech8221;. Technical report, Computer Science Department at NCSU. <<http://www.csc.ncsu.edu/faculty/rodman/Segmentor.pdf>>, 2004.

## Index of Keywords and Terms

**Keywords** are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

**E** Elec 301, § 1(1), § 2(3), § 3(7), § 4(11),  
§ 5(13), § 6(17), § 7(21)

**H** Harmonics, § 3(7), § 4(11)

**L** Length, § 7(21)

**P** pitch, § 1(1), § 2(3), § 3(7), § 4(11), § 7(21)

**R** Randomizer, § 7(21)

**S** Stallyn, § 8(23)

**T** Team, § 8(23)

**W** Wyld, § 8(23)  
wyld stallyns, § 1(1), § 2(3), § 3(7), § 4(11),  
§ 5(13), § 6(17), § 7(21)

## Attributions

Collection: *Speech Synthesis*

Edited by: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/col10253/1.7/>

License: <http://creativecommons.org/licenses/by/2.0/>

Module: "Perfect Pitch: Using Software to Alter Your Voice"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12553/1.6/>

Pages: 1-2

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: [http://creativecommons.org/licenses/by/1.0](http://creativecommons.org/licenses/by/1.0/)

Module: "The Pitch Correction Algorithm: An Overview"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12539/1.5/>

Pages: 3-5

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: [http://creativecommons.org/licenses/by/1.0](http://creativecommons.org/licenses/by/1.0/)

Module: "Harmonic Detection"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12555/1.5/>

Pages: 7-10

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: [http://creativecommons.org/licenses/by/1.0](http://creativecommons.org/licenses/by/1.0/)

Module: "Reconstructing A DFT With A Pitch Shift"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12552/1.3/>

Pages: 11-12

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: [http://creativecommons.org/licenses/by/1.0](http://creativecommons.org/licenses/by/1.0/)

Module: "Voice Randomization"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12556/1.4/>

Pages: 13-15

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: <http://creativecommons.org/licenses/by/2.0/>

Module: "Length Changer"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12554/1.7/>

Pages: 17-19

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: [http://creativecommons.org/licenses/by/1.0](http://creativecommons.org/licenses/by/1.0/)

Module: "Speech Synthesis Summation"

By: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

URL: <http://cnx.org/content/m12551/1.3/>

Page: 21

Copyright: Robert Ahlfinger, Brenton Cheeseman, Patrick Doody

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Wyld Stallyn Team"

By: Brenton Cheeseman, Robert Ahlfinger, Patrick Doody

URL: <http://cnx.org/content/m12550/1.3/>

Pages: 23-24

Copyright: Brenton Cheeseman, Robert Ahlfinger, Patrick Doody

License: <http://creativecommons.org/licenses/by/1.0>

## **Speech Synthesis**

This is a course that covers techniques for altering speech or sound in several different ways using signal processing.

## **About Connexions**

Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.