Connexions module: m12554

# LENGTH CHANGER\*

## Robert Ahlfinger Brenton Cheeseman Patrick Doody

This work is produced by The Connexions Project and licensed under the Creative Commons Attribution License  $^{\dagger}$ 

#### Abstract

This module discusses one technique used to change the length of a signal without changing its pitch.

#### 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>\*</sup>Version 1.7: Aug 12, 2005 4:19 pm GMT-5

<sup>&</sup>lt;sup>†</sup>http://creativecommons.org/licenses/by/1.0

Connexions module: m12554 2

### **DFT** of Length Changer Signals

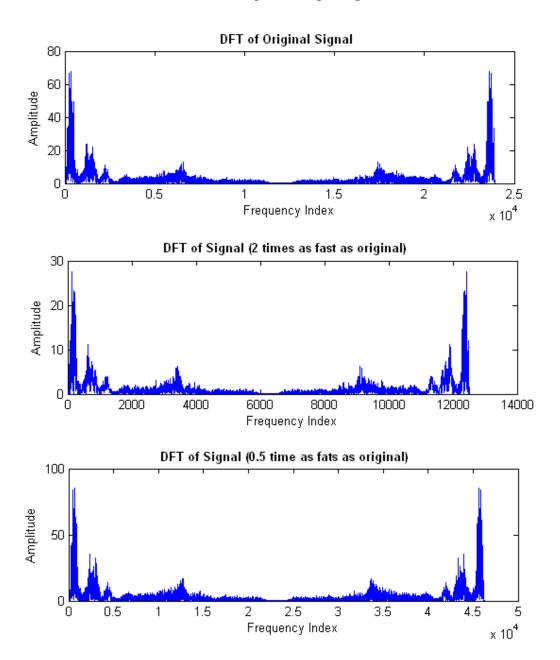


Figure 1

Connexions module: m12554 3

#### 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>1</sup>
Length Changed Voice	Slower <sup>2</sup> Faster <sup>3</sup>

Table 1

 $<sup>^{1} \</sup>rm http://cnx.org/content/m12554/latest/male.wav$ 

<sup>&</sup>lt;sup>2</sup>http://cnx.org/content/m12554/latest/male2.wav

 $<sup>^3</sup>$ http://cnx.org/content/m12554/latest/male3.wav