ALTERATION OF THE FREQUENCY PROFILE OF VOICE SIGNALS

Essaid Bouktache, Chandra R Sekhar, Omer Farook
Purdue University Calumet

Abstract: This paper explores the feasibility of a frequency translation method applied to voice signals where the higher frequencies are reallocated to a lower range where the human ear is most sensitive. Each frequency component of a voiced/unvoiced speech undergoes a certain translation to a lower frequency by an amount which depends on its position on the frequency axis. Higher frequencies would have a much higher shifting factor than the lower ones. Due to this frequency translation mechanism, the speech spectrum will have most of its power distributed among the lower frequency tones. Because of its predominance of lower frequency components, this system could be found useful in hearing aids, voice mail applications, telephone, or radio communications. This speech processor is being implemented on the Analog Devices ADSP-2101 fixed-point digital signal processor, while the main principle is illustrated using MATLAB.

I. Introduction

Even though the normal human hearing range can extend to about 15 KHz or more, speech is confined mainly in the range which extends from 100 Hz to about 5 KHz[1]. In elderly people, the ear looses its sensitivity to the high-pitched tones, resulting in poor and inadequate hearing. Since, even with age, the ear is more sensitive to lower frequencies, it would be desirable to shift most of the energy contained in the upper frequencies to a lower range, provided the speech information is fairly preserved. This is in contrast with conventional hearing aids, for example, which rely mainly on amplification and filtering. Excessive amplification of both low and high frequencies could result in severe damage of the remaining hearing of an elderly person. By forcing the speech power to be reallocated within the low and mid-frequencies, unnecessary amplification can be avoided.

This paper addresses the alteration of the frequency spectrum of speech signals using a nonlinear frequency mapping technique, where the lower frequencies undergo no change, or very little, in this transformation process, whereas the higher tones are subject to more shifting. The general principle consists of computing the frequency spectrum of spoken words using the FFT (Fast Fourier Transform) algorithm, manipulating that spectrum so that most of the energy is transferred to a lower frequency range by application of the frequency mapping technique to be presented here, and finally taking the inverse FFT of the transformed spectrum.

This method is still being implemented on the Analog Devices ADSP-21O 1 fixed-point digital signal processor. The implementation is the most critical part since the system uses nonlinear frequency mapping, and must respond to red-time situations, where, for example, a spoken word into the system must be heard back altered, but understandable, reflecting the change in its frequency spectrum. The challenge remains in the choice

of related parameters such as establishing the optimum time window needed to process the speech signal, and the degree of spectrum alteration which can be done without severely affecting the speech intelligibility.

This paper begins with a review of some FFT fundamentals, followed by a brief analysis of speech. The frequency mapping method, which is the main concern of this paper, is then presented and illustrated using MATLAB. Real-time results on the ADSP-2101 platform are still being investigated.

II. FFT Fundamentals

The mapping method to be discussed uses the FFT and its inverse as the essential tool. Therefore, it is appropriate to start with a brief review of some FFT fundamental properties.

1. Time Record and Frequency Resolution

When taking the FFT of a time domain signal x(t), a finite portion of the signal waveform must be digitized resulting in a finite series x(n) with N discrete points. When the FFT algorithm is applied to this set of points, it assumes that this portion is periodic in the time domain, whether or not the original signal (such as speech) is periodic. This portion of equally spaced points which are T seconds apart, is called a Time Record, where T is the sampling interval. Call this time-record t_R and you have $t_R = NT$. The FFT algorithm, which computes the discrete Fourier transform of the time series x(n) results in another periodic series X(k) which represents the spectrum of the signal in the frequency domain, with N discrete and equally spaced components, ΔF apart from each other, ΔF being the frequency resolution. The frequency resolution is directly related to the time record by $\Delta F = 1/t_R$. Hence, the larger the time record the better the frequency resolution. The sampling frequency f_S is such that $f_S = NAP$, with $f_S = 1/T$. Finally, the frequency spectrum is periodic, with period f_S.

2. FFT Example

The following example illustrates the above FFT parameters. Figure 1 shows a time-domain signal x(t) which represents a summation of nine sinewaves as given by the equation:

$$x(t) = \sum_{m=1}^{m=9} m \sin(2\pi m f_1 t) \text{ with } f_1 = 100\text{Hz}.$$

Figure 2 is the spectrum of this unsampled signal, showing only the positive frequencies.

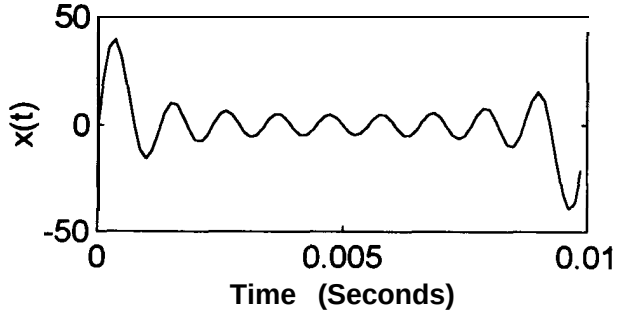

Figure 1. Time-Domain Signal x(t) for FFT Example.

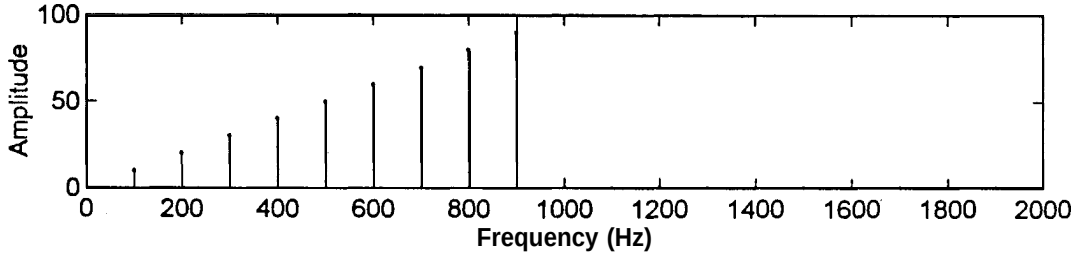

Figure 2. Spectrum of the Unsampled Signal x(t).

As can be seen in Fig. 2, the lowest frequency is f_L=100Hz, and the highest f_H = 900Hz. Assume x(t) is sampled at f_s = 2KHz (which satisfies the sampling theorem to sample at least twice the highest frequency). Taking the FFT of the corresponding sequence results in the periodic spectrum shown in Fig.3, with period fs . Again, only the positive side is shown.

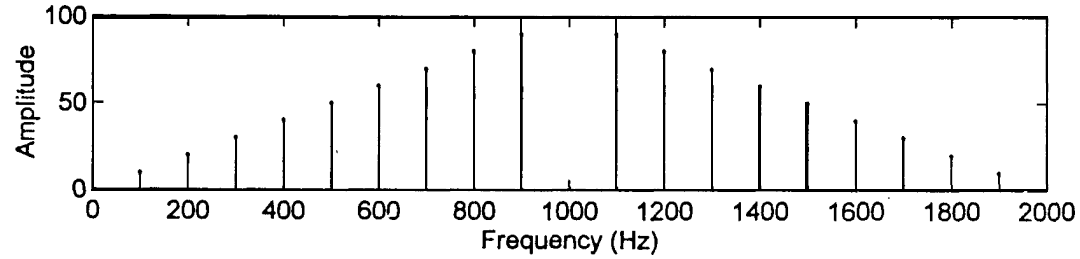

Figure 3. A 20-point FFT of the Sampled Signal x(n).

As expected, the frequency resolution in this case is just the frequency increment f_1, so ΔF = 100Hz. The time record is $t_R = 1/\Delta F = 10ms$, as can be seen in Fig. 1. The number of points in the FFT is $N = f_s/\Delta F = 2000/100 = 20$ points, which is also the number of intervals between O and fs in Fig. 3, and the number of samples in the time-domain sequence, since $t_R = NT = 20/2000 = 10ms$. We will use the same example in a later section to demonstrate the frequency translation mechanism.

III. Voiced and Unvoiced Speech

Since the frequency mapping technique to be presented in the next section is applied to speech signals, this section covers some background about the way human speech is produced. In general, speech can be synthesized into two different types of sounds: voiced and unvoiced. Voiced speech is produced by a train of air pulses exciting the vocal cords. On a short time basis (say 50ms or less), these pulses are periodic with a fundamental carrier frequency which differentiates one speaker from another. For example, the same word can be spoken by a thin voice (high carrier frequency) or by a thick voice (low carrier frequency) and remains understandable. Fundamentally, one can process the speech signal to switch from one carrier frequency to the other. Voiced speech is predominant in spoken vowels. On the other hand, unvoiced speech can be seen as a random, white noise produced by air turbulence in the vocal tract. Examples of unvoiced speech include the sound of words starting with letters such as “s”, "th", “f’, “t”, “p”, etc., basically the sound of all consonants.

Speech can be seen as a combination of a slowly varying AM signal and a slowly varying FM (Frequency Modulation) signal[1], the FM carrier being a characteristic of the speaker. Now, if we are not concerned about speaker recognition, we can move a higher carrier frequency to a lower one, as long as the speech information conveyed by the AM and FM effect is fairly preserved. This is done directly in the spectrum of the original signal, which is the subject of the next section.

IV. Frequency Mapping

In this section, we discuss the method used to alter the speech spectrum in such a way that the higher end of the spectrum is reallocated to a much lower range. The degree of alteration is really a factor to be determined experimentally, so we will not discuss the exact value here.

1. System Block Diagram

The block diagram of the entire system is shown in Fig. 4. The process involves digitizing the signal, taking an N-point FFT, performing a frequency mapping by insertion of zeros, and finally taking an inverse M-point FFT, where M is larger than N by a factor which depends on the number of zeros inserted.

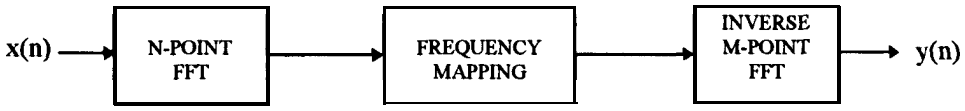

Figure 4. Block Diagram of the Spectrum Alteration System.

2. Spectrum Compression

The principle of the frequency alteration method consists of maintaining the sampling frequency fs constant and inserting a certain number of complex zeros in the spectrum of the sampled signal x(n). It is imperative to keep fs constant, otherwise the desired frequency compression would not take place. To illustrate, go back to Fig. 3 which shows the spectrum obtained with a 20-point FFT. The lowest frequency present in the signal is the component at $f_L = 100$Hz, whereas the highest frequency is $f_H = 900$Hz. Assume we want to shift f_H down to $f_h = 500$Hz, for example, and keep f_L unchanged. We also want to keep the same number of frequency components between f_L and f_H, therefore resulting in a compressed spectrum. In this example, there are only 9 spectral lines defining the entire signal. In order to make this method work in a more general sense, the following steps apply:

1) Determine the number of intervals between f_L and f_H :
   $N_{LH} = (f_H - f_L)N/f_S = (900\text{-}100)20/2000 = 8$
2) Determine the frequency resolution after compression:
   $\Delta f = (f_h - f_L)/N_{LH} = (500\text{-}100)/8 = 50$Hz
3) The new equivalent time record is:
   $t_r = 1/\Delta f = 1/50 = 20$ms
4) The new number of points M between O and fs in the compressed spectrum is:
   $M = f_S/\Delta f = 2000/50 = 40$ points
5) Determine the number of zeros to be inserted between O and f_L in the original spectrum:

a) Number of intervals before compression:

$f_L/\Delta F = 100/100 = 1$

b) Number of intervals after compression:

$f_L/\Delta f = 100/50 = 2$; (Note that AF=1OO, but $\Delta f=50$)

c) Number of zeros to be inserted:

$N_{Z1} = (f_L/\Delta f) - (f_L/\Delta F) = 2\text{-}1 = 1$ zero

6) Determine the number of zeros to be inserted between f_H and fs - f_H in the original spectrum:

a) Number of intervals before compression:

$(fs - 2f_H)/\Delta F = (2000\text{-}2x900)/100 = 2$

b) Number of intervals after compression:

$(f_S - 2f_h)/\Delta f = (2000\text{-}2x500)/50 = 20$

c) Number of zeros to be inserted:

$N_{Z2} = (fs - 2f_h)/\Delta f - (fs - 2f_H)/\Delta F = 20 - 2 = 18$ zeros

7) Take the inverse M-point FFT of the augmented spectrum (Here, M = 40 points).

V. MATLAB Simulation

Using MATLAB, it is easy to manipulate the spectrum of Fig. 3 by adding the complex zeros at appropriate places. It is important to add the zeros to the complex spectrum which is the direct result of the FFT. In contrast, before plotting the spectral lines of Fig. 3, the absolute value (magnitude) of the complex-valued spectrum was taken by specifying the MATLAB command *z=abs(fft(x,20))*, whereas the command *y=fft(x,20)* leaves the spectrum in its complex form. Using the MATLAB vector notations, one zero was added between O and 100Hz and 18 zeros were added between 900Hz and 1100Hz to this complex spectrum. The new compressed spectrum is now the frequency representation of a signal whose highest frequency is 500Hz, and whose lowest frequency remains unchanged at 100Hz. The same technique could be applied to shift this lower component as well.

Figure 5 depicts the magnitude of this new compressed spectrum, noting that the sampling frequency was deliberately kept constant. Figure 6 shows the corresponding signal in the time-domain as a result of taking the inverse 40-point FFT of the augmented complex spectrum. Note that since the frequency resolution in the new spectrum is smaller, the equivalent time record of the time-domain signal is increased from 10ms to 20ms for this particular example, since the time record is inversely proportional to the frequency resolution. In a real application, this is equivalent to listening to a recorded signal at a lower speed. Compare Fig. 1 and Fig. 6 and observe that in Fig. 6 the general shape of the original modulation is fairly preserved. As stated earlier, the factors which control the amount of distortion that could result when applied to speech signals is a subject of experimentation.

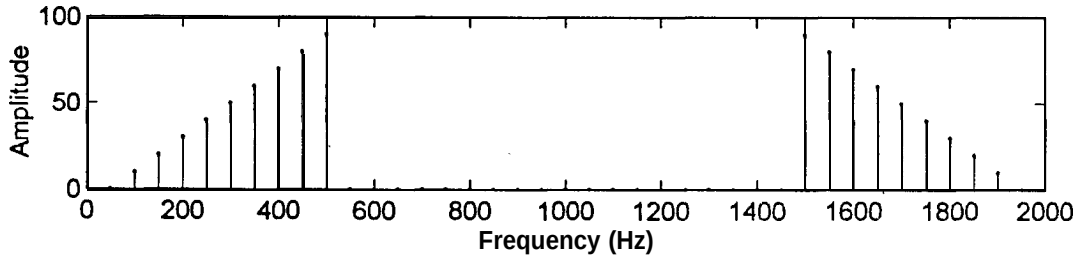

Figure 5. Spectrum of the Augmented Signal.

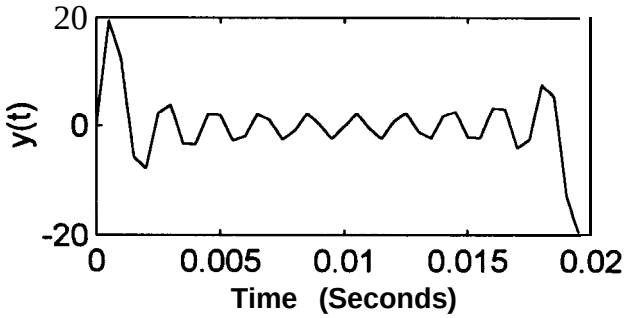

Figure 6. Equivalent Time-Domain Signal of the Augmented Spectrum.

VI. Conclusion

We have demonstrated a method by which a given frequency spectrum could be altered by inserting complex zeros in the frequency domain. The end result is a compressed spectrum, where the higher frequencies are reallocated to a much lower area, while the number of spectral lines and their magnitudes are kept unchanged. The corresponding time signal, when working with speech, exhibits a lower carrier frequency in the FM part of the modulation, since the highest frequency present in the signal is also lower. Also note that this method does not represent conventional demodulation, since the amount of frequency shift is not constant for each spectral line. Here, each frequency component undergoes a down-shift which depends on its position on the frequency axis. This algorithm can be easily implemented on a DSP hardware platform like the ADSP-2101 (or the ADSP-2181, which has an instruction cycle of 30ns). The hardware issues and other related parameters will be discussed in a subsequent paper.

VII. References

[1] Leonard M. Weinstein, "The Voice Spectrum Would be Mapped to Frequencies where the Ear Remains Sensitive", NASA Tech Briefs, p. 45, December 1994.

[2] Doug Hall, "Analysis-Synthesis Methods Lower Encoder Rates to Unheard of Levels", Personal Engineering & Instrumentation News, Vol. 12, No 7, July 1995.

[3] Robert F. Kubichek, "Using Matlab in a Speech Signal Processing Class", Computers in Education Journal, Vol. V, No. 3, pp. 2-5, July-September 1995.

[4] Hewlett Packard, "The Fundamentals of Signal Analysis", Application Note 243.

[5] Vinay K. Ingle, John G. Proakis, Digital Signal Processing Laboratory Using the ADSP-2101 Microcomputer, Prentice Hall, 1991.

[6] L.R. Rabiner, and R.W. Schafer, Digital Processing of Speech Signals, Prentice Hall, 1978.

[7] J.R. Deller, J.G. Proakis, and J.H.L. Hansen, Discrete-Time Processing of Speech Signals, Macmillan, Inc., 1993.

VIII. Biographies

ESSAID BOUKTACHE is an Assistant Professor in the Department of Electrical Engineering Technology at Purdue University Calumet, Hammond, Indiana. Dr. Bouktache received his MS and Ph.D. in Electrical Engineering from The Ohio State University in 1980 and 1985, respectively. His research and teaching interests include Digital Signal Processing, Computer Networks, and Digital Communications. Professor Bouktache has been with Purdue since 1992 and he is a member of IEEE and ASEE.

CHANDRA R. SEKHAR is the Head of the Department of Electrical Engineering Technology at Purdue University Calumet. Professor Sehkar earned a Bachelor's Degree in Chemistry from the University of Madras (India), a diploma in instrumentation from Madras Institute of Technology and a Master's Degree in Electrical Engineering from the University of Pennsylvania. Professor Sekhar's primary teaching and research focus is in the areas of biomedical and process control instrumentation and clinical engineering.

OMER FAROOK is a member of the faculty of the Electrical Engineering Technology Department at Purdue University Calumet. Professor Farook received the Diploma of Licentiate in Mechanical Engineering in 1970. He further received a B. S.E.E. and an M. S.E.E. in 1978 and 1983 respectively from the Illinois Institute of Technology. Professor's Farook current interests are in the areas of microprocessor-based systems, computer interfacing, data communication, and the design of software/firmware using C, C++, and assembly language.