VOCAL TRACT MODULATION OF INSTRUMENTAL
SOUNDS BY DIGITAL FILTERING

by

Tracy Lind Petersen
Copyright © 1976 Tracy L. Petersen

Abstract

This paper presents a method for the digital simulation
of a musical source being articulated by a model of the human
vocal tract. Sounds so processed take on the quality of human
speech. A time varying digital filter is formulated from a
digital recording of speech input. This time varying filter
is then applied to a musical source. Suggestions for the use
of this tool in musical composition are presented, and it is
shown that a wide variety of results are possible.

Introduction

Human speech may be analyzed and synthesized with a device
known as a vocoder. A thorough description of the development
of this device in its various forms is given in the book by
Flanagan [1]. The work presented here describes modifications
of basic vocoder principles for the purpose of molding sounds
in a creative way, and while many different vocoder models are
possible, the specific model described here will be the linear
prediction vocoder [2]. Figure 1 shows a generalized description
of elementary vocoder functions. The speech to be analyzed is
input to the analyzer which produces a set of filter parameters,
a gain term, and pitch information. The filter parameters
represent the vocal tract, the gain term is an energy scale
factor, and for voiced sounds (such as vowels) the pitch
information represents the approximate frequency at which the
vocal chords are vibrating. For unvoiced sounds (such as
fricatives) the pitch information is usually supplied as white
noise. This analysis is performed at discrete intervals in
time—the typical update interval being about 10 milliseconds.

It has been shown by Wakita [3] that a one-to-one
 correspondence exists between the linear prediction filter
parameters and the cross-sectional areas of an acoustic tube
model of the vocal tract, and in fact that the varying cross-
sectional areas of the tube model can be mathematically derived
Figure 1. A generalized vocoder model.

Figure 2. An interesting modification of figure 1.
from the filter parameters. Thus when these parameters are supplied to the synthesizer the synthesis filter models the frequency response of the vocal tract. The pitch information, supplied either as a pulse train spaced at the pitch rate or as white noise, is then filtered in the same sense as if the pitch signal were being played out through an acoustic tube model of the vocal tract. Since a new filter configuration is modeled approximately every 10 milliseconds this time varying filter produces a signal which approximately reconstructs the input speech.

Figure 2 suggests an interesting modification to the generalized vocoder model of figure 1. In figure 2 the pitch information has been replaced by a signal unrelated to the input speech and is shown in figure 2 as musical source. In examining this modification it will be found that since pitch information is filtered by the synthesizer as a time varying model of the vocal tract, any other signal supplied to the pitch input channel will also be filtered as though it were being played out through the vocal tract model. Thus a musical instrument may be processed to articulate human speech. Some signals, according to their spectral characteristics are more clearly articulated than others, and more will be said about techniques for improving speech quality from signals not well suited for speech production.

Implementation

Implicit in the implementation of a digital vocoder is the assumption that all signals operated on are sampled signals existing in a digital form. Thus it becomes convenient to describe the associated frequency functions in terms of Z-transform notation. The reader should be informed that a knowledge of Z-transforms is not a prerequisite to implementation as FORTRAN subroutines for performing both the analysis and the synthesis are conveniently supplied in the text by Markel and Gray [2]. The analysis process begins by buffering in a short section (about 25 milliseconds) of the sampled input speech \(x_n\). This input data of \(N\) points is then multiplied by an appropriate
Figure 3. The more rapidly varying function is the short time frequency spectrum of a section of input speech. The smoother curve superimposed shows the linear prediction spectral match determined by a 14 stage filter.
weighting function such as a Hamming window, and the short term autocorrelation function \( R_{|i-j|} \) is computed from the \( N \) weighted speech samples \( x_n \) as

\[
R_{|i-j|} = \sum_{n=0}^{N-1} x_n \cdot x_{n+i-j}.
\]

Now \( M \) simultaneous linear equations defined as

\[
\sum_{i=0}^{M} a_i \cdot R_{|i-j|} = 0, \quad j=1,2,\ldots,M \quad (\text{Eq. 1})
\]

are solved for the coefficients \( a_i \), and it can then be shown [2] that the inverse of the desired filter has been determined through a process of least squares minimization as

\[
A_M(z) = \sum_{i=0}^{M} a_i \cdot z^{-i}
\]

where \( M \) specifies the number of filter stages, \( z = \exp(j\omega T) \), \( j = \sqrt{-1} \), \( \omega \) is the radian frequency, and \( T \) is the sampling period. A reasonable value for \( M \) is the sampling frequency/1000 + 4, so for example a sampling frequency of 10 kHz would imply a value for \( M \) of 14.

Figure 3 shows two frequency functions superimposed. The rapidly varying function is the short time spectrum of a section of input speech, and the smooth curve is its linear prediction spectral match determined by a 14 stage filter. This smoothed spectral match is the magnitude frequency function \( |1/A_M(z)| \) and represents the magnitude frequency response of the vocal tract. Filter parameters and a gain term are produced by the analysis when equation 1 is solved recursively [2],[3]. If a new polynomial in \( z \) is defined as

\[
B_m(z) = A_m(1/z) \cdot z^{-m-1}
\]

then it can also be shown that the following relations hold,

\[
A_m(z) = A_{m+1}(z) - B_m(z) \cdot u_{m+1} \quad (\text{Eq. 2.1})
\]

\[
B_{m+1}(z) = z^{-1} \left[ B_m(z) + A_m(z) \cdot u_{m+1} \right] \quad (\text{Eq. 2.2})
\]

where the \( u_m \) are defined as reflection coefficients in the literature and are in fact the filter parameters in figures 1 and 2. Equations 2.1 and 2.2 define the synthesis lattice
Figure 4. mth stage of two-multiply lattice synthesis filter.

Figure 5. M stages form the complete synthesis filter.
filter structure shown in figures 4 and 5. Figure 4 shows the mth stage of the lattice while figure 5 shows the lattice cascade \( G_N(z) = 1/A_H(z) \). Input to the lattice is shown as \( X(z) \) and the desired output is \( X(z)/A_M(z) \). The filtering operation may be followed by observing that

\[
X(z) = A_M(z) \cdot X(z)/A_M(z)
\]

and that in the final stage

\[
A_0(z) \cdot X(z)/A_M(z) = 1 \cdot X(z)/A_M(z) = z \cdot B_0(z) \cdot X(z)/A_M(z).
\]

If the modification in figure 2 is implemented then \( X(z) \) in the above equations is simply the Z-transform representation of a section of the musical source signal to be filtered. Once a set of reflection coefficients have been determined from a section of input speech then samples from the musical source equal in length to the analysis update interval are input to the lattice filter. The process is then repeated by moving the analysis window down the time line by the update interval, deriving a new set of reflection coefficients and filtering the next section of musical source. Each output sample from the filter is multiplied by the gain term from the current analysis.

It has been shown that the filter parameters and gain term are recomputed at regular intervals which means that they all change as a function of time in a step-like fashion. To mask abrupt changes that may occur in the filtering process it is desirable to interpolate as a function of time, either linearly or along some smooth curve, both the filter parameters and the gain term between each initial and target set. This means that a new set of interpolated parameters are supplied to the synthesizer for each sample of the signal to be filtered.

Preprocessing The Musical Source

The dynamic filtering which has been described will have a more pronounced effect on the signal being processed if that signal tends to be spectrally flat. This means that the energy of the signal in all frequency bands tends to be similar. The sound of a large choir for example tends to have more spectral flatness than the sound of a French horn. It is possible to
render a signal spectrally flat by inverse filtering and this type of preprocessing applied to a musical source will greatly enhance the intelligibility of speech produced from it. Inverse filtering by itself can have drastic effects on a signal, and will cause a violin to sound more like a bagpipe. Nevertheless, if a sound does not process well, preprocessing by inverse filtering is an alternative which can produce interesting results.

Inverse filtering simply involves applying the described method of analysis to a signal and deriving the filter parameters or reflection coefficients. This signal is then filtered with its own filter parameters only its path through the lattice network is reversed. This configuration is derived simply by rewriting equation 2.1 as

\[ A_{m+1}(z) = A_m(z) + B_m(z) \cdot u_{m+1}. \]

Now the Z-transform of the signal to be filtered, \( X(z) \), enters the lattice as \( X(z) \cdot 1 = X(z) \cdot A_0(z) \) and emerges as \( X(z) \cdot A_M(z) \).

Variations On a Theme

Since this type of processing is directed to some creative end it is fair and even desirable to experiment with some variations. It was stated that the smooth curve in figure 3 represents the magnitude frequency response of the linear prediction filter \( 1/A_m(z) \). As the number of stages \( M \) in the filter increase, its magnitude frequency response will approach the actual magnitude frequency response of the input analysis signal. As the smooth curve becomes more and more like the rapidly varying one the filter takes on more of the actual pitch information from the analysis input signal. A reasonable upper bound for \( M \) on a signal sampled at 10 kHz is \( M=64 \).

The analysis update interval may be lengthened or shortened. As it becomes smaller the vocal tract model is articulated with increased accuracy. As the update interval is lengthened the articulation will gradually become blurred.

The gain term of the filter may be set constant. This greatly reduces intelligibility of synthetic speech but the filtered signal retains very definite speech-like qualities.
Inverse filtering a signal as previously described, with different values of M for the M stages of the inverse filter, can produce interesting results.

Finally, of course there is no reason why the signal input for analysis could not be music rather than speech. It is possible to simulate a harp being played through a clarinet....

Conclusion

The methods described provide a glimpse into the potential power of digital signal processing for creating new sounds. The theme throughout has of course been the use of a time varying digital filter and there is a great deal of room for experimentation and research in applying these techniques to the very exciting problems facing the composer of computer music. In implementing these ideas the composer will find it useful to design his algorithm so that analysis parameters corresponding to a given section of speech are stored on disk. This way any number of signals may be processed without having to reperform that particular analysis. Once a palette of sounds has been generated and stored the composer has the option of composing from that palette.

REFERENCES

About The Author:

Tracy L. Petersen holds degrees in music, mathematics, and computer science. He has been a professional violinist as well as a computer scientist, and his compositions have been performed on concerts in the United States and on European radio. He is presently involved in the University of Utah Music Department's computer music project.