EFFICIENT ANALYSIS/SYNTHESIS OF PERCUSSION MUSICAL INSTRUMENT SOUNDS USING AN ALL-POLE MODEL

Michael W. Macon [1,2] *Alan McCree* [1] *Wai-Ming Lai* [1] *Vishu Viswanathan* [1]

[1]DSP Research and Development Center, Texas Instruments, Dallas, TX 75265-5474
[2]Dept. of ECE, Oregon Graduate Institute, Portland OR 97291-1000

ABSTRACT

It is well-known that an impulse-excited, all-pole filter is capable of representing many physical phenomena, including the oscillatory modes of percussion musical instruments like woodblocks, xylophones, or chimes. In contrast to the more common application of all-pole models to speech, however, practical problems arise in music synthesis due to the location of poles very close to the unit circle. The objective of this work was to develop algorithms to find excitation and filter parameters for synthesis of percussion instrument sounds using only an inexpensive all-pole filter chip (TI TSP50C1x). The paper describes analysis methods for dealing with pole locations near the unit circle, as well as a general method for modeling the transient attack characteristics of a particular sound while independently controlling the amplitudes of each oscillatory mode.

1 INTRODUCTION

It is well-known that the impulse response of resonant bodies like percussion instruments can be modeled as a sum of damped sinusoids whose frequencies correspond to physical dimensions of the instrument [1]. This makes all-pole digital filters attractive for computationally-efficient synthesis of such sounds. In addition, the popularity of the linear predictive coding (LPC) model for speech [2] has driven manufacturers to create efficient and inexpensive hardware implementations of all-pole filters, like the TI TSP50C1x speech synthesis chip [3]. Synthesizing speech and musical instrument sounds on the same hardware is attractive for many consumer applications.

In recent work [4], a multi-channel all-pole model was applied to synthesis of piano sounds; slightly more general models can be shown to model a wide variety of instruments [5]. Other work on percussion instrument analysis includes [6], where *additive synthesis* of individual partials was used to model heavily damped percussion instruments, and [7], where parallel resonance models were used.

This paper addresses two problems in analysis of percussion instrument sounds. The first, described in Section 2, involves estimating the location of a set of poles near the unit circle to represent the instrument resonances. The second, described in Section 3, involves finding an impulsive excitation that models the transient attack of a particular note, while *simultaneously* exciting each resonant mode to its proper amplitude to maintain timbral characteristics of the tone.

2 ALL-POLE MODEL ANALYSIS

2.1 Mode isolation

The first step in the analysis algorithm is to isolate individual resonant components in the signal and find the frequency and bandwidth of a pole that models each resonance. The problem of finding damped complex exponentials in a signal is well-studied in statistical signal processing [8]. However, many methods are difficult to apply in this case because of long data records and proximity of the poles to the unit circle. In our algorithm, poles are isolated by performing "peak-picking" on the Fourier magnitude spectrum to select the most prominent components in the signal and then modulating each component to DC.

The number of partials that can be synthesized is severely limited by the hardware (6 pole pairs for the TSP50C1x), so only the most significant modes can be modeled. The following iterative algorithm was devised to make a reasonable selection. First, the signal spectrum $X(e^{j\omega})$ is computed via an FFT, and a smooth spectral envelope $X_{cep}(e^{j\omega})$ is computed by cepstral liftering [9]. The frequency ω corresponding to the largest component in $|X(e^{j\omega})|/|X_{cep}(e^{j\omega})|$ is chosen as a peak location, after which the spectrum is weighted in the neighborhood of ω to make further selection of components in this region less likely. The selection is repeated until 6 modes have been selected.

The weighting algorithm attempts to compromise between choosing the largest amplitude components and choosing components that are maximally spread in frequency. This is motivated both by masking effects in the ear and by implementation issues: roundoff noise problems in the fixed-point hardware implementation tend to be much less severe when poles are spaced further from each other in frequency.

For each resonant frequency ω_i chosen by the peak-picking algorithm, the signal $x_i[n]$ corresponding to the single mode is separated from the rest of the signal $x[n]$ by computing

$$x_i[n] = h[n] * (x[n] + j\hat{x}[n])e^{j\omega_i n} \qquad (1)$$

where "$*$" represents convolution, $\hat{x}[n]$ is the Hilbert transform of $x[n]$, and $h[n]$ is the impulse response of a lowpass filter. The cutoff frequency of $h[n]$ is set so as to attenuate other modes but not influence each mode's decay envelope (a cutoff of 100-200 Hz was sufficient). Given that extraneous frequency components have been adequately filtered out, the complex demod-

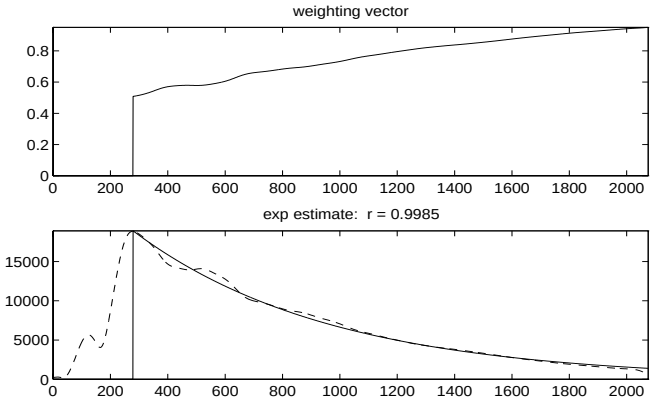

Figure 1. Pole radius estimation. top: weighting vector; bottom: mode envelope (dashed) and exponential fit

ulated partial $x_i[n]$ will have a smooth amplitude envelope $|x_i[n]|$ that can be used to estimate the pole radius (i.e., bandwidth).

2.2 Pole radius estimation

The pole radius is estimated by finding a correlation coefficient for each component amplitude envelope $x_{env}[n] = |x_i[n]|$. Empirically, it was found that using a weighting function to emphasize the less variable "tail" of the exponential decay produces better results. The weighting function $w[n]$ is computed as

$$w[n] = \frac{1}{1 + \tilde{x}_{env}[n]}$$

where $\tilde{x}_{env}[n]$ is a smoothed version of $x_{env}[n]$ normalized to the range $[0, 1]$. The correlation coefficient is then computed using a weighted least squares minimization. Figure 1 shows the weighting function $w[n]$, the envelope $x_{env}[n]$, and the function

$$v[n] = a_0 r^{n-n_0} \tag{2}$$

where n_0 is the time offset from the beginning of the signal to the maximum of the envelope, and a_0 is an initial amplitude. This value a_0 is found via a simple least-squares minimization of the error between the functions $a_0 r^{n-n_0}$ and $x_{env}[n]$.

3 EXCITATION MODELING

Maintaining the correct amplitudes of each mode relative to the others is essential to maintaining the correct timbre (tone color) for the instrument. However, the initial amplitudes a_0 of each mode of oscillation cannot be controlled by the pole locations—they are a function of the *input* to the system. Because the poles are located very close to the unit circle, using a single impulse $\delta[n]$ to excite the filter can produce mode amplitudes that are radically different from the original sound.

This section describes two methods for finding a desirable excitation for the all-pole filter. The *initial condition method* finds the minimum-length sequence to excite the modes properly; the *projection method* extends this to find an excitation sequence that can also model the transient attack characteristics of the instrument.

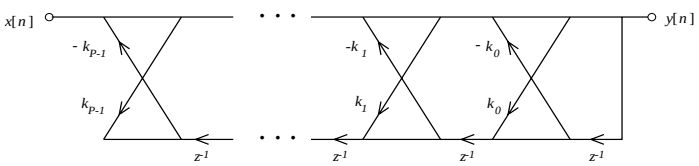

Figure 2. All-pole lattice filter

3.1 Initial condition method

This approach specifies a set of initial conditions for the delay elements of the filter, such that the modes are properly excited when the filter is run from this initial state. This is analogous to the physics of many instruments; e.g., pulling a guitar string to an initial state and releasing it excites certain modes more than others, depending on where the string is plucked along the neck of the guitar [1].

In the hardware, a lattice filter structure is used [9], as shown in Figure 2. To find initial conditions for the filter, it is advantageous to write the lattice filter as a state-space system:

$$\begin{aligned} \mathbf{x}_n &= \mathbf{A}\mathbf{x}_{n-1} + \mathbf{B}u[n] \\ y[n] &= \mathbf{C}\mathbf{x}_n \end{aligned}$$

where $u[n]$ is the filter input and $y[n]$ is the filter output. P is the number of poles in the system, and $\mathbf{x}_n$ is a $P \times 1$ state vector containing the values at time n in the filter delay registers across the bottom branch of Figure 2. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ describe the lattice filter and depend only on the filter tap weights $\{k_i\}$.

The problem at hand is to find an initial state vector $\mathbf{x}_{-1}$ such that each mode of oscillation will have the proper amplitude in the output $y[n]$ for $n \geq 0$. The modes of the filter can be isolated from each other by performing an eigendecomposition of the matrix $\mathbf{A}$,

$$\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$$

where $\mathbf{S}$ is a matrix with the eigenvectors of $\mathbf{A}$ in its columns and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. The matrix $\mathbf{S}$ will be invertible if and only if the filter has nonrepeated poles [10], and this is guaranteed by the peak-picking algorithm. The eigenvectors of $\mathbf{A}$ correspond to the modes of the system, and the eigenvalues correspond to the rate of decay of each mode.

Since the eigenvectors are linearly independent, we can adjust the amplitudes and phases of the modes independently in the initial state by making $\mathbf{x}_{-1}$ a weighted linear combination of the eigenvectors $\{\mathbf{v}_k\}$ [11],

$$\mathbf{x}_{-1} = \sum_{k=0}^{P-1} g_k \mathbf{v}_k \tag{3}$$

where

$$g_k = \frac{a_k e^{j\phi_k}}{\mathbf{C}\mathbf{v}_k}$$

and a_k and ϕ_k are the desired amplitude and phase for the kth mode of the system, as found in Equation (2). (The phases ϕ_k are somewhat arbitrary.) This gives a set of initial delay register values that will excite the modes to the desired amplitudes.

An equivalent method uses an input $u[n]$ of length P, where P is the number of poles. This method relies on constructing a *controllability matrix* [11],

$$\mathbf{E} = \left[\begin{array}{cccccc} \mathbf{A}^{P-1}\mathbf{B} & \cdots & \mathbf{A}^2\mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{B} \end{array} \right] \qquad (4)$$

and finding the input $\mathbf{u} = [u[0], \ldots, u[P-1]]^T$ that drives $\mathbf{x}_n$ to the desired state at time P. The solution for the desired input $\mathbf{u}$ is then $\mathbf{u} = \mathbf{E}^{-1}\mathbf{x}_P$.

3.2 Projection method

The impact of a mallet, clapper, or other object striking an instrument produces a brief transient signal that does not fit a low-order all-pole model. The realism of a synthesized note can be enhanced by using a transient signal of a few hundred samples as the excitation, found by inverse filtering the input signal with the analyzed pole frequencies and radii and truncating the residual. When this excitation is used as an input to the lattice filter, however, there is again no guarantee that the modes of the system will be excited to their proper relative amplitudes. The method described here overcomes this problem.

Given a length N excitation signal $u_D[n]$, the target state at time N must be specified to insure that the resulting oscillatory modes will have the proper amplitudes and phases. Since we seek an excitation that is as close as possible to the inverse filter residual $u_D[n]$, it is advantageous to set the phases of each mode at time N to be as close as possible to the *actual* phases that result from using $u_D[n]$ as the system input.

The desired amplitude at time N is easily found by $a_N = a_0 r^{N-n_0}$. The phases at time N are found as follows. The approximate frequencies of the filter output are known from the peak-picking analysis, and the decay constants of the modes are generally large enough that the sinusoid amplitudes can be considered almost constant over a small interval. The filter response to the input $u_D[n]$ just after the excitation is "turned off" can be approximated by

$$y_D[n] \approx \sum_{k=1}^{P/2} c_k e^{j\omega_k n} + c_k^* e^{-j\omega_k n} \qquad (5)$$

over some interval $N+1 \le n \le N+M$. An optimal solution can be found by performing a least squares fit of coefficients $\{c_k\}$ to the data $\{y[N+1], \ldots, y[N+M]\}$. The desired phases $\{\phi_k\}$ can then be found from the phase angles of the complex coefficients $\{c_k\}$. Finally, given the target amplitudes a_k and phases ϕ_k at time N, the target state $\mathbf{x}_N$ can be found via the sum of eigenvectors in Equation (3).

Now, given the target state $\mathbf{x}_N$ and a desired input sequence $u_D[n]$ ($\mathbf{u}^D$ in vector notation), the task is to find an input $\mathbf{u}$ that lies as close as possible to $\mathbf{u}^D$ and excites the modes to their proper amplitudes. Borrowing the notation for the controllability matrix of Equation (4), the problem can be phrased as follows:

> *Given $u_D[n]$ and a target state $\mathbf{x}_N$, find an input $u[n]$ such that*
>
> $$\mathbf{x}_N = \mathbf{E}\mathbf{u} \qquad (6)$$
>
> *is satisfied and the error*
>
> $$\varepsilon = \sum_{n=0}^{N} (u_D[n] - u[n])^2 \qquad (7)$$
>
> *is minimized over the range of all possible inputs $u[n]$.*

Since Equation (6) represents an underdetermined system of equations when the excitation length N is greater than the number of poles P, it has no unique solution. However, any solution of (6) must be of the form $\mathbf{u} = \mathbf{u}^+ + \mathbf{u}^{\mathcal{N}}$, where $\mathbf{u}^+$ is in the row space of $\mathbf{E}$ and $\mathbf{u}^{\mathcal{N}}$ is in the nullspace of $\mathbf{E}$, denoted $\mathcal{N}(\mathbf{E})$. The solution $\mathbf{u}^+$ is unique; thus the problem above can be solved by first finding $\mathbf{u}^+$, then finding a vector $\mathbf{u}^{\mathcal{N}} \in \mathcal{N}(\mathbf{E})$ that lies as close as possible to the difference vector $\mathbf{u}^D - \mathbf{u}^+$.

The row space component can be found by computing the *pseudoinverse* of $\mathbf{E}$

$$\mathbf{E}^+ = \mathbf{Q}_2 \mathbf{\Sigma}^+ \mathbf{Q}_1^T \qquad (8)$$

where $\mathbf{Q}_2$, $\mathbf{\Sigma}^+$, $\mathbf{Q}_1^T$ are found by performing a singular value decomposition (SVD) of the matrix $\mathbf{E}$ [10]. The row space solution is then

$$\mathbf{u}^+ = \mathbf{E}^+ \mathbf{x}_N \qquad (9)$$

The vector $\mathbf{u}^+$ is the "minimum energy" solution to Equation (6).

To find the nullspace component $\mathbf{u}^{\mathcal{N}}$, the difference vector $\mathbf{u}^D - \mathbf{u}^+$ must be projected onto the nullspace of $\mathbf{E}$. The matrix $\mathbf{Q}_2^T$ from the SVD contains a basis for the nullspace of $\mathbf{E}$ in its last $N-r$ columns. A new matrix $\mathbf{V}$ can be created by putting these nullspace basis vectors into its columns. Then, the projection of the difference vector onto the nullspace can be written

$$\mathbf{u}^{\mathcal{N}} = \mathbf{V}\mathbf{V}^T(\mathbf{u}^D - \mathbf{u}^+) \qquad (10)$$

Finally, these two components can be combined into the final solution $\mathbf{u}_{opt} = \mathbf{u}^+ + \mathbf{u}^{\mathcal{N}}$, which can be shown to satisfy (6) and minimize the error in (7). An example of such a decomposition for a xylophone note is shown in Figure 3.

It can be seen that the nullspace input $\mathbf{u}^{\mathcal{N}}$ looks very much like the desired input $\mathbf{u}^D$, but results in a filter output that is *identically zero* after it is "turned off." The input $\mathbf{u}^+$ is rather small in comparison, yet it is responsible for *all* of the nonzero filter response after the input is turned off.

4 IMPLEMENTATION

A fixed-point simulation was developed to test the effects of roundoff noise on the algorithm, and this was used to scale the excitation to avoid register overflow. To improve accuracy in the fixed-point synthesis implementation, the reflection coefficients are quantized to their 12 bit representation *before* inverse filtering and computing the excitation projection. It is important to note that the inaccuracies of the pole location quantization and inverse filtering are compensated for by the projection technique – the projection of the residual *guarantees* that the modes will be properly excited to recreate the timbre of the original sound.

In experiments with several instrument sound samples, it was found that very good results are obtained for sounds containing 6 or fewer significant modes (the upper limit of the TSP50C1x hardware capability), and having pole radii smaller than $r = .999$. Reasonable choices were usually made by the peak-picking algorithm, but a few required manual selection of significant modes.

For some sounds, especially xylophone and woodblock, using a 100–200 sample excitation sequence (at 8 kHz sampling rate) made a drastic difference in the realism of the synthesized note. For instruments excited by a nearly ideal impulse (e.g., metallic instruments like chimes or bells) the initial condition method was sufficient to model the transient attack portion of the signal.

Systems with several poles spaced close together and having radii very close to 1 (e.g., low frequency church bells with a decay of several seconds) tended to have difficulties with roundoff noise and limit cycles in the fixed-point implementation. Although the conditions under which these effects occur could be more thoroughly analyzed, the constraints of the hardware make them hard to address.

The analysis computation associated with the projection operation in Equation (10) becomes significant as N (the number of samples in the excitation sequence) becomes large. Values of $N < 500$ were not problematic on a moderate-power workstation, and this is sufficient under the severe memory constraints of the target application. (The complexity of the end-user synthesis is fixed.)

In summary, the algorithm described in this paper provides a means for independent control of factors influencing the timbre of synthesized percussion sounds. By its implementation in readily-available hardware, the method has been shown to make efficient, high-quality synthesis of such sounds possible for many consumer applications.

REFERENCES

[1] A. H. Benade, *Fundamentals of Musical Acoustics*. Dover Publications, Inc., 1990.

[2] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, pp. 561–580, April 1975.

[3] Texas Instruments Incorporated, Dallas, TX, *TSP50C1x Speech Synthesizer Design Manual*, 1990.

[4] J. Laroche and J. L. Meillier, "Multichannel excitation/filter modeling of percussive sounds with application to the piano," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 329–344, April 1994.

[5] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, pp. 74–91, Winter 1992.

[6] J. Laroche, "A new analysis/synthesis system of musical signals using Prony's method: Application to heavily damped percussive sounds.," in *Proc. ICASSP*, pp. 2053–2056, IEEE, April 1989.

[7] P. Depalle, *Analyse, Modélisation et Synthèse des Sons Fondées sur le Modèle Source/Filtre*. PhD thesis, Université du Maine, LeMans, France, 1991.

[8] S. M. Kay, *Modern Spectral Estimation*. Engelwood Cliffs, NJ: Prentice Hall, 1988.

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

[10] W. G. Strang, *Linear Algebra and Its Applications*. Academic Press, 1988.

[11] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1980.

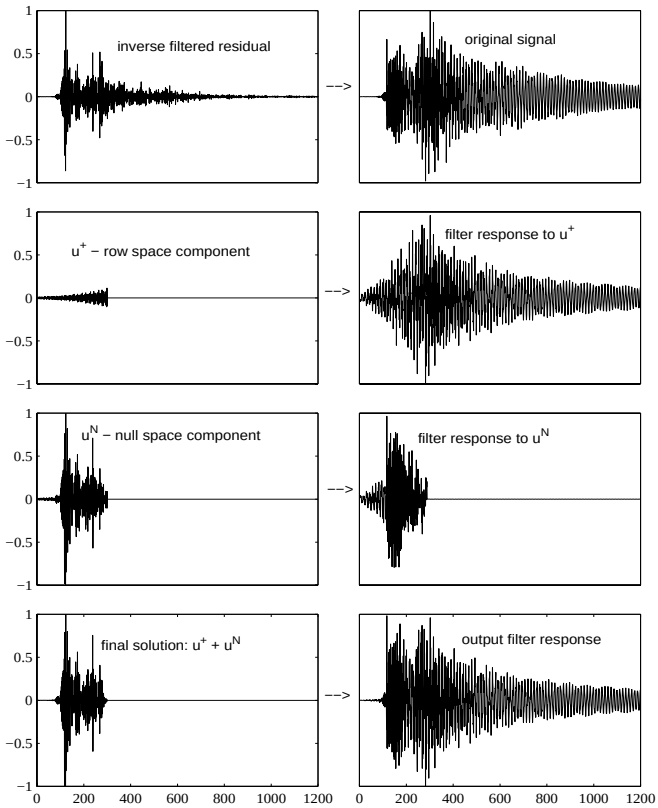

Figure 3. Plots showing various elements of excitation decomposition for a xylophone note. Left hand side are excitation signals; right hand side are filter responses to each excitation. The decomposition separates the excitation into a row space component that controls mode amplitudes and a nullspace component that does not excite the filter resonances.