MUSIC SIGNAL SYNTHESIS USING SINUSOID MODELS AND SLIDING-WINDOW ESPRIT

Anders Gunnarsson, Irene Y.H. Gu

Dept. of Signals and Systems, Chalmers Univ. of Technology, Gothenburg, 42196, Sweden

ABSTRACT

This paper proposes a music signal synthesis scheme that is based on sinusoid modeling and sliding-window ESPRIT. Despite widely used audio coding standards, effectively synthesizing music using sinusoid models, more suitable for harmonic rich music signals, remains an open issue. In the proposed scheme, music signals are modeled by a sum of damped sinusoids in noise. A sliding window ESPRIT algorithm is applied. A continuity constraint is then imposed for tracking the time trajectories of sinusoids in music and for removing spurious spectral peaks in order to adapt to the changing number of sinusoid contents in dynamic music. Simulations have been performed to several music signals with a range of complexities, including music recorded from banjo, flute and music with mixed instruments. The results from listening and spectrograms have strongly indicated that the proposed method is very robust for music synthesis with good quality.

1. INTRODUCTION

Audio synthesis and coding are one of the important issues in multimedia communications and other digital multimedia applications such as digital music synthesis. Current audio coding standards include MPEG-1 Layer III (or "MP3" specification), MPEG-2, and MPEG-4 high-efficiency advance audio coding (HE-AAC) standards for multi-channel sound via the transmission of compressed stereo/mono audio plus a low-rate side-information channel [1]. Stereo or multi-channel audio signal is either encoded by splitting signal into a filterbank and LPC, or using MDCT (Modified Discrete Cosine Transform) domain processing. MDCT converts time domain sampled audio waveforms into frequency-domain, where frequency components are allocated with different bits according to their audibility determined by the masking thresholds below which the sound components are non-audible.

There is also a growing demand for digital synthetic music that can be employed in parallel to music signals generated by music instrument. One of the special features for audio signals generated by music instruments such as guitar, flute and piano is that they primarily consist of sinusoids due to several fundamentals and their harmonics. Some music analysis and synthesis methods have been proposed, for example, methods based on physical modeling [2], on the wave scattering methods for solving partial differential equations [3]. Methods based on sinusoid model of music signals include using LPC spectra or STFT followed by peak extraction and partial tracking [4, 5]. Both LPC and STFT suffer from low frequency resolution for estimating sinusoids in the frequency-domain. Despite sinusoid models may give better characterization of harmonic rich music signals, effectively synthesizing music based on such a model remains a challenging issue for any real applications. In this paper we propose a sinusoid model-based music synthesis scheme, where music signals are modeled by a sum of damped sinusoids in noise. A sliding-window ESPRIT algorithm is used, followed by imposing a continuity constraint for tracking the true sinusoid components in the music signal and removing spurious peaks, in order to adapt to the dynamics of sinusoid contents in music. Only these tracked damped sinusoids are included in the music signal model and are subsequently used for re-synthesis. Our reported scheme is novel and robust, in the best of our knowledge, in terms of using ESPRIT for dynamic music synthesis under sinusoid models and of good synthetic music quality.

2. CHARACTERIZING MUSIC SIGNALS

2.1. Damped Sinusoids for Music Signal Modeling

Since most music signals generated by music instruments (e.g. piano, flute and guitar) consist of several fundamentals and harmonics, an exponentially damped sinusoid model is used for characterizing such type of music signals. Under the model a music signal $s(n)$ is described by

$$z(n) = \sum_{i=1}^{p} A_i e^{-\alpha_i n} \cos(2\pi\omega_i n + \phi_i) + w(n) \qquad (1)$$

where $w(n)$ is the model noise assuming to be zero-mean white. The unknowns in the model are the number (p) of sinusoids (or, the model order), the magnitude A_i, the frequency ω_i, the damping factor α_i and the initial phase ϕ_i for each sinusoid, $i = 1, 2, \cdots, p$. Since music signals modeled by (1) are nonstationary, a sliding-window ESPRIT method is then applied for estimating the time-varying model parameters.

2.2. Constraint: continuity of sinusoids in music signals

Apart from the damped sinusoid model assumption, another assumption imposed to the music signals is that the changes of sinusoids in time are continuous both in frequencies and in magnitudes. This assumption is due to the fact that the mechanical movement of music instrument or the change in sound production chamber cannot be happened instantly. This assumption is utilized in tracking the time trajectories of candidate sinusoids in music signals and in removing spurious spectral peaks due to inaccurate model order selection.

3. SYSTEM DESCRIPTION

Based on the above model, an analysis and synthesis system is proposed for music signals. The proposed system consists of the following processing: In the analysis phase, the music signal is first divided into overlapped blocks, each containing 20-30ms of data. An ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) is then applied to each block of music signals for estimating the parameters associated with the damped sinusoids, including magnitude, frequency, initial phase and damping factor of each sinusoid. The estimated *candidate* sinusoids are then tracked along the time direction using the continuity constraint so that spurious peaks are removed. Only the parameters of tracked sinusoids are used in the music signal model of that block. The sliding-window ESPRIT is applied to overlapped data blocks. In the synthesis phase, the estimated parameters of tracked sinusoids in each block are used to synthesize the music signal of that block based on the damped sinusoid model. Care is also taken for maintaining the power of music signal in each block to ensure the continuity of music signal through the neighboring blocks.

4. SLIDING-WINDOW ESPRIT AND TRACKING OF DAMPED SINUSOIDS

4.1. Sliding-Window ESPRIT

Since the music signal is nonstationary, the signal is first divided into small blocks such that each block of data is approximately stationary. For each block of data, an ESPRIT algorithm is then applied [6].

Let the samples $z(n)$, $n = 0, 1, \ldots$, in the music signal sequence be divided into blocks of fixed size L. The size of blocks is usually determined empirically such that the data within each block can be considered as approximately stationary. Let the overlap of adjacent blocks be K, $K < L$, and the m-th sample in the j-th block be $z^{(j)}(m)$, $m = 0, 1, \ldots, L-1$, $j = 1, 2, \ldots$. Then the data sample in the j-th block, $z^{(j)}(m)$, is related to the sample $z(n)$ in the original data sequence by $z^{(j)}(m) = z\left(m + (j-1)(L-K)\right)$. The index m in the blocked data $z^{(j)}(m)$ is hence related to the actual time index n in the original data sequence $z(n)$ by $n = m + (j-1)(L-K)$. For each block of data, the damped sinusoid parameters are then estimated by the LS-ESPRIT method described below.

4.2. LS-ESPRIT Method: parameter estimation

Since the frequency components of signal are exponentially damped sinusoids, LS-ESPRIT (Least Squares - ESPRIT) can be used to resolve closely-spaced frequency components of a signal with high resolution.

ESPRIT decomposes a signal into a sum of sinusoids using the signal subspace-based approach. Let us consider a signal $z(n)$ as being the sum of p exponentially damped sinusoids in additive noise $w(n)$. The number of sinusoids are assumed to be known before applying the algorithm. The ESPRIT method for estimating the parameters of the model is implemented as follows [6]:

1. For $\mathbf{y}(n) = [z(n) \cdots z(n+L-1)]^T$, the data covariance matrix is computed by $\mathbf{R} = \frac{1}{M}\sum_{n=1}^{M} \mathbf{y}(n)\mathbf{y}^T(n)$, where $\mathbf{R}$ is of size M, $L \geq M > 2p$. Or, one can first generate the data sample matrix by using the Matlab function $\mathbf{Y} = hankel(y(1:M), y(M:L))$ then compute $\mathbf{R} = \mathbf{Y}\mathbf{Y}^T$.
2. The eigenvalues λ_i and the corresponding eigenvectors $\mathbf{s}_i$ of $\mathbf{R}$ are found, $i = 1, 2, \cdots, 2p$. The eigenvalues are arranged in a decreasing order.
3. Considering the first $2p$ eigenvectors, matrices $\mathbf{S}$, $\mathbf{S}_1$ and $\mathbf{S}_2$ are formed:

   $$\mathbf{S} = [\mathbf{s}_1 \cdots \mathbf{s}_{2p}],\ \mathbf{S}_1 = [\mathbf{I}_{M-1}\ \ 0]\mathbf{S},\ \mathbf{S}_2 = [0\ \ \mathbf{I}_{M-1}]\mathbf{S}$$

   where $\mathbf{I}_{M-1}$ is the identity matrix of size $(M-1)$.
4. The eigenvalues of the matrix $\psi = (\mathbf{S}_1^T\mathbf{S}_1)^{-1}\mathbf{S}_1^T\mathbf{S}_2$ are found. These eigenvalues $(c_1, \ldots, c_{2p})$ determine the frequencies f_i and the damping factors α_i:

   $$f_i = \frac{f_s\ angle(c_i)}{2\pi},\ \ \alpha_i = -f_s \ln(|c_i|) \qquad (2)$$

   where f_s is the sampling frequency.

It is worth to notice that the ESPRIT algorithm uses complex exponentials instead of sinusoids, and the equivalent expression of (1) in complex exponentials can easily be yielded by noting the relation $\cos(\omega_i n) = \left(e^{j\omega_i n} + e^{-j\omega_i n}\right)/2$. Since each sinusoid consists of a positive and a negative spectral line, for a music signal consisting of p sinusoids the model order in the ESPRIT algorithm is set to be $2p$.

For computing the remaining parameters in the model, the following equation is solved using L signal samples ($L > M$):

$$\mathbf{x} = \mathbf{V}\mathbf{h} \qquad (3)$$

where

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ c_1 & c_2 & \cdots & c_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ c_1^{L-1} & c_2^{L-1} & \cdots & c_{2p}^{L-1} \end{bmatrix},$$

$$\mathbf{x} = [z(0) \quad \cdots \quad z(L-1)]^T, \ \mathbf{h} = [h_1 \quad h_2 \quad \cdots \quad h_{2p}]^T$$

The least squares solution to (3) is

$$\mathbf{h} = (\mathbf{V}^H\mathbf{V})^{-1}\mathbf{V}^H\mathbf{x} \tag{4}$$

Having computed $\mathbf{h}$, the amplitude A_i and the initial phase ϕ_i are calculated as:

$$A_i = 2|h_i|, \quad \phi_i = angle(h_i) \tag{5}$$

4.3. Tracking Sinusoids

Once the parameters of sinusoid candidates are estimated in each block of data, a tracking algorithm is then applied. For tracking the valid sinusoids in music signal, a constraint on time continuity of sinusoids is imposed. This is performed in the algorithm as follows. A threshold th_{df} is set for maximum possible frequency change in the consecutive blocks for each individual sinusoid. Also a minimum length th_{len} for a valid time trajectory of sinusoid is set. Further, to tolerant estimation error due to noise and tracking error in some individual blocks, a time trajectory is allowed to be broken for a very short duration up to a maximum length th_{break}. These thresholds are determined empirically.

4.4. Other Parameter Settings

Block size L: The size of the data block is chosen such that data samples in each block are approximately stationary. The size of data block is set to be about 20-30 ms in our tests.
Autocorrelation Matrix Size M: The size M of data autocorrelation matrix is the summed value of signal subspace dimension ($2p$) and the noise subspace dimension ($M - 2p$). For data length $L \gg 2p$, M is usually set to be $M \approx L/2.5$.
Step Size: The step size that each sliding window is shifted forward is a tradeoff between the speed at which the parameters are being updated and the computational cost. In our tests, 50% window overlap is chosen (i.e. a step size of $2/L$ samples).
Model Order p: The number of sinusoids in (1) usually varies from different sliding windows. However, for each given music signal sequence we choose the same order for all windows of data. The principle is that the order p should be chosen sufficiently high so that the remaining noise $w(n)$ in (1) is close to white. This is also dependent on the underlying complexity of music instrument which generates the specific signal. Since the MSE (mean square error) of the estimated signal decreases monotonically when the model order increases, one may set a desired threshold for selecting the model order. In our tests, the model order is empirically chosen in between 16 and 24 for music signals that are down-sampled to 8KHz.

4.5. Simulations and Results

To test the proposed methods, analysis and synthesis of music signals with a variety of complexities were conducted. All test signals were down-sampled to 8KHz. The window size for data blocks for all music signals was set to be 25ms and the overlap between neighboring windows was 50%. First, the music signal is analyzed. The tracked time-varying parameters were then used for re-synthesizing.

Example 1: Analysis/synthesis of a music signal from 'banjo'.
In this example, a simple music signal was used for the simulation. The signal was generated by a pluck of a single string on the music instrument 'banjo'. The signal lasted approximately 1.5 seconds. The sliding-window LS-ESPRIT algorithm was applied to estimate the damped sinusoid components, where the number of sinusoids was set to $p = 16$ (or, $2p$=32 in complex exponentials). The 1st row of Fig.1 shows the results from applying to Banjo signal. The original signal, the synthesized signal and the residuals between these two signals are included (1st column). One can observe that the residuals are rather small after the algorithm converges (after about 500 samples). The estimated frequencies and magnitudes of sinusoids in time from the sliding-window ESPRIT are shown in the 2nd column of Fig.1, where the color indicates the magnitudes of sinusoids. One can clearly see that the sinusoids in Banjo signal are almost stationary (i.e., in horizontal lines), starting from strong magnitudes (in red) and gradually fading out (in blue). As the sinusoids fade, one can observe that more spurious peaks from ESPRIT are due to the noise, which are more randomly scattered. The sub-figure in the 3rd column shows the tracked frequencies of sinusoids, where the parameters used for the tracking sinusoids were set to be $th_{df} = 5Hz$, th_{len}=10 (blocks) and th_{break}=2 (blocks). We can observe that most selected data is correct, there are cases of spectral line split when the music signal fades out. These tracked sinusoids were then used for synthesis except that the low frequency curve near zero frequency (dc component) was not used. For comparing the results, the 4th and 5th columns in Fig.1 show the spectrograms of the original and the re-synthesized banjo signals. One can also observe that some split of estimated sinusoids resulted in the bumps in the spectrogram of synthetic signal. Further, one can observe that the dc component is removed from the synthesized signal.

Example 2: Analysis/synthesis of a flute signal.
In this example, the music signal was generated from a flute. The results are included in the 2nd row of Fig.1, including the original signal waveform overlapped with re-synthesized and residual waveforms of the flute signal; the results from the sliding-window ESPRIT ($2p$=32) and the tracked sinusoids where the tracking parameters were set to th_{df}=10Hz, th_{len}=5 and th_{break}=2; and finally the spectrograms of the original and the re-synthesized flute signal.

Example 3: Analysis/synthesis of mixed music contents.
In this example, the music signal was extracted from a segment of music from a Björk's music CD ("it's all so quiet"),

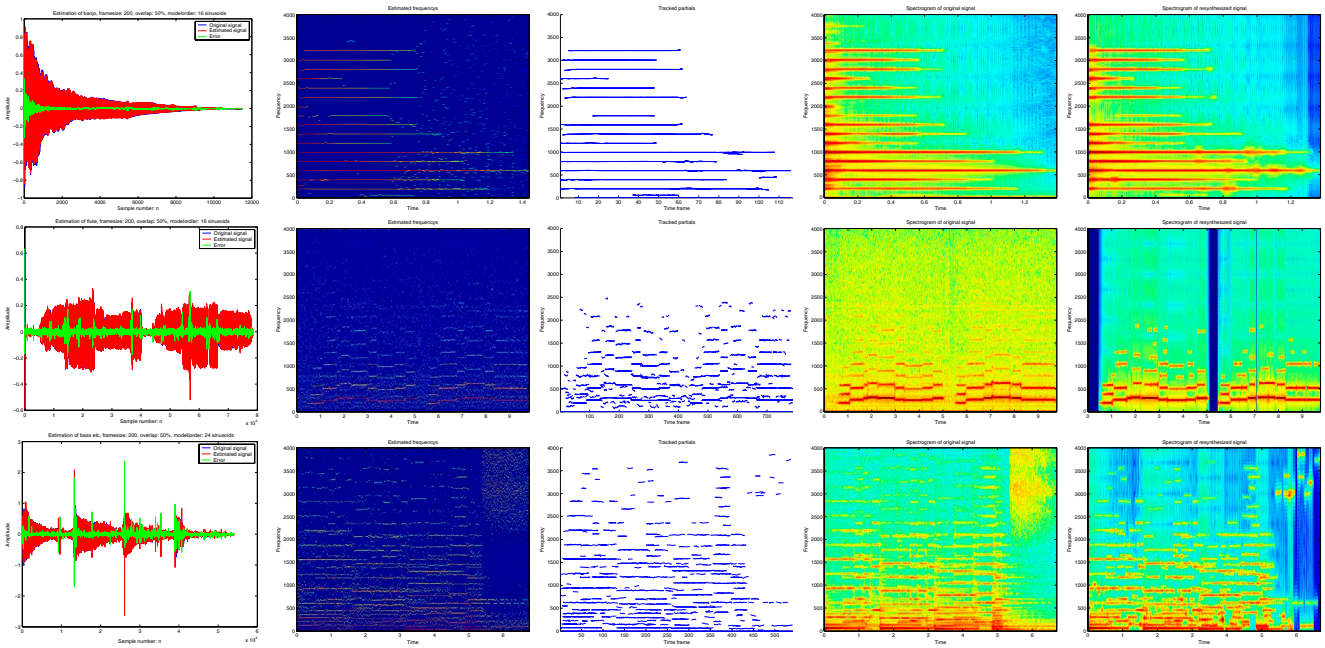

Fig. 1. Test results from using the proposed method. Rows 1 to 3 correspond to: the results from banjo signal, flute signal, and a segment of Björk's song (near the end of the spectrogram) plus music signal. Columns 1-5 correspond to: overlapped original (blue) and synthesized signal (red, or darkest gray values) and residuals signal (green, or light gray values close to the horizontal axis); results from the sliding-window ESPRIT; tracked sinusoids for the signal; spectrogram of original signal; spectrogram of re-synthesized signal using tracked sinusoids (blue color in spectrograms corresponds to the lowest energy in signals).

containing mixed signals from multiple music instruments. The 3rd row contains the corresponding analysis and synthesis results. This includes the original signal waveform overlapped with re-synthesized and residual waveforms, the results from the sliding-window ESPRIT ($2p$=48) and the tracked sinusoids where the tracking parameters were set to th_{df}=10Hz, th_{len}=10 and th_{break}=2; and finally the spectrograms of the original and the re-synthesized music signals.

From our simulation results as well as listening tests which compared between the original and the synthesis music signals, the proposed method is shown to have provided satisfactory results and to be very robust for a range of music signals.

5. CONCLUSIONS

A novel music signal synthesis scheme is proposed by applying the sliding-window ESPRIT for estimating the parameters of damped sinusoid candidates followed by tracking the sinusoid trajectories in music signals. Although ESPRIT method itself is not new, and sinusoid models have been studied previously for audio signals, to authors' knowledge, our scheme is novel for successfully applying ESPRIT to dynamic music signal synthesis based on damped sinusoid models. Our simulations performed on music signals with a range of complexity have shown that the proposed scheme is very robust for re-synthesis of music signals. In particular, the proposed system has obtained good sound quality to segments of dynamic and complex music signals from mixed instruments, which shows a good potential for music synthesis using methods outside the range of audio coding standards.

6. REFERENCES

[1] "MPEG-4 Standard for coding moving pictures and audio", International organization for standardisation, ISO/IEC JTC1/SC29/WG11.

[2] V.Välimäki, H.Penttinen, J.Knif, M.Laurson, C.Erkut, Sound synthesis of the harpsichord using a computationally efficient physical model, EURASIP Journal JASP, No.7, pp.934-948, 2004.

[3] S.Bilbao, "wave scattering methods for solving partial differential equations", thesis, available at http://ccrma.stanford.edu/ bilbao/master/goodcopy.html.

[4] M.Lagrange, S.Marchand, J-B.Rault, "Using Linear Prediction to Enhance the Tracking of Partials", In proc. of ICASSP 2004, Canada.

[5] M.Lagrange, S.Marchand, J-B.Rault, "Tracking partials for the sinusoidal modeling of polyphonic sounds", in proc. of ICASSP conf., USA, 2005.

[6] A.Eriksson, P.Stoica, T. Soderstrom, "Second-order properties of MUSIC and ESPRIT estimates of sinusoidal frequencies in high SNR scenarios", IEE Proc. Radar and Signal Processing, vol.140,pp.266-272, 1993.