Sinusoidal and Residual Decomposition and Residual Modeling of Musical Tones Using the QUASAR Signal Model

Yinong Ding
DSP Research & Development Center
Texas Instruments, Inc., MS 8374
Dallas, TX 75265-5474, USA
ding@ti.com

Xiaoshu Qian
Department of Electrical Engineering
University of Rhode Island
Kingston, RI 02881, USA
qian@ele.uri.edu

Abstract
A novel approach to the analysis and synthesis of musical tones based on a Quadratic polynomial-phase Sinusoids And Residual (QUASAR) signal model is presented. Taking advantage of the fact that musical signals are usually analyzed off-line, the proposed approach estimates the sinusoidal parameters using a global waveform fitting criterion and uses LPC-based waveform coding techniques for synthesizing residuals. An example of piano note is included to demonstrate the advantages of the proposed approach in terms of memory use (data reduction), computational complexity, and synthesis quality.

1 Introduction
In recent years, music and speech synthesis using a sinusoidal model has been an active and interesting research topic. In their Spectral Transformation System (STS or M&Q) [1], McAulay and Quatieri of MIT Lincoln Laboratory proposed to use a sinusoidal model for speech analysis and synthesis. In 1990, Serra and Smith developed a Spectral Modeling System (SMS) [2]. SMS achieved several advancements over M&Q for music applications. One significant contribution of SMS is the proposal for a sinusoidal and stochastic decomposition of musical tones and the use of a noise-driven source filter model for synthesis of the stochastic component. In both M&Q and SMS, sinusoids are modeled (parameterized) by their amplitude, frequency and phase. In order to account for the time-varying nature of the signal being analyzed, analysis is performed on a frame-by-frame basis. At the synthesis time, sinusoidal parameter estimates obtained at frame boundaries are interpolated to obtain their values at each sample point. In [1], McAulay and Quatieri also proposed to use a cubic phase interpolation algorithm to synthesize sinusoids. In our study of M&Q and SMS, several problems related to the signal model and the analysis and synthesis algorithms are identified. First, the parameter estimates obtained by a frame-based approach are “time-localized”, not being able to take full advantage of the fact that most musical tones are processed off-line. Second, an analysis by synthesis approach is not employed. Therefore the synthesis interpolation errors cannot be taken into account while model parameters are estimated. In addition, the cubic phase interpolation algorithm has an inherent artifact of frequency track “oscillation” in some scenarios and is computationally expensive [3, 4]. Third, a noise-driven source filter model for synthesis of stochastic components generally does not work well due to the loss of phase coherence or alignment between the sinusoidal and the stochastic part [3].

In this paper, we introduce a Quadratic polynomial-phase Sinusoids And Residual (QUASAR) signal model for music analysis and synthesis. The sinusoidal parameters can be estimated using an analysis-by-synthesis approach and a global waveform fitting criterion. We propose to use LPC-based waveform coding techniques for synthesizing the residual obtained by subtracting the sinusoidal part from the original.

2 Signal Models
Various natural signals, such as music and speech, as well as many signals used in communications, radar, sonar, and other man-made signals, can be represented by a sum of sinusoids. A commonly used sinusoidal model is given in (1), where a signal is modeled as a sum of sinusoids with time varying amplitudes, $A_m(t)$, frequencies, $\omega_m(t)$ and phases, $\theta_m(t)$.

$$s(t) = \sum_{m=1}^{M} A_m(t) \sin(\omega_m(t) t + \theta_m(t)),$$

$$\theta_m(t) = \int_0^t \omega_m(\sigma) d\sigma + \eta_m + \Phi(\omega_m(t)).$$

(1)
In the above signal model, \( M \) is the number of sinusoids, which could also be a function of time in some applications. The time-varying phase \( \theta_m(t) \) contains the time-varying frequency contribution, a fixed phase offset, \( \eta_m \), which accounts for the fact that the sine waves will generally not be in phase, and a time-varying frequency dependent term, \( \Phi(\omega_m(t)) \). In the speech production model, \( \Phi(\omega_m(t)) \) is determined by the phase function of the vocal tract.

The signal model in (1) is also the one used in M&Q and SMS. To extract the time-varying amplitudes, frequencies and phases of the component sine waves, a conventional approach is to analyze the signal on a frame-by-frame basis. At the synthesis stage, the parameters estimated at frame boundaries are interpolated to obtain their values at each sample point. A frame-based approach suffers from two major limitations. First, it estimates signal parameters only using the "local" data, unable to “look-ahead” and "look-back". Therefore the parameter estimates are “time-localized”. Secondly, a frame-based approach has to use some kind of window to truncate the data. The window effect thus limits the estimation resolution. Further more, the interpolation for synthesis usually produces errors, especially if the signal parameters do not vary slowly with respect to the analysis frame rate, the frame rate parameter estimates can not be reliably interpolated using lower-order evolution models such as linear interpolation for amplitudes.

For music synthesis applications, musical notes are usually pre-recorded. This allows the estimation of model parameters to be performed in a globally optimized sense, and this also allows an analysis-by-synthesis approach to be employed so that the interpolation errors can be taken into account while the model parameters are estimated. We show how these can be implemented in Section 3.

### 2.1 Cubic phase interpolation

In signal model (1), sinusoids are parameterized by time-varying amplitudes, frequencies, and phases. The parameter values at frame boundaries are estimated at the analysis time and are interpolated at the synthesis stage. For many years, researchers have used a cubic phase interpolation algorithm to synthesize sinusoids. The idea of applying a cubic polynomial to interpolate the phase between frame boundaries was independently proposed by McAulay and Quatieri in their Spectral Transformation System [1] and by Almeida and Silva in their harmonic sine wave synthesizer [5]. The cubic phase interpolation algorithm is now reviewed below.

Assume the frequency and phase estimates of the \( m \)th frequency track at two ends of the \( m \)th frame are

\[
\{\hat{\omega}_m^n, \hat{\theta}_m^n\} \quad \text{and} \quad \{\hat{\omega}_m^{n+1}, \hat{\theta}_m^{n+1}\},
\]

then a cubic phase interpolation function can be postulated as follows,

\[
\hat{\theta}_m^n(\tau) = a_m^n + b_m^n \tau + c_m^n \tau^2 + d_m^n \tau^3,
\]

where \( \tau = t - t_m \), and \( t_m \) is the starting time of the \( m \)th frame. In order to guarantee the phase function values match the measured (estimated) frequencies and phases at the frame boundaries, the cubic polynomial coefficients \( a_m^n, b_m^n, c_m^n \) and \( d_m^n \) are obtained by solving a set of linear equations, and the results are shown below [1], (the track subscript, \( m \), has been omitted for clarity.)

\[
a^n = \hat{\theta}^n, \quad b^n = \hat{\omega}^n, \quad c^n = \hat{\theta}^{n+1} - \hat{\theta}^n - \hat{\omega}^n T + 2\pi P,
\]

and \( e^n \) satisfy the following matrix equation

\[
\begin{bmatrix}
  c^n (P) \\
  d^n (P)
\end{bmatrix} =
\begin{bmatrix}
  \frac{1}{\pi} & \frac{1}{\pi} \\
  \frac{1}{2\pi} & -\frac{1}{2\pi}
\end{bmatrix}
\begin{bmatrix}
  \hat{\theta}^{n+1} - \hat{\theta}^n - \hat{\omega}^n T + 2\pi P \\
  \hat{\theta}^{n+1} - \hat{\omega}^n
\end{bmatrix}.
\]

The integer \( P \) in the above equation is given by

\[
P = \frac{1}{2\pi} \left( \hat{\theta}^n + \frac{\hat{\omega}^n + \hat{\omega}^{n+1}}{2} T - \hat{\theta}^{n+1} + \epsilon \right),
\]

where \( T \) (and throughout of the paper) is the frame length, and \( \epsilon \) is the smallest number that makes \( P \) an integer. It is clear that \( |\epsilon| < \pi \). From (5) and (6), we can obtain

\[
a^n = \frac{\hat{\omega}^{n+1} - \hat{\omega}^n}{2\pi}, \quad b^n = \frac{\hat{\theta}^{n+1} - \hat{\theta}^n}{2\pi},
\]

This cubic phase interpolation algorithm seems to have gained a wide acceptance in the speech and music signal processing community. However, in our experiments with a variety of musical instrument sounds, it is noticed that the interpolated frequency tracks resulted from the cubic interpolation scheme tend to exhibit small “oscillations” which is especially conspicuous when the difference of the frequency measurements at frame boundaries is small. This phenomenon of frequency track oscillation is illustrated using synthetic data in Figure 1(a), where the frequency measurements at the frame boundaries are constant, \( f_0 = 2000 \) Hz, and the phase measurement at the \( n \)th frame is \( (1 + 0.012e_n)|2\pi f_{0n}T + \eta_0 \) (module 2\( \pi \)). The 0.012\( e_n \),
is the perturbation term used to model phase measurement errors. $e_n$ is independently and identically distributed Gaussian random variable with zero mean and unit variance. $\theta_0$ is a random initial phase, uniformly distributed over $[-\pi, \pi]$. Also shown in Figure 1(b) is the corresponding phase track (solid line). Both frequency and phase measurements are denoted by the circle symbols.

Figure 1: Frequency and phase tracks obtained using the cubic (solid line) and the least squares quadratic (dashed line) phase interpolation algorithms (a) Frequency tracks; (b) Phase tracks.

The phenomenon of the frequency track “oscillation” shown in Figure 1(a) is actually predictable. From the interpolation formulas (3), (4), (5), (7), the frequency derivatives (the second order derivatives of the instantaneous phase) at the frame boundaries are given by

$$\\omega'(0) = \frac{\Lambda^{n+1} - \Lambda^n}{T} + \frac{\Lambda^{n+1} - \Lambda^n}{T^2}$$

$$\\omega'(T) = \frac{\Lambda^{n+1} - \Lambda^n}{T} - \frac{\Lambda^{n+1} - \Lambda^n}{T^2}$$

We can see that when the difference of the frequency measurements at frame boundaries is small, the first term in (8) is smaller in magnitude than the second one. The $\omega'(0)$ and $\omega'(T)$ have opposite signs, forcing the frequency track changes its slope at frame boundaries, and thus displaying the phenomenon of “oscillation”. This phenomenon, though hardly noticeable in many applications, is definitely not desired, and nor is there any physical basis to justify its existence.

2.2 Least squares quadratic phase interpolation

Due to inevitable estimation errors, the frequency and phase measurements shown in (2) will not be consistent in general. Imposing a cubic phase model and forcing the interpolated parameters to match the error-corrupted estimates will end up with creating artifacts. In addition, using cubic phase model increases computational complexity of a synthesis system. Motivated by producing smoother frequency tracks and reducing computational complexity, we model the instantaneous phase of a sinusoidal component as a quadratic polynomial. At the nth frame, the phase function is repre-
sented by
\[ \theta^n(\tau) = a^n + b^n \tau + c^n \tau^2. \] (9)
Assume we have N frames. There will be 3N unknowns (coefficients) to be determined from the frequency and phase measurements. However, due to the frequency and phase continuity constraints at frame boundaries:
\[ \begin{align*}
\theta^n(T) &= \theta^{n+1}(0), \\
\omega^n(T) &= \omega^{n+1}(0), & n = 0, 1, \ldots, N - 2,
\end{align*} \] (10)
the total number of unknowns is reduced to 3N − 2(N − 1) = N + 2. Note that the phase function \( \theta(t) \) is a piecewise quadratic polynomial, it can also be expressed by a linear combination of \( N + 2 \) quadratic basis spline functions (B-splines) [6]. That is
\[ \theta(t) = \sum_{n=-2}^{N-1} a^n B_n(t), \] (11)
where \( B_n(t) \) is the nth quadratic B-spline function as illustrated in Figure 2. We determine the \( N + 2 \) coefficients, \( a^n, n = -2, -1, 0, \ldots, N - 1 \) by minimizing the following weighted squared error
\[ \varepsilon = \lambda \sum_{n=0}^{N} \left[ \theta(t_n) - \hat{\theta}^n \right]^2 + (1 - \lambda) \sum_{n=0}^{N} \left[ \omega(t_n) - \hat{\omega}^n \right]^2 T^2. \] (12)
The weighting factor \( \lambda \) in the above equation can be interpreted as the importance index for the “phase mismatch” relative to the “frequency mismatch” at frame boundaries or as the confidential level to the accuracy of the phase measurements. Readers are referred to [5] for details on how to find the combination coefficients, \( a^n, n = -2, -1, 0, \ldots, N - 1 \) when the above error minimization criterion is employed.

Applying this least squares quadratic phase interpolation algorithm with \( \lambda = \frac{1}{2} \) to the same synthetic data as for the example in Figure 1, we can get a much smoother frequency track with less computations. The resulting frequency and phase tracks, denoted by dashed lines, are plotted in Figure 1(a) and (b), respectively.

The difference between the least squares quadratic phase interpolation and the “magnitude only” reconstruction [1] is obvious, though the instantaneous phase function of the latter is also a second order polynomial. In the proposed quadratic phase interpolation algorithm, both frequency and phase measurements are taken into account and are made “consistent” in a least square sense, while in the magnitude-only reconstruction, the phase measurements are totally ignored and the phase is simply the integral of the instantaneous frequency.

### 2.3 The QUASAR signal model

Based on above discussions, we propose to model a musical signal as a sum of quadratic polynomial-phase sinusoids and residual, i.e.,
\[ s(t) = \sum_{m=1}^{M} A_m(t) \sin(\theta_m(t)) + r(t), \]
\[ \theta_m(t) = a_m(t) + b_m(t) t + c_m(t) t^2. \] (13)
In this model, \( r(t) \) is the residual of \( s(t) \). Since \( r(t) \) usually sounds very noisy, it is also called the stochastic part or noise of \( s(t) \). \( r(t) \) contains important instrument information which can not be well represented by sinusoids such as the breath noise in a flute or saxophone, and the bowing friction noise of a violin. Without this information, the synthesized musical tones tend to sound artificial. Introducing a residual part into the signal model avoids using a very high number of sinusoids to model the noise-like part of the signal, and thus makes the model more parsimonious.

It should be pointed out that the model for the sinusoidal part in (13) is different from that in (1). Though many functions, including the phase function in (1) can be generally represented with a polynomial based on the Taylor expansion theory, the model in (13) specifically assumes that the sinusoidal phase is a second order polynomial, and the coefficients of the polynomial are the model parameters to be (directly) estimated from the data. In the next section, we propose a novel approach to estimating quadratic phase sinusoidal parameters for musical application. We obtain these parameter estimates so that the given signal (data) fits the model in a least square sense. It is well known that the least squares estimator is equivalent to an maximum likelihood estimator if the noise is white Gaussian.

### 3 Estimation of Model Parameters

In the previous section, we discussed the rationales to use a quadratic polynomial phase sinusoidal model for the sinusoidal part of musical signals. In this section, we discuss how to estimate the model parameters from music data. Using polynomial phase sinusoidal model for signal processing is not new. It has applications in radar, sonar, and communications as well. A generic approach to finding the estimates of the phase coefficients and amplitudes of polynomial phase components is to derive a maximum likelihood estimation algorithm. Due to its computational burden, the maximum likelihood estimator is not practical. The computationally efficient alternatives have to be exploited.
In [7], a suboptimal algorithm based on a new transformation - the discrete polynomial-phase transformation (DPT) was proposed. For our application, considering the fact that musical notes are usually pre-recorded and analyzed off-line, we can seek a more suitable solution to the estimation problem.

### 3.1 Estimation of Sinusoidal Parameters

As pointed out earlier, most existing analysis algorithms do not take an analysis-by-synthesis approach. Therefore any errors resulted from the interpolation process for synthesis can not be recovered. This problem needs to be addressed when deriving our estimation algorithm. In order to do this, and by noting that each sinusoidal component has a nominal frequency, denoted by \( \omega_m \), whose contribution to the instantaneous phase is \( \omega_m t \), we first rewrite the signal model (13) as

\[
 s(t) = \sum_{m=1}^{M} A_m(t) \sin(\omega_m t + \phi_m(t)) + r(t), \tag{14}
\]

where \( \phi_m(t) \) is the phase deviation track of the \( m \)th sinusoidal component. It should be emphasized that \( \omega_m, \ m = 1, \cdots, M \) are by no means constrained to be harmonically or quasi-harmonically related. We then express the time-varying amplitude and phase deviation tracks as linear combinations of linear and quadratic spline functions, respectively, i.e.,

\[
 A_m(t) = \sum_{n=0}^{N} A_{m}^{n} A_n(t), \tag{15}
\]

\[
 \phi_m(t) = \sum_{n=-2}^{N} \alpha_{m}^{n} B_n(t). \tag{16}
\]

In the above equations, \( A_n(t) \) and \( B_n(t) \) are the \( n \)th linear and quadratic B-spline function, respectively, as illustrated in Figure 3 and 2, where \( n \) is the data frame index. From Equations (14) - (16), we see that the number of sinusoidal parameters has reduced from three to two per sinusoid per frame compared with any existing sinusoidal model-based music synthesis algorithms or systems. This represents more than 33% of savings in memory use and is a significant advantage for using the proposed signal model.

The objective of the estimation algorithm is to obtain estimates of \( A_{m}^{n} \) and \( \alpha_{m}^{n} \) based on the following error minimization criterion:

\[
 \left( \hat{A}_{m}^{n}, \hat{\alpha}_{m}^{n} \right) = \arg \min_{A_m A_{m}^{n} \alpha_{m}^{n}} \sum_{k=0}^{N L-1} |s(k) - \hat{s}(k)|^2, \tag{17}
\]

where

\[
 \hat{s}(k) = \sum_{m=1}^{M} \left( \sum_{n=0}^{N} A_{m}^{n} A_n(k) \right) \sin \left( \omega_m k + \sum_{n=-2}^{N} \alpha_{m}^{n} B_n(k) \right), \tag{18}
\]

\( k \) is the sample index, \( L \) is the number of samples in a frame of length \( T \), and the sampling rate is assumed to be 1 for convenience.

From Equation 17, we see that the sinusoidal parameters are estimated based on a waveform (data) fitting criterion as opposed to the energy fitting one used by M&Q and SMS. We also see that the error to be minimized is the energy of the residual for a given number of sinusoids. It is worth noting that the synthesis of sinusoids is implicitly performed while sinusoidal parameters are estimated. Because the synthesis process has been built into the analysis, any interpolation errors due to the use of the linear and quadratic evolution models (15) and (16) for amplitude and phase, respectively, will be included in the residual.

The completed estimation procedure for sinusoidal parameter is provided in [3, 4], and is diagrammed in Figure 4.

### 3.2 Modeling of the Residual

As stated in Section 2, the residual plays an important role in synthesis of musical tones. Without the residual, a synthesized musical tone tends to sound artificial. In addition, it is desirable to change and/or modify the residual for music synthesis. An effective modeling and analysis/synthesis of the residual is the key to achieve this objective.

#### A. Obtaining the Residual

From (13), we see that the residual can be obtained by subtracting the sinusoidal part from the original signal. It is well known that the synthesis of sinusoids dominates the computation of a sinusoidal model based music synthesizer, and the calculation of the instantaneous phase is the major portion for generating sinusoids. It is observed in [3] that the use of a quadratic phase model allows the instantaneous phase to be calculated recursively using two additions instead of two additions and two multiplications, and the recursive algorithm can not be applied when a cubic phase model is used due to distortions caused by accumulated numerical errors. This is not only because that a higher order polynomial has more numerical problems but also because that the coefficient of the cubic term is usually very small for musical tones, and thus is extremely
sensitive to the numerical errors. Due to these factors, the computational complexity for obtaining sinusoidal phase can be reduced by more than 50\% [3].

Once the sinusoidal part is reconstructed, it is straightforward to get the residual by subtracting the sinusoidal part from the original as shown in Figure 4.

B. Synthesis of the Residual

When proposing to decompose a musical tone into a deterministic and stochastic part, Serra and Smith also proposed to use a noise-driven source filter model for the analysis and synthesis of stochastic components [2]. The filtering can be implemented in time domain or in frequency domain. Correspondingly, the magnitude of the filter frequency response, or the magnitude spectrum is made an approximation to the spectrum envelope of the stochastic signal. Subsequent research on the problem, though varying in some aspects such as the proposed use of the noise perception model [8], followed the basic model with a pure random noise as the filter excitation. An advantage of using random noise as excitation is that it eliminates the need of memory for storing the residual. However, it is our experience (and of many others [8]) that this noise-driven source filter model generally does not work well due to the loss of phase coherence between the sinusoidal and the stochastic part. As a result, the sum of the sinusoidal and stochastic part often sounds like two separate signals. In order to maintain the phase coherence (waveform coherence) and still achieve data reduction, waveform coding-based coding techniques can be used for the residual analysis and synthesis.

Among various waveform coding algorithms, the Multi-pulse Excitation Linear Prediction (MPLP) for low-bit rate speech coding [9] is of particular interest. MPLP makes no prior assumption about the nature of its input, known as the excitation signal. In MPLP, the excitation signal consists of a sequence of pulses for all speech classes - including voiced and unvoiced. The coding algorithm makes no attempt to make its excitation periodic or non-periodic. On the other hand, we found that, the residual, when properly obtained, does not contain such musical information as pitch, loudness, and other “intelligently important” information. What it carries is the information about the characteristics of the instrument, providing the “gesture” of musical sounds. The nature of the residual along with the features of MPLP make this coding algorithm a good candidate for the modeling of musical signal residuals. Another similar, but much simpler coding algorithm is the Regular Pulse Excitation (RPE) algorithm [10].

A description of the MPLP algorithm in matrix arithmetic is given in [3], where a procedure for finding the locations and amplitudes of a given number of pulses in a given time interval different from that used in [9] is provided. The algorithm block diagram of MPLP is depicted in Figure 5, where \( A(z) \) is a qth order time-varying filter which can be determined by using linear prediction (LP) techniques. \( A(z/\gamma) \) is often called the error-shaping filter because it changes the spectrum shape of the error signal to be minimized. The vector \( b \) and \( p \) contain the amplitudes and locations of Q samples in an L sample long data frame, respectively. Normally, \( Q < L \).

![Figure 4: Analysis block diagram of the proposed analysis/synthesis algorithm](image)

4 Sound Example

In this section, we present an example of analysis and synthesis of musical tones. In this example, the signal is a piano note (G3, mezzo forté), and the optional parameter optimization step shown in Figure 4 was not performed. For sinusoidal parameter estimation, the analysis frame length T is set to about 3 milliseconds. Figure 6 shows the periodograms of the original note, the reconstructed sinusoidal part and the residual, respectively. From this figure, we see that the number of sinusoids is 8. The nominal frequencies of these 8 sinusoids are marked with + signs on the periodogram of the original note. Shown in Figure 7 are the signal waveforms of the original note, the reconstructed sinusoidal part and the residual, respectively. In order to show the details of the waveforms, only 0.1 second long of sam-
Figure 5: Block diagram of MPLP. (a) Analyzer, (b) Synthesizer.

5 Conclusions

We have proposed a new signal model for music analysis and synthesis in this paper. From our extensive experiments with a variety of musical instrument sounds including piano, trumpet, violin, flute, bassoon, we found that with the unique quadratic polynomial-phase sinusoids and residual signal model, along with the analysis and synthesis algorithms introduced in this paper, a very good sinusoidal and stochastic decomposition can be achieved. Compared with the M&Q and SMS algorithms, the proposed approach reduces memory use for storing sound parameters by more than 33% and reduces computations for calculating sinusoidal phase by about 50%. We are currently working on the implementation of the proposed algorithm on a Texas Instruments’ fixed-point DSP, and will report our results in the future.
Figure 7: Waveforms of the piano note G3, mezzo forte

Figure 8: Original and re-synthesized residuals of a piano note

References


