

MIT Open Access Articles

Sinewave parameter estimation using the fast fan-chirp transform

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Dunn, Robert B., Thomas F. Quatieri, and Nicolas Malyska (2009). Sinewave parameter estimation using the fast fan-chirp transform. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2009 (Piscataway, N.J.: IEEE): 349-352. © 2009 IEEE

As Published: <http://dx.doi.org/10.1109/ASPAA.2009.5346528>

Publisher: Institute of Electrical and Electronics Engineers

Persistent URL: <http://hdl.handle.net/1721.1/59315>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



SINEWAVE PARAMETER ESTIMATION USING THE FAST FAN-CHIRP TRANSFORM*

Robert B. Dunn, Thomas F. Quatieri, Nicolas Malyska

MIT Lincoln Laboratory
244 Wood Street, Lexington, MA 02420
[rbd,quatieri,nmalyska]@ll.mit.edu

ABSTRACT

Sinewave analysis/synthesis has long been an important tool for audio analysis, modification and synthesis [1]. The recently introduced Fan-Chirp Transform (FChT) [2,3] has been shown to improve the fidelity of sinewave parameter estimates for a harmonic audio signal with rapid frequency modulation [4]. A fast version of the FChT [3] reduces computation but this algorithm presents two factors that affect sinewave parameter estimation. The phase of the fast FChT does not match the phase of the original continuous-time transform and this interferes with the estimation of sinewave phases. Also, the fast FChT requires an interpolation of the input signal and the choice of interpolator affects the speed of the transform and accuracy of the estimated sinewave parameters. In this paper we demonstrate how to modify the phase of the fast FChT such that it can be used to estimate sinewave phases, and we explore the use of various interpolators demonstrating the tradeoff between transform speed and sinewave parameter accuracy.

Index Terms— Sinewave Analysis/Synthesis, Fan-Chirp Transform, Sinewave Phases

1. INTRODUCTION

An audio signal can be represented as a sum of time-varying sinusoids which have continuously varying amplitudes, frequencies and phases [1]. The parameters of these sinewaves are typically estimated on a frame-by-frame basis by sampling the peaks of the short-time Fourier transform magnitude to determine the amplitudes, frequencies and phases of the component sinusoids. The Fan-Chirp Transform (FChT) is a generalization of the Fourier transform in which the harmonically related basis functions have linear frequency modulation [2,3]. Audio signals often have harmonics with significant frequency modulation and the FChT has been shown to provide better resolution than the Fourier transform for estimation of sinewave parameters representing such signals [4]. A fast version of the FChT operates by reformulating the FChT as the Fast Fourier Transform (FFT) of a time-warped signal, thus significantly reducing computation [3]. Unfortunately, this fast FChT, as presented in [3], has a phase function that does not correspond to the phase of the original continuous-time FChT and it cannot be directly used to estimate sinewave phases. In this paper we describe the source of this phase difference and show how to modify the implementation of the fast FChT so that its phase function will match that of the original

continuous-time transform. The fast FChT requires time-warping the input signal and this is accomplished via interpolation. The choice of interpolation function naturally affects both the accuracy and speed of the overall transform. We investigate how the tradeoff between accuracy and speed for the fast FChT affects the accuracy of estimated sinewave parameters.

This paper is organized as follows. We start by describing the continuous-time FChT in section 2 and demonstrating how it can be reformulated as the Fourier transform of a time-warped signal. Section 3 presents the discrete-time FChT and shows the steps necessary to perform the fast FChT as well as explaining the phase correction required for sinewave analysis. An example of sinewave analysis/synthesis demonstrating the importance of phase correction is given in section 4. Finally, in section 5 we examine the computational reduction made possible with the fast FChT and explore how the choice of interpolator used in the fast FChT provides a tradeoff between transform speed and sinewave parameter accuracy.

2. CONTINUOUS-TIME FAN-CHIRP TRANSFORM

This section describes the continuous-time FChT and shows how it can be reformulated as the Fourier transform of a time-warped signal.

2.1. Fan-Chirp Transform Definition

For the sake of simplicity and because in practice we are interested in working exclusively with finite-duration signals, we use the following definition of the Fan-Chirp Transform (FChT) for a signal $x(t)$, centered at the origin, with duration T :

$$X(f, \alpha) \triangleq \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \sqrt{|\phi'_\alpha(t)|} \exp(-j2\pi f \phi_\alpha(t)) dt \quad (1)$$

where the phase function $\phi_\alpha(t)$ is defined as:

$$\phi_\alpha(t) \triangleq (1 + \frac{1}{2}\alpha t) t \quad (2)$$

and $x(t)$ is non-zero only on the interval $-\frac{T}{2} \leq t \leq \frac{T}{2}$. In order to prevent the derivative of the phase function $\phi_\alpha(t)$ (frequency of the basis functions) from becoming zero the chirp rate α is constrained to $|\alpha| < \frac{2}{T}$.

* This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2.2. Fourier Transform of a Time-Warped Signal

The Fan-Chirp Transform can be reformulated as a Fourier transform by warping the time axis according to the function $\tau = \phi_\alpha(t)$ [2,3]. With this variable substitution the FChT becomes:

$$X(f, \alpha) = \int_{\phi_\alpha(-\frac{T}{2})}^{\phi_\alpha(\frac{T}{2})} \tilde{x}(\tau) \tilde{\rho}(\tau) \exp(-j2\pi f \tau) d\tau \quad (3)$$

where $\tilde{x}(\tau)$ is a time-warped version of $x(t)$ and $\tilde{\rho}(\tau)$ is a scaling function on the time-warped axis. Equation (3) is easily seen to be the Fourier transform of the product $\tilde{x}(\tau)\tilde{\rho}(\tau)$ and this leads to efficient implementation in discrete time using the FFT. We use the inverse of the warping function $\phi_\alpha(t)$, which we denote $\psi_\alpha(\tau) = \phi_\alpha^{-1}(\tau)$, to compute the time-warped input signal $\tilde{x}(\tau) = x(\psi_\alpha(\tau))$. Since the time warping function $\phi_\alpha(t)$ is quadratic, its inverse function $\psi_\alpha(\tau)$ has two solutions. The solution of interest is:

$$\psi_\alpha(\tau) = \frac{-1}{\alpha} + \frac{\sqrt{1+2\alpha\tau}}{\alpha} \quad (4)$$

(the other solution for $\psi_\alpha(\tau)$ reverses the time axis reflecting it about the point $-\frac{1}{\alpha}$.) The scaling function in equation (3) can be shown to be:

$$\begin{aligned} \tilde{\rho}(\tau) &= \sqrt{|\phi'_\alpha(\psi_\alpha(\tau))|} \psi'_\alpha(\tau) \\ &= \frac{1}{\sqrt{1+2\alpha\tau}} \end{aligned} \quad (5)$$

where prime denotes derivative.

3. DISCRETE-TIME IMPLEMENTATION

We first define the FChT of a discrete-time signal and then derive a fast implementation based on time warping and computing an FFT. It is assumed that the input signal $x[n]$ has an odd length N and is centered at the origin. Centering the signal at the origin allows correct estimation of sinusoid phases directly from the FChT output. While it is possible to operate with an even length signal, correct phase estimation would in that case require adjustments that add unnecessary complication.

3.1. Direct Form Discrete-Time Implementation

The discrete-time FChT can be realized directly from equations (1) and (2) as:

$$X[k, \hat{\alpha}] = \sum_n x[n] \sqrt{|1 + \hat{\alpha}n|} \exp(-j2\pi \frac{k}{K} (1 + \frac{1}{2} \hat{\alpha}n)n) \quad (6)$$

where n is the time sample index with range $-\frac{N-1}{2} \leq n \leq \frac{N-1}{2}$, N is the length in samples of $x[n]$, k is the frequency sample

index, K is the number of frequency samples, the normalized chirp rate is $\hat{\alpha} = \alpha T_s$, and T_s is the sampling period. We note that the phase of the discrete-time FChT in equation (6) corresponds to a sampled version of the phase of the continuous-time FChT in equations (1) and (2). The FChT can be readily evaluated with equation (6), however, due to their inherent linear frequency modulation some of the basis functions are undersampled. This problem is substantially diminished by over-sampling $x[n]$ by a factor of two before computing the FChT [2,3].

3.2. Time-Warped Discrete-Time Implementation

As shown in section 2.2 the FChT can be computed by first time warping and scaling the input signal and then computing the Fourier transform of the result. In discrete time equation (3) is written as:

$$X[k, \hat{\alpha}] = \sum_m \tilde{x}[m] \tilde{\rho}[m] \exp(-j2\pi \frac{k}{K} m) \quad (7)$$

where the range of m will be determined below. Equation (7) can easily be evaluated as the FFT of the product $\tilde{x}[m]\tilde{\rho}[m]$.

The discrete-time signal $\tilde{x}[m]$ is created by uniformly sampling the continuous time signal $\tilde{x}(\tau)$. Due to time warping, the signal $\tilde{x}(\tau)$ has greater bandwidth than $x(t)$ when $\hat{\alpha}$ is non-zero and it is necessary to sample $\tilde{x}(\tau)$ with a higher sampling rate than $x(t)$ in order to diminish the effects of aliasing. We define the number of samples of $\tilde{x}[m]$ to be M where M is chosen to be appropriately larger than N as discussed in [2,3]. The range of $\tilde{x}(\tau)$ is $\phi_\alpha(-\frac{T}{2}) \leq \tau \leq \phi_\alpha(\frac{T}{2})$ which can be shown to have a duration T . In [3] the time instants at which the time warped signal $\tilde{x}(\tau)$ is sampled are defined as:

$$\tau_m = \phi_\alpha\left(-\frac{T}{2}\right) + \left(m + \frac{1}{2}\right) \frac{T}{M} \quad \text{for } 0 \leq m < M \quad (8)$$

where $\frac{T}{M}$ is the sampling period on the time-warped axis. The time-warped signal $\tilde{x}[m]$ is obtained by sampling the continuous time signal $x(t)$:

$$\tilde{x}[m] = x(t_m) \quad (9)$$

$$t_m = \psi_\alpha(\tau_m) \quad (10)$$

In practice, the signal $x(t)$ is not available but its discrete-time counterpart $x[n]$ is available and the required samples $x(t_m)$ are readily interpolated from $x[n]$.

The definition in equation (8) of the time instants, τ_m , at which the time-warped signal is sampled, is natural in that it selects a set of M samples that are symmetrically spaced between the end points of the time-warped signal, and the indices of the time-warped signal match the indices of the FFT buffer. Unfortunately this definition has the side effect of adding an implicit delay to the discrete-time signal $\tilde{x}[m]$. From (8) we can see that the delay at sample $m = 0$ is:

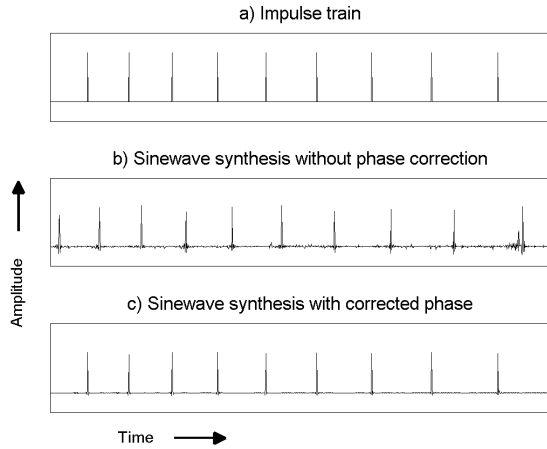


Figure 1: Example analysis/synthesis of an impulse train with linear-FM.

$$\tau_0 = \phi_\alpha \left(-\frac{T}{2} \right) + \frac{T}{2M}. \quad (11)$$

This delay does not change the scale factor $\tilde{\rho}(\tau)$ which in [3] is computed from τ_m (which itself is not delayed). It also does not change the magnitude of the resulting FChT. The delay does, however, change the phase of the FChT such that it does not match the phase of the discrete-time FChT given in equation (6) and does not match the phase of the continuous time FChT given in equations (1) and (2). It can be shown that the phase shift, $\Delta\theta(f)$, added to the fast FChT due to the delay τ_0 is:

$$\begin{aligned} \Delta\theta(f) &= 2\pi f \phi_\alpha(\psi_\alpha(\tau_0)) \\ &= 2\pi f \tau_0 \end{aligned} \quad (12)$$

The phase shift in equation (12) must be removed from the fast FChT before it can be used to properly estimate phases for sinewave analysis/synthesis. In discrete-time this phase shift can be evaluated as:

$$\Delta\theta[k] = \pi \frac{k}{K} \left(\frac{N}{M} - \left(1 - \frac{1}{4} \hat{\alpha} N \right) N \right) \quad (13)$$

where k is the frequency index and K is the length of the FFT. The phase of the fast FChT can be modified to match the phase of the FChT in equation (6) by multiplying the FFT output by $e^{-j\Delta\theta[k]}$. Observe that the delay, τ_0 , generally corresponds to a non-integer number of samples and thus it cannot be removed by a circular rotation of the samples in the FFT buffer.

An alternative method allows us to avoid the need for phase correction. The samples of the time-warped signal can be redefined such that there is no delay and phase correction is not required. This is accomplished by redefining τ_m as:

$$\tau_m = m \left(\frac{T}{M} \right) \quad (14)$$

The range of m in (14) is derived from the relationship $\phi_\alpha \left(-\frac{T}{2} \right) \leq \tau \leq \phi_\alpha \left(\frac{T}{2} \right)$, which yields:

$$M \left(\frac{1}{8} \hat{\alpha} N - \frac{1}{2} \right) \leq m \leq M \left(\frac{1}{8} \hat{\alpha} N + \frac{1}{2} \right) \quad (15)$$

The bounds of m in (15) correspond to the limits of the input signal $x[n]$ after it has been mapped onto the time-warped axis. Although m takes on only integer values, the bounds of m are generally not integer values due to the time warping. The minimum value of m is therefore the lowest integer that is greater than or equal to the lower bound. The maximum value for m is obtained by adding $(M-1)$ to the minimum value of m . The range of m includes both negative and positive values so the signal $\tilde{x}[m]\tilde{\rho}[m]$ must be placed in the FFT buffer by using the indices modulo the length of the FFT.

3.3. Scaling Function

The scaling function $\tilde{\rho}(\tau)$ in equation (5) requires evaluation of a 4th root. This computation can be simplified by evaluating the counterpart of $\tilde{\rho}(\tau)$ on the original time axis:

$$\begin{aligned} \rho(t) &= \tilde{\rho}(\phi_\alpha(t)) \\ &= \sqrt{\left| \phi'_\alpha(\psi_\alpha(\phi_\alpha(t))) \right| \psi'_\alpha(\phi_\alpha(t))} \\ &= \frac{1}{\sqrt{1 + \alpha t}} \end{aligned} \quad (16)$$

Using the above equation we can compute the product $\tilde{x}[m]\tilde{\rho}[m]$ by equating the signals before and after time warping:

$$\tilde{x}[m]\tilde{\rho}[m] = x(t_m)\rho(t_m) \quad (17)$$

where $x(t_m)$ is interpolated from $x[n]$ and $\rho(t_m)$ is computed using equation (16). The fast FChT is then completed by taking the FFT of $\tilde{x}[m]\tilde{\rho}[m]$.

4. SINEWAVE ANALYSIS/SYNTHESIS EXAMPLE

In sinewave analysis/synthesis correct estimation of the sinewave phases requires that the analysis window be centered at the origin [1]. If the analysis window is shifted from the origin, as results from equation (8), the estimated sinewave phases will contain a phase shift that has linear frequency dependence and a quadratic dependence on the length of the analysis window T . While in essence, this phase shift is simply a delay, if it is not corrected the result is much worse than a delayed output signal. The problem here is that on any given frame the measured sinewave parameters contain an accurate representation of the signal at the center of the analysis window (which becomes a delayed moment in time); these parameters, however, contain a much worse representation of the signal at the original (non-delayed) moment in time. Additionally, in sinewave analysis the duration of the analysis window varies with the pitch of the input signal so the delay in (11) will vary from one frame to the next. While one could account for the delay by appropriately reformulating the synthesis equations, this approach is mathematically tedious and would

Fast FChT Interpolation VS Parameter Error					
Method		Speed	ΔA	Δf	$\Delta \theta$
FFT		x1119	0.6 dB	67.7%	16.1%
Direct form FChT		x 1	0.2 dB	7.6%	0.4%
Fast FChT w/	Ideal (sinc)	x 5.1	0.2 dB	7.6%	0.4%
	Cubic Spline	x 63	1.1 dB	8.7%	0.4%
	Convolutional	x 78	1.8 dB	8.4%	0.7%
	Shape Preserving	x 111	5.8 dB	43.5%	50.0%

Table 1: Comparison of speed and accuracy of FFT, FChT, and Fast FChT with four interpolation methods: ideal, cubic spline, cubic convolutional and cubic shape preserving. Speed is relative to the direct form FChT, ΔA is median amplitude error, Δf is median frequency error, and $\Delta \theta$ is median phase error (for the highest harmonic).

require variable frame-rate synthesis which is undesirable. Instead, we recommend implementing the fast FChT using equations (14) and (15) to avoid any difficulty.

An example demonstrating this phase problem is shown in Figure 1 which compares sinewave analysis/synthesis using the fast FChT both with and without phase correction for an impulse train with a linearly decreasing fundamental frequency. A segment of the input signal is shown in 1a. Sinewave analysis followed by synthesis without phase correction, shown in 1b, results in a signal that is not only delayed but is also a rather poor reconstruction where the impulses have variable heights and some of the dispersed energy appears as noise. When the sinewave phases are corrected, as shown in 1c, both of these problems are removed.

5. SINEWAVE PARAMETER ESTIMATION

The fast FChT requires interpolation to compute the samples of the time-warped signal. We examined how different interpolation methods affect the speed and accuracy of sinewave parameter estimation with a matlab implementation of the fast FChT. In addition to ideal interpolation (sinc function) we tested three cubic interpolation methods: classic spline, convolutional [5], and shape preserving [6]. In matlab these cubic interpolation methods correspond to 'spline', 'v5matlab', and 'pchip' respectively. The speed of the transform depends on the length of the input signal and the number of frequency samples computed. In our test of transform speed we used parameters typical of our current high-quality sinewave analysis system: an average analysis window (20.0 ms), 16 kHz sampling, and 1025 frequency samples (from DC to and including π). The sinewave parameter errors are measured as median amplitude error (ΔA) in dB, median frequency error (Δf) as a percentage of a harmonic frequency bin, and median phase error ($\Delta \theta$) as a percentage of π radians. We report errors for the highest frequency harmonic of the test signal because this harmonic has the largest frequency modulation and additionally, it is most affected by interpolation methods that tend to reduce high frequency content.

A synthetic sum of harmonic sinusoids was used as a test signal so the true sinewave parameters could be determined at any point in time. The pitch of this test signal was varied sinusoidally between 80 and 180 Hz with five cycles per second. The results of our experiments, shown in Table 1, demonstrate that, in general, as

processing time is reduced the sinewave parameter errors are increased. Analysis using the FFT is seen to be up to three orders of magnitude faster than the FChT but the estimated parameters have the largest frequency error (over 50%) and second largest phase error. The fast FChT with ideal interpolation has the lowest errors (along with the direct form of the FChT) and is five times faster than the direct form of the FChT. Use of classic cubic spline interpolation increases the speed by an order of magnitude over ideal interpolation but this increases amplitude error from 0.2 to 1.1 dB. Cubic shape preserving interpolation has the greatest speed but has rather large errors and this method is not recommended.

6. CONCLUSION

In this paper we demonstrated how to modify the fast FChT and implement it in a manner that preserves the phase requirements of sinewave parameter estimation. We have shown that use of the fast FChT reduces computation time by a factor of five (using ideal interpolation) with no increase in sinewave parameter error. Additionally, we have shown that the choice of interpolation methods used in the fast FChT allows a tradeoff between the speed of the transform and the accuracy of the estimated sinewave parameters.

7. REFERENCES

- [1] T.F. Quatieri and R.J. McAulay, "Audio Signal Processing Based on Sinusoidal Analysis/Synthesis," chapter in Applications of Digital Signal Processing to Audio and Acoustics, M. Kahrs and K. Brandenburg, eds., Kluwer Academic Publishers, Boston, MA, 1998.
- [2] M. Képesi, and L. Weruaga, "Adaptive Chirp-Based Time-Frequency Analysis of Speech Signals," *Speech Communication*, Elsevier, Vol. 48, pp. 474-492, 2006.
- [3] L. Weruaga and M. Képesi, "The Fan Chirp Transform for Non-Stationary Harmonic Signals," *Signal Processing*, Elsevier, Vol. 87, pp. 1504-1522, 2007.
- [4] R.B. Dunn and T.F. Quatieri, "Sinewave Analysis/Synthesis Based on the Fan-Chirp Transform" *IEEE WASPAA*, New Paltz, NY, pp. 247-250, 2007.
- [5] R.G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans. ASSP*, Vol. ASSP-29, No. 6, Dec. 1981, pp.1153-1160.
- [6] F.N. Fritsch and R.E. Carlson, "Monotone Piecewise Cubic Interpolation," *SIAM J. Numerical Analysis*, Vol. 17, 1980, pp.238-246.