

## Real-Time Fundamental Frequency Estimation by Least-Square Fitting

Andrew Choi

**Abstract**—For real-time applications, a fundamental frequency estimation algorithm must be able to obtain accurate estimates from short signal segments. Characterization of the error function of fitting a sinusoid to the signal segment allows its spectrum to be deduced and the algorithm to be implemented efficiently.

**Index Terms**—Fundamental frequency estimation, least-square methods, rectangular window, pitch detection, pitch-to-MIDI converters.

### I. INTRODUCTION

A pseudoperiodic sound produced by a musical instrument is composed chiefly of harmonic components whose frequencies are integer multiples of a *fundamental frequency*. The problem of fundamental frequency estimation (FFE) is central to the automatic analysis of musical signals. *Pitch-to-MIDI converters* are devices through which conventional musical instruments can be attached to digital synthesizers as controllers. The sound from the musical instrument is analyzed to determine the notes played on it and MIDI<sup>1</sup> messages are generated and sent to cause these notes to be played on the synthesizer. In the application of FFE algorithms to pitch-to-MIDI converters, the fundamental frequencies must be estimated in real time. This paper will focus on FFE for monophonic signals.

A number of performance measures affect the suitability of an FFE algorithm for use in real time. The *response time* is the delay between the instant a note is played on the instrument and the instant the synthesizer begins to generate the corresponding sound. Assuming that the delay due to the synthesizer is negligible, the response time is equal to the sum of the length of the initial signal segment analyzed and the computation time of the FFE algorithm. Ideally, the response time must be so short that the player of the instrument does not perceive the delay. A response time longer than 35 ms will make playing the instrument unnatural for the performer [1]. To illustrate the difficulty of this problem and the scale of the involved quantities, consider the note E2—the lowest note that can be played on the guitar—which has a fundamental frequency of 82.4 Hz. A 15-ms segment of this note contains only a little more than a complete cycle of the waveform. A computation time of 15 ms for the FFE algorithm will result in a response time of 30 ms. An FFE algorithm for this application must therefore correctly determine the fundamental frequency from such a short initial signal segment and have a small computational requirement. Economic and engineering constraints have resulted in commercial pitch-to-MIDI converters whose response times are well over 30 ms for notes with low pitches [1]. Delays on these converters are very noticeable to the player of the instrument.

Another important performance measure of an FFE algorithm for the type of applications being considered is the resolution at which it can distinguish neighboring frequencies. For instruments that produce

notes with discrete fundamental frequencies, the resolution only needs to be 12 notes per octave, i.e., an error tolerance of  $\pm 0.5 ({}^{12}\sqrt{2} - 1)$ , or  $\pm 3\%$ . A higher resolution is necessary when small and continuous changes of pitch in time, such as that resulting from the execution of vibrato, glissando, and pitch bends, need to be detected. Some FFE algorithms have lower recognition resolution inherently because they subdivide the frequency range into bins at linear intervals.

### II. PREVIOUS WORK

Many existing FFE algorithms for musical signals generate estimates reliably when the signal segment analyzed is sufficiently long. *Frequency-domain* FFE algorithms operate by performing spectral analysis on sequential segments of the signal and applying a pattern matching technique to the spectrum to determine each segment's fundamental frequency. Brown [2] computes the cross-correlation of the constant- $Q$  transform of a segment of the signal with a fixed comb pattern. The calculation of the constant- $Q$  transform and a fast algorithm for approximating it are considered in [3] and [4], respectively.

The accuracy of a frequency-domain FFE algorithm depends on the quality of its spectral analysis step. Since the constant- $Q$  transform and the Fourier transform actually compute the spectrum of the *periodic extension* of a signal segment, a window function must first be applied to minimize spectral leakage due to discontinuity at the ends of the observation interval (see, for example, [5]). Applying such a window function to a signal segment with low fundamental frequency that already contains very few samples removes much information at the two ends of the window which is useful in determining its spectrum. The FFE algorithm described in this paper avoids this problem by using a rectangular window function.

An alternative approach for designing FFE algorithms is based on computing an autocorrelation between the waveform and a delayed version of itself and determining the fundamental frequency by maximizing the degree of their similarity. Cook *et al.* [1] use a least mean square adaptive algorithm to determine the coefficients of a filter that predicts a segment of a signal from an earlier segment. The phase of the filter is computed from these coefficients, which is then used to estimate the period.

A hybrid technique, described in [6], first determines a coarse estimate of the fundamental frequency using a frequency-domain algorithm. The phase change of the component closest in frequency to the coarse estimate between two segments of the signal separated by one sample is then used to estimate the fundamental frequency accurately.

The effect of segment lengths of signals on the accuracy of FFE algorithms is studied in [7]. A dynamic programming algorithm is described for matching harmonics to peaks in the constant- $Q$  transform of the signal. The result is an FFE algorithm capable of correctly handling short signal segments. However, since it uses the constant- $Q$  transform, it has the same disadvantages of frequency-domain algorithms described above, namely, that much information about the spectrum is discarded by the application of the window function. A modification to the autocorrelation-based algorithm in [1] that enhances its performance in real time is also considered in [7].

Central to the FFE algorithm introduced in this paper is a spectral analysis method based on least-square fitting. It will be described in the next section. Two properties of the least-square fitting are crucial for the spectral analysis algorithm: one that allows the sinusoidal components of the input signal to be identified and one that allows

Manuscript received July 12, 1995; revised September 16, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dennis R. Morgan.

A. Choi is with Department of Computer Science, University of Hong Kong, Hong Kong (e-mail: choi@cs.hku.hk).

Publisher Item Identifier S 1063-6676(97)01902-0.

<sup>1</sup>MIDI is a standard protocol for communication among synthesizers.

the algorithm to be implemented efficiently. Section IV describes the FFE algorithm and analyzes its computation time. The experimental results of its application to real signals are presented in Section V.

### III. A LEAST-SQUARE SPECTRAL ANALYSIS ALGORITHM

Let the discrete signal segment to be analyzed be denoted by  $x(n)$ ,  $n = 1, \dots, N$ . In general, this signal can be composed of a number of harmonic components. Suppose a sinusoidal function of the form  $\hat{x}(n) = a \sin(\omega n) + b \cos(\omega n)$  is to be matched to  $x(n)$ . The frequency of the sinusoid is  $\omega$ , given in radians per sample, i.e.,  $\omega = 2\pi f/s$ , where  $f$  and  $s$  are the frequency of the sinusoid in Hz and the sampling frequency of  $x(n)$  in samples per second, respectively. The values of  $a$  and  $b$  determine the amplitude and phase of  $\hat{x}(n)$ . The square error of the match, a function of  $\omega$ ,  $a$ , and  $b$ , is given by  $e = \sum_{n=1}^N (\hat{x}(n) - x(n))^2$ . For a given value of  $\omega$ , a pair of values can be chosen for  $a$  and  $b$  to minimize the square error of the match by setting

$$\frac{\partial e}{\partial a} = a \sum_{n=1}^N \sin(\omega n) \sin(\omega n) + b \sum_{n=1}^N \cos(\omega n) \sin(\omega n) - \sum_{n=1}^N x(n) \sin(\omega n) = 0$$

$$\frac{\partial e}{\partial b} = a \sum_{n=1}^N \sin(\omega n) \cos(\omega n) + b \sum_{n=1}^N \cos(\omega n) \cos(\omega n) - \sum_{n=1}^N x(n) \cos(\omega n) = 0.$$

The solution to this pair of equations is given by  $a^* = (QX - RW)/(PR - Q^2)$  and  $b^* = (QW - PX)/(PR - Q^2)$ , where

$$P = \sum_{n=1}^N \sin(\omega n) \sin(\omega n)$$

$$Q = \sum_{n=1}^N \cos(\omega n) \sin(\omega n)$$

$$R = \sum_{n=1}^N \cos(\omega n) \cos(\omega n)$$

$$W = - \sum_{n=1}^N x(n) \sin(\omega n)$$

$$X = - \sum_{n=1}^N x(n) \cos(\omega n).$$

Let  $e^*(\omega)$  be the value of  $e$  when  $a$  and  $b$  are set to  $a^*$  and  $b^*$ , respectively, for a given value of  $\omega$ . In other words,  $e^*(\omega)$  is the minimum square error of matching a sinusoid of frequency  $\omega$  to the signal segment  $x(n)$  over all possible amplitudes and phases for such sinusoids.

The problem can be restated in vector form as one of minimizing  $e = \|\mathbf{x} - A\mathbf{x}'\|^2$  over all possible values of the complex variable  $A$ , where  $x'(k) = \exp(j\omega k)$ . The value of  $e$  is minimized when  $A = \langle \mathbf{x}, \mathbf{x}' \rangle / \|\mathbf{x}'\|^2$ , and this minimum value is given by  $e^*(\omega) = \|\mathbf{x}\|^2 - |A|^2 \|\mathbf{x}'\|^2$ . Thus, the value of  $e^*(\omega)$  depends inversely on the square of the magnitude of  $A$ , which is related by a constant to  $\langle \mathbf{x}, \mathbf{x}' \rangle$ , the discrete time Fourier transform (DTFT) of  $\mathbf{x}$ . A more exact characterization of the shape of  $e^*(\omega)$  allows the frequencies of the sinusoidal components of the signal segment to be computed

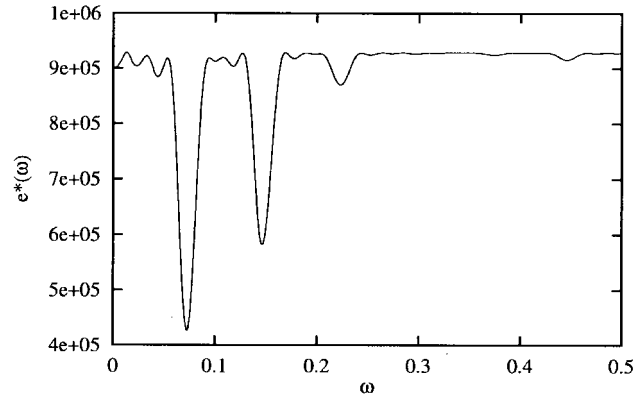


Fig. 1. Plot of  $e^*(\omega)$  for a 300-sample segment of a C4 note for  $0 \leq \omega \leq 0.5$ .

more efficiently and accurately than using the fast Fourier transform (FFT) algorithm alone.

Fig. 1 shows a plot of the function  $e^*(\omega)$  for a typical 300-sample signal segment of a C4 note sampled from an electric guitar in the range  $0 \leq \omega \leq 0.5$ ; the function is relatively flat for  $\omega > 0.5$ . This figure illustrates two properties of the shape of this function, which will be stated without proof since they are needed in the description of the spectral analysis algorithm. Their proofs can be found in [8].

#### A. Property 1

Each “significant” trough in the function  $e^*(\omega)$  corresponds to a sinusoidal component of the signal segment  $x(n)$ . The value of  $\omega$  of the minimum point at the bottom of a trough is equal to the frequency of the corresponding sinusoidal component.

In Fig. 1, the three deepest troughs reach their minimum values at  $\omega = 0.074$ ,  $\omega = 0.147$ , and  $\omega = 0.224$ , respectively. Since the sampling frequency is 22 255 samples per second, these correspond to sinusoidal components with frequencies 262.10 Hz, 520.67 Hz, and 793.41 Hz, respectively. The frequency of C4 is 261.63 Hz, and these are therefore approximately the fundamental, second harmonic, and third harmonic of the C4 note, respectively.

#### B. Property 2

The “width” of each significant trough in the function  $e^*(\omega)$  is at least  $2\pi/N$ , and this width is independent of the frequencies of the sinusoidal components of the input signal, provided that these are located sufficiently far apart from each other.

The value of  $2\pi/N$  is not surprising since it is the bin width for an FFT of length  $N$ . In the example, since the value of  $N$  is 300, the width of each significant trough will be greater than 0.0209. Since musical signals contain harmonic components whose frequencies are integer multiples of the fundamental frequency, the troughs corresponding to them are regularly spaced in  $\omega$ . Troughs will be separated from each other if the fundamental frequency is somewhat larger than  $2\pi/N$ , which corresponds to 74.18 Hz for  $N = 300$  provided  $S = 22\,255$  samples per second.

Since the fundamental frequency can occur at any value of  $\omega$ , it is impractical to evaluate the function  $e^*(\omega)$  at a large number of values of  $\omega$  in order to identify the troughs. However, Property 2 makes it necessary to compute  $e^*(\omega)$  only at values of  $\omega$  that are evenly spaced at a distance of  $2\pi/(3N)$  apart. Doing so guarantees that at least three consecutive values of  $\omega$  will “fall into” each trough, where the function value in the middle is smaller than those of its two neighbors. An additional test is used to eliminate troughs that are too shallow: a simple test is to admit only troughs corresponding

**Algorithm 1**

1. Divide the frequency range to be analyzed, say  $[0, 0.5]$ , into  $M = 0.5/(2\pi/(3N))$  intervals by choosing  $\omega_i = 2\pi i/(3N)$  for  $i = 0, 1, \dots, M$ .
2. Compute  $e^*(\omega_0), e^*(\omega_1), \dots, e^*(\omega_M)$ . Identify those values of  $\omega_i$  for which  $e^*(\omega_{i-1}) > e^*(\omega_i) < e^*(\omega_{i+1})$ .
3. Eliminate troughs that are too shallow by selecting only those intervals in step 2 for which the function value at the middle point is significantly smaller than those of its two neighbors, e.g.,  $0.9e^*(\omega_{i-1}) > e^*(\omega_i) < 0.9e^*(\omega_{i+1})$ .
4. For each interval selected in step 3, apply the golden ratio search (or successive parabolic interpolation) to locate the value of  $\omega$  of the minimum in the trough by searching the interval  $[\omega_{i-1}, \omega_{i+1}]$ .

Fig. 2. The least-square spectral analysis algorithm.

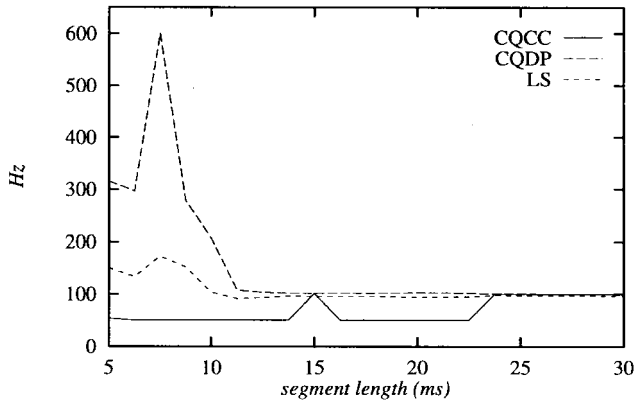


Fig. 3. Fundamental frequency estimates for a G2 note.

to points whose function values are less than 90% of those of their two neighbors. For our application the highest note to be recognized is G5 (784Hz). Experiments show that accurate estimates can still be obtained if the calculations are limited to the range  $0 \leq \omega \leq 0.5$ , corresponding to 0 to 1771 Hz. For  $N = 300$ ,  $e^*(\omega)$  must therefore be evaluated at  $0.5 / (2\pi/(3N)) \approx 72$  points.

Since the value of  $e^*(\omega)$  within each trough is unimodal, once a trough is detected, the golden ratio search or the successive parabolic interpolation [9] can be used to obtain the value of  $\omega$  at which the minimum occurs. Either technique converges on the minimum value by successively reducing the interval in which it is known to lie. Although they are not necessary in the FFE algorithm below, the amplitudes of the sinusoidal components can be obtained at this time by solving a linear optimization problem. The resulting spectral analysis algorithm, based on least-square fitting, will be referred to as Algorithm 1 below and is summarized in Fig. 2.

**IV. COMPUTATION TIME ANALYSIS**

After the frequencies of the minimum points of the deepest troughs in the error function  $e^*(\omega)$  have been obtained, the fundamental frequency is estimated as follows. Let  $\omega_1, \omega_2, \dots, \omega_k$  be these frequencies, where  $\omega_1 < \omega_2 < \dots < \omega_k$ . In our experiments, only the three deepest troughs are used (these are identified by the three smallest values of  $e^*(\omega)$ ). Therefore,  $k \leq 3$ . When the fundamental frequency of the signal is low, the trough in  $e^*(\omega)$  that corresponds to the fundamental may be distorted near  $\omega = 0$  and Algorithm 1 may fail to detect it. Assuming that the harmonic components corresponding to the fundamental may or may not be detected, two cases must be distinguished: that  $\omega_1, \omega_2, \dots, \omega_k$  correspond to i)

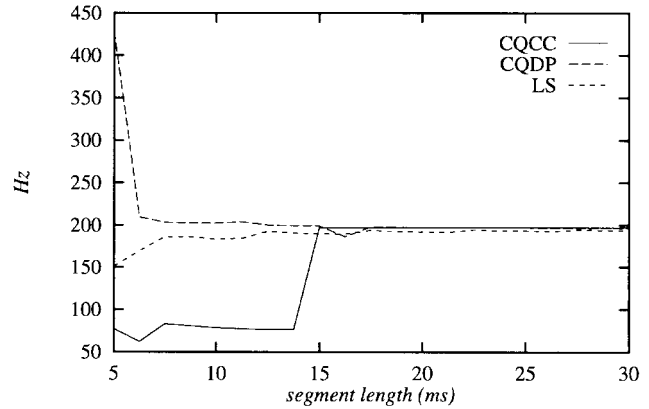


Fig. 4. Fundamental frequency estimates for a G3 note.

the fundamental, second, third, ..., and  $k$ th harmonics of the signal, respectively; or ii) the second, third, ...,  $k$ th, and  $(k+1)$ th harmonics of the signal, respectively. In the first case, the estimate of the fundamental frequency is the average of  $\omega_i/i$  over  $i = 1, 2, \dots, k$  and in the second case, it is the average of  $\omega_i/(i+1)$  over  $i = 1, 2, \dots, k$ . The estimate corresponding to the set of values with a smaller standard deviation is reported by the FFE algorithm, since the standard deviation measures the consistency among the troughs in estimating the fundamental frequency. This scheme avoids octave errors for input signal segments with simple frequency spectrums, as for those used in the experiment.

The response time of the FFE algorithm is the sum of the length of the initial signal segment analyzed and its computation time. Since it uses the simple algorithm described above for estimating fundamental frequencies, it spends the majority of its computation time on spectral analysis. The spectral analysis algorithm (Algorithm 1 in Fig. 2) in turn spends most of its time on computing  $e^*(\omega)$  for the different values of  $\omega$  in Steps 2 and 4. Instructions for implementing the loops and tests in Algorithm 1 and for generating the fundamental frequency estimates require a negligible amount of time relative to that for evaluating  $e^*(\omega)$  and will be ignored in the following analysis. For a given value of  $\omega$ , the steps for computing  $e^*(\omega)$  require the numbers of instruction cycles tabulated in Table I.

In the table,  $t$  denotes the number of instruction cycles needed for computing the sine or cosine function and  $d$  denotes that for a division. The formulas given in the table assume that the target DSP processor is capable of single-cycle multiply-and-accumulate operations. An example of such a processor is the Analog Devices ADSP-2100. For this processor,  $t = 25$  and  $d = 33$ .

Since the values of  $\sin(\omega n)$  and  $\cos(\omega n)$ , for  $k = 1, 2, \dots, N$  and  $P, Q$ , and  $R$  depend only on  $\omega$  and not on  $x(n)$ , Step 2 of Algorithm 1 evaluates  $e^*(\omega)$  for the same  $M$  values of  $\omega$  for any signal segment analyzed. These values can thus be precomputed and stored in a table. The evaluation of  $e^*(\omega)$  for each value of  $\omega$  then requires only the computation of (iii), (iv), and (v) in Table I, which takes  $8N + 2d + 9$  instruction cycles. Therefore the entire Step 2 of Algorithm 1 requires  $M(8N + 2d + 9)$  instruction cycles. Alternatively, Step 2 can be computed using an FFT algorithm.

In Step 4 of Algorithm 1,  $e^*(\omega)$  is evaluated for different values of  $\omega$  that depend on the fundamental frequency of the given  $x(n)$ . Therefore, the evaluation of  $e^*(\omega)$  for a single value of  $\omega$  requires all five steps in Table I, or  $2Nt + 11N + 2d + 9$  instruction cycles. An iterative algorithm such as the golden ratio search or successive parabolic interpolation is used to find the minimums of the troughs. Since the troughs of the function  $e^*(\omega)$  can be closely approximated by quadratic curves, the successive parabolic interpolation technique

TABLE I  
INSTRUCTION CYCLES REQUIRED BY ALGORITHM 1 FOR EACH VALUE OF  $f$

	Values computed	Instruction cycles
(i)	$\sin(\omega n)$ and $\cos(\omega n)$ for $k = 1, 2, \dots, N$	$2Nt$
(ii)	$P$ , $Q$ , and $R$	$3N$
(iii)	$W$ and $X$	$2N$
(iv)	$a^*$ and $b^*$	$2d + 9$
(v)	$e^*(f)$	$6N$

can locate the minimum value of a trough with high accuracy in only a few iterations. The computation of  $e^*(\omega)$  in Step 4 for all the different values of  $\omega$  requires a total of  $L(2Nt + 11N + 2d + 9)$  instruction cycles, where  $L$  is the product of the number of troughs examined and the number of iterations for each trough. A typical value for  $L$  is 12.

The total number of instruction cycles required for the evaluation of  $e^*(\omega)$  for all the different values of  $\omega$  in Algorithm 1 is therefore  $M(8N + 2d + 9) + L(2Nt + 11N + 2d + 9)$ . When  $N = 300$ ,  $M = 72$ ,  $d = 33$ ,  $t = 25$ , and  $L = 12$ , a total of 398 700 instruction cycles are needed. A 30-MIPS DSP processor will require 13.29 ms to process such a signal segment. If  $N$  is reduced to 200, the same processor requires 7.59 ms to process each signal segment. Note that this estimate does not take into account the instructions necessary for loop maintenance. The actual running time will be somewhat longer.

## V. EXPERIMENTS

A set of experiments was performed to study the real-time performance of the new FFE algorithm. The test inputs were sampled from an electric guitar at a sampling rate of 22 255 samples per second and the notes G2 (98.1 Hz), G3 (196.0 Hz), G4 (392.0 Hz), and G5 (784.0 Hz) were used. These notes range over most of the playing range of the guitar and have lower frequencies than test inputs used by previous papers on FFE. The results shown are representative of the results of repeated runs of the experiment and for notes with different fundamental frequencies.

Three FFE algorithms were tested: algorithm CQCC, which is based on the constant- $Q$  transform and cross-correlation [2], algorithm CQDP, which is based on the constant- $Q$  transform and dynamic programming [7], and algorithm LS, the FFE algorithm introduced in this paper, which performs spectral analysis by Algorithm 1 in Fig. 2 and FFE by the simple decision algorithm described in the previous section. The "comb" pattern for CQCC consists of eight components, which was chosen to optimize its performance.

These algorithms were implemented in C++ in UNIX and did not run in real time. The beginning of the signal was identified automatically by a simple threshold algorithm on the power in an initial segment of the signal.

The three FFE algorithms were applied to the initial segments of the sampled notes with lengths between 5 and 30 ms, in increments of 1.25 ms. Figs. 3, 4, 5, and 6 plot the fundamental frequency estimates reported by these algorithms as functions of the lengths of the initial segments analyzed, for notes G2, G3, G4, and G5, respectively.

For notes with low fundamental frequencies, represented by G2, algorithm LS performs better than algorithm CQDP by requiring a slightly shorter initial segment to achieve an accurate estimate. Its accuracy for segments that are even shorter is also higher and it is more stable than algorithm CQDP (i. e., the estimates do not fluctuate as much). Both algorithm LS and algorithm CQDP require a much shorter initial segment (by 15 ms) than algorithm CQCC to obtain accurate estimates.

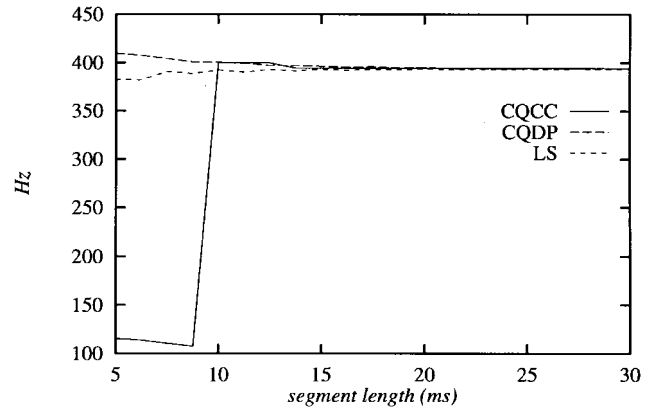


Fig. 5. Fundamental frequency estimates for a G4 note.

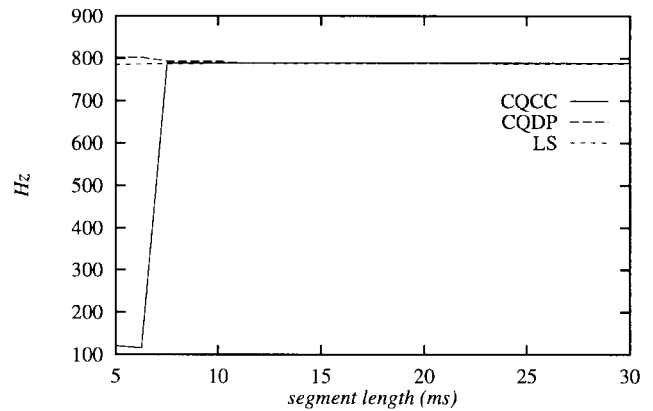


Fig. 6. Fundamental frequency estimates for a G5 note.

For notes with intermediate fundamental frequencies, represented by G3, both algorithm LS and algorithm CQDP converge to correct estimates of the fundamental frequency when the length of the initial signal segment is greater than 7.5 ms. In fact, algorithm CQDP produces an estimate with a slightly smaller absolute error than that produced by algorithm LS for the 6.25-ms initial signal segment. However, the absolute errors of algorithm LS are much smaller than those of algorithm CQDP for initial signal segment lengths of 5 ms or smaller. Also, a fluctuation of the estimate occurs at 16.25 ms for algorithm CQDP, due to its less stable nature. Both algorithms require initial signal segments that are about 5 ms shorter than those required by algorithm CQCC to obtain accurate estimates.

The plots for G4 and G5, representative of notes with higher fundamental frequencies, show that algorithm LS converges somewhat more quickly than algorithm CQDP as the length of the signal segment increases. All three FFE algorithms perform well when the initial signal segments are longer than 10 ms in length for these notes with high fundamental frequencies.

## VI. SUMMARY

A spectral analysis algorithm based on least-square fitting was described. It performs its function by applying a rectangular window function so that it is capable of correctly analyzing relatively shorter signal segments. The algorithm minimizes the square error of fitting a sinusoid to the signal segment. Two properties of the error function of the least-square fitting allow the frequencies of the sinusoidal components of the signal segment to be determined and the troughs to be detected by evaluating the error function at a small, evenly-

spaced set of frequencies. An FFE algorithm based on this spectral analysis algorithm was also described. Since it has higher accuracy than previous FFE algorithms for short signal segments, it is more suitable for use in real time. The computation time for this FFE algorithm was analyzed. Experimental results that demonstrate the real-time performance of the algorithm were also described.

#### REFERENCES

- [1] P. Cook, D. Morrill, and J. Smith, "A MIDI control and performance system for brass instruments," in *Proc. ICMC 1993*, Tokyo, Japan, pp. 130–133.
- [2] J. Brown, "Musical fundamental frequency tracking using a pattern recognition method," *J. Acoust. Soc. Amer.*, vol. 92, pp. 1394–1402, Sept. 1992.
- [3] ———, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, pp. 425–434, Jan. 1991.
- [4] J. Brown and M. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *J. Acoust. Soc. Amer.*, vol. 92, pp. 2698–2701, Nov. 1992.
- [5] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, pp. 51–83, Jan. 1978.
- [6] J. Brown and M. Puckette, "A high resolution fundamental frequency determination based on phase changes of the fourier transform," *J. Acoust. Soc. Amer.*, vol. 94, pp. 662–667, Aug. 1993.
- [7] A. Choi, "On the improvement of the real-time performance of two fundamental frequency recognition algorithms," in *Proc. 1st Brazilian Symp. Computer Music*, Caxambu, MG, Brazil, Aug. 1994, pp. 27–32.
- [8] ———, "Real-time fundamental frequency estimation by least-square fitting.," Dept. Comput. Sci., Univ. Hong Kong, Tech. Rep. TR-95-03, June 1995.
- [9] G. Forsythe, M. Malcolm, and C. Moler, *Computer Methods for Mathematical Computations*. Englewood Cliffs, NJ: Prentice-Hall, 1977.