James W. Beauchamp, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA
Robert C. Maher, University of Nebraska-Lincoln, Lincoln, Nebraska, USA
Rebekah Brown, Indiana University, Bloomington, Indiana, USA

Presented at the 94th Convention 1993 March 16–19 Berlin

AES

AN AUDIO ENGINEERING SOCIETY PREPRINT

Detection of Musical Pitch from Recorded Solo Performances

James W. Beauchamp, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA
Robert C. Maher, University of Nebraska-Lincoln, Lincoln, Nebraska 68588-0511, USA
Rebekah Brown, Indiana University, Bloomington, Indiana 47405, USA

Our frequency-domain-based pitch detector graphs musical pitch as the log of detected fundamental frequency on a Unix computer. Recorded virtuosic solo performances are difficult to track when they contain frequency-obscuring reverberation, or when they are dry, since extraneous noises can then interfere. We tested our detector on dry and reverberated performance versions of two musical passages and found it to work quite well on dry versions, but performance was degraded on the reverberated versions. Also, we used the spectral and fundamental frequency data to resynthesize the recordings, with the advantage that most extraneous noises and reverberation were eliminated. The theory of the detector and possibilities for improving its performance are discussed.

1 System Overview

Our objective is to develop a tool for accurately detecting and graphing the fundamental frequency or musical pitch as a function of time for solo performance. Although this area has had a long history of activity [1-4], no circuit or computer algorithm has been shown to work in a completely reliable fashion, particularly for rapid passages recorded in either a studio or reverberant environment. Two applications are automatic music transcription and resynthesis with altered timbre, tempo, and pitch range. Also of interest are musicological applications where the detected output can be used to study timing and intonation of performances.

Rapid musical passages present peculiar problems since notes are often very short and not necessarily stable. In other words, a large proportion of each note may consist of attack and decay transients, where the pitch is moving towards or away from a stable value. With studio recordings, noises caused by various mechanical devices (such as violin bows and clarinet keys) and the performer (e.g., breath noises) can cause problems, whereas in a hall the predominant factor seems to be the echo of previous notes occurring simultaneously with the current one, causing a chord effect.

Our current solution to the problem is a program, implemented on a NeXT computer, which attempts to find the best match to a harmonic series within a time-variant spectrum at each time frame. The spectrum is found using an algorithm originally developed by McAulay and Quatieri [5] to determine spectral peaks which exceed a certain threshold. Then the fundamental frequency is estimated by a harmonic search and match method we call the *two-way mismatch* method. A special graphing program, which works with either Tek4014 or Postscript protocols,

is used to plot the log of the estimated fundamental frequency, so that it is displayed as a graph of musical note value (e.g., G#, Bb); in this way, the graph can be directly interpreted in terms of standard equal-tempered pitch.

Additional programs can take the spectrum and fundamental frequency data and produce a *harmonic analysis file*, which is essentially the same as the original time-varying spectrum file except that all non-harmonic information is removed. When this file is resynthesized using additive synthesis, the result is ideally not only free of noise but also, in the case of reverberated recordings, would appear to be void of echo.

Fig. 1 depicts the frequency detection and optional resynthesis process.

2 Time-Variant Spectrum Analysis

For each frame a Kaiser-windowed, zero-padded FFT is taken on the input signal. The windows are generally 512 samples long, the FFT 1024 samples, and the hop size is 128 samples. After high frequency emphasis is applied, the peaks above a certain threshold (given by the user) are identified, and their amplitudes and frequencies are refined by quadratic interpolation. More details for this general procedure are given in [5-8]. Our program for carrying out this procedure is called **mqan**. The result is a series of frames with a variable number of spectral components, called "peaks", stored in an *mq file*.

3 Fundamental Frequency Determination

The program **fcheck** is used to process an mq file to produce a file consisting of a list of fundamental frequencies (FFs), one for each frame (called a *ff file*). The user specifies a minimum and maximum FF for the search range and the number of harmonics to use for the subsequent error calculation. Then for each frame, fcheck scans through the search range of FFs and for each trial frequency performs a calculation called the *two-way mismatch error*. The frequency which produces the least error is the chosen FF value for that frame. The error calculation works as follows:

First, the amplitudes of all peaks in the current frame are normalized by the maximum amplitude (giving the set $\{a_k\}$). The user has already specified the number of harmonics N (typically 10). Then, for each harmonic n of the current trial FF, the difference Δf_{1n} between the harmonic's frequency and the nearest peak frequency is calculated. An error E_1 based on the $\{\Delta f_{1n}\}$, the $\{a_k\}$, and the harmonic frequencies is then calculated. This completes the *predicted-to-measured error* calculation. Next, for each peak in the frame, the difference Δf_{2n} between the peak's frequency and the nearest trial harmonic is calculated. An error E_2 based on the $\{\Delta f_{2n}\}$, the $\{a_k\}$, and the peak frequencies is then calculated. This completes the *measured-to-predicted error* calculation. The total error is then $E = w_1E_1 + w_2E_2$, a function of the trial FF. The weights w_1 and w_2, as well as various constants in the formulas for E_1 and E_2, are chosen by trial and error to give good results over a variety of test signals.

The FF search algorithm for finding the minimum value of E proceeds in two stages: First, the entire FF search range is scanned by semitone (5.9%) increments, and the local minima are noted. Second, a much finer step size is used in the region of each local minimum to find the true global minimum and to improve the accuracy of the FF detection. The limit of this refinement can be set to converge the final FF within some arbitrary figure such as 1 Hz. We find it necessary to perform a detailed search of most local minima found in the first stage, rather than just the global minimum, because the depths of minima can not be absolutely determined from results of the first, coarse increment stage.

Figures 2 and 3 graphically depict the two-way mismatch algorithm. Fig. 2 shows how the trial (predicted) harmonics are matched to the measured spectral peaks (tracks). Fig. 3 shows how the measured peaks are matched to the trial harmonics.

Fig. 4 shows two typical curves of error E as a function of trial FF. For the first curve, the decision is clear cut. However, the decision is not so obvious for the second curve, where it could be said that there are two possible candidates for the "best" FF value. However, in our current implementation, we take the FF corresponding to the lowest value of E.

The two-way mismatch method is designed to identify the best harmonic sequence of peaks, based on minimizing the mismatch error. If a frame contains peaks due to extraneous or non-harmonic sources, the performance of the pitch detector will be degraded since the signal no longer matches the basic assumption on which the mismatch error method is based, i.e., that the signal consists solely of harmonic partials.

4 Solo Passage Results

Two passages were tested: 1) the first 8 bars (87 notes, E_4 to E_6) of the *Partita III* for Unaccompanied Violin by J. S. Bach; and 2) a 2 bar ($\approx$ 22 notes, Db_5 to Db_6) fragment from the third movement (*Abîme des oiseaux*) of Oliver Messiaen's *Quatuor pour la fin du temps* for unaccompanied clarinet.

We denote p as *musical pitch*, which can be defined in terms of frequency f using

$$p = 9 + 12 \log(f/27.5)/\log(2) .$$

Continuous values of p are graphed vs. time, and its integer values are identified by corresponding musical notes on the vertical axis. Note that the frequency for C_0 corresponds to p = 0, for $C\#_0$ we have p = 1, and that middle C (C_4), which is f = 261.6 Hz, corresponds to p = 48.

For the *Partita* we tested three versions. The first was generated by a computer using a waveform consisting of 8 equal-amplitude harmonics and an amplitude envelope having .02 s rise and fall times. For this signal, the FFs were perfect equal-tempered frequencies where

$$f = 27.5 \times 2^{(p-9)/12} .$$

The accuracy of the overall pitch detector system for this "ideal" signal is demonstrated in Fig. 5. Only tiny variations from perfection occur.

Fig. 6 shows the result for a studio violin recording, performed by a very competent violinist. All of the notes are there, but the graph is obscured by glitches which occur between notes. (These may be caused by the scrapes of the bow on the string.) Many of the glitches are very short and could be removed by a subsequent deglitching operation. This operation would be based on the assumption that a frequency for a given frame is not valid if it is sufficiently different than the previous frame and stays different for only 1 or 2 frames. To remove the glitch, the frequencies of these frames could be replaced by the frequency of the frame preceding the glitch.

Fig. 7 shows the result for a reverberated CD performance by a well known violinist. Most of the 87 notes register correctly, but a good number (around 15) are missed. Glitches between notes are not as common (one suspects because the reverberation smooths transitions), but now glitches occur even during notes which register, while some notes (notably the bottom E4s) are missed altogether. Clearly reverberation makes the pitch detection task more difficult for this example. Tweeking of the mismatch error function or aspects of the algorithm might yield substantial improvement, but we did not do that in this case.

All notes register in the detection of a studio recording of a clarinet performing the Messiaen fragment, as shown in Fig. 8. However, there are occasional glitches just before and just after rests in the score, which possibly could be eliminated based on the weak amplitudes that occur at those points.

As shown in Fig. 9, the result is surprisingly good for a reverberated LP performance of the Messiaen fragment, but in this case we tuned the detection algorithm a bit. All notes seem to register, but there are a couple of extra little "blips" and the very rapid grace notes are rather obscure. Also, pitches linger on through the rests, not necessarily inappropriately, due to the reverberation.

5 Synthesis from Spectral Peak and Fundamental Frequency Data

The program **harmformat** is used to process spectral peak data produced by mqan, as guided by the FF data produced by fcheck, to produce a harmonic analysis file consisting of harmonic amplitudes and frequencies for consecutive frames. This can then be further processed by an additive synthesis program **addsyn** (see [8] for a more detailed description) to create a sound file.

Audition of the addsyn-produced sound file, in comparison to the original, provides useful insight into the quality of the pitch tracking. In many cases, the glitches which appear so prominent in the musical pitch vs. time plots are hardly noticeable in the synthesis, probably because the glitches occur at low amplitude or are extremely short.

In the case of studio recordings, the synthesized result is, for the most part, much cleaner than the original; low level bow scrapes and other extraneous noises are eliminated. The result may not be as natural-sounding as the original, but this could be improved by adding reverberation. For

our two studio recordings all notes seem to be present. Most glitches are not audible, except for some violin scrapes that are quite loud (but still acceptable to the violin aficionado) in the original recording.

In the case of hall recordings, the sound overlay (chord) effect of the reverberation is removed. However, again, the effect is not entirely natural, as the original amplitude envelopes of the source instruments have been corrupted by the reverberation effect. Also, unless the FF detection algorithm is carefully adjusted, some synthesized notes may be incorrect.

6 Conclusions and Extensions

The computer algorithms employed for musical pitch detection of solo passages described in this paper produce promising results which in the future may be substantially improved with relatively simple changes to the code. As it is, for studio recordings, we achieved close to 100% "hits" (notes correctly detected) . The number of "false alarms" (apparent notes or glitches that do not belong) which occurred in the pitch vs. time graphs is not acceptable, but we suspect that many of these can be removed either on the basis of brevity or low amplitude. For reverberated samples, we only achieved a high hit-to-false alarm ratio when the algorithm was carefully adjusted.

Assuming that the score is known in advance (or that the pitches are otherwise determined), the pitch vs. time graphs can be a useful tool for determining timing and accurate pitch data. A useful extension to the display program would be the ability to "zoom in" on an individual note or group of notes in order to get more accurate readings. One of the present authors (Rebekah Brown) is focusing on determining the precise intonation of each note in rapidly performed passages. For this project, she has been using a sound editor on the NeXT computer to segment individual notes into separate files, and then she applies a pitch-synchronous phase vocoder program (see [8] for details) to achieve a more accurate, glitchless, determination of frequency vs. time information. The use of phase to determine frequency is inherently more accurate than spectral peak detection, but it does require that an approximate value for the frequency is known in advance.

It may not be possible to provide definitive measurements of the fundamental frequency in highly transient situations. A reasonable guess is that the method described in this article is limited to 0.5% accuracy in real situations. It appears that the phase vocoder method, as applied to individual notes, can give values which are accurate to 0.05% for an "ideal" input waveform, when frames in the interior of notes are averaged. For real sounds, the phase vocoder can give an accurate average value, provided an approximate frequency is known in advance. We hope in the future to do further testing of the pitch detection accuracy using synthetic input signals which more closely mimic acoustically-generated signals.

Another possible improvement to the detection process would be to utilize frame-to-frame frequency tracking. This is one aspect of the McAulay-Quatieri (MQ) algorithm which we are not currently utilizing. While it is true that an individual MQ track does not always correspond to the same harmonic throughout the life of the track, the tracks could be used to help decide which of several candidate local minima in the mismatch error function is most likely to be the correct one.

Acknowledgements

We wish to thank Christopher Kriese for writing the **pitchit** program, which performs graphic display of the musical pitch vs. time data. This work was supported, in part, by the Research Board of the University of Illinois at Urbana-Champaign and by the Research Council and Department of Electrical Engineering at the University of Nebraska-Lincoln.

8 References

[1] P. A. Tove, L. Ejdesjö, and A. Svärdström, "Frequency and Time Analysis of Polyphonic Music", *J. Acoust. Soc. Am.*, vol. 41, no. 5, pp. 1265-1271 (1967).

[2] M. Piszczalski and B. A. Galler, "Predicting musical pitch from component frequency ratios", *J. Acoust. Soc. Am.*, vol. 66, no. 3, pp. 710-720 (1979).

[3] B. Doval and X. Rodet, "Estimation of Fundamental Frequency of Musical Signals", *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 3657-3660 (1991).

[4] J. C. Brown, "Musical fundamental frequency tracking using a pattern recognition method", *J. Acoust. Soc. Am.*, vol. 92, no. 3, pp. 1394-1402 (1992).

[5] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 744-754 (1986).

[6] J. O. Smith and X. Serra, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation", *Proc. 1987 Int. Computer Music Conf.*, pp. 290-297, Int. Computer Music Assn., San Francisco (1987).

[7] R. Maher and J. Beauchamp, "An Investigation of Vocal Vibrato for Synthesis", *Applied Acoustics*, vol. 30, pp. 219-245 (1990).

[8] J. W. Beauchamp, "Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds", *Audio Engineering Soc. Preprint* (1993).

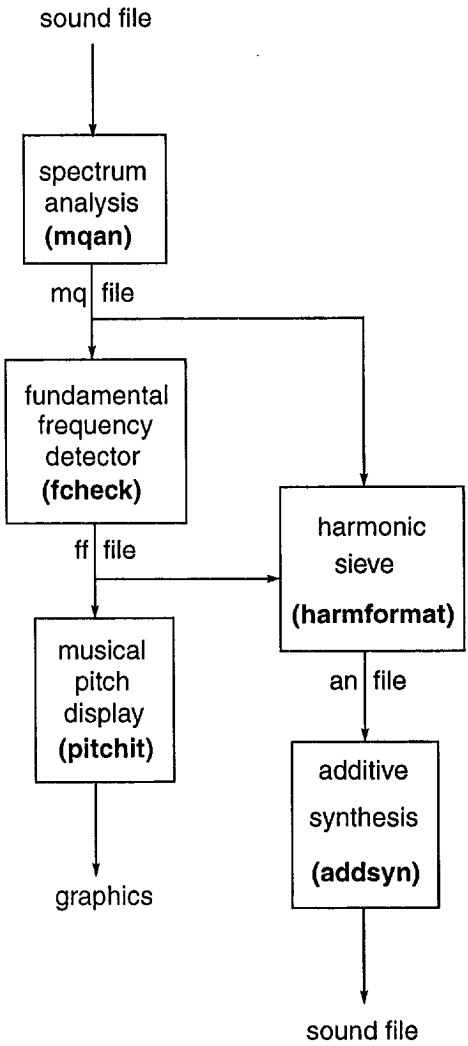


Fig. 1 Musical pitch detection and harmonic resynthesis system.

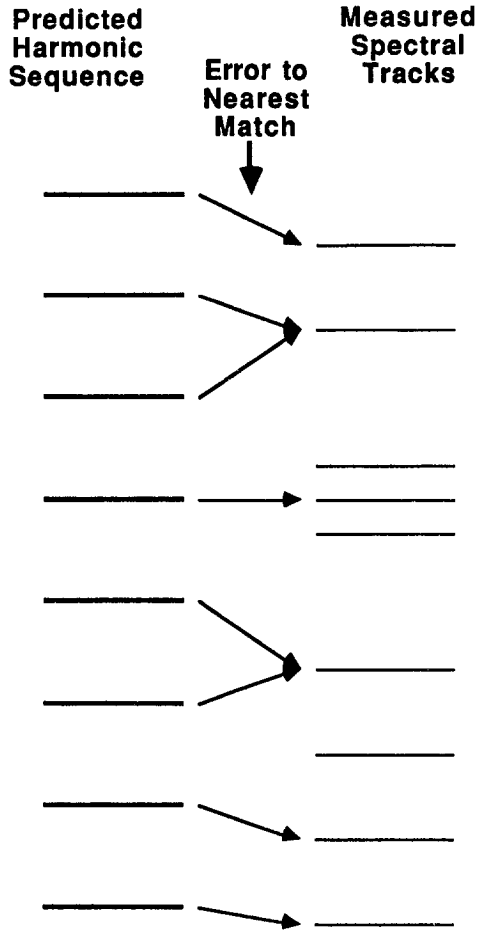


Fig. 2 Two-way mismatch method: The predicted-to-measured error calculation.

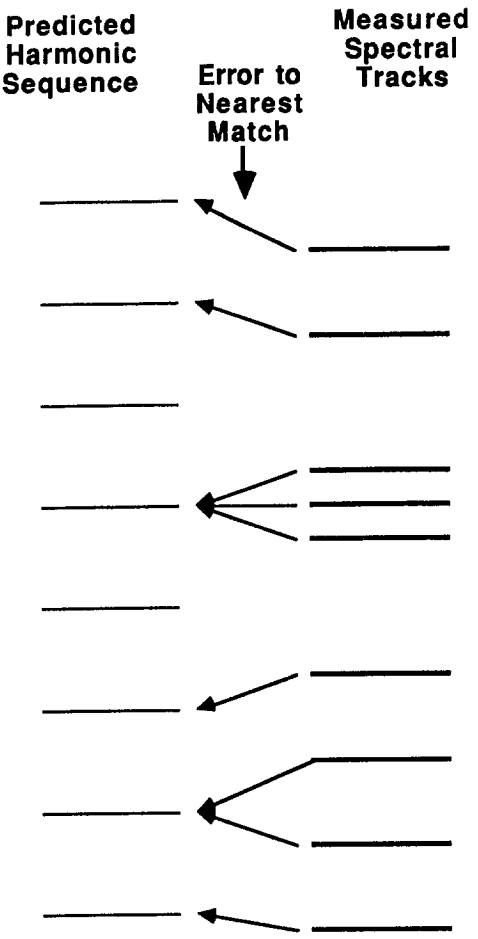
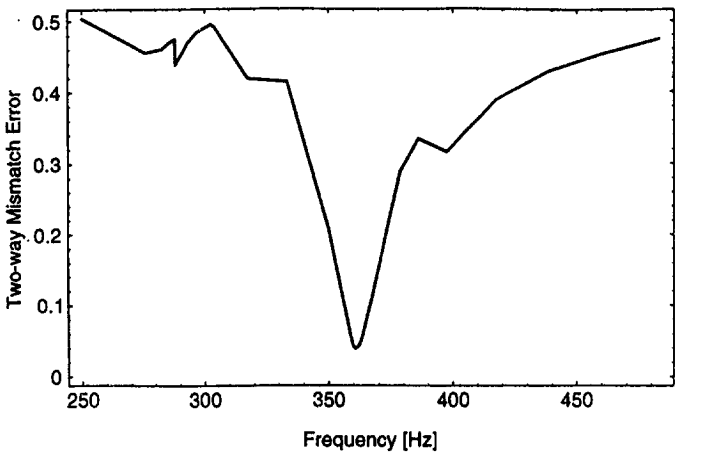


Fig. 3 Two-way mismatch: The measured-to-predicted error calculation.



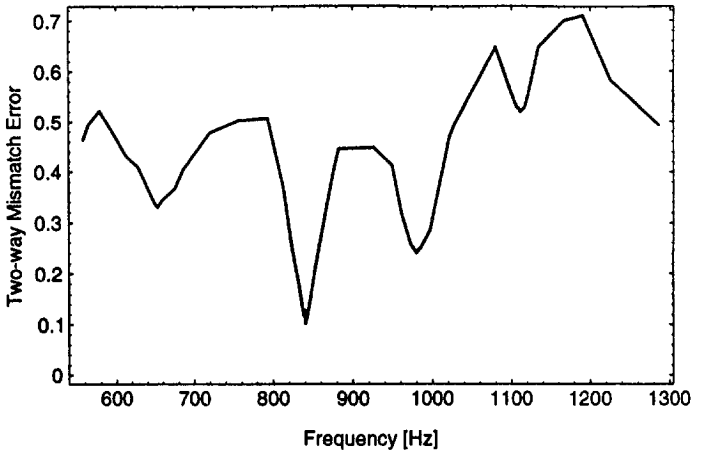


Fig. 4 Two examples of Two-Way Mismatch Error vs. Trial Fundamental Frequency functions. The selected fundamental frequency corresponds to the minimum value of the error.

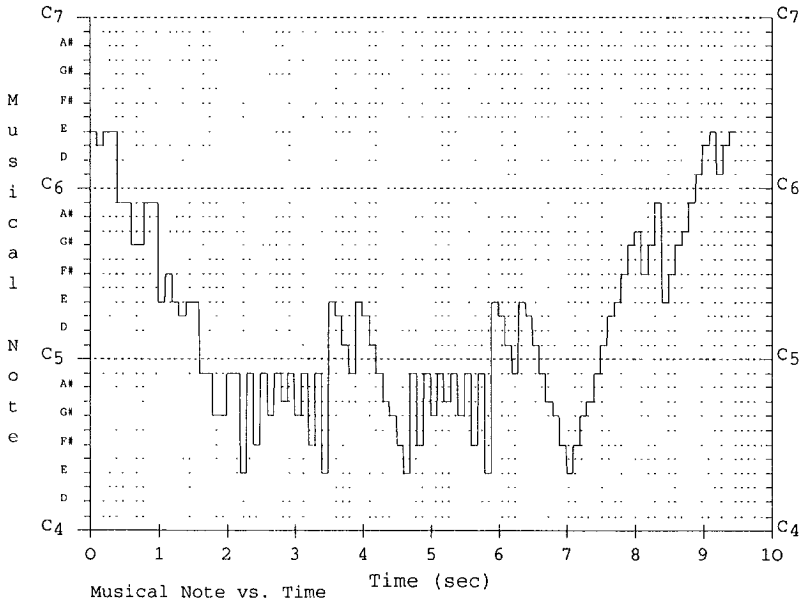


Fig. 5 Musical Note value vs. time for pitch detection of a computer-generated input signal synthesized with perfect equal-tempered frequencies: first 8 bars of Bach's *Partita III*.

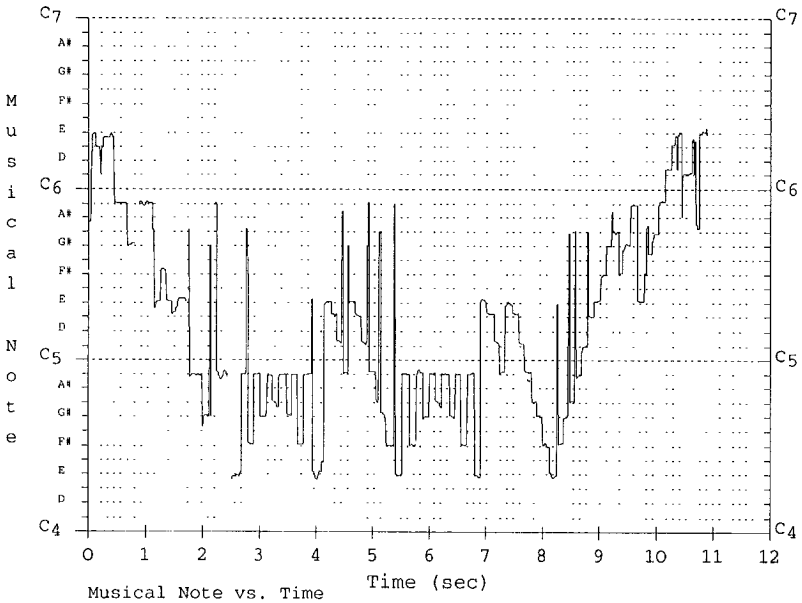


Fig. 6 Musical Note value vs. time for studio recording of violin performance of first 8 bars of Bach's *Partita III*.

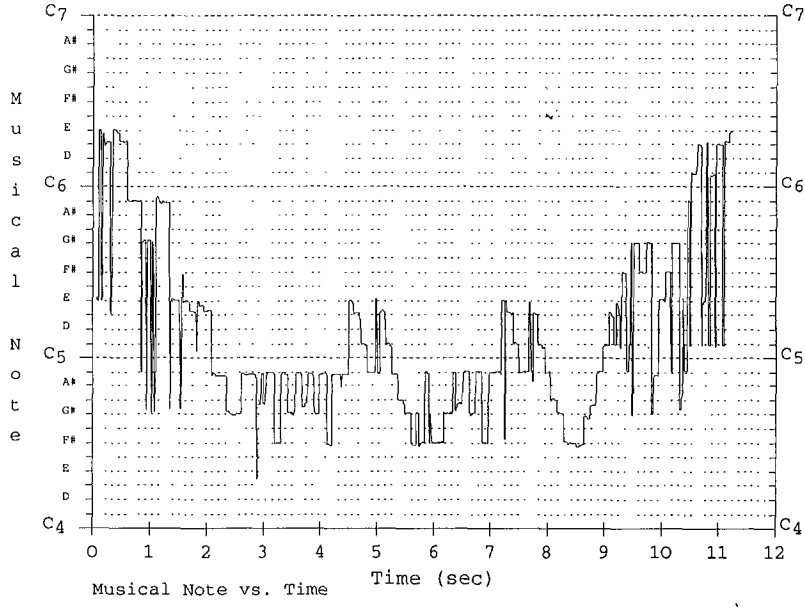


Fig. 7 Musical Note value vs. time for reverberated recording of violin performance of first 8 bars of Bach's *Partita III*.

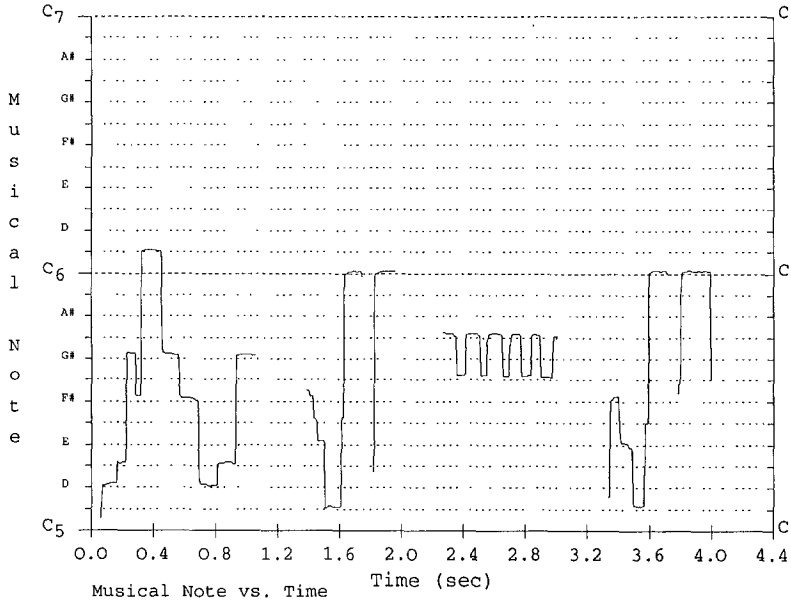


Fig. 8 Musical Note value vs. time for studio recording of clarinet performance of 2 bar fragment from Messiaen's *Quatuor pour la fin du temps.*

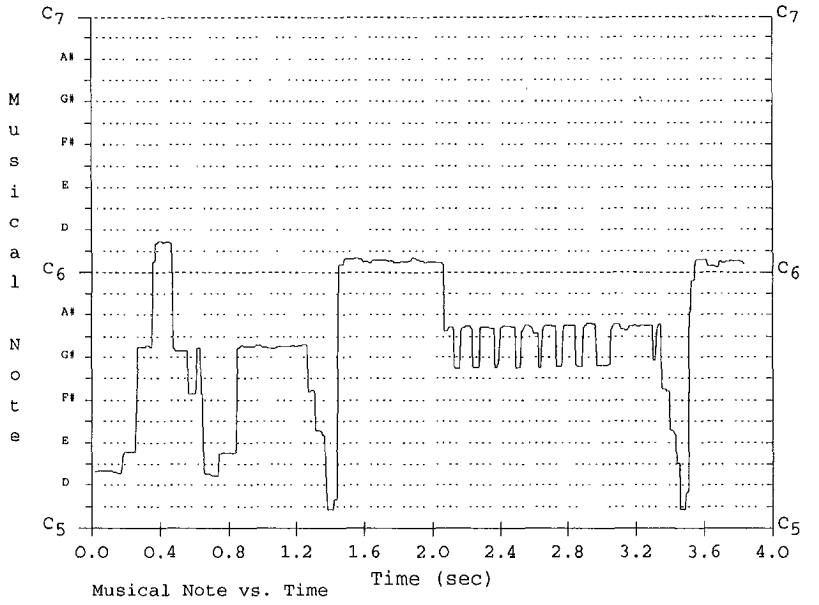


Fig. 9 Musical Note value vs. time for reverberated recording of clarinet performance of 2 bar fragment from Messiaen's *Quatuor pour la fin du temps.*