From: Proceedings of the 1983 International Computer Music Conference.

An Automated Approach To Tuning

Roger B. Dannenberg and Arthur H. Benade

May 83

Abstract

Conventional keyboard or computer tuning systems suffer from either a lack of "natural" harmonic intervals, or the inability to support modulation. In contrast to conventional fixed-pitch systems, a variable-pitch tuning system allows small changes in pitch to obtain desirable intervals within a framework where modulation is possible. Such a system is practical only when the correct pitch variations can be determined without elaborate notation by the composer. To solve this problem, an algorithm is proposed that computes pitch from a conventional score. A modification to the basic algorithm allows a controlled amount of "equal-temperedness", and similar algorithms can be applied to microtonal scales.

1. Introduction

Many investigators over several centuries have considered the use of a wide variety of tunings and scales in music. All of the studies of which we know consider only fixed assignments of pitch to scale steps; however, the computer makes it practical to consider variable tuning systems, where small pitch adjustments are made to compensate for the "imperfections" inherent in any fixed-pitch tuning system. The motivation for this work is found in the practice of instrumental and vocal performers who routinely make such pitch adjustments, automatically and often without conscious efforts ([2], Sections 15.3 and 15.4).

We shall begin with a quick review of the problems of fixed-pitch tuning systems. Then, an alternative called *variable-pitch* tuning systems is described. We then consider the problem of making a computer play "in tune" using a variable-pitch system, and present one algorithm which seems promising. Finally, some of the remaining problems are described.

2. Fixed-pitch systems

Ordinarily, a tuning system consists of a finite set of ratios ranging from 1:1 to 2:1 and a reference corresponding to the ratio 1:1. The system generates a set of pitches whose corresponding frequencies are in the given ratios. The frequencies may also be scaled by any power of two, corresponding to octave transpositions. This usually gives new intervals which are perceptually significant in the context of conventional music[1]. For example, the conventional equal-tempered, 12-tone tuning system consists of the ratios $\{ 2^{i/12}$,for $0 \leq i \leq 11\}$ and the reference frequency 440Hz.

It is desirable for a tuning system to satisfy two properties:

1. The system should generate "natural" harmonic intervals.
2. The system should allow modulations from one key to another.

The first property is important because harmonic intervals do not produce beating, a readily

[1] Although it is possible (and sometimes desirable) to consider "stretched" or "compressed" octaves which do not correspond to power-of-two frequency ratios, we will not consider them in this paper.

recognized and perceptually important phenomenon ([2], Section 14.4). Thus, a harmonic interval of 1.5 (a musical fifth) has a different perceptual status from ratios such as 1.505 or 1.492. Most scales are based at least partly upon these special (harmonic) relationships between pitches. The second property suggests that transposition must be possible without abandoning the tuning system. Unfortunately, no "fixed" tuning system can meet both of these requirements perfectly. To illustrate this point, let us try to design a tuning system that satisfies both properties.

Suppose we begin with the assumption that our tuning system must include the interval of the perfect fifth, corresponding to frequency ratios of 3:2. Now, if we want to be able to modulate to this pitch, we must include the fifth relative to the ratio 3:2, or $3^2:2^2$, which becomes 9:4, or 9:8 after octave scaling. To modulate again, we must include the ratio 27:16, and so on. If we continue in this fashion, we will generate an infinite set of ratios $\{3^i:2^i\}$, which for all practical purposes represents a pitch continuum after octave scalings.

In contrast, an equal-tempered tuning system solves the modulation problem because it divides the octave equally, but the solution is obtained only by sacrificing beat-free "natural" tunings of intervals. For example, the major third in the 12-tone, equal-tempered system is a rapidly beating interval about 14 cents wider than the ratio 5:4. This is simply not acceptable for some music.

3. Variable-pitch tuning systems

An alternative to fixed-pitch tuning systems is to allow small pitch variations from a given set of pitches. Slight pitch adjustments are made to correct for otherwise "out-of-tune" intervals. We note that this is exactly how two brass players, for example, can play a beat-free major third when their instruments are designed to play a wider, equal-tempered third. The players simply adjust their tuning slides, or "lip" the notes to the appropriate pitches.

Because the equal-tempered tuning system contains many good approximations to harmonic intervals, only slight adjustments are needed to obtain beat-free intervals. It has been suggested that musicians may choose pitches approximately 10 cents above and below the equal-tempered pitch in addition to the equal-tempered pitch itself, because the additional pitch choices allow the musician to produce very nearly beat-free intervals with other scale tones ([2], Section 15.4).

Computers can, of course, generate pitches that are essentially arbitrary, so producing a desired pitch is no problem. However, it is not always trivial to determine what pitch is appropriate. We suspect that the problem of pitch determination accounts for the popularity of fixed-pitch tuning systems in computer-generated music. In the next section, we will discuss a method for automating pitch selection. We will assume that a composer has specified a score using notes from a nominal 12-tone, equal-tempered scale, but that pitches are to be adjusted in accordance with their harmonic significance.

Although we will describe a specific algorithm, we hope the reader will keep in mind that other algorithms are certainly possible, and that greater flexibility can be obtained if the composer is allowed to override the algorithm by specifying exact values. This is desirable for a number of reasons, including intentional dissonance, and situations where beating is not apparent ([2], Section 15.6).

4. Automating pitch selection

In this section, an algorithm for selecting pitches is described. We will assume that there is a direct correspondence between pitch and frequency (such as with tones whose partials are harmonic) and from here on we will speak in terms of *frequency* rather than pitch.

The algorithm consists of several major steps. In the first step, each set of simultaneously sounding notes is analyzed harmonically to yield a root and then a chord type. In the second step, this information is used to determine initial *relative* frequencies among the notes. In the next step, common tones and root movement are analyzed to determine an initial *absolute* frequency for the notes. In the final step, the initial absolute frequencies are modified to prevent a long term drift of the basic pitch region as the music progresses.

Each step of the algorithm is applied to each chord in the score, from the beginning to the end of the score. (Each step is applied to the first chord, then each step is applied to the second, and so on.) A chord is defined as the beginning of any note, since that point marks a change in the set of notes presently sounding. At the conclusion of the algorithm, a precise frequency will be specified for each note in the score.

4.1. Step one: harmonic analysis

Frequency selection requires an analysis of the notes to be played. Initially, only vertical aspects of the score are considered, that is, we will base the analysis solely on the notes presently sounding. Hindemith's algorithm [5] for determining the root of a chord is applied (getting the "wrong" answer for the root is not critical, and other root-finding algorithms are possible). Hindemith's algorithm is as follows:

First, eliminate from consideration the upper note of any octave relationship. Then find the "best" interval in the chord, where intervals are ranked from "best" to "worst" as follows: fifth, fourth, major third, minor sixth, minor third, major sixth, major second, minor seventh, minor second, major seventh. If there are two equal intervals in the chord, and no interval ranks higher, then the best interval is the lower of the two. Interestingly, Hindemith's ranking of intervals corresponds closely to an ordering of intervals according to the prominence with which mistuning advertises itself. (See Table 14.1 in Benade [2], page 274).

If the best interval is a fifth, third, or second, then the chord root is the lower note of the best interval. For best intervals of fourth, sixth, or seventh, the chord root is the upper note.

For a few chords, Hindemith makes exceptions, but we will ignore these. Also, Hindemith does not assign a root to two-note chords (intervals) or to a single pitch or one with octave doublings. We will extend the algorithm to cover these cases as follows:

1. If only one interval remains after eliminating doubled tones, then the interval is considered the best interval, and the root is determined accordingly.

2. If only a tritone is present, the root is the lowest note of the interval.

3. If only a single note is present, the note is considered to be the root.

Having found the root, we now represent the chord as the set of all intervals above the root, ignoring octaves. For example, the chord {C_4, G_4, E_5} has the root C_4, and contains the intervals {unison, perfect fifth, major third}. Note the following:

- The interval of the tenth is treated as a third.
- There are 12 possible intervals, including unison.
- Since the root-finding algorithm always assigns the root to be a note in the chord, the unison interval will always be present.
- Since the unison interval is always present, there are 2^{11} or 2048 possible chords or interval combinations in this scheme.

4.2. Step two: relative frequency determination

For each of the 2048 chord types, we store the desired frequency ratios for the intervals of the chord. For example, for the chord type {unison, major third, perfect fifth}, we might store the ratios {1:1, 5:4, 3:2}. These ratios will be used to determine the actual pitches that are played.

It may appear that we have ignored the presence of, for example, the minor third between the note labeled 5:4 and the one labeled 3:2. However, this interval is implicit in the chord type, and also implicit in the corresponding frequency ratios. This implicit interval must be taken into account when the frequency ratios are established.

To summarize the algorithm up to this point: we take a vertical array of notes, or chord, and apply a modification of Hindemith's algorithm to determine the root. The chord is then classified by the intervals that it contains. Next, appropriate frequency ratios for the notes of the chord are determined by table lookup, using the interval set as index.

4.3. Step three: absolute frequency determination

Now we must determine absolute frequencies for the notes of the chord. Given the relative frequencies from the previous step, it is only necessary to determine the frequency for one note; all the others can be computed using the assigned ratios. We will base the pitch on common tones and octave relationships with the previous chord.

If there is one note in common between the present chord and the previous one, use the frequency for that note in the previous chord as the frequency for the corresponding note in the present chord. If there are no common tones, but there are two notes with an octave relationship between the present and previous chords, then use the frequency of the note from the previous chord to determine the frequency of the note in the present chord. For example, if the previous chord had an A_3 whose frequency was 221Hz, and the present chord has an A_4 (nominally 440Hz), then we would assign the frequency of 442Hz (2 x 221Hz) to the A_4, since it has an octave relationship with a note in the previous chord.

This step is generalized to handle the case where there is more than one common tone. In this situation, we would like to minimize the pitch changes in the common tones. To accomplish this, first choose one of the common tones and compute frequencies for the present chord as in the previous paragraph. Next, subtract the mean increase in frequency of common tones, expressed in cents, from each note of the present chord. The effect is to "center" the new chord so that the common tones move a minimal distance[2].

If there is no common tone or octave, then look for the relationship of the fifth or fourth, and apply the same resolution for cases where there are multiple relationships between chords. If there are no fifth or fourth relationships either, or if we are dealing with the first chord of the score, then chose the nominal equal-tempered frequency for the root.

---

[2] A variant of this algorithm could be to leave the pitch of common tones unaltered. The same computations would be used to find the other pitches.

4.4. Step four: preventing drift

If we use common tones to determine the reference pitch for successive chords, there is the possibility that, after a number of modulations, the pitch will drift away from our original set of reference pitches (the equal-tempered scale based on A440). For example, consider the following sequence of chords: $\{C_4, E_4\}$, $\{E_4, G_4\#\}$, $\{G_4\#, C_5\}$, $\{C_4, E_4\}$, ... Each of these is analyzed as a major third with the root in common with the third of the previous chord. If the ratio assigned to the third is 5:4, then the second C in the sequence will have a frequency which is $5^3/4^3$ times that of the first C. That is 125:64 or about 41 cents short of the expected octave.

To keep our frame of reference from drifting far off course due to modulations, we perform a final adjustment to the frequencies determined in the previous step. First, we compute *drift* as follows:

$$\delta = 1/n \sum_{i=1}^{n} (f_i - g_i)$$

where δ stands for drift (in cents), f_i is the chosen frequency for the i^{th} note of the chord (in cents), and g_i is the nominal equal-tempered frequency for the same note (in cents).

After computing the drift, the entire chord is transposed by the amount $(-\delta \times \alpha)$, where $0 \leq \alpha \leq 1$. This has the effect of damping any cumulative drift, and α controls the rate of damping[3].

4.5. Further parameterization

Our goal is to extend the power of the composer, not to take power away from him. So far, we have developed a scheme to help the composer generate music that is "in tune" in the sense that, with properly initialized tables, beating can be reduced or eliminated. There are situations, however, where a composer might want to increase beating. We can introduce another parameter, β, to provide this kind of control. When $\beta = 0$, the tuning is produced by our algorithm as described. When $\beta = 1$, frequencies from the reference tuning (equal temperament) are used. For other values

[3] We could go all out at this point by recognizing that we have constructed a digital high-pass filter which lets momentary perturbations of pitch pass through, but in the long run cancels any cumulative drift. One could easily construct a higher order filter, parameterized by cutoff frequency rather than α

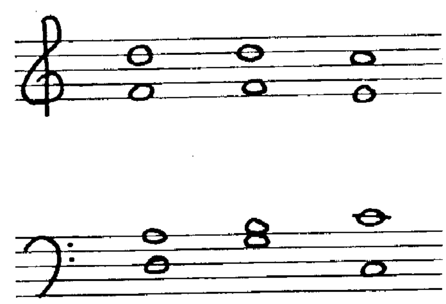
of β, frequencies are interpolated or extrapolated accordingly. The final frequencies are thus:

$$h_i + \beta \times (g_i - h_i),$$

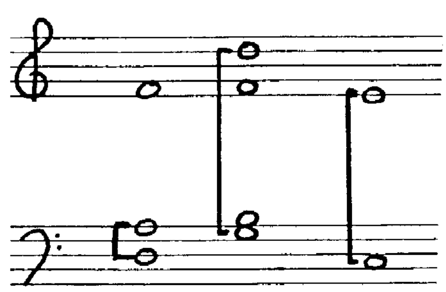
where $h_i = f_i - \delta \times \alpha$,
g_i is the reference tuning for the i^{th} note, and
f_i is the frequency for the i^{th} note of the chord
before drift compensation

5. An example

Let us apply the algorithm to the simple ii-V7-I chord progression:



In Step 1, we eliminate octave doublings and find the "best" interval in each chord:



The roots are D, G, and C as expected. The chord types can then be represented by intervals above the root:

{minor third, perfect fifth}
{major third, perfect fifth, minor seventh}
{major third}

Applying Step 2 to the first chord, we obtain the ratios 6:5 for the minor third, and 3:2 for the fifth (or whatever tuning the table-builder wanted for a major triad). Since this is the first chord, we choose the equal-temperament tuning for the root, and compute the other pitches:

D_3 = 146.832 Hz (equal temperament)

A_3	= 220.248	(3/2 x 146.832)
F_4	= 352.397	(6/5 x 2 x 146.832)
D_5	= 587.328	(4 x 146.832)

Now, we can move on to the second chord, a dominant seventh. Suppose the table gives the following ratios: 5:4, 3:2, and 1.789 for the third, fifth, and seventh, respectively. (The ratio for the seventh was chosen to make this example interesting.) In Step 3, we find two common tones, D and F. We will use D for our initial reference. From the ratios, we compute the following initial frequencies:

D_5	= 587.328 Hz	(reference from previous chord)
G_3	= 195.776	(2/3 x 1/2 x 587.328)
B_3	= 244.720	(5/4 x 195.776)
F_4	= 350.245	(1.789 x 195.776)

In Step 4, we compute the mean increase in frequency of the common tones. For the D, the increase is zero, and for the F, the increase is – 10.60954 cents, yielding a mean of – 5.3048 cents, which is *subtracted*[4] from our initial frequencies. The new frequencies, from lowest to highest are:

G_3	= 196.378 Hz
B_3	= 245.471
F_4	= 351.320
D_4	= 589.131

Notice how the interval from F to D is widened by about 10 cents from the previous chord: the D is raised and the F is lowered.

Finally, we apply Step 4 to adjust the frequencies toward the nominal equal-tempered pitch. The drift computation is illustrated below:

	Step 3 Freq. (Hz)	Equal-Tempered Freq. (Hz)		Difference (cents)
G_3	196.378	195.998		3.35327
B_3	245.471	246.942		– 10.34356
F_4	351.320	349.228		10.33978
D_4	589.131	587.329		5.30352
			sum	8.65301
			mean	**2.16325**

Thus, the frequencies from Step 3 are a little over 2 cents sharp on the average, relative to equal

[4]That is, the frequencies are increased by 5.3048 cents, or multiplied by 1.00307.

tempered tuning. Assuming $\alpha = 0.1$, we lower the frequencies 0.216325 cents by multiplying each by 0.999875. The final frequencies are:

G_3 = 196.353
B_3 = 245.440
F_4 = 351.276
D_4 = 589.057

This completes the computation for the dominant seventh chord. For Step 3 of the last chord, there is no common tone, so the initial reference is based on fifth and fourth relationships (B to E, G to C, and F to C). The simplest way to do the computation is to transpose the B, G, and F of the previous chord by a fifth or fourth (3:2 or 4:3), and use the resulting frequencies as if they were common tones. These "implied" or "virtual" common tones will be:

E_4 = 327.253 Hz (4/3 x 245.440)
C_4 = 261.804 (4/3 x 196.353)
C_5 = 526.914 (3/2 x 351.276)

The remainder of the computations are similar to those for the previous chord.

6. Other considerations

The tabular method of determining frequency ratios may not apply to sounds with inharmonic spectra. Ideally, we would like to model the listener in order to estimate the perceived beating. The problem then becomes one of adjusting frequencies to minimize the beating function. The formulation of a suitable model, however, is likely to be very difficult.

Similarly, the strength of various harmonics and the octave of the pitch affect the perceived beating [7], but these factors are ignored by our algorithm in order to make the computation more tractable.

Another consideration is that our algorithm considers only a very local context in determining what constitutes an appropriate tuning for a chord. In some cases, however, one might want to consider chords in a more global context. For example, in tonal music, the "correct" tuning for a chord may depend upon its relationship to the tonic in addition to its chord type. Further study of performers may reveal other criteria for pitch selection.

7. Conclusions

In this paper, we have discussed the problems of fixed-pitch tuning systems and advocated the use of a more flexible, variable-pitch approach. We have described an approach to tuning that changes the role of the computer from passive instrument to active performer and interpreter. We have also presented an algorithm that can be used to automate the choice of pitch.

The algorithm seems to make reasonable pitch selections, but our experience with it is limited. We plan to determine experimentally the extent to which listeners perceive the subtle tuning adjustments made by our algorithm. To gain insight into the pitch-selection problem, one could also analyze the way real performers choose and adjust pitches. For example, reports by Sundberg [6], Hagerman [4], and Ternstrom [8] discuss measurements of pitch in performances by instruments, barbershop quartets, and choirs. Max Mathews has instrumented a violin and measured pitches in a performance situation.

Much has been said about microtonal scales and the division of the octave into more than 12 intervals. For example, the paper by Balzano [1] uses a group-theoretic approach to suggest that structures analogous to conventional diatonic scales and triads exist in microtonal pitch systems. A set of compositions by Easley Blackwood [3] explores the equal tunings of 13 through 24 notes, illustrating interesting possibilities for harmony and modulation. An unfortunate characteristic of these alternate tuning systems, as with the 12-tone equal-tempered tuning, is that many intervals are "out of tune". By applying the principles presented in this paper, one could automate the pitch adjustments that might make microtonal music more palatable to the listener and more versatile to the composer.

In a broader context, this study is but one example of the use of the computer as *performer/interpreter* rather than simply *instrument*. Computers will become more helpful to composers when they can interpret not just pitch, but rhythm, tempo, articulation, dynamics, and other parameters in an intelligent and musically meaningful way.

References

[1] G. J. Balzano.
The Group-theoretic Description of 12-Fold and Microtonal Pitch Systems.
Computer Music Journal 4(4):66-84, 1980.

[2] A. H. Benade.
Fundamentals of Musical Acoustics.
Oxford University Press, 1976.

[3] E. Blackwood.
Twelve Microtonal Etudes for Electronic Music Media.
Stereo LP Recording available from Easley Blackwood, 5300 S. Shore Drive, Chicago, IL 60615.
score published by G. Schirmer.

[4] B. Hagerman and J. Sundberg.
Fundamental Frequency Adjustment in Barbershop Singing.
Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR 1/1980), April, 1980.

[5] P. Hindemith.
Craft of Musical Composition, Book 1, Theoretical Part.
Schott, 1942.

[6] J. Sundberg.
In tune or not? A study of fundamental frequency in music practise.
Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR 1/1982), April, 1982.

[7] E. Terhardt.
Pitch, consonance, and harmony.
Journal of the Acoustical Society of America (55), 1974.

[8] S. Ternstrom and J. Sundberg.
Acoustical factors related to pitch precision in choir singing.
Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR 2-3/1982), October, 1982.