Adaptive tunings for musical scales

William A. Sethares
University of Wisconsin, Madison, Wisconsin 53706-1691

(Received 27 October 1993; accepted for publication 5 April 1994)

A fixed, octave-based musical scale cannot remain faithful to the consonant simple integer ratio intervals and simultaneously be modulated to all keys. It is possible to reconcile these competing criteria, however, if the notes of the scale are allowed to vary. This paper presents a method of adjusting the pitches of notes dynamically, an *adaptive tuning*, that maintains fidelity to a desired set of intervals and can be modulated to any key. The adaptive tuning algorithm changes the pitches of notes in a musical performance so as to maximize consonance, which is calculated based on recent perceptual work. The algorithm can operate in real time, is responsive to the notes played, and can be readily tailored to the timbre (or spectrum) of the sound. This can be viewed as a generalization of the methods of just intonation, but it can operate without specifically musical knowledge such as key and tonal center and is applicable to timbres with nonharmonic spectra as well as the more common harmonic timbres.

PACS numbers: 43.75.Bc

INTRODUCTION

Numerous musical scales have been proposed through the past several centuries including meantone; Pythagorean; various just intonations; scales by Werkmeister, Vallotti, and Young, Partch, and Carlos; and the now ubiquitous 12-tone equal-temperament scale.[1] Many of these scales were created as an attempt to minimize the inevitable inconsistencies (commas, dieisis, and schismas, for example) that arise when trying to design a single fixed scale to remain faithful to a desired set of intervals and to be transposable simultaneously to all keys. This paper suggests a different approach, one that allows the tuning to change dynamically as the music is performed; we call such a scale an *adaptive tuning*. Adaptive tunings pursue a single tuning strategy in which the notion of musical "key" loses much of its significance, yet can maintain a desirable set of intervals (such as the small integer intervals) irrespective of starting tone.

Carlos[2] and Hall[3] have introduced quantitative measures of the ability of fixed scales to approximate a desired set of intervals. Since different pieces of music contain different intervals, and since it is mathematically impossible to devise a single fixed scale in which all intervals are perfectly in tune, Hall[3] suggests choosing tunings based on the piece of music to be performed. Polansky[4] suggests the need for a "harmonic distance function," which can be used to make automated tuning decisions, and points to Wagge's[5] "intelligent keyboard," which utilizes a logic circuit to choose automatically between alternate versions of thirds and sevenths depending on the musical context. The adaptive tuning concept uses a measure of consonance as its "distance function" to change the pitches of notes dynamically (and in real time) as the music is performed. Partch[6] wrote "it is conceivable that an instrument could be built that would be capable of an automatic change of pitch throughout its entire range." This paper shows how to realize this concept. In addition, the adaptive tuning is responsive to the timbre of the instruments as they are played.

The adaptive tuning algorithm exploits the results of recent work[7] on the perception of consonance, the "principle of local consonance," which is based on a functional parametrization of the tonal consonance curves of Plomp and Levelt.[8] Local consonance describes the relationship between the timbre of a sound (its spectrum) and a set of intervals (scales or chords) in which the sound can be played most consonantly. For a sound with a given timbre, the "dissonance function" $D(\alpha)$ describes the perceived dissonance of the sound when played at intervals α. Values of α at which local minima of the dissonance function occur are intervals that are (locally) maximally consonant. Such values are called points of local consonance and tend to be musically desirable intervals. For instance, simple integer ratios are points of local consonance for harmonic timbres. The adaptive tuning algorithm calculates the (gradient of the) dissonance at each time step and adjusts the tuning of the notes toward the nearest point of local consonance.

This paper proposes a new (adaptive) solution to the longstanding problem of scale formation. The tuning algorithm is implementable in software or hardware and can be readily incorporated into electronic music studios. Just as many MIDI synthesizers have built-in alternate tuning tables that allow the musician to play in various just intonations and temperaments, an adaptive tuning feature could be added to sound modules. The musician could then effortlessly play in a scale that continuously adjusts to the timbre and the performance in such a way as to maximize consonance.

Section I briefly reviews the concept of consonance and dissonance curves. The adaptive tuning algorithm is then derived as an approximate gradient descent of a cost function defined by the dissonance of the currently sounding tones. Section III analyzes the algorithm and gives several examples of its behavior. Section IV compares the adaptive tuning algorithm to just intonation and to the standard 12-tone equal-tempered scale. Section V suggests several ways that the algorithm might be implemented, and the final section concludes.

© 1994 Acoustical Society of America

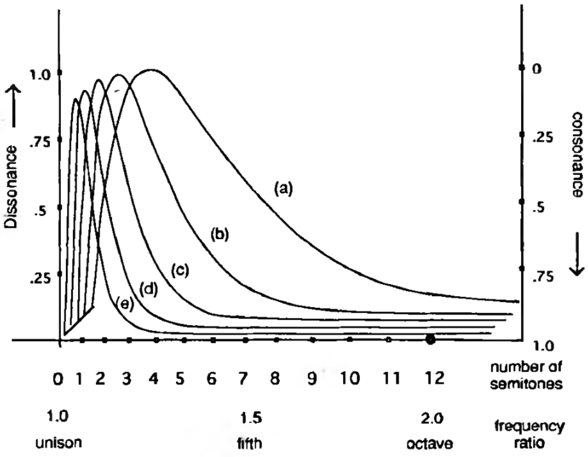


FIG. 1. Dissonance curves for pure sine waves as a function of frequency difference. The consonance and dissonance scales are arbitrary. (a) Base frequency 125; (b) base frequency 250; (c) base frequency 500, (d) base frequency 1000, (e) base frequency 2000.

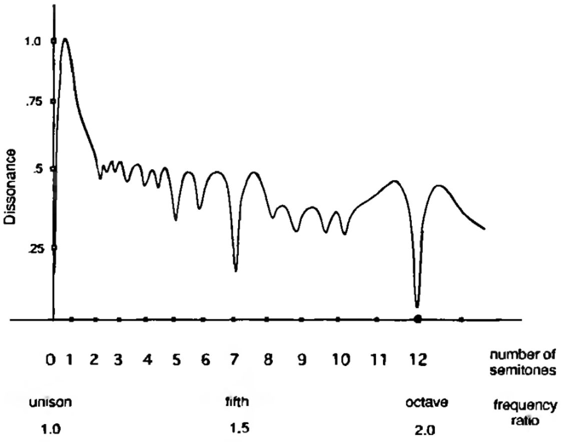


FIG. 2. Dissonance curve for the timbre with nine harmonic partials. The horizontal axis represents frequency difference. Dots mark the location of notes in the standard 12-tone equal-tempered scale. Dissonance values are normalized so that the largest value occurs at unity.

I. A MEASURE OF DISSONANCE

The psychoacoustic work of R. Plomp and W. J. M. Levelt[8] provides a basis on which to build a measure of consonance and dissonance that can be used to guide the adaptation. Plomp and Levelt asked volunteers to rate the perceived dissonance of pairs of pure sine waves, giving curves such as Fig. 1, in which the dissonance is minimum at unity, increases rapidly to its maximum somewhere near one quarter of the critical bandwidth, and then decreases steadily back toward zero. When considering timbres that are more complex than pure sine waves, dissonance can be calculated by summing up all the dissonances of all the partials and weighting them according to their relative amplitudes. This leads to curves such as Fig. 2, which shows the dissonance curve for a timbre with nine harmonic partials. Note that the local minima occur near many of the steps of the 12-tone equal tempered scale (actually, they fall on the nearby simple integer ratios, as shown in Table I). Similar curves can be drawn for nonharmonic timbres.

To be concrete, the dissonance between a sinusoid of frequency f_1 with amplitude v_1 and a sinusoid of frequency f_2 with amplitude v_2 can be parametrized as

$$d(f_1,f_2,v_1,v_2)=v_1v_2(e^{-as|f_2-f_1|}-e^{-bs|f_2-f_1|}), \qquad (1)$$

where

$$s=d^*/[s_1\ \min(f_1,f_2)+s_2], \qquad (2)$$

$a=3.5$, $b=5.75$, $d^*=0.24$, $s_1=0.021$, and $s_2=19$ are determined by a least-squares fit. The amplitude term v_1v_2 ensures that softer components contribute less to the total dissonance measure than those with larger amplitudes, d^* is the interval at which maximum dissonance occurs, and the s parameters in (2) allow a single functional form to interpolate smoothly between the various curves of Fig. 1 by sliding

TABLE I. Notes of the equal-tempered musical scale compared to locations of points of local consonance of the dissonance curve for a nine partial harmonic timbre, and compared to the just intonation major scale from Wilkinson.[1] Septimal (sept.) scale values from Partch.[6]

Note name	Power $\alpha=\sqrt[12]{2}$	12-tone equal temperament	Minima of dissonance curve	Just intonation (Wilkinson[1])	
C	$\alpha^0=1$		1	1:1	unison
C♯	$\alpha^1=1.059$			16:15	just semitone
D	$\alpha^2=1.122$		1.14 (8:7=sept. maj. 2)	9:8	just whole tone
D♯	$\alpha^3=1.189$		1.17 (7:6=sept. min 3)		
			1.2 (6:5)	6:5	just min. 3
E	$\alpha^4=1.260$		1.25 (5:4)	5:4	just maj. 3
F	$\alpha^5=1.335$		1.33 (4:3)	4:3	just perfect 4
F♯	$\alpha^6=1.414$		1.4 (7:5=sept. tritone)	45:32	just tritone
G	$\alpha^7=1.498$		1.5 (3:2)	3:2	perfect 5
G♯	$\alpha^8=1.587$		1.6 (8:5)	8:5	just min. 6
A	$\alpha^9=1.682$		1.67 (5:3)	5:3	just maj. 6
A♯	$\alpha^{10}=1.782$		1.75 (7:4=sept. min. 7)	16:9	just min. 7
B	$\alpha^{11}=1.888$		1.8 (9:5=large just maj. 7)	15:8	just maj. 7
C	$\alpha^{12}=2.0$		2.0	2:1	octave

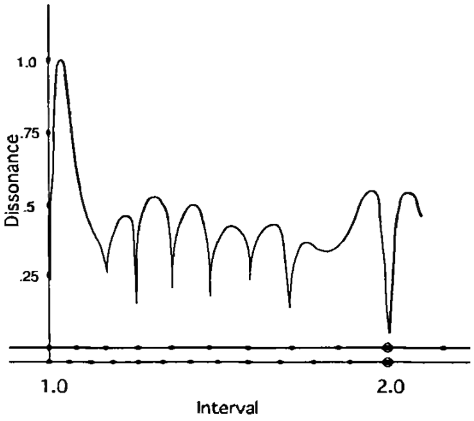


FIG. 3. Dissonance curve for a nonharmonic timbre with partials at 1, β^9, β^{14}, β^{18}, β^{21}, β^{25}, β^{27}, and β^{30}, where β=the ninth root of 2. This timbre is appropriate for nine-tone equal temperament, since minima of the dissonance curve occur at many of the nine-tone scale steps (top horizontal axis) and not at the steps of the 12-tone scale steps (bottom axis). The dissonance values (vertical axis) are arbitrary.

the dissonance curve along the frequency axis so that it begins at the smaller of f_1 and f_2 and by stretching (or compressing) it so that the maximum dissonance occurs at the appropriate frequency. See Ref. 7 for a derivation, justification, and discussion of this model.

More generally, a timbre F with base (or fundamental) frequency f is a collection of n sine waves (or partials) with frequencies $(f,a_2f,...,a_nf)$ and amplitudes $(v_1,v_2,...,v_n)$, where the a_i have been ordered from lowest to highest and all are greater than 1. The intrinsic dissonance of the timbre F is the sum of the dissonances of all pairs of partials

$$D_F=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d(a_if,a_jf,v_i,v_j), \tag{3}$$

with the convention that $a_1=1$. When two notes of the timbre F are played simultaneously at an interval α, the resulting sound has a dissonance that is the same as that of a timbre with frequencies

$$(f,a_2f,...,a_nf,\alpha f,\alpha a_2f,...,\alpha a_nf).$$

Equation (3) can then be used to calculate the intrinsic dissonance of this concatenated timbre. Equivalently, define the timbre αF to contain the frequencies $(\alpha f,\alpha a_2f,...,\alpha a_nf)$ with amplitudes $(v_1,v_2,...,v_n)$. The dissonance of F at the interval α is

$$D_F(\alpha)=\frac{1}{2}(D_F+D_{\alpha F})+\sum_{i=1}^{n}\sum_{j=1}^{n} d(a_if,\alpha a_jf,v_i,v_j), \tag{4}$$

and the dissonance curve generated by the timbre F is $D_F(\alpha)$ over an appropriate range of α.

While (3) is generally more useful for calculations, the dissonance curve (4) allows a simple visualization of the behavior of the adaptive tuning algorithm. Figures 2 and 3 show the dissonance curves for a harmonic and a nonharmonic timbre. Suppose the musical score (or the performer) commands two notes F_1 and F_2, which form an interval α. The dissonance $D_F(\alpha)$ of this interval can be viewed directly from the dissonance curve. The algorithm essentially "slides down" the curve until it reaches the nearest local minimum α^*, which is a point of local consonance. The algorithm retunes the notes F_1 and F_2 to F_1^* and F_2^* so that the actual interval sounded is precisely $F_1^*/F_2^* = \alpha^*$.

II. ALGORITHM STATEMENT

The algorithm must have access to the timbre (or spectrum) of the sounds it is to adjust. This information may be built-in (as in the case of a musical synthesizer or sampler that inherently knows the timbre of its notes) or it may be calculated (via a Fourier transform, for instance). The algorithm adjusts the pitch of each note so as to decrease the dissonance until the nearest point of local consonance is reached. This modified set of pitches (or frequencies) is then output to a sound generation unit. Thus, whenever a new musical event occurs, the algorithm calculates the optimum pitches so that the sound locally minimizes the dissonance.

There are several possible ways that the necessary adjustments can be carried out. Consider the simple case of two notes with pitches F_1 and F_2 (with $F_1<F_2$). With no adaptive tuning, the interval F_2/F_1 will sound. The simplest adaptive strategy would be to calculate the dissonances of the intervals $F_2/F_1+\epsilon$ for various values of ϵ (appropriate ϵ's could be determined by the bisection method, for instance). The point of minimum dissonance is given by that value of ϵ for which the dissonance is smallest. The pitches of F_1 and F_2 are then adjusted by an appropriate amount, and the more consonant interval sounded.

This simple "search" technique is inefficient, especially when it is necessary to calculate the dissonance of several notes sounded simultaneously. The "gradient descent" method[9] is a better way of finding the nearest local minimum of the dissonance curve. Suppose that m notes, each with timbre F, are desired. Let $f_1<f_2<\cdots<f_m$ represent the fundamental frequencies (pitches) of the notes. A "cost" function D is defined to be the sum of the dissonances of all the intervals at a given time,

$$D=\frac{1}{2}\sum_{l=1}^{m}\sum_{k=1}^{m}\sum_{p=1}^{n}\sum_{q=1}^{n} d(a_pf_l,a_qf_k,v_p,v_q). \tag{5}$$

An iteration is then conducted which updates the f_i by moving "downhill" over the m dimensional surface D. This is

$$\left\{\begin{matrix}\text{new}\\ \text{frequency}\\ \text{value}\end{matrix}\right\}=\left\{\begin{matrix}\text{old}\\ \text{frequency}\\ \text{value}\end{matrix}\right\}-\{\text{step size}\}\{\text{gradient}\},$$

where the gradient is an approximation to the partial derivative of the cost with respect to the ith frequency. The minus sign insures that the algorithm looks for a local minimum (rather than a local maximum). The adaptive tuning algorithm is

```
do
    for $i=1$ to $m$
        $f_i(k+1)=f_i(k)-\mu \frac{dD}{df_i(k)}$        (6)
    endfor
until $|f_i(k+1)-f_i(k)|<\delta \ \forall i,$
```

where k is an iteration counter. Thus the frequencies of all notes are modified in proportion to the change in the cost and to the step size μ until convergence is reached, where convergence means that the change in all frequencies is less than some specified δ. Some remarks are in order.

(i) δ should be chosen based on the tuning accuracy of the sound generation unit.

(ii) As a practical matter, it is often advisable to fix the frequency of one of the f_i's and to allow the rest to adapt relative to this fixed pitch.

(iii) It is sensible to carry out the adaptation with a logarithmic step size, that is, one that updates the frequency in "cents" rather than directly in hertz.

(iv) It is straightforward to generalize the algorithm to retune any number of notes, each with its own timbral structure.

(v) A detailed discussion of the calculation of $dD/df_i(k)$ is given in the Appendix.

(vi) There are many ways to carry out the minimization of D. An iterative algorithm is proposed because closed-form solutions for the minima are only possible in the simplest cases.

(vii) If desired, the adaptation can be slowed by choosing the step size small. Outputting intermediate values causes the sound to slide into the point of maximum consonance. This is one way to realize Darregs vision of an "elastic" tuning.[10]

III. BEHAVIOR OF THE ALGORITHM

Any iterative procedure raises issues of convergence, equilibria, and stability. Since the adaptive tuning algorithm is defined as a gradient descent of the dissonance D, such analysis is conceptually straightforward. However, the function D is complicated, its error surface is multimodal, and exact theoretical results are only possible for simple combinations of simple timbres. Accordingly, the analysis focuses on a few simple settings, and examples are used to demonstrate which aspects of these simple settings generalize to more complex (and hence more musically interesting) situations. Theorem 1 shows the close relationship between the behavior of the algorithm and the surface formed by the dissonance curve. In effect, the behavior of the algorithm is to adjust the frequencies of the notes so as to make a controlled descent of the dissonance curve. The simplest possible case considers two notes F and G, each consisting of a single partial.

Theorem 1: Let f_0 and g_0 be the frequencies of two sine waves, with $f_0<g_0$. Apply the adaptive tuning algorithm (6). Then (a) $g_0>(1-s_1)f_0-s_2$ implies that $|g_{k+1}-f_{k+1}|>|g_k-f_k|$ for all k; and (b) $g_0<(1-s_1)f_0-s_2$ implies that $|g_{k+1}-f_{k+1}|<|g_k-f_k|$ for all k, where the s_i are defined in (2).

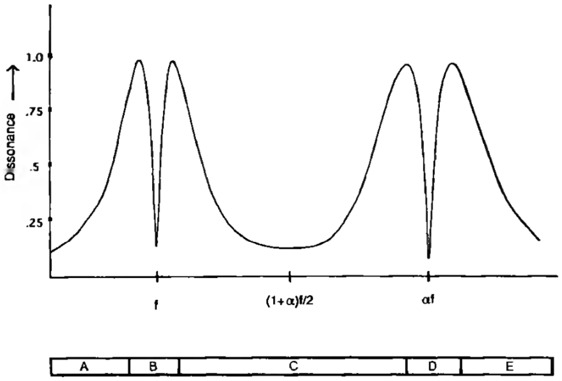


FIG. 4. Dissonance between a note with two fixed partials at f and αf, and a note with a single partial g, as a function of g. There are five possible behaviors as the adaptive tuning algorithm is iterated, depending on the starting frequency. If g is in region A, g drifts toward zero; in region B, g merges with f; in region C, g has minimum at $(1+\alpha)f/2$; in region D, g merges with αf; or, in region E, g drifts toward infinity.

Proof: All proofs are relegated to the Appendix.

In case (a), the f_k and g_k grow further apart as time evolves, while in case (b), f_k approaches g_k and the algorithm drives the dissonance $d(f_k,g_k)$ to its global minimum at some point where $f_k=g_k$. To see this graphically, picture the algorithm evolving on one of the single humped dissonance curves of Fig. 1. If the initial difference between f and g is small, then the algorithm descends the near slope of the hump, driving f and g closer until they merge. If the difference between f and g is large, then the algorithm descends the far side of the hump and the dissonance is decreased as f and g move further apart. The two partials "drift away" from each other. (This is conceptually similar to the "parameter drift" of Ref. 11, where descent of an error surface leads to slow divergence of the parameter estimates.) Together, (a) and (b) show that the point where $g=(1-s_1)f-s_2$ (the top of the hump) is an unstable equilibrium.

For more complex timbres, more interesting (and useful) behaviors arise. Theorem 2 shows how interlaced partials can avoid both drifting and merging.

Theorem 2: Consider two notes F and G. Suppose that F consists of two partials fixed at frequencies f and αf with $\alpha>1$, and that G consists of a single partial at frequency g_0 that is allowed to adapt via the adaptive tuning algorithm (6). For simplicity, assume that s of (2) is identically equal to 1. Then (a) there are three stable equilibria: at $g=f$, at $g=\alpha f$, and at $g=(1+\alpha)f/2$; (b) if $g_0 \ll f$, then $|g_{k+1}-f|>|g_k-f|$ for all k; and (c) if $g_0 \gg \alpha f$, then $|g_{k+1}-\alpha f|>|g_k-\alpha f|$ for all k.

Figure 4 shows the corresponding error surface. The regions of convergence for each of the possible equilibria are shown below the horizontal axis. As in Theorem 1, when g is initialized far below f or above αf (in regions A or E), then g drifts away, while if g starts near enough to f or αf (in regions B or D), then g ultimately merges with f or αf. The interesting new behavior in Theorem 2 occurs in region C,

TABLE II. Converged major and minor chords differ depending on the number of harmonic partials they contain.

Initial notes (12 tone)	Initial frequencies	Converged frequencies for 2–3 partials	Converged frequencies for 4 partials	Converged frequencies for 5–16 partials
Minor				
C	523	523	523	523
E♭	622	647	641	627
G	784	784	784	784
Major				
C	523	523	523	523
E	659	647	641	654
G	784	784	784	784

where g is repelled from both f and αf and becomes trapped at a new minimum at $(1+\alpha)f/2$. In fact, this behavior is generic—sandwiched partials typically reduce dissonance by assuming intermediate positions. This is fortunate, since it gives rise to many of the musically useful properties of adaptive tunings. In particular, sets of notes with interlaced partials do not tend to drift apart since it is difficult for partials to cross each other without a rise in dissonance.

To be concrete, consider notes F with partials at frequencies $(f_0, f_1, \ldots, f_n)$ and G with partials at frequencies $(g_0, g_1, \ldots, g_m)$. Suppose that g_i is sandwiched between f_j and f_{j+1},

$$f_j + \delta^* < g_i < f_{j+1} - \delta^*,$$

for some δ^* larger than the distance to the top of the hump in Fig. 1, and suppose that all other partials are far away:

$$f_{j-1} \ll f_j, \quad f_{j+1} \ll f_{j+2},$$

$$g_{i-1} \ll f_j, \quad f_{j+1} \ll g_{i+1}.$$

Then the dissonances (and their gradients) between g_i and the f_i are insignificant in comparison to the dissonances between g_i and the nearby frequencies f_j and f_{j+1}. Thus g_i acts qualitatively like the g of Theorem 2 as it is adjusted by the adaptive tuning algorithm toward some intermediate equilibrium. Of course, the actual convergent value depends on a complex set of interactions among all the partials, but g_i tends to become trapped, since approaching either f or αf requires climbing a hump of the dissonance curve and a corresponding increase in dissonance.

As more partials are adapted, the error surface increases in dimension and becomes more complex. Notes evolve on an m-dimensional sheet that is pocketed with crevices of consonance into which the algorithm creeps. Even a quick glance at the Appendix shows that the number of equations grows rapidly as the number of interacting partials increases. To examine the results of such interactions in a more realistic situation, Table II reports converged values (in hertz, accurate to the nearest integer) for triads played with harmonic timbres with varying numbers of partials. In each case, the algorithm is initialized with fundamental frequencies that correspond to the 12-tone equal tempered notes C, E^b, G (a minor chord) or to C, E, G (a major chord), and the algorithm is iterated until convergence. No "drifting" notes or divergence occurs because the partials of the notes are interlaced. In all cases, the fifth (the interval between C and G) remains fixed at a ratio of 1.5:1. For simple two and three partial notes, the major and minor chords merge, converging to a "middle third" that splits the fifth into two parts with ratios 1.21 and 1.24. With four partials, the middle third splits the fifth into two nearly equal ratios of 1.224.

For notes with five or more partials (up to at least 16), the two initializations evolve into distinct musical entities. The major chord initialization converges to a triad with ratios 1.2 and 1.25, while the minor chord initialization converges to a triad with the inverted ratios 1.25 and 1.2. These are consistent with the minor and major thirds of the just intonation scale, suggesting that performances in the adaptive tuning are closely related to a just intonation when played with harmonic timbres of sufficient complexity.

Thus musical theory (in this case just intonation) is intimately related to perceptions of consonance, assuming an ideal harmonic structure. When timbres deviate from a harmonic structure, however, music theory and consonance differ. It is easy to hear that the ear prefers consonance over music theory. A particularly striking example is provided by the use of stretched (and/or compressed) timbres.[12,13] Consider a complex nonharmonic timbre with partials at

$$f, \ 2.1f, \ 3.24f, \ 4.41f, \ \text{and } 5.6f,$$

which are the first five partials of the stretched timbre defined by

$$f_n = fA^{\log_2 n},$$

for $A = 2.1$. As shown in Table III, an initial set of notes at C, E, G, C converges to notes with fundamental frequencies that are completely unrelated to "normal" 12-tone equal tempered intervals based on the semitone $\sqrt[12]{2}$. Rather, they are notes of a "stretched" scale that is closely related to the stretched semitone $\beta = \sqrt[12]{2.1}$. Thus a major chord composed of notes with stretched timbres converges to a stretched major chord. Similarly, the minor chord converges to a stretched minor chord. Recall that in Ref. 7 a scale and a timbre were defined to be related if the timbre generates a dissonance curve with local minima at the scale steps. Using this notion of related scales and timbres, the action of the algorithm can be described succinctly: *The adaptive tuning*

TABLE III. Using five partial "stretched" timbres, the adaptive tuning algorithm converges to "stretched" major and minor chords, $\beta = \sqrt[12]{2.1}$.

Initial notes (12 tone)	Initial frequency of fundamental	Convergent values	Ratio	Nearest stretched scale tone
C	523	523	1.0	β^0
E♭	622	635	1.21	$\beta^3=1.20$
G	784	808	1.54	$\beta^7=1.54$
C	1046	1099	2.1	$\beta^{12}=2.1$
C	523	523	1.0	β^0
E	659	665	1.27	$\beta^4=1.28$
G	784	808	1.54	$\beta^7=1.54$
C	1046	1100	2.1	$\beta^{12}=2.1$

algorithm automatically retunes notes so as to play in the scale related to the timbre of the notes.

IV. RELATION TO JUST INTONATION AND TO 12-TONE EQUAL TEMPERAMENT

Since harmonic timbres are related to a scale composed of simple integer ratios, using the adaptive tuning (AT) strategy is similar to playing in a just intonation (JI) major scale. Table I compares intervals in the 12-tone equal temperament, in major just intonation scales (from Refs. 1 and 6), and the locations of the local minima of the dissonance curve for a timbre with nine harmonic partials. The intervals in just intonation are quite similar to the locations of the local minima. In particular, the consonance curve agrees with the "septimal" scales of Partch for seconds, tritones, and the minor seventh, but with the just major scale for the major seventh. Local minima occur at both the septimal and the just thirds.

Significant differences arise, however, when the tonal center of the piece changes. Consider a musical fragment that cycles through the "circle of fifths":

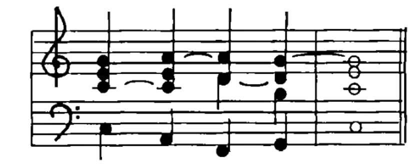
C G D A E B F♯ C♯ G♯ D♯ A♯ F C.

When played in JI in the key of C, this progression sounds very "out-of-tune." For instance, major thirds are harmoniously played at intervals of 5:4 in the keys near C, but are sounded as 32:25 in A and E and as 512:405 in F♯. Some fifths are impure also; the fifth in the D♯ chord, for instance, is played as 40:27 rather than the desired 3:2. Such inaccuracies are readily discernible to the ear and sound quite out-of-tune and dissonant. Problems such as this are inevitable for any nonequal fixed tuning.[3] The adaptive tuning is able to maintain the simple 5:4 and 3:2 ratios throughout the musical fragment because it is not a fixed set of notes.

One might consider "switching" from JI in C to JI in G to JI in D, etc., using the local musical key to determine which JI scale should be used at a given instant. This would cure the immediate problem for this example. Unfortunately, it is not always easy to determine the proper local key of a piece, nor even to determine if and when a key change has occurred. The AT automatically adjusts the tuning to the desired intervals with no *a priori* knowledge of the musical key required. When used with harmonic timbres, it is reasonable to view the AT as a way to interpolate continuously between an appropriate family of just intonations.

A subtler problem is the existence of the "syntonic comma" as illustrated in Fig. 5. Hall[3] points out that if this musical fragment is played in JI with the tied notes held at constant pitch, then the instrument finishes lower than it begins. Equal temperament prevents this drift in tonal center by forcing the mistuning of many of the intervals away from their just small integer ratios. The adaptive tuning maintains the just ratios and the tonal center remains fixed. This is possible because the pitches of the notes are allowed to vary microtonally. For instance, the C note in the second chord is played at 264 Hz while the "same" note in the first chord is played at 261.5 Hz.

One of the major advantages of the 12-tone equal tempered scale over JI is that it can be transposed to any key.



Frequencies when played in Just Intonation with held notes:	392.5	436---	436	387.5--	387.5
	327	327	290.5-	290.5	323
	261.5-	261.5	290.5	242	258.5
	131	109	87	96.5	129
Frequencies when played in 12-tone equal temperament:	392	440	440	392	392
	329.5	329.5	293.5	293.5	329.5
	261.5	261.5	293.5	247	261.5
	131	110	87.5	98	131
Frequencies when played in adaptive tuning:	392.5	440	438.5	391	392.5
	327	330	292	294	327
	261.5	264	292	245	261.5
	131	110	87.5	98	131
Ratios when played in adaptive tuning and in just intonation:	6/5	4/3	3/2	4/3	6/5
	5/4	5/4	1/1	6/5	5/4
	2/1	6/5	5/3	5/4	2/1

FIG. 5. An example of the "syntonic comma" in just intonation; the piece ends about 21 cents lower than it begins. 12-tone equal-temperament maintains the pitch by distorting the simple integer ratios. The adaptive tuning microtonally adjusts the pitches of the notes to maintain simple ratios and to avoid the wandering pitch. All frequency values are rounded to the nearest 0.5 Hz.

The adaptive tuning strategy shares this advantage, as demonstrated by the circle of fifths example. Both 12-tone and the AT can be played starting on any note (in any key). The 12-tone equal scale has sacrificed consonance so that (say) all the C notes can have the same pitch. As before, the adaptive tuning algorithm modifies the pitch of each note in each chord slightly in order to increase the consonance. Thus the C note in the C chord has a (slightly) different frequency from the C notes in the G♯ and F chords.

When restricted to a single key (or to a family of closely related keys) JI has the advantage that it sounds more consonant than 12-tone equal (at least for harmonic timbres), since all intervals in 12-tone equal are mistuned somewhat from the simple integer ratios. The AT shares this advantage with JI. Thus the difference between a piece in AT and the same piece played in 12-tone equal is roughly the same as the difference between JI and 12-tone, for pieces in a single key when played with harmonic timbres. Whether this increase in consonance is worth the increase in complexity (and effort) is much debated, although the existence of groups such as the "Just Intonation Network"[14] is evidence that some find the differences worthy of exploration.

A major advantage of the adaptive tuning approach becomes apparent when the timbres of the instruments are nonharmonic, that is, when the partials are not harmonically related. Consider a "bell-like" or "gong-like" instrument with the nonharmonic spectrum of Fig. 3 which was designed for play in nine-tone equal temperament using the techniques of Ref. 7. The dissonance curve is significantly different from the harmonic dissonance curve of Fig. 2. The most consonant intervals occur at steps of the nine-tone equal-tempered scale (the top axis) and are distinct from the simple integer ratios. The 12-tone equal scale (shown in the bottom axis) does not contain a close approximation to most of these consonant

TABLE IV. Using the nine-tone timbre of Fig. 3, the adaptive tuning algorithm converges to minima of the related dissonance curve. The major chord converges to a chord with nine-tone scale steps 1, 3, and 5. $\beta=\sqrt[9]{2}$.

Initial notes (12 tone)	Initial frequency of fundamental	Convergent values	Convergent ratio	Nearest nine-tone scale step
C	523	528		
E♭	622	617	1.17	$\beta^2=1.17$
C	523	528		
E	659	659	1.26	$\beta^3=1.26$
C	523	518		
F	698	705	1.36	$\beta^4=1.36$
C	523	513		
F♯	739	755	1.47	$\beta^5=1.47$
C	523	528		
G	783	777	1.47	$\beta^5=1.47$
C	523	523		
G♯	830	830	1.59	$\beta^6=1.59$
C	523	519		
A	880	888	1.71	$\beta^7=1.71$
C	523	533		
A♯	932	915	1.71	$\beta^7=1.71$
C	523	528		
E	659	660	1.26	$\beta^3=1.26$
G	783	770	1.47	$\beta^5=1.47$

intervals. Table IV demonstrates the behavior of the AT algorithm when used with this nine-tone timbre. Pairs of notes are initialized at standard 12-tone intervals; the algorithm compresses or expands them to the nearest point of local consonance. In all cases, the converged values are intervals in nine-tone equal temperament. Similarly, a standard major chord converges to the first, third, and fifth scale steps of the nine-tone scale.

The adaptive tuning strategy can be viewed as a generalization of just intonation in two directions. First, it is independent of the key of the music being played, that is, it automatically adjusts the intonation as the notes of the piece move through various keys. This is done without any specifically "musical" knowledge such as the local "key" of the music. Second, the adaptive tuning strategy is applicable to nonharmonic timbres as well as harmonic, thus broadening the notion of "just intonation" to include a larger palette of sounds.

Consonance is only one aspect of music, and the use of adaptive tunings has no influence on musical essentials such as rhythm and melody. Even within the realm of harmony, it would be naive to suggest that maximizing consonance is always desirable. A dissonance-free rendition of Stravinsky's "Rite of Spring" would surely lose much of its impact.

V. IMPLEMENTATIONS AND VARIATIONS

The adaptive tuning algorithm can be readily incorporated into existing MIDI-based synthesizers and music studios in the form of software, stand alone hardware, or (ideally) as a built-in option analogous to the alternate tuning tables currently found in many MIDI-based sound modules. There are several ways that adaptive tunings could be added to (or incorporated in) a computer based music environment. These include (i) software to manipulate Standard MIDI Files (or the equivalent) [in such an implementation, the musician or composer generates a Standard MIDI File (SMF); the adaptive tuning algorithm is implemented as a software program that reads the SMF, adapts the tuning of the notes as described above, and writes a modified SMF file that can subsequently be played via standard sound modules or manipulated further by the musician/composer in a sequencer program]; (ii) a stand-alone piece of hardware (or software to emulate such hardware) that interrupts the flow of MIDI data from the controller (for instance, the keyboard), adapts the tuning as described above, and outputs the modified notes; (iii) the adaptive tuning strategy can be incorporated directly into the sound generation unit (the synthesizer or sampler); and (iv) direct manipulation of digitized sound.

The software strategy (i) has the advantage that it may be simply and inexpensively added to any computer-based electronic music system. The disadvantage is that it is inherently not a real-time implementation. On the other hand, both the stand-alone approach (ii) and the built-in approach (iii) are capable of real-time operation. The current MIDI specification has no reserved commands for retuning notes. Even so, both (i) and (ii) could be implemented using "pitch bend" commands. An unfortunate side effect of this is that the note commands would need to be rechanneled, thus increasing the complexity of the system setup for the operator. Since the algorithm is most effective when it has access to the timbre (or spectrum) of the sound, both (i) and (ii) require that the operator input the timbre. Of course, a frequency analysis module could be added to the software/hardware, but this would increase the complexity of the unit. The built-in solution (iii) does not suffer from any of these complications (indeed, the synthesizer inherently "knows" the timbre of the sound it is producing) and is consequently preferred for MIDI implementation, though it would clearly require a commitment by musical equipment manufacturers. Approaches (i) and (ii) have been implemented in software and were used to generate the examples given in this paper.

The adaptive tuning could also be implemented in hardware (or software to emulate such hardware) that directly manipulates digitized sound. The device would perform an appropriate FFT (fast fourier transform, or equivalent) to determine the current spectrum of the sound, run the adaptive algorithm to modify the spectrum, and then return the modified spectrum to the time domain with an inverse FFT. The device could be operated off-line or in real time if sufficient computing resources are devoted to the task.

Throughout this paper, the adaptive tuning algorithm has been stated in terms of an optimization problem based on dissonance curves solvable by gradient descent methods. Other optimization problems based on other psychoacoustic measures of sound quality and solvable by other types of algorithms are also possible. Indeed, as the state of knowledge of psychoacoustic phenomena increases, new adaptive tuning algorithms seem likely.

The adaptive tuning algorithm is stated in terms of an optimization problem, that of adjusting a set of m notes, each with n partials, to the nearest local minimum of the dissonance curve. Simple search and gradient techniques are only two of many possible methods capable of solving this optimization problem. Other methods include "random search methods" (for instance, simulated annealing[15] and the genetic algorithm[16]), methods based on linear (or nonlinear) programming, or other optimization solving algorithms. A particularly natural approach might be in terms of a neural network structure. The specific goal of the optimization problem (as the minimization of the dissonance curve of Ref. 7) is only one of many possible optimization criteria that might be useful in creating adaptive tuning algorithms. For instance, other parametrizations of the Plomp–Levelt[8] curves would lead to closely related optimization problems. Factors other than consonance and dissonance could also be used, for instance, one might choose an adaptation based on some aspect of tonal fusion,[17] or one based on some aspect of auditory masking.[18] Alternately, adaptive tunings could be based on some set of *a priori* desirable intervals, rather than directly on properties of the auditory system. Algorithms could also be derived that consist of weighted versions of more than one of the above criteria.

VI. CONCLUSIONS

The adaptive tuning strategy provides a new solution to the longstanding problem of scale formation. Just intonations (and related scales) sacrifice the ability to modulate music through multiple keys, while 12-tone equal temperament sacrifices the consonance of intervals. Adaptive tunings retain both consonance and the ability to modulate, at the expense of (real-time) microtonal adjustments in the pitch of the notes. The adaptive tuning approach is based on data that encodes basic human perceptions, the tonal consonance curves of Plomp and Levelt.[8] As such, it may lay a claim to cultural independence, at least to the extent that such fundamental perceptions are intercultural.

The behavior of the adaptive tuning algorithm was described in terms of notes continuously descending a complex multidimensional landscape studded with dissonant mountains and consonant valleys. For harmonic timbres, the adaptive tuning acts like a just intonation that automatically adjusts to the key of the piece, with no specifically musical knowledge required. For nonharmonic timbres, the adaptive tuning automatically adjusts the frequencies of the tones to the nearest point of local consonance, providing an automated way to play in the scale related to the timbre of the sound. Adaptive tunings are determined by the timbre of the sounds and by the piece of music performed; chords and melodies tend to become more "in tune with themselves."

APPENDIX

This Appendix derives concrete expressions for the update terms of the adaptive tuning algorithm (6) and details proofs of the theorems. Only the terms in D [see (5)] that include f_i need to be considered when calculating the gradient dD/df_i. Thus

$$\frac{dD}{df_i}=\frac{d}{df_i}\left(\frac{1}{2}\sum_{k=1}^{m}\sum_{p=1}^{n}\sum_{q=1}^{n} d(a_pf_i,a_qf_k,v_p,v_q)\right.$$

$$\left.+\frac{1}{2}\sum_{k=1}^{m}\sum_{p=1}^{n}\sum_{q=1}^{n} d(a_pf_k,a_qf_i,v_p,v_q)\right) \qquad \text{(A1)}$$

$$=\sum_{k=1}^{m}\sum_{p=1}^{n}\sum_{q=1}^{n}\frac{d}{df_i}d(a_pf_i,a_qf_k,v_p,v_q), \qquad \text{(A2)}$$

since $d(f,g,v,w)=d(g,f,v,w)$ and the derivative commutes with the sums. Calculating the derivative of the individual terms $(d/df_i)d(f,g,v,w)$ in (A2) is complicated by the presence of the absolute value and min functions in (1) and (2). The function is not differentiable at $f=g$, and changes depending on whether $f>g$ or $g>f$. Thus we define the function

$$\frac{d}{df}d(f,g,v,w)=\begin{cases} vw\left[\dfrac{-ad^*}{(fs_1+s_2)}\exp\left(\dfrac{ad^*(f-g)}{fs_1+s_2}\right)+\dfrac{bd^*}{(fs_1+s_2)}\exp\left(\dfrac{bd^*(f-g)}{fs_1+s_2}\right)\right], & \text{if } f>g,\\[2ex] vw\left[\dfrac{ad^*(gs_1+s_2)}{(fs_1+s_2)^2}\exp\left(\dfrac{ad^*(f-g)}{fs_1+s_2}\right)-\dfrac{bd^*(gs_1+s_2)}{(fs_1+s_2)^2}\exp\left(\dfrac{bd^*(f-g)}{fs_1+s_2}\right)\right], & \text{if } f<g,\\[2ex] 0, \quad \text{if } f=g, \end{cases} \qquad \text{(A3)}$$

which is a close approximation to the desired derivative. Then an approximate gradient is readily computable as the triple sum (A2) of elements of the form (A3).

Two simplifications are made to streamline the results. A single dissonance function is assumed for all frequencies, and all partials are presumed to have unit amplitudes. Thus $s=1$ in (2) and $v_1=v_2=1$ in (1). Taking the derivative of (1) with respect to $x=|f_2-f_1|$, and equating it to zero shows that the point of maximum dissonance occurs at $d^*=\ln(a/b)/(a-b)$.

Proof of *Theorem 1*: From (A3), the updates for f and g are

$$f_{k+1}=f_k-\mu\left[\frac{ad^*(g_ks_1+s_2)}{(f_ks_1+s_2)^2}\exp\left(\frac{ad^*(f_k-g_k)}{f_ks_1+s_2}\right)\right.$$

$$\left.-\frac{bd^*(g_ks_1+s_2)}{(f_ks_1+s_2)^2}\exp\left(\frac{bd^*(f_k-g_k)}{f_ks_1+s_2}\right)\right], \qquad \text{(A4)}$$

$$g_{k+1}=g_k+\mu\left[\frac{ad^*}{(f_ks_1+s_2)}\exp\left(\frac{ad^*(f_k-g_k)}{f_ks_1+s_2}\right)\right.$$

$$\left.-\frac{bd^*}{(f_ks_1+s_2)}\exp\left(\frac{bd^*(f_k-g_k)}{f_ks_1+s_2}\right)\right]. \qquad \text{(A5)}$$

The term in brackets in (A5) is positive whenever

$$a\ \exp\left(\frac{ad^*(f_k-g_k)}{f_ks_1+s_2}\right)>b\ \exp\left(\frac{bd^*(f_k-g_k)}{f_ks_1+s_2}\right), \qquad \text{(A6)}$$

which is true whenever

$$\ln(a)+\frac{ad^*(f_k-g_k)}{f_ks_1+s_2}>\ln(b)+\frac{bd^*(f_k-g_k)}{f_ks_1+s_2}. \qquad \text{(A7)}$$

Rearranging gives

$$\frac{\ln(a)-\ln(b)}{a-b}>\frac{d^*(f_k-g_k)}{f_ks_1+s_2}. \qquad \text{(A8)}$$

Since the left-hand side is equal to d^*, this can be rewritten

$$f_ks_1+s_2>f_k-g_k. \qquad \text{(A9)}$$

Thus $g_k>(1-s_1)f_k-s_2$ implies that $g_{k+1}>g_k$. Similarly, $f_{k+1}<f_k$, which together show (a). On the other hand, if $g_k<(1-s_1)f_k-s_2$, an identical argument shows that $g_{k+1}<g_k$ and $f_{k+1}>f_k$ for all k. □

Proof of Theorem 2: The total dissonance for this case includes three terms, $D_{\text{total}}=d(f,g)+d(f,\alpha f)+d(g,\alpha f)$. Since α and f are fixed, $d(f,\alpha f)$ is constant, and minimizing D_{total} is the same as minimizing $d(f,g)+d(g,\alpha f)$. Combining (A2) and (A3) under the assumption that $s=1$, and assuming that $f<g<\alpha f$, the update for g is

$$g_{k+1}=g_k-\mu(ae^{-a(\alpha f-g_k)}-be^{-b(\alpha f-g_k)}$$

$$-ae^{-a(g_k-f)}+be^{-b(g_k-f)}). \qquad \text{(A10)}$$

This has an equilibrium when $\alpha f-g_k=g_k-f$, that is, when $g=[(1+\alpha)/2]f$. Calculation of the second derivative shows that it is positive at this point as long as $f/2(\alpha-1)\gg 1$, which holds for all reasonable f and α. Hence this is a stable equilibrium. Note that if $s=1$ is not assumed, then a much more complex update arises for g. This will have an equilibrium near, but not at, $(1+\alpha)f/2$.

Because of the nondifferentiability of the dissonance function at $f=g$, we cannot simply take the derivative at this point. The strategy to show that $f=g$ is stable is to show that if $g=f+\epsilon$ for some small $\epsilon>0$ then the update decreases g, while if $g=f-\epsilon$ for some small $\epsilon>0$ then the update increases g. Supposing that $g>f$, and assuming that $f(\alpha-1)\gg 1$, the gradient is approximately

$$ae^{-af(\alpha-1)}-be^{-bf(\alpha-1)}-a+b. \qquad \text{(A11)}$$

Since b is about twice the size of a, this is positive. Similarly, for $g=f-\epsilon$, the gradient is approximately

$$ae^{-af(\alpha-1)}-be^{-bf(\alpha-1)}+a-b, \qquad \text{(A12)}$$

which is negative. Consequently, $f=g$ is a local stable point. The point where $\alpha f=g$ is analyzed similarly. Analogous arguments to those used in theorem 1 show that for $g\ll f$, g decreases, while for $g\gg\alpha f$, g increases. □

[1] S. R. Wilkinson, *Tuning In* (Hal Leonard, Milwaukee, WI, 1988).

[2] W. Carlos, "Tuning: at the crossroads," Computer Music J. 11, 29–43 (1987).

[3] D. E. Hall, "Quantitative evaluation of musical scale tunings," Am. J. Phys. **XX**, 543–552 (1974).

[4] L. Polansky, "Paratactical tuning: an agenda for the use of computers in experimental intonation," Comput. Music J. **11**, 61–68 (1987).

[5] H. M. Wagge, "The intelligent keyboard," 1/1 **1**, 12–13 (1985).

[6] H. Partch, *Genesis of a Music* (Da Capo, New York, 1974).

[7] W. A. Sethares, "Local consonance and the relationship between timbre and scale," J. Acoust. Soc. Am. **94**, 1218–1228 (1993).

[8] R. Plomp and W. J. M. Levelt, "Tonal consonance and critical bandwidth," J. Acoust. Soc. Am. **38**, 548–560 (1965).

[9] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," Proc. IEEE **64**, 1151–1162 (1976).

[10] I. Darreg, in private communication with J. Chalmers (1994).

[11] W. A. Sethares, D. A. Lawrence, C. R. Johnson, Jr., and R. R. Bitmead, "Parameter drift in LMS adaptive filters," IEEE Trans. Acoust. Speech, Signal Process. **ASSP-34**, 868–879 (1986).

[12] M. V. Mathews and J. R. Pierce, "Harmony and nonharmonic partials," J. Acoust. Soc. Am. **68**, 1252–1257 (1980).

[13] F. H. Slaymaker, "Chords from tones having stretched partials," J. Acoust. Soc. Am. **47**, 1469–1571 (1968).

[14] 1/1 (Journal of the Just Intonation Network), 535 Stevenson St., San Francisco, CA.

[15] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines* (Wiley-Interscience, New York, 1990).

[16] S. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, New York, 1989).

[17] A. Bregman, *Auditory Scene Analysis* (MIT, Cambridge, MA, 1990).

[18] W. J. Strong and G. R. Plitnik, *Music, Speech, Audio* (Soundprint, Provo, UT, 1992).