

Playing Music in Just Intonation – A Dynamically Adapting Tuning Scheme

Karolin Stange^{1,2}, Christoph Wick³, and Haye Hinrichsen¹

¹ Fakultät für Physik und Astronomie, Universität Würzburg
Am Hubland, 97074 Würzburg, Germany

² Hochschule für Musik, Hofstallstr. 6–8, 97070 Würzburg, Germany

³ Fakultät für Mathematik und Informatik, Universität Würzburg
Am Hubland, 97074 Würzburg, Germany

E-mail: hinrichsen@physik.uni-wuerzburg.de

Abstract. We introduce a dynamically adapting tuning scheme for microtonal tuning of musical instruments, allowing the performer to play music in just intonation in any key. Unlike previous methods, which are based on a procedural analysis of the chordal structure, the suggested tuning scheme continually solves a system of linear equations without making explicit decisions. In complex situations, where not all intervals of a chord can be tuned according to just frequency ratios, the method automatically yields a tempered compromise. We outline the implementation of the algorithm in an open-source software project that we have provided in order to demonstrate the feasibility of the tuning method.

1. Introduction

The first attempts to mathematically characterize musical intervals date back to Pythagoras, who noted that the consonance of two tones played on a monochord can be related to simple fractions of the corresponding string lengths [1, 2]. Physically this phenomenon is caused by the circumstance that oscillators such as strings do not only emit their fundamental ground frequency but also a whole series of partials (overtones) with integer multiples of the ground frequency. Consonance is related to the accordance of higher partials, i.e., two tones with fundamental frequencies f and f' are perceived as consonant if the m -th partial of the first one matches with the n -th partial of the second, meaning that $mf = nf'$ (see Fig. 1). The impression of consonance is particularly pronounced if m and n are small. Examples include the perfect octave ($m/n = 2/1$), the perfect fifth ($3/2$), and the perfect fourth ($4/3$). On the other hand, if a consonant interval is slightly out of tune, the resulting mismatch of almost-coinciding partials leads to a superposition of waves with slightly different frequencies, perceived of dissonance.

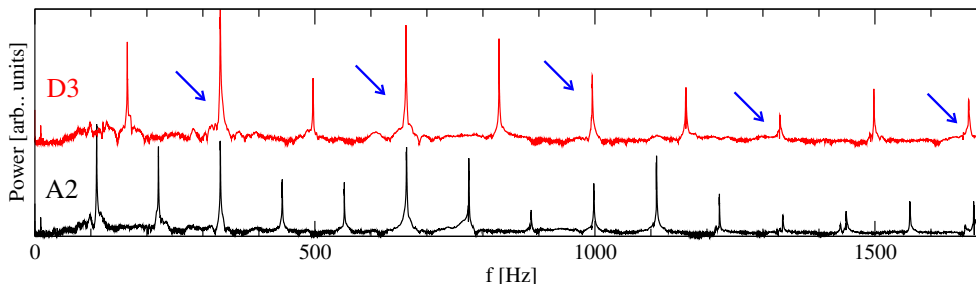


Figure 1. Consonance of a perfect fifth. The figure shows a measured power spectrum of the piano keys A2 (110 Hz) and D3 (165 Hz). The perfect fifth is perceived as consonant because many partials of the corresponding natural harmonic series coincide (marked by the arrows in the figure).

When fretted instruments and keyboards came into use it was necessary to define a system of fixed frequencies in an octave-repeating pattern. Since the frequency ratios of adjacent intervals multiply, this immediately confronts us with the fundamental mathematical problem that multiplication and prime numbers are incommensurate in the sense that powers of prime numbers never yield other simple prime numbers. For example, it is impossible to match k perfect fifths with ℓ octaves since $(3/2)^k \neq (2/1)^\ell$ for all $k, \ell \in \mathbb{N}$. Mathematically speaking, the concatenation of musical intervals (by multiplying their frequency ratios) is an operation that does not close up on a finite set of tones per octave.

Fortunately the circle of fifths approaches a closure up to a small mismatch [3]: when stacking twelve perfect fifths on top of each other the resulting frequency ratio $(3/2)^{12}$ differs from seven octaves (ratio 2^7) only by a small amount; explaining why the Western chromatic scale is based on twelve semitones per octave. Nevertheless the remaining difference of $\approx 1.4\%$ (23.46 cents \ddagger), known as the Pythagorean comma, is clearly audible and cannot be neglected in a scale with fixed frequencies. Likewise, a sequence of four perfect fifths transposed back by two octaves $((3/2)^4/2^2)$ differs from a perfect third ($5/4$) by the so-called syntonic comma of 21.51 cents.

Since it is impossible to construct a consistent musical scale that is based exclusively on pure beatless perfect intervals, one has to seek out suitable compromises. Over the centuries this has led to a fascinating variety of tuning systems, called temperaments, which reflect the harmonic texture of the music in the respective epoch [4]. With the increasing demand of flexibility the equal temperament (ET) finally prevailed in the 19th century and has established itself as a standard temperament of Western music. In the ET the octave is divided into twelve equally sized semitones of constant frequency ratio $2^{1/12}$. The homogeneous geometric structure of the ET ensures that all interval sizes are invariant under transposition (displacements on the keyboard). This means that music can be played in any key, differing only in the global pitch but not in the harmonic texture.

\ddagger In music theory, a cent (c) is defined as 1/100 of a semitone in the equal temperament.

However, this high degree of symmetry can only be established at the expense of harmony [5]. In fact, the only perfect interval in the ET is the octave with the frequency ratio 2/1 while all other intervals are characterized by irrational frequency ratios, deviating from the just intervals. For some intervals the variation is quite small and hardly audible, e.g., the equally tempered fifth differs from the perfect fifth by only two cents. For other intervals, however, the deviations are huge and clearly audible if not even disturbing. For example, the minor third in the ET is almost 16¢ smaller than the natural frequency ratio 6:5. The same applies to the major third which is about 14¢ greater than the ratio 5:4. These discrepancies may explain why the ET was only accepted with reluctance by musicians over many centuries. Only in the 19th century the growing harmonic dimensionality of music combined with an increasing intonational tolerance of the audience allowed the ET to become a new tuning standard.

Just Intonation

Although musical temperaments provided a good solution for most purposes, many musicians, music theorists and instrument makers have been searching over many centuries for possible ways to overcome the shortcomings of temperaments and for playing music solely on the basis of pure intervals, referred to as *just intonation* (JI) [6]. Tuning an instrument in just intonation means to adjust the twelve pitches of the octave in such a way that all frequency ratios with respect to a certain reference frequency f^* are given by simple rational numbers. For example, a possible choice would be [10]:

semitones	0	1	2	3	4	5	6	7	8	9	10	11
f/f^*	1	16/15	9/8	6/5	5/4	4/3	45/32	3/2	8/5	5/3	9/5	15/8

Just intonation always refers to a particular chordal root, the so-called keynote. In its own reference scale, JI sounds very consonant if not even sterile, but unfortunately a transposition in other scales is not possible. For example, tuning a piano in just intonation with keynote C, music played in C-major would be perfectly harmonic while playing in other scales would sound dissonant. The same applies to modulations from one key to another. Thus, for good reasons, just intonation has the reputation of being absolutely impractical.

To overcome this problem, a possible solution would be to enlarge the number of tones per octave. Important historical examples are for example a keyboard with 19 keys per octave suggested by the renaissance music theorist G. Zarlino (1558), and the two-manual archichembalo by N. Vicentino (1555) with 36 keys per octave [9]. This development culminated in 1914 with the construction of a harmonium with 72 keys per octave by Arthur von Oettingen [11]. More recently various types of electronic microtonal keyboards have been developed [12]. However, as one can imagine, such instruments are very difficult to play.

Temperaments are primarily relevant for keyboard instruments (such as piano, harp, organ) and fretted instruments (e.g. lute and guitar), where all tones are tuned statically in advance. In comparison many other instruments (e.g. string instruments) allow the musician to recalibrate the pitches during the performance, and the same applies of course to the human voice. Musicians playing such instruments tune the pitches *dynamically* while the music is being played. By listening to the harmonic consonance and its progression, well-trained musicians are able to estimate the appropriate frequency intuitively and to correct their own pitch instantaneously. Usually the played notes are a compromise between just intervals and the prevailing ET [6]. By dynamically adapting the pitches, this allows one to significantly improve the harmonic texture. In this respect the following quotation of the famous cellist Pablo Casals [14] seems to make a lot of sense:

“Don’t be scared if your intonation differs from that of the piano. It is the piano that is out of tune. The piano with its tempered scale is a compromise in intonation.”

Dynamically Adapting Tuning Schemes

Is it possible to mimic the process of dynamical tuning by constructing a device which instantaneously calculates and corrects the pitches like a singer in a choir? This idea can be traced back to the early days of electronic keyboard instruments. Since then various implementations have been suggested, the most important ones including *Groven-Max* [15], *Justonic Pitch Palette*TM [16], *Mutabor* [17], *Hermode Tuning*TM [18], and *TransFormSynth* [19]. These will be described in the following.

- One of the first pioneers of dynamic tuning schemes was the Norwegian microtonal composer and music-theorist Eivind Groven [15]. In 1936 he constructed a pipe organ driven by an electric circuit of relays used in telephone switchboards at that time. The organ had three sets of pipes differing by a syntonic comma, thus representing three lines of Euler’s Tonnetz [6, 20]. Playing a chord on the manual, the electro-mechanical logical circuit computed the optimal arrangement of the chord in this section of the Tonnetz and triggered the pipes accordingly. In 1995 this method was implemented on a computer, redirecting the output of a MIDI keyboard to three MIDI pianos differing in pitch by a syntonic comma [21].
- *Justonic Pitch Palette*TM was a proprietary software produced by Justonic Tuning, Inc. based on a patented method developed by J. William Gannon and Rex A. Weyler [16]. This tuning method is dynamic in so far as the artist himself can change the keynote frequency f^* during the performance by hand. The corresponding frequencies are then retrieved from a table and sent to a microtonal synthesizer. The selection of the keynote requires additional hardware such as an extra manual.
- *Mutabor* is a microtonal software project initiated by M. Vogel at the University

of Darmstadt [17]. The first version, referred to as Mutabor I, was designed as “*a system with 171 steps per octave for electronic keyboard instruments*”. Depending on the actual chord being played, the program calculates the keynote and tries to tune all frequencies in pure fifths, fourths, and thirds. However, this leads to audible frequency shifts between subsequent chords. From 1987 on Mutabor II improved this method, aiming to produce a usable PC-software for MIDI devices and allowing the user to develop individual tuning algorithms. In a third stage starting in 2006, Mutabor has evolved into a full-fledged microtonal programming language. Nevertheless, for music with a greater harmonic complexity, the keynote is not always detected reliably. As a solution, Mutabor offers the user to pre-determine the succession of keynotes in a separate MIDI file.

- *Hermode Tuning*TM is a commercial adaptive tuning scheme developed by Werner Mohrlök [18]. To our knowledge it is the only adaptive tuning scheme that has reached a wider dissemination, ranging from implementations in church organs to plugins for software packages such as *Cubase*TM. Instead of determining a keynote, the algorithm tunes intervals between adjacent tones of a given chord instantly to just ratios. At the same time the global pitch is adjusted in such a way that the difference to the usual ET is minimized. This reduces the disturbing frequency shifts whenever the keynote changes. Hermode tuning also tries to compensate the so-called pitch drift (see section 4.3).
- *TransFormSynth* is a freely available software-based synthesizer developed by William A. Sethares [19]. Unlike all other approaches – including the one presented here – which are based on the idea of dynamically modifying the fundamental frequencies and thereby the whole series of corresponding partials, Sethares proposes to keep the fundamental frequencies fixed (e.g. according to the ET). Instead, his algorithm modifies the frequencies of the higher partials such that they match. As a result, even though the overtone spectra are distorted, the synthesized sound is nevertheless perceived as consonant[§]. As far as we can see, this method is restricted to synthesizers which allow the overtone spectra to be specified individually, but it cannot be applied to ordinary musical instruments with natural harmonic overtone spectra.

All these methods are similar in that they analyze a given chord and then make decisions for tuning the frequencies, i.e. they are defined in procedural terms. Depending on the harmonic context, these decisions can be quite complex with different possible solutions for the same situation.

In the present paper, we propose an alternative adaptive tuning scheme based on a different method. Instead of making a sequence of *if-then* decisions, our method is defined mathematically and amounts to continuously solving a system of linear

[§] A similar phenomenon occurs in the context of piano tuning. Since the overtone spectra of stiff steel strings are slightly inharmonic, piano tuners stretch the tuning in order to compensate these deviations.

equations. As is described below, the system of equations may be viewed as a resistor network or likewise as a mechanical network of springs representing the interval sizes. Roughly speaking, each spring prefers to relax into a state where its length corresponds to the natural size of a pure interval and it will do so whenever this is possible, producing a chord in just intonation. However, if the spring network is so complex that it is impossible for all springs to simultaneously be situated in their ground state, the system will approach a non-trivial state under tension, representing a tempered harmonic compromise. This happens automatically without making any explicit decisions and resembles the way in which musicians find the best possible intonation. As another advantage, our method finds a harmonic compromise for *all* intervals in a given chord, not only of the intervals between adjacent tones.

To demonstrate the technical feasibility of the dynamically adjusting tuning scheme proposed in this paper, we implemented the tuning algorithm in an open-source application which is available for various platforms including mobile devices. This software will be described in more detail at the end of the paper.

2. Definitions and Notations

2.1. Frequencies and the Equal Temperament

In the following we consider a standard chromatic keyboard with keys enumerated from left to right by the index $k \in \{0, \dots, K - 1\}$.^{||} For traditional instruments the corresponding frequencies f_0, \dots, f_{N-1} are static throughout the performance, meaning that they have to be tuned beforehand according to a certain temperament. As mentioned above, Western music of today is predominantly based on the *equal temperament* (ET) with twelve equally-sized semitones, defined by

$$f_k^{\text{ET}} := f_{k_0} 2^{(k-k_0)/12}, \quad (1)$$

where k_0 denotes the index of the reference key and f_{k_0} the corresponding reference frequency (usually A4 with $f_{k_0} = 440$ Hz).

An interval between two tones k and k' is characterized by a certain frequency ratio $f_{k'}/f_k$. In the ET this ratio is given by $f_{k'}^{\text{ET}}/f_k^{\text{ET}} = 2^{(k'-k)/12}$. The main advantage of the ET is that this ratio depends only on the difference $k' - k$, meaning that the frequency ratios are invariant under transpositions (key changes $k \rightarrow k + \text{const}$). Therefore, apart from the global pitch, the ET sounds identical in all keys.

^{||} For example, in the Midi norm [23] k runs from 0 to 127 with the reference key A4 located at $k_0 = 69$.

$k' - k$	Interval	$f_{k'}^{\text{ET}}/f_k^{\text{ET}}$	$R_{k,k'}^{\text{JI}} = m/n$	$\Phi_{k,k'}^{\text{JI}}[\zeta]$	$\Phi_{k,k'}^{\text{ET}}[\zeta]$	Deviation $\phi_{k,k'}^{\text{JI}}[\zeta]$
0	Unison	1	1	0	0	0
1	Semitone	1.0595	16/15	111.73	100	+11.73
2	Major Second	1.1225	9/8	203.91	200	+3.91
3	Minor Third	1.1892	6/5	315.64	300	+15.64
4	Major Third	1.2599	5/4	386.31	400	-13.69
5	Fourth	1.3348	4/3	498.04	500	-1.96
6	Tritone	1.4142	45/32	590.22	600	-9.78
7	Fifth	1.4983	3/2	701.96	700	+1.96
8	Minor Sixth	1.5874	8/5	813.69	800	+13.69
9	Major Sixth	1.6818	5/3	884.36	900	-15.64
10	Minor Seventh	1.7818	9/5	1017.60	1000	+17.60
11	Major Seventh	1.8878	15/8	1088.27	1100	-11.73
12	Octave	2	2	1200	1200	0

Table 1. Possible choice for frequency ratios of justly tuned chromatic intervals. The corresponding interval sizes in just intonation (JI) are compared with the standard equal temperament (ET). Note that the choice of m/n for a given interval is not always unique. For example, the minor seventh can be tuned according to the ratio 9/5 or 7/4 (see Table 2 in Sect.4).

2.2. Consonance and Just Intonation

Two tones are perceived as consonant (justly intonated) if the corresponding frequency ratio $f_{k'}/f_k$ is given by a simple rational number $R = m/n$. For example, an octave has the frequency ratio 2:1 while the perfect fifth corresponds to 3:2 (see Table 1). The impression of consonance is particularly pronounced if m and n are small.

Just intonation (JI) is a tuning scheme where the frequencies f_k are tuned according to certain rational numbers with respect to a certain keynote k^* in an octave-repeating pattern:

$$f_k^{\text{JI}} = R_{k,k^*}^{\text{JI}} f_{k^*}^{\text{JI}}. \quad (2)$$

A possible choice of the fractions R_{k,k^*}^{JI} is given in Table 1. With respect to the keynote the resulting interval ratios $f_k^{\text{JI}}/f_{k^*}^{\text{JI}}$ are exactly those listed in the table. However, in contrast to the ET these frequency ratios are not invariant under key changes. For example, for keynote C the fifth C-G has the correct frequency ratio 3:2 but the fifth D-A has the ratio $40/27 \simeq 1.48$ which is clearly too small. Therefore, as already outlined in the introduction, JI as a statically tuned temperament can only be used in the scale referring to its keynote (and a few complementary keys) while it sounds dissonant in most other keys.

2.3. Logarithmic pitches and interval sizes

The corresponding frequency ratios multiply when combining several musical intervals in a chord. Therefore, as already pointed out by Huygens in 1691 [3,24], it is convenient

to consider the *logarithm* of frequency ratios, quantified in units of cents. Using this convention we define pitches and interval sizes as follows:

- The *absolute pitch* Λ_k of the key k is defined as the cent difference between the key k and the reference key k_0 (A4):

$$\Lambda_k := 1200 \log_2 \left(\frac{f_k}{f_{k_0}} \right) = 1200 \left(\log_2 f_k - \log_2 f_{k_0} \right), \quad (3)$$

where $\log_2 f = \log f / \log 2$ denotes the logarithm to the base 2. For example, the pitches in the ET (1) are simply given by multiples of 100 cent:

$$\Lambda_k^{\text{ET}} = 100(k - k_0). \quad (4)$$

For the actual *pitch deviation* of the key k relative to the ET we use the notation

$$\lambda_k := \Lambda_k - \Lambda_k^{\text{ET}}. \quad (5)$$

- The microtonal *absolute interval size* $\Phi_{k,k'}$ between two keys k and k' is defined as the corresponding pitch difference in units of cents:

$$\Phi_{k,k'} = \Lambda_{k'} - \Lambda_k = 1200 \left(\log_2 f_{k'} - \log_2 f_k \right). \quad (6)$$

In the ET (1) the interval sizes are given by the number of semitones times 100:

$$\Phi_{k,k'}^{\text{ET}} = \Lambda_{k'}^{\text{ET}} - \Lambda_k^{\text{ET}} = 100(k' - k). \quad (7)$$

For the actual *interval size deviation* from the ET we use the notation

$$\phi_{k,k'} := \Phi_{k,k'} - \Phi_{k,k'}^{\text{ET}} = \lambda_{k'} - \lambda_k. \quad (8)$$

For JI a list of possible values for $\Phi_{k,k'}^{\text{JI}}$ and $\phi_{k,k'}^{\text{JI}}$ is given in Table 1.

3. Vertical Intonation – Adaptive Tuning of a Single Chord

Adaptive tuning schemes are confronted with two important aspects of tuning. On the one hand, each new chord has to be intonated ‘vertically’, that is, one has to tune the relative pitches between simultaneously played notes. On the other hand, subsequent chords have to be intonated relative to each other in ‘horizontal’ (temporal) direction according to the harmonic progression, as it will be discussed in the following section.

3.1. Vertical tuning at first glance

In order to tune a given chord vertically, we want to determine the pitches in such a way that the resulting interval sizes are equal or at least close to the ideal ratios of JI. In other words, for a chord consisting of N keys $\{k_1, k_2, \dots, k_N\}$ we have to find pitches $\{\Lambda_{k_1}, \dots, \Lambda_{k_N}\}$ such that the interval sizes Φ_{k_i, k_j} agree as much as possible with the values $\Phi_{k_i, k_j}^{\text{JI}}$ listed in Table 1.

Most of the existing approaches mentioned in the introduction consider only the intervals between *adjacent* tones of a chord. Contrarily our method also takes intervals between non-adjacent tones into account on equal footing with the others. This means that for a chord consisting of N tones there are $N(N - 1)/2$ possible intervals which have to be tuned as just as possible.

As there are $N(N - 1)/2$ intervals but only N degrees of freedom, it is clear that it is not always possible to find a consistent solution where all interval sizes Φ_{k_i, k_j} match exactly with the given cent differences $\Phi_{k_i, k_j}^{\text{JI}}$. For example, the triad C-E-G can be tuned in just intonation while the triad C-E-G $^\sharp$ cannot.¶ In the latter case the algorithm should render an acceptable tempered compromise. In fact, this is basically what musicians do: they do not solve complicated puzzles of number theory, instead they simply adjust their own pitch on an intuitive basis in such a way that the best possible harmonic compromise is achieved.

The solution proposed in this paper is based on a simple idea which can be explained as follows. As sketched in Fig. 2, we consider a fictitious battery-resistor network, where each battery has a voltage corresponding to the ideal pitch difference $\Phi_{k_i, k_j}^{\text{JI}}$ of JI. If the chord can be tuned justly (e.g. as a major triad) the voltages will adjust exactly at the corresponding pitches and the currents passing the resistors are zero. Otherwise, for chords which cannot be tuned perfectly, the network will produce a certain compromise which depends on the specific choice of the resistors. As already pointed out above, the dissipated power can be regarded as a measure how strongly this tuning compromise is tempered.

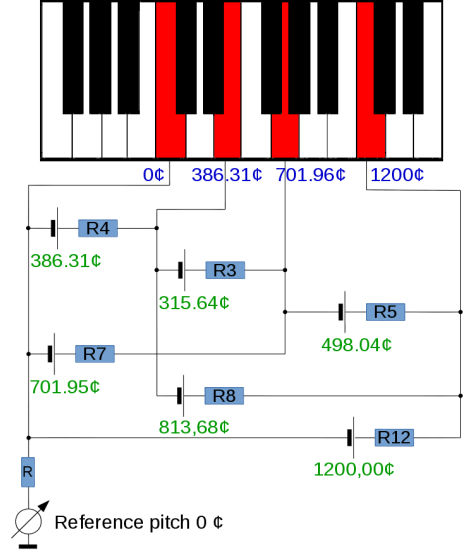
3.2. Mathematical formulation

Consider a chord of N tones with key indices k_1, k_2, \dots, k_N in increasing order. The chord consists of $N(N - 1)/2$ intervals with index pairs $i, j \in \{k_1, \dots, k_N\}$ ordered by $i < j$. The task would be to tune the pitches Λ_i (with $i \in \{k_1, \dots, k_N\}$) in such a way that the pitch differences $\Lambda_j - \Lambda_i$ deviate as little as possible from the ideal pitch differences $\Phi_{k_i, k_j}^{\text{JI}}$ listed in Table 1, or equivalently, that the differences $\lambda_j - \lambda_i$ deviate as little as possible from $\phi_{i, j}^{\text{JI}} := \phi_{k_i, k_j}^{\text{JI}}$. We solve this problem by minimizing the squared deviations as follows. Denoting by $\vec{\lambda} = (\lambda_{k_1}, \dots, \lambda_{k_N})^T$ the vector of pitch deviations from the ET of the pressed keys, we define a deviation potential by

$$V[\vec{\lambda}] = \frac{1}{4} \sum_{\substack{i, j \in \{k_1, \dots, k_N\} \\ i < j}} w_{ij} \left(\lambda_j(t) - \lambda_i(t) - \phi_{ij}^{\text{JI}} \right)^2, \quad (9)$$

¶ C-E-G: Tuning the major third C-E to the ratio $5/4$ and the minor third E-G to the ratio $6/5$, the resulting fifth C-G has the ratio $\frac{5}{4} \cdot \frac{6}{5} = \frac{3}{2}$, meaning that the triad is perfectly consonant in just intonation. Contrarily, for C-E-G $^\sharp$, where two major thirds are combined, the resulting augmented fifth would have the frequency ratio $\left(\frac{5}{4}\right)^2 = \frac{25}{16}$ which differs from the ratio $\frac{8}{5}$ listed in the table. This means that this chord cannot be tuned consistently in just intonation.

Figure 2. Simplified sketch of the vertical tuning scheme suggested in the present paper. The figure shows a keyboard on which a C-Major triad is played. Viewing these keys as electrical contacts we place a battery in series with a resistor between each of the $4 \cdot 3/2 = 6$ possible intervals. Assuming that each battery has a voltage equal to the ideal pitch difference $\Phi_{k_i, k_j}^{\text{JI}}$ in JI, the resistor network will attain an equilibrium according to Kirchhoff's laws where the voltages at the key contacts represent the desired microtonal pitches. If all electrical currents in the network vanish (as in the present example) the chord is tuned exactly in JI. If not, the specific choice of the resistors determines a tempered compromise where the dissipated power measures how much the chord is tempered. The system is coupled to an external voltage which controls the reference pitch.



which is just the sum of all quadratic deviations of the interval sizes weighted by certain factors w_{ij} , assuming that $w_{ii} = 0$. The weights can be chosen freely and can be viewed as the conductivity of the resistors in Fig. 2. Their purpose is to determine the ‘rigidity’ of the respective interval in the tuning process. In practice it is meaningful to assign a high weight factor to intervals with simple fractional ratios. In addition, the weight factors may also take the actual volume of the participating tones into account.

In a more compact notation, the deviation potential (9) can be written in the bilinear form

$$V[\vec{\lambda}] = \frac{1}{2} \vec{\lambda} \cdot \mathbf{A} \vec{\lambda} + \vec{b} \cdot \vec{\lambda} + c, \quad (10)$$

where \mathbf{A} is a symmetric $N \times N$ matrix and \vec{b} is a vector with the components

$$A_{ij} = \begin{cases} -w_{ij} & \text{if } i \neq j \\ \sum_{\ell} w_{i\ell} & \text{if } i = j \end{cases}, \quad b_i = \sum_j w_{ij} \phi_{ij} \quad (11)$$

and where

$$c = \frac{1}{4} \sum_{i,j} w_{ij} \phi_{ij}^2 \quad (12)$$

is a constant. The optimal pitches $\vec{\lambda}^{\text{opt}}$, in which we would like to tune the chord, correspond to a situation where $V[\vec{\lambda}]$ is minimal, that is $\vec{\nabla}_{\lambda} V = \vec{0}$, leading to the system of equations $\mathbf{A} \vec{\lambda} + \vec{b} = \vec{0}$. Thus, if \mathbf{A} was invertible, the solution would be given by

$$\vec{\lambda}^{\text{opt}} = -\mathbf{A}^{-1} \vec{b}. \quad (13)$$

Thus, the whole tuning process amounts to solving a system of linear equations. Finally, the potential

$$V[\vec{\lambda}^{\text{opt}}] = c - \frac{1}{2} \vec{b} \cdot \mathbf{A}^{-1} \vec{b} \quad (14)$$

evaluated at $\vec{\lambda} = \vec{\lambda}^{\text{opt}}$ gives the dissipated power, telling us to what extent the result is tempered.

However, inspecting \mathbf{A} one can easily see that the column sum is zero, hence the matrix does not have full rank and thus it is not invertible. This can be traced back to the fact that the potential is defined in pitch *differences*, leaving the absolute pitch of the chord undetermined. This can be circumvented easily by coupling the network to an external source which determines the global concert pitch, as described in Appendix A.

4. Horizontal Intonation – Adaptive Tuning in the Harmonic Progression

Normally we perceive combinations of tones as ‘in tune’ or ‘out of tune’ if they are played simultaneously. However, we are also capable of recognizing the consonance of subsequently played tones, provided that the elapsed time in between is not too large. Apparently our sense of hearing is able to memorize sounds and their spectra for a short while. In empirical studies it was found that this psychoacoustic intonational short-term memory is characterized by a typical time scale of about three seconds [25, 26].

If the chords are tuned separately as described in the last section, the change of the keynote may lead to unpleasant intonational discontinuities between chords. This requires that the intonational memory has to be taken into account by correlating the pitches of subsequent chords in a harmonic progression. In the following we describe how the intonational memory can be incorporated in the suggested framework of adaptive tuning.

4.1. Intonational memory

Pressing a key k the instrument produces a sound with the time-dependent intensity (volume) $I_k(t)$ which decays to zero when the key is released. In order to implement the intonational memory, we introduce a *memory function* $M_k(t)$ interpreted as the ‘virtual intensity’ at which the sound of a key k is memorized. When a new key is pressed $M_k(t)$ is initially set to the actual intensity $I_k(t)$. Thereafter, it follows $I(t)$ adiabatically by means of the first-order differential equation

$$\frac{dM_k(t)}{dt} = \frac{1}{\tau_M} \left(I_k(t) - M_k(t) \right), \quad (15)$$

where $\tau_M \approx 3\text{s}$ is the characteristic time scale of the intonational memory. For example, if the volume of a key drops suddenly to zero after releasing a key, $M(t)$ will decrease

exponentially as e^{-t/τ_m} .

This simple model of the intonational memory can be improved further by observing that it also takes some time to recognize the pitch of a newly pressed key. In fact, it is quite easy to memorize the pitches of long sustained notes while individual short notes at high tempo are much harder to remember. This suggests that there is another typical time scale τ_R for recognizing the pitch of a sound which may be taken into account by considering the dynamics

$$\frac{dM_k(t)}{dt} = \begin{cases} \frac{1}{\tau_R} (I_k(t) - M_k(t)) & \text{if } M_k(t) < I_k(t) \\ \frac{1}{\tau_M} (I_k(t) - M_k(t)) & \text{if } M_k(t) \geq I_k(t) \end{cases}. \quad (16)$$

Among musicians the typical value of τ_R is expected to be smaller than τ_M , and it seems that values in the order of one second are a reasonable choice.

4.2. Horizontal adaptive tuning

In order to correlate the intonation of subsequent chords, we use the same mechanism as described above for the case of vertical tuning. To this end let us consider a memorized key with the index k_m that was tuned to the pitch $\tilde{\Lambda}_m$, followed by a newly pressed key with the index k_i (including the case that the same key is pressed again). The aim is to tune the pitch $\Lambda_i(t)$ dynamically in such a way that the interval size $\tilde{\Lambda}_m - \Lambda_i(t)$ approximates as much as possible the ideal interval size $\Phi_{k_i, k_m}^{\text{JI}}$ of JI, as listed in Table 1. In other words, we have to determine $\vec{\lambda}$ in such a way that $\tilde{\lambda}_m - \lambda_i(t)$ deviates as little as possible from $\phi_{im}^{\text{JI}} := \phi_{k_i, k_m}^{\text{JI}}$. This leads to simply adding a memory term in the potential

$$V[\vec{\lambda}] = \frac{1}{4} \sum_{i,j} w_{ij}(t) (\lambda_j(t) - \lambda_i(t) - \phi_{ij}^{\text{JI}})^2 + \frac{1}{2} \sum_i \sum_m \tilde{w}_{im}(t) (\tilde{\lambda}_m - \lambda_i(t) - \phi_{im}^{\text{JI}})^2 \quad (17)$$

where k_i, k_j with $i, j \in \{1, \dots, N\}$ run over all playing keys while k_m with $m \in \{1, \dots, M\}$ runs over the memorized keys. Here $\tilde{w}_{im}(t)$ is a time-dependent weight factor reflecting the actual intensity of the key k_i and the memorized intensity of the key k_m . Again this potential can be written in the form (10) with

$$\mathbf{A}_{ij} = \begin{cases} -w_{ij} & \text{if } i \neq j \\ \sum_{\ell} w_{i\ell} + \sum_m \tilde{w}_{im} & \text{if } i = j \end{cases} \quad (18)$$

$$b_i = \sum_j w_{ij} \phi_{ij} + \sum_m \tilde{w}_{im} (\phi_{im} - \tilde{\lambda}_m) \quad (19)$$

$$c = \frac{1}{4} \sum_{i,j} w_{ij} \phi_{ij}^2 + \frac{1}{2} \sum_i \sum_m \tilde{w}_{im} (\phi_{im} - \tilde{\lambda}_m)^2 \quad (20)$$

and its minimum is attained at $\vec{\lambda}^{\text{opt}} = -\mathbf{A}^{-1}\vec{b}$. Note that this method automatically finds a tempered compromise if the keynotes of a harmonic progression are incompatible.

4.3. Pitch Drift

One of the major disadvantages of dynamical tuning schemes with temporal correlation is the gradual migration of the overall pitch. For example, playing a full chromatic scale of 12 semitones with fixed sizes $\Phi_{k,k+1}^{\text{JI}}=111.73\text{c}$ one ends up with 1340.76c , which is more than a half tone higher than an octave. In practice the pitch drift meanders in positive and negative direction, depending on the actual harmonic progression.

The pitch drift can be reduced by admitting different interval sizes, as will be explained in the next section. For example, 12 semitones with alternating sizes of 111.73c and 92.18c (forming six whole tones of 3.91c) add up to 1223.46c which is much closer to a just octave of 1200c . But even this improvement does yet not eliminate the pitch progression entirely.

It is therefore meaningful to implement an additional pitch drift compensating mechanism by slowly adjusting all pitches uniformly in such a way that the desired reference pitch is approached, as described by the differential equation

$$\frac{d\lambda_i}{dt} = \frac{1}{\tau_c} \left(\left(\frac{1}{N} \sum_{i \in \{k_1, \dots, k_N\}} \lambda_i \right) - \lambda^{\text{ref}} \right), \quad (21)$$

Here $\lambda^{\text{ref}} = 1200 * \log_2(f_{k_0}/440 \text{ Hz})$ is the global pitch versus 440 Hz while τ_c defines the typical time scale on which the compensation takes place. This time scale should be chosen such that it makes the compensation not noticeable for the listener.

5. Dealing with Non-Unique Interval Sizes

So far the method outlined above determines the best possible tuning result for a fixed table of interval sizes (see Table 1). However, the frequency ratios in JI are not unique; rather there are various possible choices for certain intervals, defining different variants of just intonation (see Table 2). It turns out that the tuning result can be improved by finding the best possible solution among these variants [27].

The advantage of admitting several variants can be explained by the following example. Two successive major seconds with the ratio of $9/8$ ($+3.9\text{c}$) make up a major third with the ratio of $81/64$ ($+7.8\text{c}$) which differs significantly from the perfect ratio of $5/4$ (-13.7c). On the other hand, after combining two different justly intonated variants of major seconds with the ratios of $9/8$ ($+3.9\text{c}$) and $10/9$ (-17.6c) the resulting major third has exactly the just frequency ratio of $5/4$ (-13.7c).

What determines the correct choice of the interval size? In a procedural setting this is a highly complex music-theoretical problem that requires a thorough analysis of the harmonic progression. But in practice the decision for the best fitting interval is made aurally on an intuitive basis (although a general understanding of the harmonic progression is part of the process). Inspired by this observation we implement non-

$k' - k$	Interval Name	Tuning alternatives p/q ($\phi_{k,k'}^{\text{JI}}[\text{¢}]$)		
0	Unison	1/1 (0)		
1	Semitone	16/15 (+11.73)	25/24 (-29.32)	
2	Major Second	9/8 (+3.91)	10/9 (-17.60)	
3	Minor Third	6/5 (+15.64)		
4	Major Third	5/4 (-13.69)		
5	Fourth	4/3 (-1.96)		
6	Tritone	45/32 (-9.78)		
7	Fifth	3/2 (+1.96)		
8	Minor Sixth	8/5 (+13.69)		
9	Major Sixth	5/3 (-15.64)		
10	Minor Seventh	16/9 (-3.91)	9/5 (+17.60)	7/4 (-31.17)
11	Major Seventh	15/8 (-11.73)		
12	Octave	2/1 (0)		

Table 2. List of possible frequency ratios of just chromatic intervals. The third column contains the same data as in Table 1. In addition the most important alternative tuning ratios are shown.

unique interval sizes in the algorithm described above by repeating the minimization procedure for all possible combinations of the alternative interval sizes listed in Table 2 and then to take the solution with the lowest deviation from JI.

6. Open-Source Demonstration Software – Technical Details

In order to demonstrate the tuning method introduced in this paper, we initiated an open-source project called *Just Intonation*.⁺ This software allows the user to hear and play music with and without adaptive tuning. The application has been designed as an

⁺ See just-intonation.org. The source code can be downloaded from gitlab.com/tp3/JustIntonation. The present paper refers to version 1.3.1



Figure 3. Snapshot of the application *Just Intonation* in the expert mode.

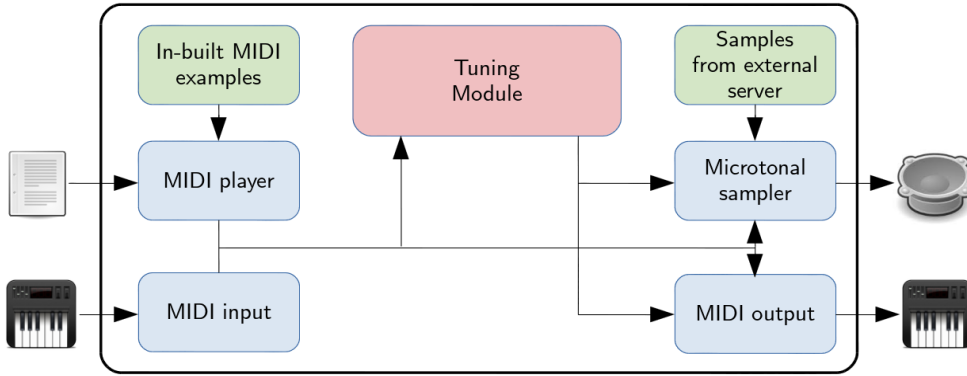


Figure 4. Basic structure of the application and its modules.

educational application rather than a professional tool for producing music. It provides a *simple mode* for getting started as well as an *expert mode* for more sophisticated experiments (see screenshot in Fig. 3).

Just Intonation is a multiplatform application for desktop computers and mobile devices written in C++ and is based on the Qt framework*. As sketched in Fig. 4, it contains various submodules which partially run in different threads and communicate with each other via Qt signal-slot messages. MIDI messages generated by an inbuilt player or an external device are sent to the tuning module and the sound-generated modules. Depending on the MIDI data the tuning module continually computes the vector $\vec{\lambda}^{\text{opt}}$ according to the formulas given above and emits the pitches via Qt signals to the audio modules. The application includes a microtonal sampler which can play triangular waves as well as non-standardized samples recorded by the authors. As the application is designed primarily for educational purposes, there is no particular emphasis on low-latency audio. Alternatively, it is possible to connect an external MIDI device which is capable of processing 'pitch-bends'. Normally the MIDI pitch-bend command modifies the frequencies of all keys uniformly. In order to circumvent this restriction, the MIDI output module remaps the incoming MIDI stream to 15 different channels, tuning each of them individually via pitch-bend.‡

The MIDI interface and the sampler work in a similar way as in many other applications. Therefore, let us focus here on the tuner module which can be found in `application/modules/tuner`. This module, realized as an instance of the class `Tuner`, runs entirely in a separate event loop of an independent thread and communicates with the application via Qt signals, ensuring thread safety. Its internal structure is shown in Fig. 5. Its main functionality is

* Qt™ is a cross-platform application framework licensed under LGPL that is used for developing application software.

‡ This restricts the output to a 15-voice polyphony. As of this writing, the MIDI output module is still in an experimental state.

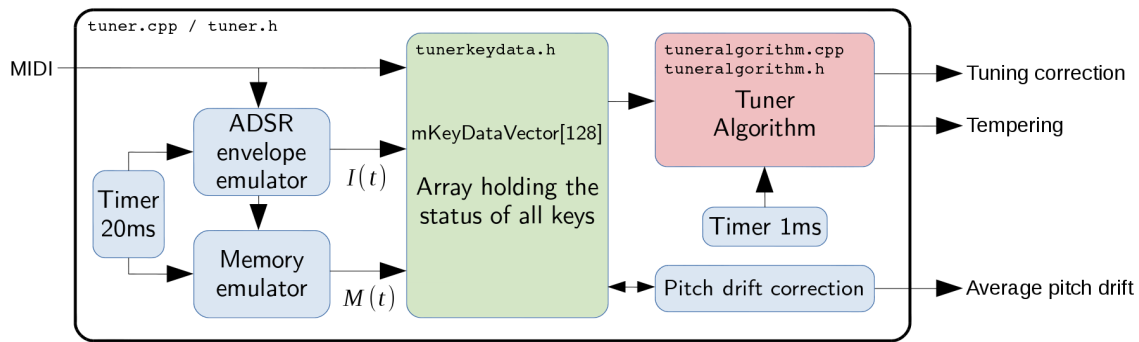


Figure 5. Internal structure of the tuner module.

- receiving MIDI signals and sending tuning signals,
- emulating the intensity $I(t)$ as well as the memory $M(t)$ for each key,
- keeping track the status of the keys in an array of type `KeyData`,
- executing the `TunerAlgorithm` every 20ms or upon incoming MIDI events, and
- managing the adiabatic pitch drift compensation.

The current status of the keys is stored in a vector named `mKeyDataVector`. The vector components hold structures of type `KeyData` with the following fields:

```
struct Keydata {
    int key;                ///< Number of the key
    bool pressed;          ///< Flag indicating a pressed key
    bool newlyPressed;    ///< Flag indicating a newly pressed key
    bool emitted;         ///< Flag indicating first emitted tuning
    double intensity;     ///< Intensity (volume) I(t)
    double memory;        ///< Psychoacoustic memory M(t)
    double sustainDampingFactor; ///< Sustain damping of the key
    double releaseDampingFactor; ///< Release damping of the key
    qint64 timeStamp;     ///< Time when the key was pressed
    double pitch;         ///< Actual pitch
    double previousPitch; ///< Previously emitted pitch
}
```

The `TunerAlgorithm` (see `application/modules/tuneralgorithm.cpp`) is the core of the application. Its essential part is a function for dynamical tuning with the signature

```
double TunerAlgorithm::tuneDynamically (KeyDataVector &keyDataVector,
                                         const QVector<double> intervals,
                                         const QVector<double> weights,
                                         bool optimizedJI)
```

The first parameter passes a non-constant reference to the vector of `KeyData` mentioned before. The second parameter contains a vector of twelve interval size deviations

(for example the values $\phi_{k-k'}^{\text{JI}}$ listed in Table 1 or the values for some different temperament), while the third parameter passes the corresponding weight factors. If the flag `optimizedJI` is true, the algorithm selects the optimal choice of interval sizes among various alternatives from a hard-coded list (see Table 2), ignoring the second parameter. However, for a better readability we sketch in the following the essential parts of the code *without* the optimization mechanism described in the previous section.

The implementation of the tuner algorithm is based on the open-source library *Eigen*†† which is used here to solve the system of linear equations described above. Under normal circumstances *Eigen* can find the solution within milliseconds, but as the number of equations grows quadratically with the number of pressed keys, an exceptionally large number of pressed keys could effectively lead to a freeze. Therefore, the first step is to identify the first N most important keys truncated by $N \leq N_{\max}$. To this end we first collect all audible and memorized keys in maps, exploiting the fact that they are automatically sorted by decreasing $I(t)$ or $M(t)$:

```
QMultiMap<double,int> audibleKeys, memorizedKeys;
for (auto &key : keyDataVector) {
    if (key.intensity>0) audibleKeys.insert (-key.intensity, key.key);
    if (key.memory>0)    memorizedKeys.insert(-key.memory, key.key);
}
```

Next we copy the most important entries to a vector named `keys`, truncating its size:

```
const int Pmax=8, Nmax=16;
QVector<int> keys = audibleKeys.values().mid(0,Pmax).toVector();
const int P = keys.size();
keys += memorizedKeys.values().mid(0,Nmax-P).toVector();
const int N = keys.size();
```

Now the size of the vector `keys` is restricted by $N \leq N_{\max}$ and it contains $P \leq P_{\max}$ audible keys followed by $N - P$ memorized keys. Note that the memorized keys may contain audible ones as well.

Defining the *significance* of a key as $I(t) + M(t)$, we create two vectors containing the pitches and the significances of the involved keys:

```
VectorXd pitch(N), significance(N);
for (int i=0; i<N; ++i) {
    KeyData& key = keyDataVector[keys[i]];
    pitch(i) = key.pitch;
    significance(i) = key.intensity + key.memory;
}
```

†† *Eigen* is a C++ library for linear algebra, see eigen.tuxfamily.org. As a template library it consists exclusively of header files. *Eigen* is included as third-party software in the present project.

In the first line `VectorXd` is a vector type provided by the library *Eigen*. Similarly, we define matrices for the number of semitones, the direction and the weight for each interval between pairs of keys contained in `keys`:

```
MatrixXi semitones(N,N), direction(N,N);
MatrixXd weight(N,N); weight.setZero();
MatrixXd interval(N,N); interval.setZero();

for (int i=0; i<N; ++i) for (int j=0; j<N; ++j) {
    semitones(i,j) = abs(keys[j] - keys[i]);
    direction(i,j) = (keys[j] >= keys[i] ? 1 : -1);
    interval(i,j) = direction(i,j) * intervals[semitones(i,j)%12];
    weight(i,j) = weights[semitones(i,j)%12] *
                std::pow(octaveWeightFactor, (semitones(i,j)-1)/12) *
                sqrt(significance(i) * significance(j));
    if (i>P or j>P) if (keys[i]!=keys[j]) weight(i,j) *= memoryWeight;
}
```

Here the twelve weights and interval sizes passed in the second and third parameter are periodically repeated on octaves, weighting each octave by an additional `octaveWeightFactor`. In addition, the weight is multiplied by the geometric mean of the corresponding significances defined above. Moreover, weight factors between currently played and memorized keys are multiplied by a constant called `memoryWeight` that can be used to control the influence of horizontal tuning. Finally, the contributions $-\tilde{\lambda}_m$ in (19) and (20) are taken into account by subtracting the pitches:

```
for (int i=0; i<P; ++i) for (int j=P; j<N; ++j) {
    interval(i,j) -= pitch(j);
    interval(j,i) += pitch(j);
}
```

Using the *Eigen* library, Eqs. (18)-(20) can be implemented as

```
VectorXd diagonal = VectorXd(weight.array().rowwise().sum());
MatrixXd A = -weight.block(0,0,P,P) + MatrixXd::Identity(P,P)*epsilon
            + MatrixXd(diagonal.head(P).asDiagonal());
VectorXd B = (interval * weight).diagonal().head(P) - epsilon*pitch.head(P);
double C = (interval.array() * interval.array() * weight.array()).sum() / 4;
```

Having prepared all matrices and vectors, the tuning algorithm just takes a single line:

```
VectorXd U = - A.inverse() * B;
```

Now the vector `U` contains the desired microtonal pitch deviations which are submitted to the sampler (together with the corresponding key numbers). According to (14), the dissipated power V , measuring the degree of tempering, can be calculated by

```
double V = C - B.dot(AI*B)/2;
```

This completes the description of the tuner algorithm. For more details please refer to the original files on gitlab.com/tp3/JustIntonation and the documentation of the source code on doxygen.just-intonation.org.

7. Outlook

The fascinating development of temperaments and the ongoing tug-of-war between just tuning and key invariance over many centuries is certainly one of the most exciting aspects in the history of music theory and practice. At the beginning of the 20th century it seemed as if the universal acceptance of the ET had finally settled this issue and, in fact, still today this is the prevailing view. In contrast, we share the opinion that the quest for a better intonation is not over yet and that the ET is probably only an intermediate rather than a final solution.

Looking back at the past 150 years it seems that the search for a better intonation oscillates between enthusiasm and disillusionment. For example, starting with the seminal work by Helmholtz [29] in 1865, who was among the first to provide a scientific basis for the sensation of music, many theorist and instrument makers at the end of the 19th century were inspired by the challenge to construct a “*Reininstrument*”, but the solutions were simply too complicated to become accepted on a broad scale. Then in the early 20th century the interest in just intonation abated, probably because of the seemingly unstoppable success of the ET.

In the second half of the last century, a renewed interest in just intonation and different ways of tuning arose alongside with an increasing attention on historical performance practices [6]. The new technologies becoming available constituted another promoting factor for this process. Following the visionary contributions by Eivind Grove [15], the emerging computer technology led to a variety of proposals, patents, and software packages which reflect the technological capabilities of the respective time. Unfortunately, apart from few exceptions, none of these approaches reached a broader dissemination, partly because of unpleasant squeaky computer sounds, and partly because the whole issue had maintained the reputation of being exotic and academic, linked to the somewhat elitist microtonal community where 12 tones per octave are considered merely as an exception rather than a rule. This may explain why the interest of the mainstream professionals in these activities declined once again during the past two decades.

Meanwhile, however, an ordinary mobile phone has become more powerful than a supercomputer in the 80's, offering new and previously undreamt-of possibilities. For example, solving a system of equations in real time, as proposed in the present work, would have been inconceivable two decades ago. Moreover, digital information technology continues to change the musical landscape and the art of instrument making entirely. The purpose of this project is to demonstrate that by now we have completely

different means at our disposal which allow us to consider different approaches and to make dynamic tuning schemes suitable for everyday use.

Finally, the emergence of electronic media enhances the interaction between different musical cultures. On the one hand, it is obvious that many traditional intonation systems throughout the world are increasingly influenced (if not even destroyed) by the Western ET. On the other hand, it should not be underestimated that this interaction also influences the Western world, and it cannot be ruled out that at some point in the future it may become fashionable to deviate from the ET. In addition, it is to be expected that the art of instrument making will continue to evolve rapidly and that on the long run the importance of statically tuned temperaments will probably decrease. All this suggests that dynamically adapting tuning scheme might become more important in the future. With our contribution we would like to point out that there is a lot open space for further research and development in this direction.

Acknowledgements

We would like to thank A. Whisnant for critical reading of the manuscript.

Appendix A. Controlling the Global Pitch

To solve the problem of non-invertibility we couple the network weakly to the global reference pitch, as indicated at the bottom of Fig. 2. This amounts to adding the term to the potential (10)

$$V[\vec{\lambda}] = \frac{1}{4} \sum_{i,j=1}^N w_{ij} \left(\lambda_j(t) - \lambda_i(t) - \phi_{ij}^{\text{II}} \right)^2 + \frac{\epsilon}{2} \sum_{i=1}^N (\lambda_i - \lambda_{\text{ref}})^2, \quad (\text{A.1})$$

where $\lambda_{\text{ref}} = 1200 \log_2(f_{k_0}/440 \text{ Hz})$ is the deviation from the reference pitch and $\epsilon \ll w_{ij}$ is a small coupling parameter, replacing Eqs. (11) and (12) by the modified expressions

$$\mathbf{A}_{ij} = \begin{cases} -w_{ij} & \text{if } i \neq j \\ \epsilon + \sum_{\ell} w_{i\ell} & \text{if } i = j \end{cases} \quad (\text{A.2})$$

$$b_i = -\epsilon \lambda_{\text{ref}} + \sum_j w_{ij} \phi_{ij} \quad (\text{A.3})$$

$$c = \frac{1}{4} \left(2N\epsilon \lambda_{\text{ref}}^2 + \sum_{i,j} w_{ij} \phi_{ij}^2 \right). \quad (\text{A.4})$$

Now the matrix \mathbf{A} is invertible and the optimal pitch differences can be computed by $\vec{\lambda}^{\text{opt}} = -\mathbf{A}^{-1}\vec{b}$. A similar modification can be made if horizontal tuning is taken into account.

References

- [1] H. E. White and D. E. White, *Physics and Music: The Science of Musical Sound*, Dover Books on Music, reprint (2014).
- [2] D. Geller, *Praktische Intonationslehre für Instrumentalisten und Sänger*, Bärenreiter (1997).
- [3] C. Huygens, *Lettre touchant le cycle harmonique*, Rotterdam (1691).
- [4] J. M. Barbour, *Tuning and Temperament: A Historical Survey*, Dover Books on Music (2004).
- [5] R. W. Duffin, *How Equal Temperament Ruined Harmony (and Why You Should Care)*, W. W. Norton & Company (2008).
- [6] R. W. Duffin, *Just Intonation in Renaissance Theory and Practice*, Music Theory Online (MTO) 12, 3 (2006).
- [7] see e.g.: *Die Musik in Geschichte und Gegenwart*, ed. by Friedrich Blume, Bärenreiter/Metzler (1998), mgg-online.com
- [8] G. Zarlino, *Le Istitutioni Harmoniche* (1558), online at IMSLP
- [9] see e.g.: Nicola Vicentino: Musica prisca caput on YouTube
- [10] see e.g.: Wikipedia: Music and Mathematics
- [11] A. von Oettingen, *Die Grundlagen der Musikwissenschaft und das duale Reinstrument*, Teubner Verlag, p. 155-361 (1917).
- [12] D. Keislar, History and Principles of Microtonal Keyboards, *Computer Music Journal* 11, 18 (1987).
- [13] Figure taken from Wikimedia commons (public domain).
- [14] Pablo Casals, in: S. Applebaum, *The Way They Play*, Vol. 1, Paganiniana Publications (1972).
- [15] D. Løberg Code, *Groven.Max: An Adaptive Tuning System for MIDI Pianos*, *Computer Music Journal* 26, 50 (2002). See also wmich.edu/mus-theo/groven/
- [16] G. J. William and R. A. Weyler, *Just Intonation Tuning*, US-Pat. CA 2182662 (1995). Commercial software distributed by *Justonic Tuning Inc* until spring 2016.
- [17] R. Wille *et al.*, www.math.tu-dresden.de/~mutabor/
- [18] W. Mohrlök, www.hermoder.com/index_de.html
- [19] W. A. Sethares, *Adaptive tunings for musical scales*, *Journal of the Acoustical Society of America* 96, 10 (1994). Free software available at homepages.cae.wisc.edu/~sethares
- [20] L. Euler, *Tentamen novae theoriae musicae ex certissimis harmoniae principiis dilucide expositae* Saint Petersburg Academy 147 (1739). See also wikipedia.org/wiki/Tonnetz.
- [21] The first public performance was realized by D. L. Code in 2001, see wmich.edu/mus-theo/groven.
- [22] “Just Intonation” is a demonstration software developed by the authors of this paper running on various platforms, see <http://just-intonation.org>.
- [23] For MIDI specifications see e.g. www.midi.org/specifications.
- [24] H.F. Cohen, *Quantifying Music: The Science of Music at the First Stage of Scientific Revolution 1580-1650*. Springer, Berlin (1984).
- [25] M. Wittmann and E. Pöppel, *Temporal mechanisms of the brain as fundamentals of communication – with special reference to music perception and performance*, *Musicae Scientiae*, Special Issue 1999-2000, 13-28
- [26] A. C. Lehmann and S. Goldhahn, *Duration of playing bursts and redundancy of melodic jazz improvisations in John Coltrane’s “Giant Steps”*. *Musicae Scientiae* 20 (2016) 345-360.
- [27] The necessity to deal with various interval sizes was pointed out to us by Ross W. Duffin (private communication).
- [28] R. Kopiez, *Intonation of harmonic intervals: Adaptability of expert musicians to equal temperament and just intonation*, *Music Perception: An Interdisciplinary Journal* 4, 383 (2003).
- [29] H. von Helmholtz, *Die Lehre von den Tonempfindungen: als physiologische Grundlage für die Theorie der Musik*, Friedrich Vieweg & Sohn, Braunschweig (1865).