

OWT: A REAL-TIME OPTIMAL TUNING APPLICATION

Wei Pan¹, Micah K. Johnson², Larry Polansky³, Daniel Rockmore^{1,4} and Douglas Repetto⁵

¹ Dartmouth College, Department of Computer Science

² MIT, Department of Brain and Cognitive Sciences

³ Dartmouth College, Department of Music

⁴ Dartmouth College, Department of Mathematics

⁵ Columbia University, Department of Music

ABSTRACT

This paper presents the Optimal Well-tempered Tuning system (OWT), a software application for computing optimal tuning systems in real-time. The optimal tuning systems are created from a framework, proposed by us, that formalizes historical and cross-cultural criteria. The parameters of the system are a fixed number of pitches, a repeat factor, an ideal tuning set, and weights for keys and intervals. This framework allows for an efficient least-squares solution to a tuning problem, thus enabling real-time control of the parameters. With this software, a user can visualize, modify, and hear optimal tuning systems while the music is playing.

1. INTRODUCTION

When J.S. Bach wrote the forty-eight preludes and fugues of *The Well-Tempered Clavier* (1722, 1740), the tuning of his keyboard was markedly different from the one we are accustomed to today. Though it is commonly thought that Bach composed the works to demonstrate the modern 12-tone equal-temperament, scholars believe Bach's particular tuning was one of the "well-temperaments" in wide use at the time [1].

Well-temperaments are tuning systems with a musical richness that arises out of a complex system of acoustical compromises. Bach thought that the concept was important enough to publicize it in the title of his masterpiece, a work written to demonstrate the possibility of writing and playing in all twenty-four major and minor keys [1, p. 35].

Although the term *well-temperament* most often refers to various tunings of the late Baroque, it can be argued that any tuning system which implements a similar set of compromises, with the goal of achieving a versatile tuning, is a well-temperament. This classification includes virtually all established tuning systems throughout the world and throughout history. Each of the many well-temperaments in use in Bach's time was distinct, yet all tried to satisfy similar criteria in order to achieve a "good" tuning. These criteria included imprecise notions such as purity of intervals and number of acceptable keys [2].

In previous work, we developed a mathematical framework for tuning and scale formation, thus making pre-

cise the historically subjective criteria for designing well-temperaments. In this paper, we demonstrate our system with a software application capable of deriving, visualizing, and performing music with optimal tuning systems in real-time. In the rest of the paper, we review the mathematical details of our framework, present the software application, and discuss its features. In addition, we have made the software freely available so that anyone can experiment with and hear optimal tuning systems [3].

2. METHODS

Tuning systems throughout history and across various cultures have used a set of complex compromises to account for some or all of the following constraints:

1. Pitch set: use of a fixed number of pitches (and consequently, a fixed number of intervals)
2. Repeat factor: use of a repeat factor for scales and for the tuning system itself (e.g., the octave)
3. Intervals: an idea of correct or ideal intervals, in terms of frequency relationships
4. Hierarchy: a hierarchy of importance for the accuracy of intervals in the system
5. Key: a higher-level hierarchy of the relative importance of the specific scales or modes begun at various pitches in the system.

Most tuning systems attempt to resolve some or all of the five constraints listed above. Our framework provides a way to state these constraints mathematically and derive an optimal solution. In our formulation, there is a weighting system that allows for the creation of any tuning system with a fixed number of pitches, repeat factor, and some set of ideal intervals. Not all tuning systems, of course, consider all of these constraints in their construction, nor are these constraints exhaustive. However, these constraints constitute an economical and musically reasonable set capable of specifying well-tempered tuning systems.

2.1. Mathematical formulation

We formalize the five constraints described above using the following variables:

1. a set of n pitches $a_1, \dots, a_n \neq 0$
2. a repeat factor $\omega > a_n$
3. a set I_1, \dots, I_n of ideal intervals
4. a set of interval weights ι_1, \dots, ι_n
5. a set of key weights $\kappa_0, \dots, \kappa_n$

In order to judge the overall tuning of a fixed set of pitches, we consider the interval matrix, i.e., the $(n+1) \times (n+1)$ matrix of all intervals generated from the set of pitches:

$$M = \begin{pmatrix} m_{0,0} & \dots & m_{0,n} \\ \vdots & \ddots & \vdots \\ m_{n,0} & \dots & m_{n,n} \end{pmatrix}, \quad (1)$$

where:

$$m_{i,j} = \begin{cases} a_j - a_i & \text{if } i \leq j, \\ \omega + a_j - a_i & \text{if } i > j, \end{cases} \quad (2)$$

with $a_0 = 0$. The diagonals of the interval matrix M hold all instances of a particular interval generated by the set of pitches. The corresponding ideal interval matrix is:

$$L = \begin{pmatrix} I_0 & I_1 & \dots & I_{n-1} & I_n \\ I_n & I_0 & \dots & I_{n-2} & I_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ I_2 & I_3 & \dots & I_0 & I_1 \\ I_1 & I_2 & \dots & I_n & I_0 \end{pmatrix}, \quad (3)$$

with $I_0 = 0$. We define an error function E to measure the sum of squared errors between the ideal intervals and the intervals generated by the set of pitches:

$$E(a_1, \dots, a_n) = \sum (M - L)^2, \quad (4)$$

where the exponentiation and summation are both applied element-wise to the matrix. Without any preference given to keys or intervals (i.e., without any weights), the optimal solution to Equation (4) is always equal-temperament. Intuitively, this solution makes sense: if all weights are equal (to one), then equal-temperament is the best tuning system. To specify preferences for certain keys and intervals, we introduce matrices of key and interval weights:

$$\mathcal{I} = \begin{pmatrix} \iota_0 & \iota_1 & \dots & \iota_{n-1} & \iota_n \\ \iota_n & \iota_0 & \dots & \iota_{n-2} & \iota_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \iota_2 & \iota_3 & \dots & \iota_0 & \iota_1 \\ \iota_1 & \iota_2 & \dots & \iota_n & \iota_0 \end{pmatrix}, \quad (5)$$

$$\mathcal{K} = \begin{pmatrix} \kappa_0 & \kappa_0 & \dots & \kappa_0 & \kappa_0 \\ \kappa_1 & \kappa_1 & \dots & \kappa_1 & \kappa_1 \\ \kappa_2 & \kappa_2 & \dots & \kappa_2 & \kappa_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \kappa_n & \kappa_n & \dots & \kappa_n & \kappa_n \end{pmatrix}. \quad (6)$$

Let the weight matrix $W = I \star K$, where the \star operator denotes a Hadamard, or element-wise product. The weighted version of the error function is then:

$$\hat{E}(a_1, \dots, a_n) = \sum W \star (M - L)^2. \quad (7)$$

For each set of constraints specified in the matrices W and L , there is a unique solution that minimizes the weighted error function \hat{E} . This solution is computed using least-squares. The result is a set of pitches a_1 to a_n , called the optimal tuning system. While the optimal tuning is unique to a given set of constraints, the converse is not true: there is not necessarily a unique set of constraints that will generate a given tuning. In other words, multiple sets of constraints can generate the same tuning, within a specified tolerance.

3. SOFTWARE

3.1. Introduction

MIDI tuning applications, such as Scala and the Java Just Intonation Calculator [4, 5], have existed for a long time. However, we feel that our optimal tuning framework is both a powerful and general tool for analyzing and creating tuning systems. Therefore, we developed a GUI-based software application based on the mathematical formulations in Section 2. We believe that this software will be beneficial to musical communities for several reasons: it will enable scholars to study and analyze different tunings in a mathematical manner; it will allow tuning-system designers to both hear and see their tuning systems; and it will allow performers to modify tuning systems in real-time by adjusting the constraints.

We call our application OWT (Optimal Well-tempered Tuning system).¹ It is a MIDI player with real-time optimal tuning adjustment that provides a graphical user interface for users to change constraint sets. From the constraints, it calculates optimal tuning results, displays them graphically and plays MIDI files with arbitrary tunings in real time.

This software uses *JAVA* technologies, and takes either the low end software synthesizer that ships with the *JAVA* runtime library or any other high end external MIDI synthesizer as a tone generator. Such an architecture allows us to distribute OWT for different platforms with ease [6].

3.2. Tuning window

The tuning window of OWT, which appears first when launching the application, is illustrated in Fig. 1. It is separated into different blocks and multiple tasks can be accomplished through this window:

1. **Entering Tuning Constraints:** The ideal intervals, interval weights and key weights are adjustable in box (a); the pitch set and the repeat factor can also be modified in box (c).

¹ The software is under active development and the latest version and manual can be downloaded from [3].

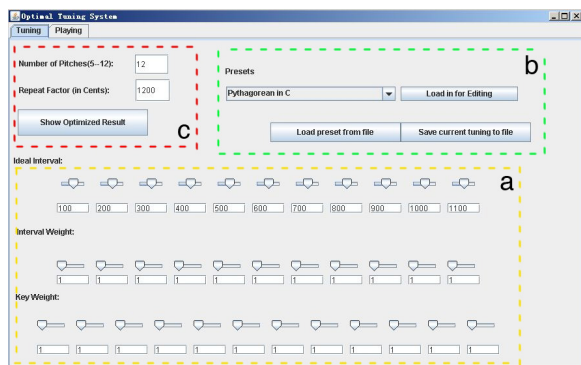


Figure 1. The tuning window allows control of (a) tuning constraints, (b) tuning parameters, and (c) preset tunings.

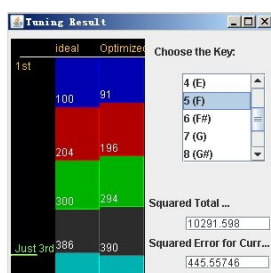


Figure 2. The tuning result window. The key of F in the W3 tuning is displayed as an optimized result.

- Loading Preset Tunings:** OWT has many built-in tuning presets for users to use directly, which can be loaded inside the upper right box (b). They range from the famous Werckmeister III (W3) to the optimal well-tempered tunings OWT1 and OWT2, mentioned in Section 4 of this paper. All presets are presented in the form of a set of constraints.
- Saving and Loading Constraints:** OWT allows tuning constraints to be saved to a file on your hard drive. This file can be loaded at a later date to continue working on a tuning system.
- Displaying an Optimal Tuning:** The tuning result window shown in Fig. 2 can be opened by clicking the “Show Optimized Result” button. The ideal intervals and the actual optimized intervals for a certain key are compared in two colored columns with the value of each interval illustrated through the height of each rectangle. Squared error measures are also available.

3.3. Playing window

A MIDI sequencer is embedded in OWT, which can be accessed through the playing window in Fig. 3. The MIDI sequencer is compatible with all formats of .MID files, and its ability to play in arbitrary tunings is constructed on the standard MIDI pitch bending mechanism [11].

The upper block (a) in Fig. 3 is a standard MIDI player interface, where users can open, play, or stop a MIDI file.

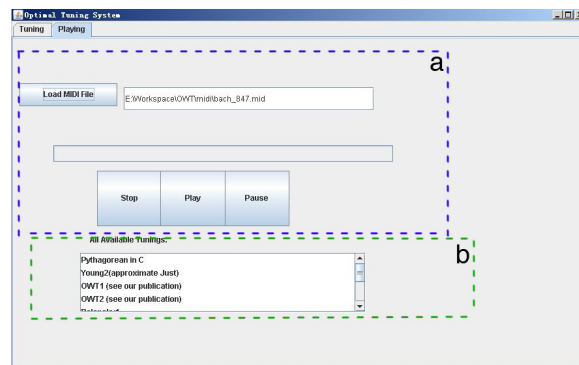


Figure 3. The playing window lets the user (a) load and play MIDI files and (b) change the tuning in real-time.

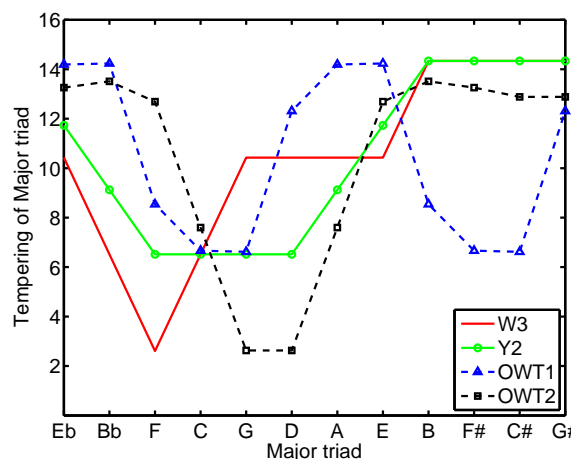


Figure 4. The tempering of the major triad for four tuning systems, two historical, Werckmeister III (W3) and Young II (Y2), and two novel, OWT1 and OWT2. All four tuning systems achieve the minimum mean-tempering of 10.43¢, but distribute the error differently amongst the keys.

The lower block (b) is for tuning selections, which allows users to switch between preset tunings.

One unique feature of OWT is that it allows for changes to the constraints, and therefore changes to the tuning system, in real-time. In the OWT interface, a user can switch to the tuning window while a MIDI file is being played simultaneously in the play window. It is very exciting to switch between W3, 12-tone equal temperament, or to any of your own tunings while Bach’s *The Well-Tempered Clavier* is being played.

3.4. Other Features

OWT also supports non-regular tunings: by adjusting the repeat factor and number of pitches, users are able to generate brand new scales with fewer than 12 notes in an octave. Users can compose MIDI files for their unique scales by following the OWT manual.

4. EXAMPLES

To explore the possibilities resulting from our framework, we created two new well-temperaments. As a way of comparing tuning systems, Rasch, Chalmers, and others have proposed several simple measurements [7, 8]. We will use the measurement proposed by Rasch in his discussion on Werckmeister's tuning systems. This measurement considers the mean-tempering of "all consonant intervals, which is equal to the mean tempering of all triads, or of all keys [7]." A simple way to compute the mean-tempering of a tuning system is to measure the absolute difference between the intervals of the major triad in each key to the ideal intervals $3/2$, $5/4$ and $6/5$. Rasch's measure is thus an error function between a tuning system and a given set of ideal intervals. This error function provides us with a meaningful way to measure the results of some simple experiments in generating new well-tempered tuning systems.

The historical tuning system Werckmeister III (W3) is exemplary in its mean-tempering of 10.43ϵ , which is the same as twelve-tone equal-temperament, and can be shown to be an absolute minima. Another historical well-temperament known as Young II (Y2), sometimes considered to be an improvement on W3 [9, 10], also achieves this minimality.

Using our framework, we generated two new optimal tuning systems with the same minimal mean-tempering of 10.43ϵ . In Fig. 4, we show the tempering of the major triad in all twelve keys for the two historical tunings mentioned above (W3 and Y2) as well as our new optimal tuning systems (OWT1 and OWT2). These new tuning systems are maximally in tune by the mean-tempering measure. They also have a great deal in common, theoretically and musically, with historical well-temperaments, yet their musical implications and structure differ in important ways from their historical models. For example, an important characteristic of Y2 is its four adjacent keys with a major-triad tempering of 6.5ϵ . Our optimal tuning system OWT1 also has four keys with a similar major-triad tempering, but these keys occur in two pairs separated by a tritone. The tuning system W3 has one key (F) with a very small major-triad tempering of 2.6ϵ . The optimal tuning system OWT2 has two keys (G and D) that achieve the same minimum. These are just two examples of the many possible tuning systems that can be generated with our software.

5. CONCLUSION

We have presented a software application for exploring optimal tuning systems. This application allows scholars to hear and see their tuning systems as well as compare them with historical systems such as Werckmeister III and Young II. It also allows performers to adjust tuning systems while the MIDI file is being played, thus enabling real-time control of tuning.

But what does this application mean with respect to the

great tuning traditions of the world? Tuners like Werckmeister and Young labored hard and long to create their complex, beautiful scales, yet similar if not identical results may be generated quickly and simply, according to specific sets of initial conditions. Rather than trivialize the work of these master tuners, this application sheds new light on their accomplishments: both Werckmeister and Young, using the tools of their time, were able to balance complex sets of compromises to achieve the minimal mean-tempering. This application simply builds from their work using modern techniques.

While this application models some constraints for tuning systems, it does not model all criteria that have influenced tuning systems throughout history. There are certainly other cultural, aesthetic, historical and intangible factors that have affected the development of tuning systems in ways that mathematics cannot model. But given the range of scales that can be generated, including many historical well-temperaments, this work suggests that the mathematics of scale tuning is a little less mysterious than had previously been thought.

6. REFERENCES

- [1] Ledbetter, D. *Bach's Well-Tempered Clavier: The 48 Preludes and Fugues*. Yale University Press, New Haven, 2002.
- [2] Barbour, J.M. *Tuning and Temperament: A Historical Survey*. Dover Publications, Mineola, NY, 2004.
- [3] Polansky, L, et. al. *Optimal Well-Temperament Supplementary Materials*. <http://eamusic.dartmouth.edu/~larry/owt/>, May 23, 2008.
- [4] Coul, M. *Scala Version 2.2*. <http://www.xs4all.nl/~huygensf/scala/>, Netherland, 2006.
- [5] *Java Just Intonation Calculator*. <http://jjicalc.sourceforge.net>, May 23, 2008.
- [6] Sun Microsystems. *Java Technology*. <http://java.sun.com>, May 23, 2008.
- [7] Werckmeister, A. *Musicalische Temperatur*. Ed. R. Rasch. Utrecht, Diapason Press, 1983.
- [8] Chalmers, J. "Some New Linear Temperaments", *Xenharmonikon* 2, 1974.
- [9] Jorgensen, O. *Tuning*. Michigan State University Press, East Lansing, 1991.
- [10] Donahue, T. *A Guide To Musical Temperament*. Scarecrow Press, Lanham, MD, 2005.
- [11] MIDI Manufacturers Assoc. *General MIDI 2 Specification, Version 1.1*, 2003.