

Syracuse University

SURFACE

Syracuse University Honors Program Capstone Projects Syracuse University Honors Program Capstone Projects

Spring 5-1-2011

Flexible Tuning Software: Beyond Equal Temperament

Erica Sponsler

Follow this and additional works at: https://surface.syr.edu/honors_capstone

 Part of the [Computer and Systems Architecture Commons](#), and the [Other Computer Engineering Commons](#)

Recommended Citation

Sponsler, Erica, "Flexible Tuning Software: Beyond Equal Temperament" (2011). *Syracuse University Honors Program Capstone Projects*. 239.

https://surface.syr.edu/honors_capstone/239

This Honors Capstone Project is brought to you for free and open access by the Syracuse University Honors Program Capstone Projects at SURFACE. It has been accepted for inclusion in Syracuse University Honors Program Capstone Projects by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Flexible Tuning Software: Beyond Equal Temperament

A Capstone Project Submitted in Partial Fulfillment of the Requirements of the Renée Crown University Honors Program at Syracuse University

Erica Sponsler

Candidate for B.S. Degree
and Renée Crown University Honors

May/2011

Honors Capstone Project in Computer Science

Capstone Project Advisor: Dr. Shiu-Kai Chin

Honors Reader: Dr. James Royer

Honors Director: James Spencer, Interim Director

Date: _____

Flexible Tuning Software: Beyond Equal Temperament

A Capstone Project Submitted in Partial Fulfillment of the Requirements of the Renée Crown University Honors Program at Syracuse University

Sierra Fox

Candidate for B. Mus. Degree
and Renée Crown University Honors

May/2011

Honors Capstone Project in Music Performance

Capstone Project Advisor: _____

Dr. Joseph Downing

Honors Reader: _____

Dr. Olukola Owolabi

Honors Director: _____

James Spencer, Interim Director

Date: _____

Abstract

The premise of this creative Capstone project was to develop a computer instrument that is capable of tuning itself flexibly in such a way as to not require a tempering of the Western scale, as is necessary for fixedly tuned instruments. The difficulty in creating such a system of tuning arises from the mathematical paradox of the *musical harmonic series*, which is the sequence of frequencies that sound naturally as overtones over a fundamental pitch. They follow a proportional pattern of 1:2, 2:3, 3:4, etc. These small-integer ratios of frequencies represent the consonant (in-tune) harmonic intervals. When an interval does not have a small-integer ratio between its notes, it is perceived as “out-of-tune,” and the two frequencies will compete with each other.

The problem becomes apparent when the ratios are used to create the Western scale on a fixed instrument. When tuning such an instrument (i.e. a piano), the intervals inherently cannot all be tuned *justly*, or according to the appropriate proportions. The nature of the ratios does not allow larger intervals to be explained exactly by the smaller intervals, though they are expected to coexist and be used simultaneously in Western music. For example, an octave in music is the equivalent of three major thirds; however, the justly tuned octave (2:1) is not equal to the sum of three major thirds (5:4); $5:4^3=125:64$, not 2:1. The difference, then, between notes tuned in terms of different intervals is known as a *comma*, and the attempts to distribute this comma are called *temperaments*.

To overcome the impossibility of perfectly consonant temperaments, we have created a computer program that can function as a self-tuning instrument by mathematically calculating the frequency of each pitch played in relation to the pitch preceding it. Though this allows every interval between sequentially played notes to be in tune, it does mean that pitches are not fixed, and rather are flexible and changing as a piece progresses. The program possesses the capability to play several historical fixed temperaments, namely Pythagorean, Quarter-Comma Meantone, Werckmeister III, and Equal Temperament. However, it also is capable of playing in two non-fixed tuning systems, distinguished as *Sequential Tuning* and *Flexible Tuning*.

The Sequential Tuning system uses the ratios in the harmonic series to tune each note proportionally to the most recent note pressed. It is ideal for educational purposes, clearly demonstrating the flexibility of the program, but is less practical for tuning purposes, as it disregards held notes and tunes solely based on the notes pressed while the note is held, creating obtrusive dissonances between any held note and the notes tuned sequentially before it is released. The Flexible Tuning system remedies this issue by first tuning notes based on a held note, and if there is no note being held, then on the most recent note pressed. This eliminates the noticeable dissonances of the Sequential Tuning system, while still tuning each note flexibly and in real time based on its interval relationships to other notes played before and at the same time as it. In this way, every interval sounding is the appropriate small integer ratio for that interval in the harmonic series.

TABLE OF CONTENTS:

Preface/Acknowledgements	<i>i</i>
Introduction to the Harmonic Series and Intervals Thereof	1
The Problem of the Syntonic Comma	5
Historical Temperaments	7
The Flexible Tuning Concept	10
Sequential and Flexible Tuning	12
Programming Cents and Hertz	13
The Issue of Assigning Ratios	15
Methods of Playing	17
Applications and Conclusion	19
Work Cited	21
Appendix A: User Manual	22
Appendix B: Pre Set Pieces	36
Summary	39

Preface/Acknowledgements

The concept for this Capstone project was based on a course on Tuning and Temperaments, taught by Dr. Joseph Downing at Syracuse University. With the information introduced in that class in the fall of 2009, our combined interest in music, and our respective enthusiasm for and studies in music theory and computer programming, we set out to create this creative Capstone Project, which we hope will provide new ways of hearing music and understanding the relationships of pitches to one another. We would like to thank our advisors, Dr. Joseph Downing and Dr. Shiu-Kai Chin, for their helpfulness and support. We would also like to thank Dr. Downing separately for lending his expertise in the areas of temperaments and the harmonic series. Our readers, Dr. Olukola Owolabi and Dr. James Royer, were truly phenomenal sources of assistance, providing support and advice throughout the process of creating our project; we are so grateful for their help and willingness to devote their time and vast knowledge to our work. We would also like to thank Robert Irwin, Nick Didkovsky, Nathaniel Weiss, and Chad Schmidt for their help in the programming facet of this project, as well as Dr. Andrew Waggoner for his assistance in the music aspect. We are additionally grateful to the Renée Crown University Honors Program at Syracuse University for providing the means and outlet for us to create this program. We hope that this program and the information it provides will create major educational opportunities for all ages, and look forward to seeing the potential applications of the program in the future.

-Sierra Fox and Erica Sponsler

Introduction to the Harmonic Series and Intervals Thereof

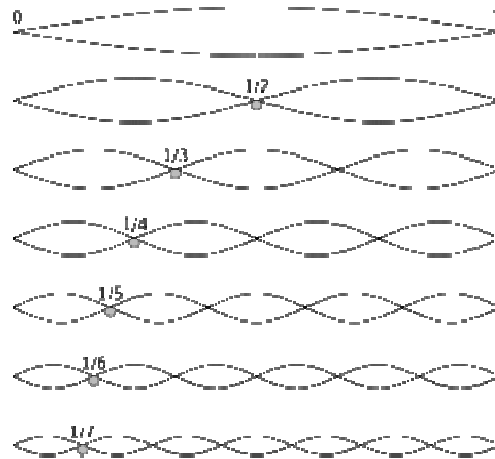
Fig. 1. Harmonic Series



The premise of this creative Capstone project was to develop a computer instrument that is capable of tuning itself flexibly in such a way as to not require a tempering of the Western scale, as is necessary for fixedly tuned instruments. The difficulty in creating a perfectly tuned system on fixed instruments rises from the mathematical paradox of the *musical harmonic series*. The harmonic series, or *overtone series*, is the sequence of frequencies that sound naturally as overtones over a fundamental pitch. The strength of the specific overtones varies depending on what instrument is playing the fundamental pitch. This variation is one factor in determining their individual *timbres*, or sound qualities; i.e., making an oboe sound different from a bassoon.

Technically, a harmonic series in math is any series of number values that are related by whole-number ratios, such as 2, 4, 6, 8, 10, or 5, 10, 15, 20, 25, etc. The overtone series specifically follows a pattern of 1, 2, 3, 4, 5, which thus has a proportional pattern of 1:2, 2:3, 3:4, etc. Figure 1 depicts the harmonic series built over the fundamental pitch of C2, through the sixteenth *partial*, or pitch in the harmonic series. Each number represents the ratio of its corresponding note to the fundamental pitch; for example, the fourth partial over a fundamental pitch has a frequency of four times that of the original pitch.

Fig. 2. Harmonic Partial on Strings



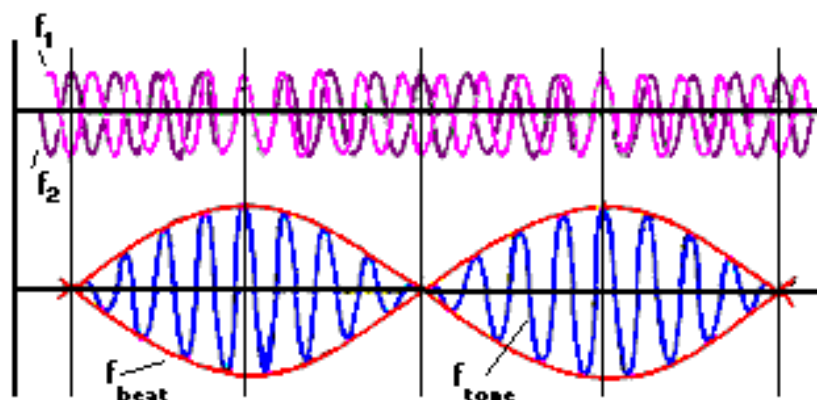
Source: Richards, Geoff. File:Harmonic Partial on Strings.svg. Digital image.

Wikipedia, the Free Encyclopedia. 8 July 2008. Web. 26 Apr. 2011.

<http://en.wikipedia.org/wiki/File:Harmonic_partial_on_strings.svg>.

The harmonic series also applies to the frequencies at which a string will vibrate when divided into the number of equal parts of that value in the series; for example, a string divided in half will sound at a frequency twice that of the original length of the string (an interval of an octave higher), and a string a third of the length of the original string will vibrate at a frequency of three times that of the original. The partials of the harmonic series can also be sounded as harmonic overtones by pressing lightly at the small-integer ratios on a vibrating string. The points on the string at which the harmonics will sound when pressed are called *nodes* (see Figure 2). As displayed in the figure, when a string is lightly pressed at a node, the string will sound a harmonic at the partial corresponding to that node. For instance, lightly pressing the harmonic node at any third of a vibrating string (either 1/3 or 2/3 down the string) will cause the third partial (an octave and a perfect fifth from the original pitch) to sound (“The Harmonic Series”).

Fig. 3. Interfering Frequencies and Beat Patterns



Beat waves form when two waves that are close in frequency combine.

Source: Moy, Chu. *Beats*. Digital image. *HeadWize - Technical Paper: The Elements of Musical Perception by HeadWize*. HeadWize, 2000. Web. 26 Apr. 2011. <http://gilmore2.chem.northwestern.edu/tech/elemsnts_tech.htm>

The small-integer ratios of frequencies in the harmonic series represent the *consonant* (in-tune) harmonic intervals. When an interval does not have a small-integer ratio between its notes, it is perceived as “out-of-tune,” and the two frequencies will compete with each other. As the frequencies create less compatible sound-waves, the two waves will interfere, creating acoustical *beats*. This is caused by the slight difference in the frequencies sounding together; the waves interact in such a way as to create a distinct wavering sound (see Figure 3). The beat pattern is dependent upon the particular interactions of the sine waves of the sounding frequencies; these beats slow and eventually diminish as the interval is tuned more closely to the appropriate small-integer harmonic ratio (Truax).

It is important to mention that some of the notes shown in the depiction of the harmonic series have unusual markings next to or above them. These markings denote pitches that are not typically part of the Western scale; they are slightly different than the pitches as notated without the markings. For example, the eleventh partial is an F quarter-sharp, which is slightly lower than an F sharp, and the thirteenth partial is a slightly flat A. The seventh and fourteenth partials are a slightly lowered B flat. Unlike the other altered pitches in most cases, this slightly low B flat is used commonly in widely accepted music, particularly as the 6:7 ratio in jazz music. This altered pitch is known in this situation as a *blues third*.

The Problem of the Syntonic Comma

The problem of tuning consonantly becomes apparent when the small-integer ratios determined by the harmonic series are used to create the Western scale on a fixed instrument. When tuning a fixed instrument (for instance, a piano), the intervals inherently cannot all be tuned *justly*, or according to the appropriate small-integer proportions. For example, a perfect octave is the equivalent of three major thirds; however, the justly tuned octave (a 2:1 ratio) is not equal to the sum of three major thirds (5:4 ratios); $(5:4)^3=125:64$, not 2:1. The difficulty in tuning arises because scales tuned by different intervals will yield different frequencies for each note, making some intervals consonant but others noticeably dissonant.

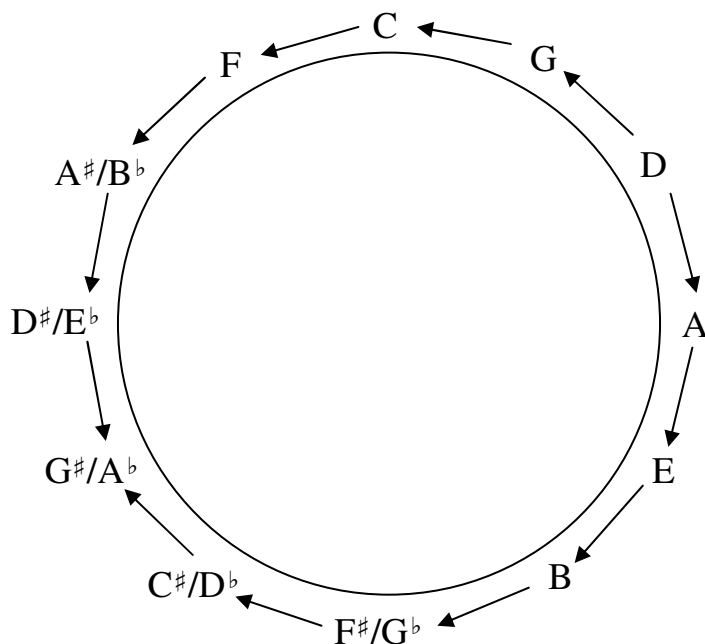
The impossibility of the ratios functioning in tandem with one another is caused by a mathematical phenomenon. The nature of the fractions of the harmonic series (specifically, the $[1/2, 1/3, 1/4, 1/5\dots]$ series formed by proportions of the wavelengths of harmonics sounded by a vibrating string) is what is known as a *divergent series*. This means that though the series approaches zero, it never converges, or reaches an absolute limit (which in this case would be zero). Ultimately, this asymptotic nature creates a mathematical paradox, which does not allow the larger intervals to be explained exactly by the smaller intervals.

Regardless of this incompatibility, the intervals are expected to coexist and be used simultaneously in Western scales and music. Unfortunately, this paradox creates significant discrepancies between justly tuned intervals and the

corresponding pitches on a statically-tuned instrument, such as a piano. For instance, as mentioned previously, the interval of an octave can be made up of three major third intervals; theoretically, tuning major thirds up from C4 should result in C5 an octave, or 2:1 ratio, above C4. When applied to actual pitches, this produces a contradiction between appropriate interval; to demonstrate, let us attempt to tune an octave by major thirds. If C4 is a frequency of 264 Hz, the E a major third above (at a 5:4 ratio) would be a frequency of 330. The major third above that (G[#]/A^b) tuned to another 5:4 ratio would be a frequency of 412.5. Another major third above that pitch, then, at another 5:4 ratio, equals 515.625 Hz, though it should equal a C of 528 Hz (2:1=264:528 Hz) in order to be a 2:1 octave above the C of 264 Hz. This discrepancy in frequencies is known as the *syntonic comma* (Bicknell). Depending how a piano is tuned this difference will manifest itself in various ways, but will always exist on a fixed instrument. In order to tune a fixedly-tuned instrument, then, it is necessary to find some place for the comma in the organization of notes on that instrument.

Historical Tunings

Fig. 4. The Circle of Fifth



Historically, the problems with tuning intervallic relationships on fixed instruments have been addressed by means of *temperaments*, or systems of incorporating the comma into the scale in various intervals. In Pythagorean tuning, one of the earliest temperaments, the scale was tuned by a sequence of perfect fifths (see Figure 4). The discovery of this system is attributed to Pythagoras, around 500 BCE. As demonstrated earlier, a sequence of fifths (a 3:2 ratio) will prove problematic, especially at the octaves. For instance, if the temperament is based on D, the frequency of D (288 Hz) is multiplied by $3/2$ to obtain the frequency of A, which is multiplied by $3/2$ to obtain the frequency of E, and so forth, tuning in succession B, F $^\sharp$, C $^\sharp$, and G $^\sharp$. Likewise, D is multiplied by $2/3$ to obtain a fifth below at G, and the notes C, F, B $^\flat$, E $^\flat$, and A $^\flat$ are tuned with the same pattern. This determines frequencies for all twelve chromatic notes of

the Western scale, which can be used to tune the rest of the piano by tuning octaves using the 2:1 ratio. However, an issue arises with A^b and G^\sharp , which are enharmonic spellings of the same note (different nominal titles for the same pitch on an instrument). When brought into the same octave they have very different frequencies. The difference between these two values is referred to as the *Pythagorean comma*, specifically. To remedy the discrepancy, the A^\square is disregarded, which makes the interval between the G^\sharp and the E^\square especially dissonant, and is commonly known as a *wolf interval*.

A similar tuning to Pythagorean was *Meantone Tuning*, but unlike Pythagorean, which tunes fifths justly but greatly compromises major thirds (tuning them to 81:64 ratios instead of to the appropriate 5:4 ratios), it tunes the major thirds justly. Instead of using a 3:2 ratio for the fifths, however, the most common type of Meantone Tuning (*Quarter-Comma Meantone*) calculates each fifth as slightly flat by one-fourth of the *syntonic comma*. In approximately the 17th century, Quarter-Comma Meantone tuning had largely replaced its similar predecessor, Pythagorean; the similarity between the two made the transition fairly easy, because notation could be shared between them (Monzo).

Especially as keyboard instruments became more prevalent, new methods of tuning were developed to mitigate the problem of the wolf interval. These new tunings took the comma and distributed it throughout various intervals in the scale, choosing which intervals to keep tuned justly and which to *temper*. This led to *well-tempered* tuning systems, such as the *Werckmeister III*, which was the temperament used by Bach when he composed his *Das Wohltemperierte Klavier*, (1722) or pieces for a well-tempered clavier. These temperaments distributed the comma among various intervals, not equally among each, but

rather just in such a way as to make it possible to play in all keys with minimal dissonance. Frequently used keys, such as C major, would have less noticeable dissonance, while less frequently used keys, such as F[#] major, would contain distinctively out of tune intervals. Each key, then, was associated with a different amount of tension caused by dissonance, and thereby a particular sense of mood or character. Composers could thus utilize these differences in their compositions, expressing particular emotions or dramatic tensions by moving from one key to another.

When the technology to register and measure frequencies was developed, it became possible to tune keyboard instruments to *Equal Temperament*. In this temperament, the octave is simply divided into twelve equal semi-tones, so that only the octave is justly tuned. Equal Temperament came into being in the late 18th century, but most musicians and tuners rejected it until the mid 19th century. Despite the dissonant nature of the majority of intervals in the Equal Tempered system, it has now become the most commonly used method of tuning, and is used for virtually all modern instruments. Because of the prevalence of the system, society has learned to accept the equally tempered intervals as correct and consonant, or in-tune. While this method of tuning is equal, it loses the characteristics of the individual keys, and the tensions that come from the varying distributions of the comma. It also sacrifices all justly tuned intervals besides the octave, whereas historical tunings were capable of playing justly tuned intervals in some keys. This causes significant discrepancies when fixed, Equal Tempered instruments are played with flexible instruments (i.e. string instruments, which generally tune the strings to just intervals by ear and tend to continue to tune to just intervals as they play).

The Flexible Tuning Concept

The purpose of this Capstone project was to create a computer program that can function as a self-tuning instrument, which is capable of overcoming the impossibility of perfectly consonant temperaments by mathematically calculating the frequency of each pitch played in relation to the pitch preceding it. Though this allows every interval between sequentially played notes to be in tune, it does mean that pitches are not fixed; rather, they are flexible and changing as a piece progresses. For example, while pitch A will be a frequency of 440 Hz at the start of a piece played using this program, as the piece goes on it changes relative to the other notes, and can (and likely will, depending on the intervals of the piece) be a completely different frequency by the end.

The program, titled *FlexTune* software, is able to tune flexibly by determining the frequency of the most recent note played based on its intervallic relationship to the notes the user played before it. The program uses two slightly different systems of tuning, which will be discussed later. In both systems, the program stores a note number (based on the MIDI numbering standard) as well as the frequency associated with it as a reference for tuning subsequent notes. The reference note is based on the notes the user plays, but the method for determining which note is the reference note is different in the two dynamic tuning systems. When a note is played, it calculates the difference between its note number and the reference note number (the number of semi-tones between the pitches). The program then references the ratio that is assigned to that particular interval, and multiplies the frequency of the reference note by the appropriate proportion. Then the program uses this information to produce a pitch that sounds at that frequency, which is the mathematically correct frequency for the note the user

played. At this point, the program updates the reference note, and is able to tune the following note in accordance with its interval relationship to the new reference pitch. In this way, the program can tune every note in real time to the exact, appropriate small-integer ratio.

Because of its flexibility, rather than a static tuning in which each pitch must maintain the 2:1 ratio at each octave, this tuning system allows all intervals played successively to have the ideal small-integer ratio. This means that all sounding intervals will be mathematically in tune. The acoustical beats present in intervals played in the fixed tuning systems do not sound when played in the flexible system. The flexibility of the program comes at what may be considered a cost, however; in many situations, the non-static nature of the system can be heard as pieces are played on the program. As a piece progresses, the pitches may sometimes make gradual, noticeable changes; the tuning of the computer instrument may slowly shift upwards or downwards, depending on the intervals and how they are tuning.

Sequential and Flexible Tuning

When developing the program, we began with the concept that the program would tune each note based on the most recent note pressed, tuned based on the ratios in the harmonic series. This creates a significant impracticality, however, in the cases of a note being held while other notes continue to play. Because this system tunes based on the last note pressed, the program disregards the held note, and tunes solely based on the notes being played while the note is held. This creates obtrusive dissonances between the held note and the notes tuned sequentially before it is released. Despite the impracticality of this method as a tuning system, it has value in the clarity with which it is able to demonstrate the shifting of pitches with a flexible system of just tuning, and so can be very useful as an educational tool. We decided to keep this tuning system in the program, and named it *Sequential Tuning*, because of the manner and order in which it tunes intervals.

We then created a second dynamic tuning system, which we titled the *Flexible Tuning* system. This Flexible Tuning system remedies this issues with the Sequential system by first tuning notes based on a held note, and if there is no note being held, then on the most recent note released. This eliminates the noticeable dissonances of the Sequential Tuning system, while still tuning each note flexibly and in real time based on its interval relationships to other notes played before and at the same time as it. In this way, every interval sounding at any given time is tuned to the appropriate small integer ratio for that interval in the harmonic series.

Programming Cents and Hertz

In the first stages of creating the FlexTune program, we attempted to use the common musical interface system, MIDI, or Musical Instrument Digital Interface, which is a widely accepted standard for transmitting, receiving, and producing digital music data. However, Java's MIDI library is pre-programmed to tune specifically to Equal Temperament, which began to present some challenges to our project. MIDI does allow modifications to its temperament, but instead of calculating notes according to their frequency, in Hertz, it calculates them in *Cents*. A cent is a unit that corresponds specifically to Equal Temperament, in which an Equal-Tempered semi-tone is equal to 100 cents. Because of its reliance on the system of Equal Temperament, and its use of cents rather than Hertz, this manner of calculating pitches was entirely counter-intuitive to the purpose of our program, and Java's MIDI capabilities did not allow for our program to tune as specifically or flexibly as we needed it to.

Fortunately, despite these problems with MIDI, we were able to find a software synthesis option called *JSyn*, which sounds pitches according to frequencies rather than cents, allowing the program to tune using the ratios as planned. Whereas the only possible way in which we could find to make the necessary changes to tuning with Java's MIDI library was to apply a *pitch-bend* to each note (creating an increasingly complicated algorithm), JSyn allowed us to work directly with the frequencies of the notes, assigning the nominal values to those frequencies and making it possible for each frequency to then serve as a reference point for the tuning of the following pitch.

As well as implementing dynamic tuning options, the program possesses the capability to play several historical fixed temperaments, namely the Pythagorean, Quarter-Comma Meantone, Werckmeister III, and Equal Temperament, all of which were discussed previously. Because these temperaments are fixed, it is not necessary to tune them according to ratios. Rather, the pitches for each note can be stored in a table, and when the note is played, the program looks up the appropriate pitch for that note number and uses JSyn to sound it at the correct frequency for that tuning system.

The Issue of Assigning Ratios

The program is able to apply proportional calculations to the notes played based on their intervals; however, in some cases, the appropriate ratios are not definitively assigned to each interval. With the development of Equal Temperament, intervals became homogenized sums of semi-tones, calculated as twelfths of the 2:1 octave; however, in the harmonic series, several versions of some intervals exist and sound as consonant intervals (small-integer ratios). For example, in a major scale, there are several intervals of a major second. The major second between the first two notes (Do and Re) is considered to be a 9:8 ratio. The major second between the second two notes (Re and Mi), however, is a smaller 10:9 ratio.

Both of these proportions are small integer ratios in the harmonic series that correspond to acceptable major second intervals. Because the computer program is incapable of making a subjective choice at every instance of a major second, determining how it is functioning tonally and thus which major second ratio it should utilize, it was necessary to give it a single ratio for each interval. Beyond merely the major seconds, other intervals (such as tri-tones) also have multiple possible small-integer ratios, such as 5:7, 7:10, 8:11, and 11:16.

Especially with the major seconds, which are very frequently-used intervals, the seemingly slight change between a 9:8 and a 10:9 ratio for the interval made a huge difference. With the larger interval, an ascending scale will gradually increase in pitch, so that each octave from the original pitch played will be slightly more than a 2:1 ratio from that note. With the small interval, however,

each octave from the original note played in an ascending scale is slightly smaller, and rather than rising, the pitch will sound like it is falling.

With such significant differences present we were reluctant to make an arbitrary choice as to which ratio would be used for each interval. While this would be convenient we would be severely limiting the usefulness of the program by disregarding the variety of consonant small-integer ratios. Therefore, for both of the unfixed tuning systems available in this program, users are able to access a Preferences Window, containing a set of ratio options for each interval in the scale. Users are given suggestions of several ratios that can be considered somewhat close to the accepted ratios for each interval, but can also choose to enter in a custom ratio. This can allow users to choose ratios slightly different from the usual proportions, or even experiment with drastically different ratios. The program will then tune all instances of that interval to the chosen ratio. When the user wishes to return to the normal intervals, he or she can reset the tunings to the default.

Methods of Playing

The FlexTune software is able to play both the historical and unfixed tuning systems based on computer or MIDI keyboard input. Users can play the computer keyboard as a piano keyboard or click the keys of the on-screen keyboard to produce pitches in any of the systems. . There is also the option of connecting an external MIDI instrument to the computer running the program. The instrument sends MIDI messages to the program, which the program recognizes and processes. Based on the MIDI note on and note off messages the program can use the same procedures it would use for other input methods, which allows the MIDI instrument to be used as program input for all static and dynamic tuning methods. These provisions allow users to play in the different temperaments/tuning systems in real time.

Users can also elect to play a selection of pre-set pieces in any of the systems. The FlexTune software is pre-set with Scott Joplin's *The Entertainer*, to give a well-known, accessible example for musical and non-musical users. It also contains several pieces by Johannes Sebastian Bach, including an arrangement of the hymn tune *Ach Gott, Vom Himmel Sich Darein*, as well as the *Prelude XXIII in B Major* from his collection of pieces for *Das Wohltemperierte Klavier*, as discussed previously. This provides an example of a piece clearly intended to be played in the Werckmeister III tuning system.

The FlexTune program also provides a window display of the frequency (in Hertz) of each note that sounds. The window shows the nominal expression of the note, the frequency value of that pitch at any given point, and also displays a

message when the tuning system is changed. This information is updated as the frequencies are calculated, and can be exported as file to provide a log of the information. This allows the user to see how the notes shift and change as they are calculated based on other notes in the unfixed systems.

Applications and Conclusion

The FlexTune program allows for the conventionally statically-tuned keyboard instrument to adapt and change in a manner that presents music in a drastically new way, allowing it to be heard with entirely consonant, in-tune intervals. The program has the potential to provide a practice opportunity for musicians playing flexibly tuned instruments, such as string instruments or trombones. In an ensemble of flexibly tuned instruments, the musicians should tune as best as they can to the actual, just intervals. However, when practicing with a fixed instrument in Equal Temperament, they are forced to modify their intervals to the incorrect ones created by that system. This program could allow them to practice tuning to correct intervals, to prepare them for ensemble rehearsals.

The most immediate and exciting potential application of this program, however, is as an educational device. It is capable of providing a demonstration of the imperfections of fixed temperaments, and also of demonstrating the rarely heard justly-tuned intervals. With the consistency and homogeneity of Equal Temperament, most people have become so acquainted with the incorrect intervals that they are not aware of the acoustical beats that signify the poor compatibility of the pitches. This program allows listeners to hear the potential of music to be truly in-tune, and, we hope, to train their ears to hear music based on its true consonance, rather than on the commonly compromised intervals that they are used to hearing. It is even possible that a shifting manner of tuning will become acceptable for instruments not limited by a static system. Even today, numerous ensembles and instruments worldwide have made the decision to

include the use of alternate/historical methods of tuning (Unruh). Significant efforts to explore such alternative tuning systems to the conventional Equal Temperament include the Taylor and Boody organ built at Yale University in 2006-2007 (which tunes in Quarter-Comma Meantone) or performances by the Hilliard Ensemble, which utilize various alternate tuning systems, such as Pythagorean (“Marquand Organ”). Especially in light of these modern uses for alternate systems of tuning, it is clear that the potential educational applications as well as practical uses for the FlexTune software are both evident and exciting.

WORK CITED

- Ach Gott, Vom Himmel Sich Darein. Digital image. IMSLP. 2007. Web. 26 Apr. 2011. <http://erato.uvt.nl/files/imglnks/usimg/1/19/IMSLP03822-4PC_3.PDF>.
- Bicknell, Steven. "Temperament: A Beginner's Guide." University at Albany - SUNY. Web. 26 Apr. 2011. <<http://www.albany.edu/piporg-l/tmpment.html>>.
- "Das Wohltemperierte Klavier I, BWV 846-869 (Bach, Johann Sebastian) - IMSLP/Petrucci Music Library: Free Public Domain Sheet Music." International Music Score Library Project (IMSLP) Portal. Web. 26 Apr. 2011. <[http://imslp.org/wiki/Das_wohltemperierte_Klavier_I,_BWV_846-869_\(Bach,_Johann_Sebastian\)](http://imslp.org/wiki/Das_wohltemperierte_Klavier_I,_BWV_846-869_(Bach,_Johann_Sebastian))>.
- "The Harmonic Series | Music Theory Lessons." Free Guitar Lessons. 2008. Web. 26 Apr. 2011. <<http://totalguitarist.com/lessons/theory/harmonics/intro/>>.
- "Marquand Organ." Yale University. 2005. Web. 26 Apr. 2011. <<http://www.yale.edu/ism/events/MarquandOrgan.html>>.
- Monzo, Joe. "Meantone." Tonalsoft®: Tonescape® - Microtonal Just-intonation Electronic Music Software. 2005. Web. 26 Apr. 2011. <<http://tonalsoft.com/enc/m/meantone.aspx>>.
- Moy, Chu. *Beats*. Digital image. *HeadWize - Technical Paper: The Elements of Musical Perception by HeadWize*. HeadWize, 2000. Web. 26 Apr. 2011. <http://gilmore2.chem.northwestern.edu/tech/elemnts_tech.htm>
- Praeludium XXIII. Digital image. IMSLP. 2007. Web. 26 Apr. 2011. <<http://216.129.110.22/files/imglnks/usimg/1/10/IMSLP02228-BWV0868.pdf>>.
- Richards, Geoff. *File:Harmonic Partial on Strings.svg*. Digital image. *Wikipedia, the Free Encyclopedia*. 8 July 2008. Web. 26 Apr. 2011. <http://en.wikipedia.org/wiki/File:Harmonic_partials_on_strings.svg>.
- Truax, Barry. "Beats." SFU.ca. Web. 26 Apr. 2011. <<http://www.sfu.ca/sonic-studio/handbook/Beats.html>>.
- Unruh, W. "Is It Possible?" Music Department Home. 2007. Web. 26 Apr. 2011. <<http://music.case.edu/~rwd/JustTuning/Contents/Possible.html>>.
- White, Crystal R. *FlexTune Software Icon*. 2011. Image.

APPENDIX A: USER MANUAL

I. System Requirements and Installation

a. System Requirements

- MacOSX
- Must have Java 1.6 or higher

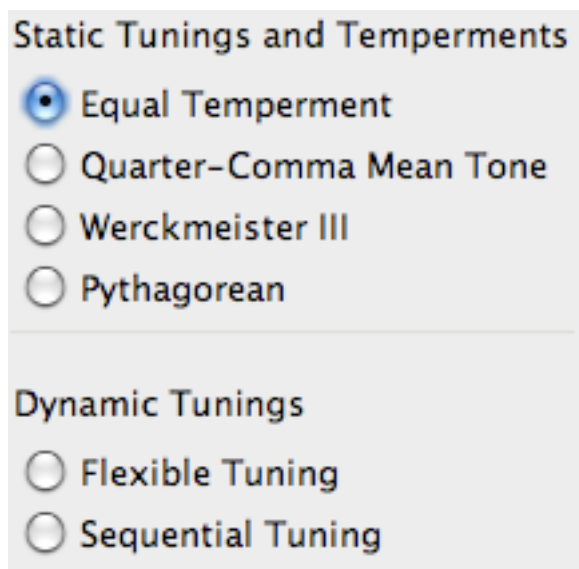
b. Installation

- Insert FlexTune Disk into your computer
- Copy FlexTune folder into your Applications folder
- FlexTune.app is inside the FlexTune folder. This file may be moved anywhere on your computer. It is suggested that you keep FlexTune.app in the FlexTune folder and create a shortcut to it if you wish to run it from another location.

II. Main Window Functions

a. Tuning Options

In the main window the tuning options panel (Figure 1) appears in the upper left hand corner. These tuning options control the tuning of anything played using the FlexTune software, whether it is one of the pre-programmed songs, notes played through the virtual keyboard, or notes played using an external MIDI device.



- Figure 1: Tuning Options Panel -

To change the tuning, just click on the radio button next to the tuning you wish to experiment with. A more detailed explanation of the tuning options available is discussed below.

i. Static Tunings

1. **Equal Temperament** – Equal Temperament is the most common musical temperament today. Each octave is broken up into twelve equal intervals making the ratio between the frequencies of any two adjacent notes equal. All frequencies are most commonly based off the standard of A being 440 Hz.
2. **Quarter-Comma Mean Tone** – Mean tone frequencies are determined very similarly to the method used in Pythagorean tuning (see below). The difference is the fifths in mean tone are tuned narrower by a quarter comma, allowing major thirds to be tuned justly.
3. **Werckmeister III** – This tuning is based on perfect fifths, but to correct for tuning disparities certain fifths are tempered a little wider, and others narrower. Specific temperament between thirds is also included in this tuning method.
4. **Pythagorean** – This is the oldest known tuning method, attributed to Pythagoras around 500 BCE. It starts with one note and then extrapolates across the keyboard by tuning perfect fifths. After all the fifths relative to the original note are tuned, all other notes can be tuned as relative octaves from one of the tuned notes. This method results in some tuning disparities the farther the notes are from each other on the circle of fifths.

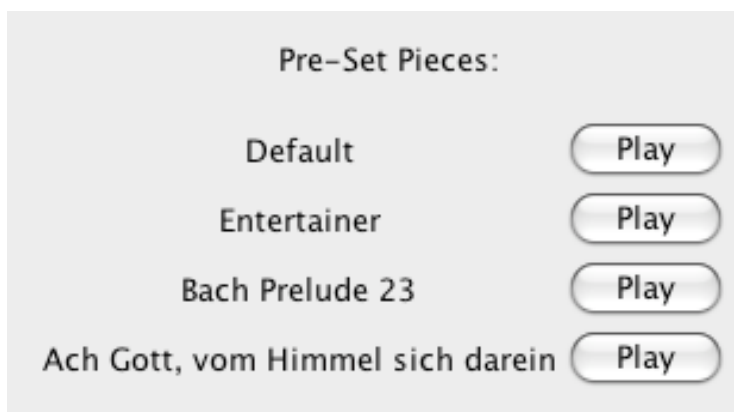
ii. Dynamic Tunings

1. **Flexible Tuning** – This tuning calculates the frequency of each note played based on the currently sounding pitches, and if no notes are sounding the frequency will be calculated on the last note that was released. This is the more practical of the two dynamic tunings because if a note is currently sounding, the new notes being played will be in tune with the held note.
2. **Sequential Tuning** – This tuning calculates the frequency of each note played based on the last note to be pressed, regardless of notes currently sounding

or in what order previous notes were released. This tuning is less practical for playing pieces in tune, but it more effectively demonstrates the flexible nature of the non-static tunings. It will often sound strange because if a note is being held, the new notes being played will not necessarily be based of the sounding note and it is likely that multiple notes sounding concurrently will not be consonant.

b. Pre-programmed Songs

There are several songs that are pre-programmed into the software that users may utilize when experimenting with different tuning options. In order to play a pre-set piece, simply click the play button next to the song you wish to hear (Figure 2).



- Figure 2: Pre-Programmed Songs Panel -

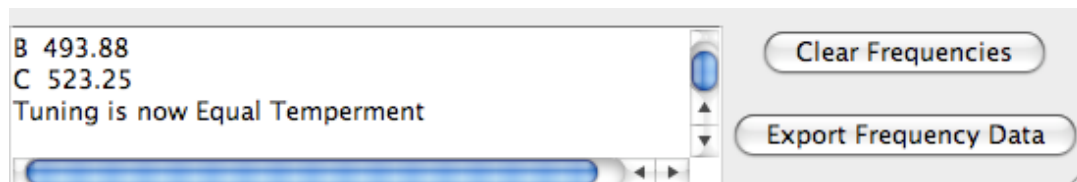
When a song is playing the play buttons will become disabled, and a stop button will appear (Figure 3). If at any point during the song you wish to stop the piece you can click the stop button. After the song has finished, or the stop button is clicked, the play buttons will once again be enabled.



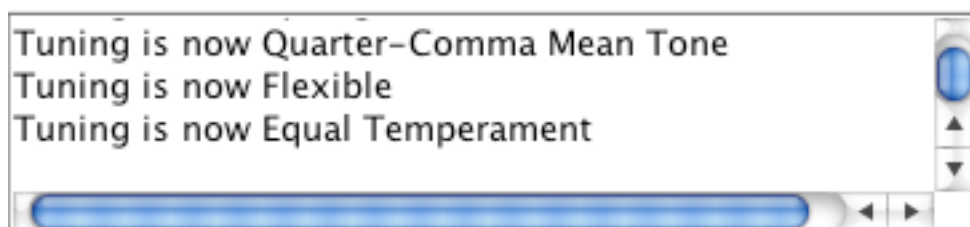
- Figure 3: Disabled play buttons, and stop button visible -

c. Frequency Display

Across the bottom of the main window there is a frequency display (Figure 4). This window contains the frequency data for every note played using the FlexTune software. Tuning change messages are also displayed in the frequency display so it is easy to tell when the tuning changed (Figure 5).

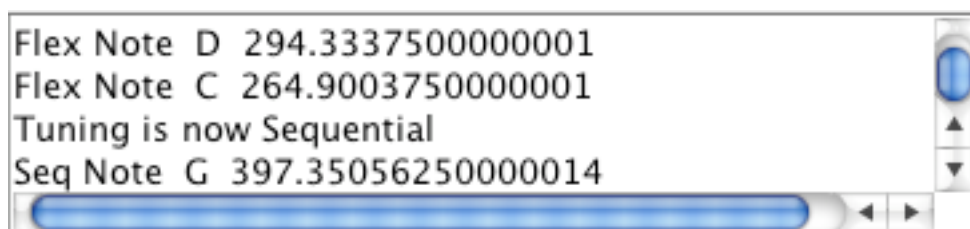


- Figure 4: Frequency Display -

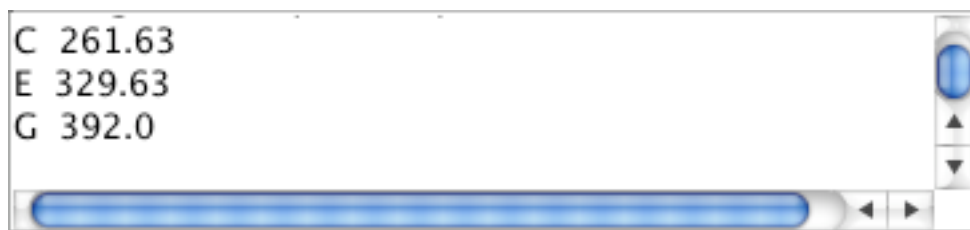


- Figure 5: Tuning change messages -

For flexible and sequential tunings the data for each note indicates which tuning method was used, the note name, and frequency played (Figure 6). For static tunings the data for each note consists of the note name and frequency (Figure 7).

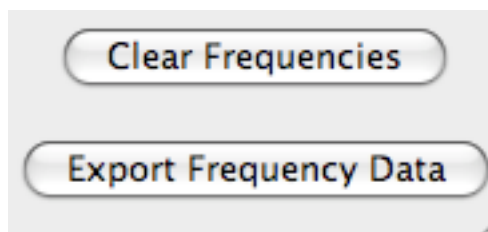


- Figure 6: Example of Flexible and Sequential Frequency information -



- Figure 7: Example of Static Tuning Frequency Information -

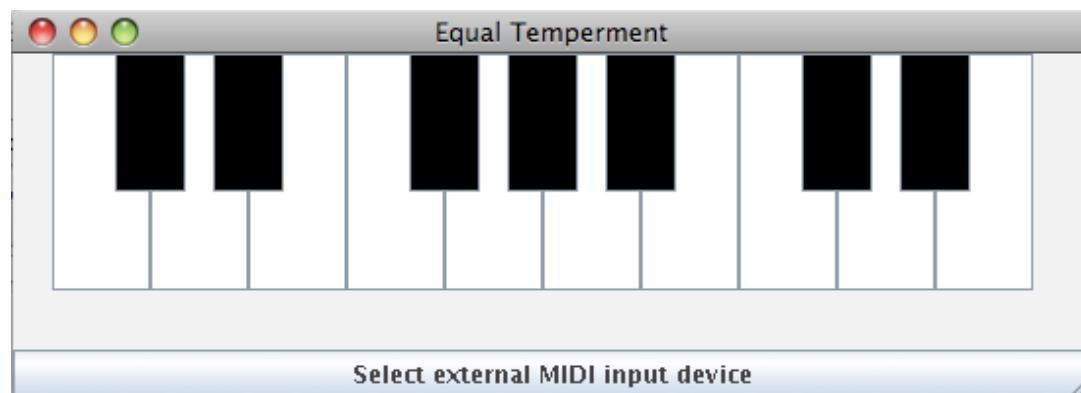
There are two buttons next to the frequency display (Figure 8). The 'Clear Frequencies' button clears the frequency display panel of all tuning change and frequency data. The 'Export Frequency Data' button will open up a save file dialog. You can choose where to save the frequency data, and all the tuning status changes and frequency data displayed in the frequency display panel will be saved to the specified location as a .txt file.



- Figure 8: Frequency display option buttons -

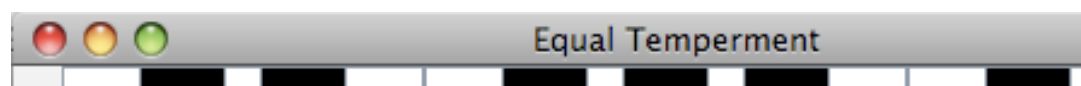
III. Musical Input

There are two ways for a user to directly control the musical input to the program. There is a virtual keyboard that appears in another window when the program opens (Figure 9). The user can control the virtual keyboard using their computer keyboard. Also, an external MIDI device may be connected to the computer, and can be set to interface with the program.



- Figure 9: Virtual Keyboard Window -

The title at the top of the keyboard window will always display the name of the tuning being used (Figure 10). Anything played through the virtual keyboard or MIDI device will be played according to the tuning displayed on the keyboard window.



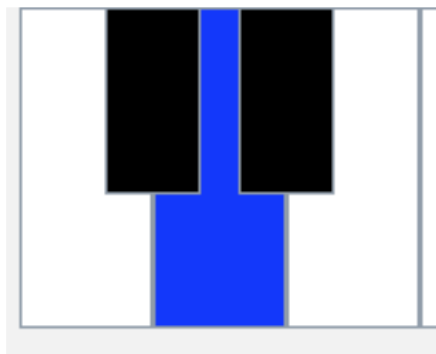
- Figure 10: Tuning Status on Keyboard Window -

a. Virtual Keyboard

The virtual keyboard is made up of buttons representing the notes from middle C to the E an octave above middle C. When clicked with a mouse, each note will sound for one second. This method is relatively dissimilar to playing a regular piano and the duration of notes cannot be controlled.

Alternately the user can type on their computer keyboard to control the virtual keyboard, and this is the recommended use. The 'a' key is mapped to middle C, the 's' key to the D above that and so on until ';', which is mapped to the E an octave above middle C. These 'home row' keys control the white notes of the keyboard. The black keys are controlled by the keys above and in between the white keys. For example, 'w' is mapped to C#, 'e' is mapped to D#, 't' is mapped to F#, and so on for the rest of the keyboard.

If in doubt you can tell which notes you are playing because they will be highlighted in blue while that note is sounding (Figure 11).



- Figure 11: Highlighted notes are currently being played -

b. External MIDI Device

Through the virtual keyboard window you can connect an external MIDI device to the program. On the bottom of the window there is a button with the text "Select external MIDI input device" (Figure 12).



- Figure 12: Select external MIDI input device button -

When you click this button a selection window is displayed (Figure 13). The program detects all MIDI devices currently connected and places the information about them in the drop down menu. If you see more MIDI devices in the list than are

connected to your computer this is probably because the program is detecting programmatic MIDI devices built into your computer. Select the MIDI device you wish to use as input to the program and click 'Ok'.



- Figure 13: Available MIDI devices selection dialog -

Once you have selected a MIDI device the program will attempt to connect to it. If this attempt is successful you will see a success message (Figure 14). If not, a failure message will be displayed (Figure 15). If the MIDI device fails to connect it means that the program does not recognize it as a valid MIDI input device. Make sure you have connected the MIDI output port correctly to the computer. If it still does not work, check the manual that came with the device for instructions on connecting it to the computer.



- Figure 14: MIDI device successfully connected -



- Figure 15: MIDI device connection error -

Once you have successfully connected the MIDI device you can use it with all the features of the FlexTune Software. Notes played within the range of the virtual keyboard will highlight in blue while being played. The current tuning selection will apply

to all notes played on the external MIDI device, and will be played through the computer's speakers. Also, all frequency data for notes played will be displayed in the frequency display panel.

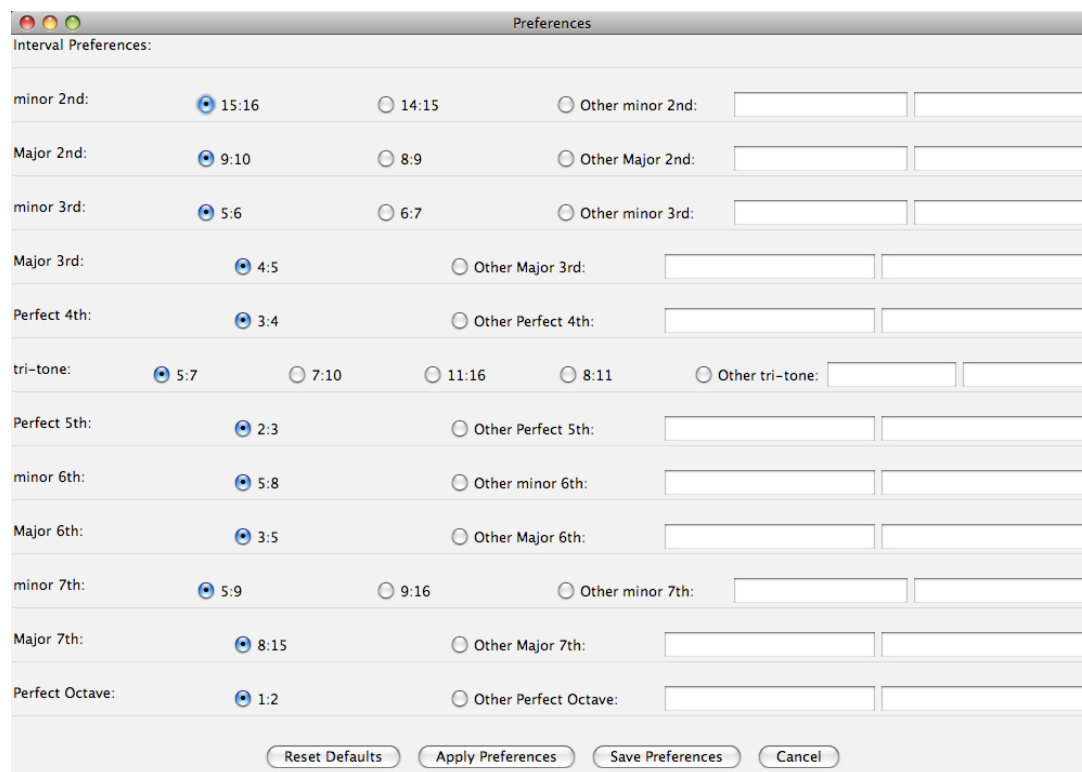
If you are using an external MIDI device with built-in speakers it is suggested that you turn the volume off as the FlexTune software will be producing sound through the computer, and anything heard through the MIDI device speakers will not be tuned according to the FlexTune software.

IV. Preferences

Intervals are perceived as in-tune when the ratio of the frequencies are the appropriate small-integer ratios. However, some intervals have more than one small-integer ratio that is an appropriate tuning based on the harmonic series. For example, a Major Second can be a 9:8 or a 10:9 ratio (9:8 is generally considered appropriate for the interval between Do and Re in a scale, whereas 10:9 is appropriate for the interval between Re and Mi).

When the computer is mathematically calculating the appropriate frequencies it cannot tell which ratios are appropriate on a case-by-case basis, partly because what ratio is considered appropriate might be subject to the opinion of the composer or performer. Therefore, included in the program is a way for the user to specify their preferences as to which ratios are used for each interval when the program is set to Flexible or Sequential tuning.

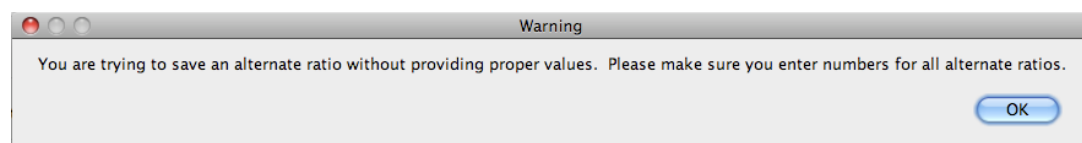
To access the preferences window, click the FlexTune drop down menu at the top of the screen and then click 'Preferences'. The Preferences screen will then be displayed (Figure 16). Here you will see several suggested ratios for each interval, and an option to specify an alternate ratio as you see fit.



- Figure 16: Preferences Window -

There are four buttons at the bottom of the Preferences window: ‘Reset Defaults’, ‘Apply Preferences’, ‘Save Preferences’, and ‘Cancel’. Clicking ‘Reset Defaults’ will reset all the radio buttons to their default locations. This will *not* save the defaults as the active ratios, it will just reset the radio button locations. In order to make the default ratios the active ratios you must click ‘Save Preferences’ or ‘Apply Preferences’. The only difference between the Save and Apply options is that Save also closes the window, and Apply will set the checked intervals to be the active intervals but will keep the window open if you wish to make more changes. ‘Cancel’ will close the window and discard any changes to the radio buttons since the last time ‘Apply Preferences’ was clicked.

When trying to set a ratio for an interval other than the suggested intervals you must type a number into the appropriate text boxes. If you don’t, when you try to apply or save the changes you will see an error message (Figure 17) and changes to the radio buttons will not be saved. If you are not specifying an alternate ratio the text boxes may be left blank.



- Figure 17: Error message for blank text box in Preferences window -

V. About

The about window (Figure 18) can be accessed through the drop down menu by clicking ‘About FlexTune’. This window contains information about the program.



- Figure 18: About FlexTune Window –

The about window has four buttons across the bottom. Clicking them will open up additional information in another program. The Capstone Summary, full written report, and User Manual are PDF files, and will be opened in your current default program for viewing PDF files. The overview video is a QuickTime movie that gives a brief overview of the concepts behind the FlexTune software as well as a demonstration on how to use the program.

VI. Troubleshooting

Described below are some common problems that may arise with the FlexTune software and how to fix them.

a. Double click the program, but it fails to run.

i. Make sure the computer is running the correct version of Java

FlexTune software requires at least Java 1.6. Java updates are free, and available in the download section of Apple’s website. If a software update has not been performed in a while it is also possible that the latest Java update will be installed the next time a software update is performed.

ii. Ensure the FlexTune resources are in the proper location

In order for the FlexTune software to run, the FlexTune folder that the application came in must be in the Applications folder. The FlexTune folder may be found on the FlexTune CD. Just drag and drop the FlexTune folder into the Applications folder. If this is not working,

check to make sure that inside the FlexTune folder there is another folder named 'extras'.

b. Notes are being played but not sounding / frequencies in the display window appear as 'null'

i. Ensure the FlexTune resources are in the proper location

In order for the FlexTune software to run, the FlexTune folder that the application came in must be in the Applications folder. The FlexTune folder may be found on the FlexTune CD. Just drag and drop the FlexTune folder into the Applications folder. If this is not working, check to make sure that inside the FlexTune folder there is another folder named 'extras'.

c. Attempting to play a pre-set piece results in a file not found error

i. Ensure the FlexTune resources are in the proper location

In order for the FlexTune software to run, the FlexTune folder that the application came in must be in the Applications folder. The FlexTune folder may be found on the FlexTune CD. Just drag and drop the FlexTune folder into the Applications folder. If this is not working, check to make sure that inside the FlexTune folder there is another folder named 'extras'.

d. Keys are being pressed but the virtual keyboard is not responding

i. Make sure the virtual keyboard window is the active window

In order for the virtual keyboard to recognize input from the computer keyboard, the virtual keyboard window must be the active window. To make the virtual keyboard the active window, simply click on it.

e. Notes are sounding, but no pre-set piece is playing and no keys are being pressed.

i. Return to virtual keyboard window and press each sounding note

Sometimes the virtual keyboard window stops being the active window while a note is being played. This may happen, for example, if another application pops up, or a different window is accidentally clicked on. If this happens, the virtual keyboard is not notified when the key is released, and does not know to stop it from playing

To remedy this problem, simply click on the virtual keyboard window and then press and release each note that is still sounding. This will reset the notes and the virtual keyboard will be back to normal.

ii. Play sounding notes on external MIDI device

Depending on the MIDI device, sometimes when a pre-set piece is playing and the MIDI device is being used simultaneously interferences can occur. This is not usually a problem, and if anything will normally result in notes *not* being played. If, however, a note is sounding without stopping it can be reset by pressing and releasing the corresponding note on the external MIDI device (or virtual keyboard if the note is in the covered range).

iii. Restart the program

If the sounding note cannot be restarted by pressing and releasing the corresponding key on the virtual keyboard or external MIDI device, restart the program by quitting and re-opening the FlexTune application.

f. Multiple keys are being pressed on the virtual keyboard and/or external MIDI device, but they are not all sounding

i. There is a limit to the number of notes that may be played simultaneously

When playing on the virtual keyboard it is possible to play a maximum of five notes at a time. When playing on an external MIDI device it is possible to play a maximum of ten notes at a time.

g. The frequency display window does not seem to be updating

i. Scroll to the bottom of the frequency display window

New messages added to the frequency display window are added to the bottom of the currently existing list. When the data is updating quickly (such as when playing a pre-set piece) the program does not always show the very bottom of the frequency display data. Scroll down in the frequency display window to see all of the frequency data.

To view the frequency data more conveniently, click the 'Export Frequency Data' button and export the frequency data to a text file.

h. External MIDI device does not appear on MIDI device list

i. Ensure it is properly connected to the computer

FlexTune software only takes in MIDI data, and does not send any data to the external MIDI device. Make sure the MIDI *in* cable is connected to the MIDI *out* port of the device, and that the cable is securely connected to the computer.

If the problem persists, consult the manual that came with the external MIDI device for instructions on connecting it to the computer.

i. External MIDI device failed to connect

i. Ensure it is properly connected to the computer

FlexTune software only takes in MIDI data, and does not send any data to the external MIDI device. Make sure the MIDI *in* cable is connected to the MIDI *out* port of the device, and that the cable is securely connected to the computer.

If the problem persists, consult the manual that came with the external MIDI device for instructions on connecting it to the computer.

j. More MIDI devices than are connected to the computer are appearing in the MIDI device list

i. The extra device(s) are most likely internal to the computer.

Java has a built-in MIDI sequencer and synthesizer. These devices, although not external, register to the computer as

valid MIDI devices, and will therefore appear on the MIDI device list. Other internal MIDI devices may exist on some computers, and they may also appear in the MIDI devices list.

k. External MIDI device does not work with FlexTune software

i. Not all MIDI devices are created equal, so some may not be compatible with FlexTune software

MIDI is a standard for transmitting, receiving, and processing musical data. However, not all MIDI devices follow the standard exactly. Depending on the MIDI device it may or may not be compatible with this program.

The only requirements for a MIDI device to be compatible with FlexTune software are:

- * It must send MIDI note on and note off messages
- * Note off messages must have a status of 128 or 144
- * Note on messages must have a status of 144
- * The MIDI note numbering standard must be followed

l. Clicking the buttons on the About window does not work

i. Ensure the computer has software installed that is capable of reading PDF files and playing QuickTime movies

When a button on the About window is clicked, the FlexTune software delegates the task of opening the associated document to the computer. If an appropriate application for viewing the document is not installed or not set as the default application for opening documents of that file type, the file will not open. The written documents require a program capable of displaying PDF files. The overview video will require a program capable of playing QuickTime videos.

Appendix B: Pre Set Pieces

Ach Gott, vom Himmel sieh darein

253.

1. 2.

By J.S. Bach

Available at <http://erato.uvt.nl/files/imglnks/usimg/1/19/IMSLP03822->

4PC_3.PDF

Public Domain

THE ENTERTAINER

♩ = 200

Track 1

Copyright © 2011 My Music Company

By Scott Joplin

Public Domain

Arranged by Lisa Mescon

PRAELUDIUM XXIII.

Prelude XXIII in B Major, from *Das Wohltemperierte Klavier I*

By J. S. Bach

Available at <http://216.129.110.22/files/imglnks/usimg/1/10/IMSLP02228->

BWV0868.pdf

Public Domain

Flexible/ Sequential Tuning: Beyond Equal Temperament

SUMMARY

The premise of this creative Capstone project was to develop a computer instrument that is capable of tuning itself flexibly in such a way as to not require a tempering of the Western scale, as is necessary for fixedly tuned instruments. The difficulty in creating such a system of tuning arises from the mathematical paradox of the *musical harmonic series*. The harmonic series, or *overtone series*, is the sequence of frequencies that sound naturally as overtones over a fundamental pitch. They follow a proportional pattern of 1:2, 2:3, 3:4, etc. These small-integer ratios of frequencies represent the consonant (in-tune) harmonic intervals. When an interval does not have a small-integer ratio between its notes, it is perceived as “out-of-tune,” and the two frequencies will compete with each other. As the frequencies create less compatible sound-waves, the two waves will interfere, creating acoustical *beats*. These beats slow and eventually diminish as the interval is tuned more closely to the appropriate small-integer harmonic ratio.

The problem becomes apparent when the ratios are used to create the Western scale on a fixed instrument. When tuning such an instrument (for instance, a piano), the intervals inherently cannot all be tuned *justly*, or according to the appropriate proportions. The nature of the fractions creates an asymptotic sequence, in mathematical terms, which ultimately does not allow the larger intervals to be explained exactly by the smaller intervals, while they are expected to coexist and be used simultaneously in Western music. For example, a perfect

octave in music is the equivalent of three major thirds; however, the justly tuned octave (2:1) is not equal to the sum of three major thirds (5:4); $5:4^3=125:64$, not 2:1. The difference, then, between notes tuned in terms of different intervals is known as a *comma*.

Historically, these problems with intervallic relationships have been addressed by means of *temperaments*; the commas are incorporated into the scale in various intervals. In Pythagorean tuning, one of the earliest temperaments, the scale was tuned by a sequence of perfect fifths. As demonstrated earlier, a sequence of fifths (a 3:2 ratio) will prove problematic, especially at the octaves. For instance, if the temperament is based on D, the frequency of D (288 Hz) is multiplied by $3/2$ to obtain the frequency of A, which is multiplied by $3/2$ to obtain the frequency of E, and so forth, tuning in succession B, F \square , C \square , and G \square . Likewise, D is multiplied by $2/3$ to obtain a fifth below at G, and the notes C, F, B \flat , E \flat , and A \flat are tuned with the same pattern. This determines frequencies for all twelve chromatic notes of the Western scale, with one major issue; A \flat and G \square , which are enharmonic spellings of the same note (different nominal titles for the same pitch on an instrument), are expressed as two different frequencies. To remedy this, the A \flat is disregarded, which makes the interval between the G \square and the E \flat especially dissonant, and is commonly known as a *wolf interval*.

Especially as keyboard instruments became more prevalent, new methods of tuning were developed to mitigate the problem of the wolf interval. These new tunings took the comma and distributed it throughout various intervals in the

scale, choosing which intervals to keep tuned justly and which to *temper*. This led to *well-tempered* tuning systems, such as the *Werckmeister III*, which was the temperament used by Bach when he composed his *Das Wohltemperierte Klavier*, or pieces for a well-tempered clavier. These temperaments distributed the comma among various intervals, not equally among each, but rather just in such a way as to make it possible to play in all keys with minimal dissonance. Frequently used keys, such as C major, would have less noticeable dissonance, while less frequently used keys, such as E^b major, would contain distinctively out of tune intervals. Each key, then, was associated with a different amount of tension caused by dissonance, and thus a particular sense of mood or character. Composers were thus able to utilize these harmonic tensions to reflect the changing emotions of a pieces, or reflect the dramatic tensions of the work.

When the technology to register and measure frequencies was developed, it became possible to tune keyboard instruments to *Equal Temperament*. In this temperament, the octave is simply divided into twelve equal semi-tones, so that only the octave is justly tuned, but because of the consistent nature of the out of tune intervals, society has learned to accept the equally tempered intervals as correct and consonant, or in-tune. While this method of tuning is equal, it loses the characteristics of the individual keys, and the tensions that come from the varying distributions of the comma, while also still sacrificing justly tuned intervals, which were at least playable in some keys in historical tunings.

The purpose of this Capstone project was to create a computer program that can function as a self-tuning instrument, which is able to overcome the

impossibility of perfectly consonant temperaments by mathematically calculating the frequency of each pitch played in relation to the pitch preceding it. Though this allows every interval between sequentially played notes to be in tune, it does mean that pitches are not fixed, and rather are flexible and changing as a piece progresses. For example, while pitch A will be a frequency of 440 at the start of a piece played using this program, as the piece goes on, it changes relative to the other notes, and can (and likely will, depending on the intervals of the piece) be a completely different frequency by the end. The program possesses the capability to play several historical fixed temperaments, namely Pythagorean, Quarter-Comma Meantone, Werckmeister III, and Equal Temperament. However, it also is capable of playing in two non-fixed tuning systems, distinguished as *Sequential Tuning* and *Flexible Tuning*.

The Sequential Tuning system uses the ratios in the harmonic series to tune each note played in proportion to the most recent note pressed. This creates a significant impracticality, however, in the cases of a note being held while other notes continue to play. Because this system tunes based on the last note pressed, the program disregards the held note, and tunes solely based on the notes being played while the note is held. This creates obtrusive dissonances between the held note and the notes tuned sequentially before it is released. Despite the impracticality of this method as a tuning system, it has value in the clarity with which it is able to demonstrate the shifting of pitches with a flexible system of just tuning, and so has distinct educational value.

The Flexible Tuning system, conversely, remedies this issue by first tuning notes based on a held note, and if there is no note being held, then on the most recent note released. This eliminates the noticeable dissonances of the Sequential Tuning system, while still tuning each note flexibly and in real time based on its interval relationships to other notes played before and at the same time as it. In this way, every interval sounding is the appropriate small integer ratio for that interval in the harmonic series.

For both of the unfixed tuning systems available in this program, users are able to access a Preferences Window, containing a set of ratio options for each interval in the scale. With the development of Equal Temperament, intervals became homogenized as sums of semi-tones, calculated as twelfths of the 2:1 octave; however, in the harmonic series, several versions of some intervals exist and sound as consonant intervals (small-integer ratios). For example, a Major Second can be a 9:8 or a 10:9 ratio (9:8 is generally considered appropriate for the interval between Do and Re in a scale, whereas 10:9 is appropriate for the interval between Re and Mi). Therefore, users are given suggestions of several ratios that can be considered somewhat close to the accepted ratios for each interval, but can also choose and enter in any ratio that they find fit. The program will then tune all instances of that interval to the chosen ratio.

The program is able to play both the historical and unfixed tuning systems based on computer or MIDI keyboard input. This allows users to play in the different temperaments/tuning systems in real time, or to hear pre-set songs played in any of the systems. It also provides a window display of the frequency

(in Hertz) of each note that sounds, which can be exported as a log of the frequencies, allowing the user to see how the notes shift and change as they are calculated based on other notes in the unfixed systems.

The most immediate potential application of this program is as an educational device, providing a demonstration of the imperfections of fixed temperaments, and also demonstrating the rarely heard justly-tuned intervals. With the consistency and homogeneity of Equal Temperament, most people have become so acquainted with the incorrect intervals that they are not aware of the acoustical beats that signify the poor compatibility of the pitches. This program allows listeners to hear the potential of music to be truly in-tune. It also has the potential to provide a practice opportunity for musicians playing flexibly tuned instruments, such as string instruments or trombones. In an ensemble of flexibly tuned instruments, the musicians should tune as best as they can to the actual, just intervals. However, when practicing with a fixed instrument in Equal Temperament, they are forced to modify their intervals to the incorrect ones created by that system. This program can allow them to practice tuning to correct intervals, to prepare them for ensemble rehearsals. Though this may cause them to differ from the typical tuning of instruments and ensemble in equal temperament, there are many ensembles (such as the Hilliard Ensemble) that work and experiment with various historical tunings, who would likely be interested in practicing with a program that gives them this flexibility. The program allows for the conventionally statically-tuned keyboard instrument to adapt and change in a

way that presents music in a drastically new way, allowing it to be heard with entirely consonant, in-tune intervals.