EjiTune: A System for Approachable Extended Just Intonation

Maxwell Miller

In today's world, tuning is frequently reduced to a binary assessment; things in tune, and things out of tune. Historically, however, the definition of what we declare 'in tune' has been a topic of debate, and there have been many changes in how we tune our instruments through time. Many of these solutions represent compromise in one or more ways, whether due to musical application, design and construction of instruments, or even aesthetic choices. With the advent of modern computing, we are free to explore new tuning systems and experiment with how we choose to compromise.

EjiTune is, at its core, a system that allows its user to explore vertical harmonies that are tuned utilizing extended just intonation ratios, while still retaining the horizontal intelligibility of melodies and other moving lines. While 5 and 7-limit systems allow for more consonant sonorities than our current 12 tone system, they come with their own share of downsides as well, such as wolf intervals or unfamiliarly sized steps[1]. EjiTune seeks to answer this by using 12 tone equal temperament as its tuning base, but then dynamically re-tunes incoming pitches into just ratios to achieve sonorities without the presence of beat frequencies. In addition to this, EjiTune has been designed to work with standard MIDI data, is written in JavaScript, and can run in the creative coding software Max 8.

Existing Tools and Systems

This tool is not the first of its kind to utilize adaptive tuning principles. A handful of different adaptive tuning systems have been built and documented, but each system seeks to answer different aspects of the tuning 'compromise'. One of the earliest of these systems was the

[1] Gann, "The Arithmetic of Listening," 111-2.

Hermode Tuning System, developed by Werner Mohrlok[2]. The Hermode system sought to achieve better intonation by utilizing just ratios, but still sound pleasing when played alongside other equal tempered instruments. Mohrlok achieved this by developing a system such that the weighted difference between any given note and its equal tempered counterpart was averaged, and the whole chord shifted accordingly. This allowed the pitch corrections it made to be relatively small, and allowed it to be incorporated into different softwares without the end user noticing drastic re-tuning while using it[3].

While the Hermode system is good at rendering general improvement to tuning, it is not very adventurous in its design, only offering approximation of the septimal seventh in a single mode, and functions most smoothly when using a tempered 9/5 minor seventh, as the septimal seventh has such a high discrepancy from 12 tone equal temperament that it causes undesirably large shifts in the other chord tones[4]. It does have another mode which is unique, however, called HMT Baroque. This mode analyzes what key center is being played and configures a tuning similar to the unequal temperaments used during the baroque period, such that the chords in the key all end up being tuned similarly, eliminating some of the undesirable shifts from chords with beating to chords without beating present in the regular HMT function[5].

Since the development of the Hermode tuning system, other systems have surfaced seeking to answer the same proposition; how can computing technology help us to improve the general quality of vertical sonority? David Code developed the Groven.Max system as a software emulation of Eivind Groven's 36-tone pure tuned organ. This system, originally conceived to filter MIDI data to a set of three player pianos, tuned a syntonic comma apart, uses

[2] Mohrlok, "The Hermode Tuning System," 1.
[3] Ibid.
[4] Ibid., 4.
[5] Ibid., 9.

a logic switching system in conjunction with tuning tables to be able to grab the correct pitches to be played given any chord[6]. Dmitri Volkov developed a VST MIDI effect called Pivotuner, which allows the user greater flexibility in how intervals are re-tuned, and primarily uses the key held the longest as being the anchor point for the re-tuning[7]. While these systems are more specialized than their Hermode predecessor, they both fail to answer how extended just intonation can be realized in a flexible manner with software. Filling this gap was the motivator for the development of EjiTune.

Guiding Principles

EjiTune is written in JavaScript, and runs in the Max programming environment. JavaScript was chosen as the language for this project as its non-strict typing of variables allows for fast and effective development, and the ease of porting to other software, such as SuperCollider, in the future. The Max environment allowed for easy management of MIDI and audio connections, and the functional prototype of the software feeds its output into a polyphonic synthesizer built in Max. Though we are utilizing the power of modern computing, there are still compromises that must be made in terms of tuning, and EjiTune is not exempt from this. Thus, there are several guiding principles for how EjiTune should function.

First, the system should be able to render noticeable improvement to existing MIDI files, and implement seamlessly with other hardware. EjiTune takes note and velocity pairs as its input, and is subsequently easy to 'interject' within other processes. Conversely, the output should be something that is simple to interpret and interfaces nicely with the software that it is running in. To accomplish this, the system outputs its data in the format of [MIDI Note,

[6] Code, "Groven.Max," 52-3.
[7] Volkov, "Pivotuner," 2.

Velocity, Frequency]. This allows the tool to still utilize the automatic voice allocation present in the poly~ object, while driving the oscillators present within the synth with a frequency in hertz.

The second guiding principle is the intelligent allowance of upper limit ratios when tuning intervals. EjiTune incorporates both the septimal seventh as well as the natural eleventh harmonic into its tuning system, and allows for both traditionally dissonant harmonies (diminished chords and clusters) as well as extended chords (such as ninth and eleventh chords) to be rendered with less beating. This use of the eleventh harmonic as well as the greater variance in intervals is something forgone by the adaptive 5-limit system aptly named Just Intonation developed by Karolin Stange, Christoph Wick, and Haye Hinrichsen at MIT, which only utilizes the 7/4 minor seventh as an alternative tuning parameter[8].

The final guiding principle of the system is the maintaining of horizontal intelligibility within melodic lines. While fixed just intonation tuning systems can allow for better realization of certain harmonies, the concurrent increase in the number of possible step sizes renders familiar melodies unrecognizable and alien. Thus, the most important aspect of the EjiTune system is that it still uses the equal temperament system as its base tuning, such that 100 cent semitones (and other resulting melodic intervals) are preserved.

Note Information and Anchor Points

Getting note information into the program is the simplest part of the process. When sent a MIDI pitch/velocity pair, EjiTune first checks to see if it is a note-off message (a note with a velocity of 0) and immediately sends the appropriate note-off information if it is. It then splices that pitch out of the note array and deletes the frequency information for it. If it is not a note-off message, however, it then appends this information onto a two dimensional array. Using a two

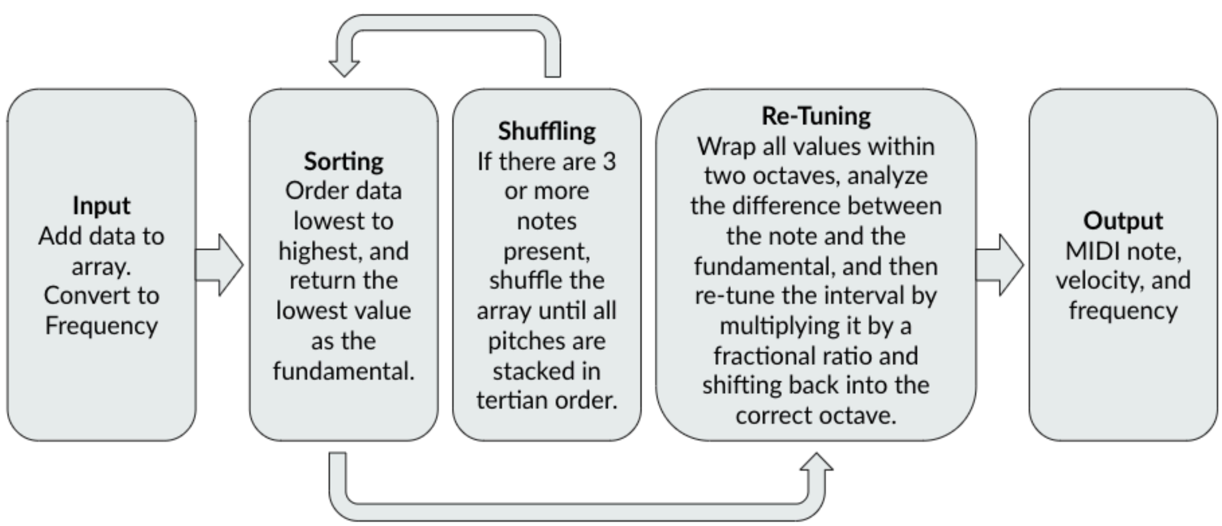
[8] Strange, Wick, Hinrichsen, "Playing Music in Just Intonation:," 57.

dimensional array allows the system to keep the original velocity information stored in the same index as the MIDI note, which aids in recall when outputting.

Since the MIDI note itself is not entirely helpful to multiply by our fractional ratios from the harmonic series, we reach the first challenge. Retaining the original note number will be helpful to track what pitches are being played on the keyboard as well as having an easy way of parsing note-off messages, so we instead convert the MIDI note to a frequency and store it in a parallel index of a second array. The base pitch for the entire tuning process is movable by the user, but defaults at A4 = 440 hertz.

It is because of this process that our system retains its anchor to 12 tone equal temperament. Since EjiTune always receives whole number MIDI notes and calculates its initial frequency from that, it behaves in a way such that playing familiar melodies such as 'Happy Birthday' and 'Mary had a Little Lamb' are rendered in the equal tempered system the user is accustomed to with no apparent tampering.

Figure 1: Data Flow in the EjiTune System



Calculating Fundamentals

While data flow within the program is relatively simple, the most difficult part of the re-tuning process comes when deciding what pitch to actually tune to. Volkov's Pivotuner system utilizes a switching method that allows it to change how it determines what 'key' it should be tuning to[9]. While this flexibility is nice, the methods available have some drawbacks. The Length Key Determiner has the advantage of allowing held notes to not be re-tuned between chord changes. The major drawback of this system is that the onset of notes must be carefully controlled, and if a performer is not attentive to how they attack notes in a chord, it is possible for the tuning of chords to shift when repeating chords without the tuning lock enabled. The second mode is the Bass Key Determiner, which uses the lowest note of the chord to determine how tuning should happen[10]. While this will give consistent tuning for repeated chords, desirable qualities such as the septimal seventh will be lost when playing the chord in an inversion. The final mode is a Chord Key Determiner, which allows the user to input chords and different permutations of notes, and subsequently dictate how they should be tuned[11]. This mode allows for musical intelligence to be applied to the tuning, at the expense of time spent for the user.

The original Hermode Tuning System uses a method similar to the Chord Key Determiner utilized by Volkov, but instead of being controlled by the user, it instead searches for predefined pitch structures built out of a small set of pure intervals[12]. Mohrlok's approach acknowledges the somewhat subjective nature of analysis and chooses to compromise the tuning of anything outside of its predetermined pitch structures to equal temperament[13]. Code's Groven.Max implementation uses a complicated logic switching system modeled after Groven's

---

[9] Volkov, "Pivotuner," 2.
[10]Ibid.
[11] Ibid.
[12] Mohrlok, "The Hermode Tuning System," 2-3.
[13] Ibid., 3.

own analog computing solution used in the original organ, and allows the user to switch between selected pitches from a 5-limit lattice or an adaptive auto-tuning system[14].

EjiTune, however, takes a more crude approach to solving the issue of determining fundamental frequency. The system takes in data, and retunes itself every time it receives a new note. There is a basic sorting algorithm that is implemented that sorts the pitches from lowest to highest. It then checks to see if there are 3 or more pitches present in the array. If there are, it then executes a shuffle function, where it calculates the interval between each pitch in the array, and shuffles pitches up the octave if the remainder of the difference between each pitch divided by 3 is greater than 1. This allows us to keep major and minor thirds, tritones, perfect fifths, major sixths, and minor sevenths intact, and continuously invert the chord if these intervals in this order are not present. There is also a special exception that detects if there is a cluster of whole tones that will then shift the second pitch in the set up the octave rather than the first. In the case of the set [0, 2, 4] this renders the tuning [1/1, 9/8, 5/4], which is more consonant than the [1/1, 35/32, 5/4] tuning rendered otherwise.

The system then checks this process to see if everything within one octave fits as a chord tone, and will run again, shuffling and checking, until it can determine what the tertian order of pitches is. There is a safety built in to this system, such that it will only attempt to iterate the shuffling process 100 times before aborting, to prevent any accidental crashes if the program is given input that it doesn't expect.

Figure 2: Intervals Available to EjiTune

Interval	P0	m2	M2	m3	M3	P4	TT	P5	m6	M6	m7	M7
Ratio	1/1	16/15 17/16	9/8	6/5 19/16	5/4	4/3 11/8	7/5 23/16	3/2	8/5 13/8	27/16	7/4	15/8 31/16

[14] Code, "Groven.Max," 54-55.

Re-tuning and Output

Once the fundamental pitch is found, the process simplifies immensely. Each note in the stored array is compared to the fundamental, and the remainder of the difference between the two divided by 24 is evaluated[15]. The system then selects the appropriate ratio based on this difference, multiplies the fundamental frequency by that ratio, and then shifts it up or down into the correct octave. These frequency values are all stored in a parallel array for ease of output.

Something of note with EjiTune is that octaves are not re-tuned based on the fundamental frequency, and are simply rendered as their equal tempered counterparts. While this will be an area of focus for future iterations of this tool, the interaction between offsets, ratio multiplication, and the cyclical nature of the re-tuning process rendered certain notes an octave apart in the incorrect register.

Output happens every time a note comes in. Because EjiTune dynamically re-tunes pitches, sometimes with fluctuations of up to a quarter tone, every new event triggers output of every pitch in the currently held array. Due to constraints within the voice allocation systems of the objects in the Max software this tool currently runs in, the tool also sends note-off messages to all active voices to prevent hanging voices as the tuning information shifts.

Tuning and the Future

The EjiTune system has yielded successful results in its preliminary prototyping and development, but still needs some refinement. As mentioned above, the repetition of the interval calculations leads to an alternating pattern of tuning as chords increase in register, which will be addressed in future implementations. The sorting and shuffle algorithms can still be polished,

---

[15] This feature has the unintended consequence of repeating its tuning patterns every two octaves. This will be changed to a logic system within the re-tuning process that will select the ratio based on register, but is not implemented yet in the current release.

and will require careful testing and tracking of input and output to find fringe cases where the triadic shuffling system breaks down.

From an aesthetic standpoint, the greatest flaw in the system is the wealth of 'microtonal inflection' that happens within held notes as harmonies are re-tuned. While some users may find these microtonal shifts desirable, they are apparent and noticeable when realizing familiar songs and can be jarring when listening to melodies that are present within polyphonic textures. Other systems, such as the bitKlavier system developed by Dan Trueman, Aatish Bhatia, and Michael Mulshine, which uses models of springs to create dynamic tuning lattices, behave much more smoothly[16]. Nevertheless, the EjiTune system is a unique implementation of extended just intonation for keyboard instruments, and accomplishes its goal of rendering extended just intervals in a context that is familiar to its user.

---

[16] Trueman, Bhatia, Mulshine, "Tuning Playfully," 73-4.

References

Volkov, Dmitri. 2023. “Pivotuner: Automatic Real-Time Pure Intonation and Microtonal Modulation.” *arXiv.org*.

Code, David Løberg. 2002. “Groven.Max: An Adaptive Tuning System for MIDI Pianos.” *Computer Music Journal* 26 (2): 50–61.

Stange, Karolin, Christoph Wick, and Haye Hinrichsen. 2018. “Playing Music in Just Intonation: A Dynamically Adaptive Tuning Scheme.” *Computer Music Journal* 42 (3): 47–62.

Wandel, Nils, Christian R. Klein, and Reinhard Klein. 2023. “Harmonic Inharmonicity: Eliminating Beats with Quantized Harmonics.” *Applied Acoustics* 213: 109602.

Lysdahl, Anne Jorunn Kydland. 2004. “Eivind Groven's Pure-Tuned Organ: Past and Future.” *Fontes Artis Musicae* 51 (2): 199–216.

Mohrlok, Werner. 2003. "The Hermode Tuning System" Accessed Dec 13th, 2023 at https://sethares.engr.wisc.edu/paperspdf/hermode.pdf

Gann, Kyle. 2019. The Arithmetic of Listening : Tuning Theory and History for the Impractical Musician / Kyle Gann. University of Illinois Press. 109-124.

Trueman, Dan, Aatish Bhatia, Michael Mulshine, and Theo Trevisan. 2019. “Tuning Playfully: Composed and Adaptive Tunings in bitKlavier.” *Computer Music Journal* 43 (2-3): 67–88.