

Muditulib, a multi-dimensional tuning library

Funs SEELEN

<http://www.muditulib.eu>

Utrecht,

European Union,

mail@funsseelen.eu

Abstract

The *Muditulib* library is introduced and explained. Muditulib is mainly a library consisting of a collection of C header files that include functions written for the purpose of tuning tonal music within the diatonic scale. This scale, as well as the library's functions, along with pitch representation systems, will be explained in detail or just shortly with reference to other literature. A music theoretical background is useful, though not necessary. Along with the core of header files an implementation for Pure Data is published. Developers are encouraged to write implementations for other synthesizers, music production platforms or any other link in the chain of tonal music production workflow.

Keywords

Tuning systems, pitch representation / MIDI, software library.

1 Introduction

Muditulib is developed to make the tuning of the common western (diatonic) scale easier, without being restricted to equal temperament of twelve tones per octave with a frequency ratio of $2 : 1$ ¹. Music theorists and mathematicians have developed many tunings for this scale through the ages. Modern software like for example SCALA² can map all possible tunings to MIDI notes. In a flexible environment like Pure Data³ one could rather easily implement such mapping oneself, so that is not the purpose of the library. Muditulib doesn't really map fixed scales to MIDI notes, but offers multiple methods to tune notes or intervals more dynami-

¹The standard equation for translating MIDI notes to frequency is $f = 440 \cdot 2^{((m-69)/12)}$, where m is the MIDI note number and f is the frequency in cycles per second. In this tuning a diatonic semitone equals a chromatic semitone. Moreover, expressed as frequency ratios, a semitone equals the square root of a whole tone, thus, on a logarithmic scale, the semitone equals half a whole tone.

²<http://www.huygens-fokker.org/scala/>

³<http://puredata.info/>

cally. In that respect Muditulib is more familiar to the Hermode tuning system⁴, although the approach is quite different. Both SCALA and Hermode will not be further explained here, for that is beyond the purpose of this paper. The next sentence deserves its own emphasized paragraph.

Muditulib has got nothing to do with microtonality, microtonal music, or microtones, whatever may be meant by those obfuscating terms.

This document is rather intended as an explanation of the software library Muditulib, along with a short summary of my research within the field of tuning and music theory, than purely as a genuine scientific article that describes research goals, methods, and conclusions. Its purpose is to propose several tuning and pitch representation systems to an audience of music software developers.

2 The diatonic scale

In order to understand the approach described here it will be helpful to explain a little bit about the diatonic system, particularly the distinction of variable steps in a scale. This is done most easily by freely citing a recent work by the present author in the next two paragraphs [Seelen, 2014].

The terms chromatic and diatonic descend from the old Greek musical system. Together with the enharmonic they formed the three tetrachords the Greek musical scales were made up from [Grout and Palisca, 1988, ch. 1]. The ancient tuning theory is clearly described by J. Murray Barbour [Barbour, 2004, ch. II]. Today's use of those terms is somehow related to that of their namegivers, although the tetrachord itself lost its value. The diatonic scale is a scale that consists of seven intervals or steps. The eighth note, or the octave, is a repetition

⁴http://www.hermode.com/index_en.html

of the first one, usually with a frequency ratio of 2 : 1. Those seven steps are divided into five larger ones, the whole tones, and two smaller ones, the semitones. The scale then created is actually the same as two Greek diatonic tetrachords on top of each other, at one side overlapping (*conjunct*) and at the other side separated by one whole tone (*disjunct*). By the chromatic scale, however, usually a division of the octave in twelve equal parts is meant, which is quite different from an accumulation of Greek chromatic tetrachords.

So far I described the historical outlines of the system. In the frequency domain the relation between the octave x and whole tone T and semitone s is as shown in equation 1, where $1 < s < T$.

$$x = T^5 \cdot s^2 \quad (1)$$

The citation [Seelen, 2014] ends here. For further reading I refer to the mentioned article. The main point is that the tonal system used as a starting point for the tuning system is a 7-tone and not a 12-tone system, as western tonal music is often incorrectly described as. This idea of a 12-tone system just evolved from practical tuning matters concerning the 7-note system. For clarity: the diatonic scale is a theoretic scale rather than a scale of fixed frequency relationships⁵.

3 Ts , a two-dimensional pitch representation system

Traditional western music notation is based on seven syllables⁶ or alphabetical characters⁷, which correspond to the graphical notes. In contrast to a one-dimensional representation like MIDI note numbers, western music notation makes a clear distinction between for example a C-sharp and a D-flat. All theoretic tone intervals consist of a number of whole tones and semitones, instead of just a number of *chromatic* semitones⁸. The system can therefore be interpreted as two-dimensional. In the previously mentioned article [Seelen, 2014] the $Ts()$ tonal representation system is proposed, consisting of

⁵E.g. the diatonic perfect fifth can be tuned to $\frac{3}{2}$, as well as $2^{(7/12)}$, and many other frequency ratios.

⁶do-re-mi-fa-so-la-ti

⁷C-D-E-F-G-A-B

⁸The word *chromatic* is emphasized for the reason that this term can be interpreted in various ways and is therefore confusing. In this case one twelfth of an octave is meant.

the two values T_n and s_n , the number of whole tones and semitones, respectively. Its advantage is that it can be easily translated to the MIDI note system as is shown in equation 2. Its reference ($Ts(0,0)$) is set equal to the C corresponding to MIDI note 0. Therefore $Ts(25,10)$ corresponds to middle C (lilypond: c').

$$m = 2 \cdot T_n + s_n \quad (2)$$

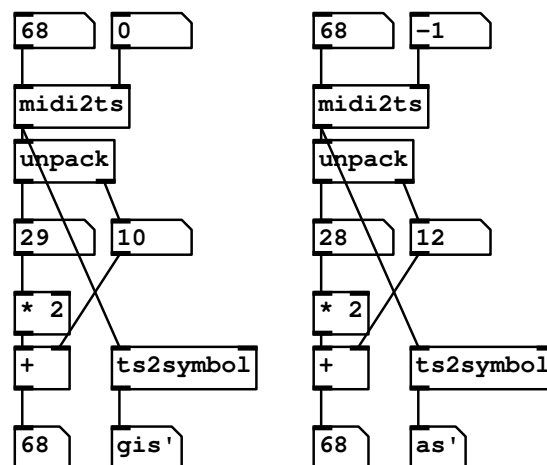


Figure 1: Examples of [midi2ts] and [ts2symbol] in Pure Data.

3.1 Ts to note name symbol

The translation of $Ts(T_n, s_n)$ to a note name symbol (the *Lilypond*⁹ standard) is done by translating the total number of steps ($T_n + s_n$) to the root character plus octave designation¹⁰. Then the deviation from the reference of the root character is calculated and translated into an amount of flattening or sharpening.

3.2 Ts to frequency

This translation can be summarized to equation 3, where f is the frequency, A is the reference frequency for $Ts(29,11)$, corresponding to the note a' , x is the frequency ratio for the octave, and r is equal to $\frac{\log s}{\log T}$ and represents the semitone to whole tone ratio.

$$f = x^{((T_n - 29) + (s_n - 11) \cdot r) / (5 + 2 \cdot r)} \cdot A \quad (3)$$

Usually x is set to 2. The ratio r then defines the kind of tuning. When r is set to about 0.6 the tuning could be said to be within the

⁹<http://lilypond.org/>

¹⁰Division by seven and its remainder (modulo).

mean tone zone. $r = 0.6$ or $r = \frac{3}{5}$ corresponds to 31-TET, for each semitone is made up from 3 and each whole tone from 5 dieses¹¹. The exact meantone ($T = \frac{1}{2}\sqrt{5}$) temperament, Pythagorean, and other examples and how their parameters are calculated are shown in table 1.

Tuning / Temperament	Equation / Calculation	Parameters
Mean tone	$\frac{5}{4} = 2^{2/(5+2r)}$	$r \approx 0.60628$
Pythagorean	$\frac{3}{2} = 2^{(3+r)/(5+2r)}$	$r \approx 0.44247$
	$r = \frac{\log s}{\log T} = \frac{\log \frac{256}{243}}{\log \frac{9}{8}}$	
Tritone temperament	$\frac{7}{5} = 2^{(3/(5+2r))}$	$r \approx 0.59006$
Stretched octave, perfect 5th and 3rds	$\frac{5}{4} = \left(\frac{3}{2}\right)^{2/(3+r)}$	$r \approx 0.63412$
	$x = \left(\frac{5}{4}\right)^{(5+2r)/2}$	$x \approx 2.01246$
19-TET	$s = 2^{\frac{2}{19}}, T = 2^{\frac{3}{19}}$	$r = \frac{2}{3}$
31-TET	$s = 2^{\frac{3}{31}}, T = 2^{\frac{5}{31}}$	$r = \frac{3}{5}$
53-TET	$s = 2^{\frac{4}{53}}, T = 2^{\frac{9}{53}}$	$r = \frac{4}{9}$

Table 1: Tuning examples of the two-dimensional system.

3.3 MIDI note numbers to Ts

Ideally the MIDI note system is skipped in all translations of tonal data. The translation from Ts to MIDI, from two to one dimension as shown in equation 2, leads to irreversible data loss. The same applies to MIDI files exported from Lilypond. However, even if composed diatonic music material wouldn't be translated into MIDI anymore, still improvisations on MIDI keyboards should be interpreted by the computer. The simplest and probably best way of doing this is to leave the decision to the performer.

¹¹The *diesis* is the interval that remains to the octave after an accumulation of three perfectly tuned ($\frac{5}{4}$) major thirds (e.g. *B-sharp* to *C*). This typically mean tone interval remainder is approximately a 31th of an octave. A.D. Fokker uses this term to indicate the smallest interval in 31-TET [Fokker and Pol, 1942]

3.3.1 User-defined

In Muditulib this is done by setting a modulation parameter (*mod*). The default ($mod = 0$) is - as a starting reference - set to two flats (*E* and *B*) and three sharps (*F*, *C*, and *G*), similar to the baroque standard. Every modulation up replaces one note in the circle of fifths by adding (1, -2) to its assigned Ts value, starting at MIDI note 3 (*E-flat* to *D-sharp*). In the opposite direction it starts at MIDI note 8 (*G-sharp* to *A-flat*). In the current implementation each modulation change is calculated from a default array at 'mod 0'.

3.3.2 Real-time pitch spelling

Another way to enrich the poor MIDI note data is the algorithmic approach. If, during a performance, a listener is able to roughly extract information about key, mode or tonality, the computer should be able too, if programmed according to a realistic cognition model. The translation from MIDI note data to staff notation or note names is called *pitch spelling* and is generally not a real-time practice. Some researchers have developed algorithms through the last decades [Longuet-Higgins and Steedman, 1971; Temperley, 2004; Cambouropoulos, 2003; Meredith, 2003; Chew and Chen, 2005; Honingh, 2006]. A very related topic is *key-finding*. In the end such procedures are all about saying something about the function of and relation between tonal events. For in some styles of music it is not always clear in which tonal direction the music will develop, when no true sense of tonality in that certain moment is present, a perfect real-time solution is theoretically impossible. The duty of the algorithm, however, is not offering perfect sheet music, but offering input for a real-time controlled dynamic tuning. Errors are acceptable, at least in cases where the human perception is uncertain. Any uncertain choice of the human tonal perception corresponds to the same uncertainty of the algorithm, ideally. An algorithm, developed by the present writer, based on memory, predicting, counting, averaging, and interval comparing, will be included in the library.

4 Tts , adding a third variable

Just intonation is a tuning approach in which all tone intervals are based on integer relationships. Pythagorean tuning can be considered 'just'. It is based on a perfect fifth (3 : 2) and a perfect octave (2 : 1) ratio. All intervals are then made up from powers of prime

numbers two and three¹². As the mean tone temperament showed, however, the perfect major third ratio is 5 : 4, but adding the number five to the tuning system introduces a problem. A major third cannot be divided into two equal whole tones within just intonation, for the mean tone is not an interval based on an integer relationship. Therefore, the major third is divided into a large and a small whole tone. This way, thirds, both major and minor, perfect fifths as well as octaves, and therefore all octave inversions of the mentioned intervals, can be tuned correctly¹³.

4.1 *Tts* to frequency

Tuning the *Tts* values is not a great deal, for there seems to be only one perfect solution, in which all octaves, perfect fifths, and thirds are tuned the most ideal way. Frequency ratios then should be $T = \frac{9}{8}$, $t = \frac{10}{9}$, and $s = \frac{16}{15}$. The translation to frequency can be best summarized by equation 4, where A is the reference a' at *Tts*(17, 12, 11)¹⁴.

$$f = \left(\frac{9}{8}\right)^{(T_n-17)} \cdot \left(\frac{10}{9}\right)^{(t_n-12)} \cdot \left(\frac{16}{15}\right)^{(s_n-11)} \cdot A \quad (4)$$

However, still some adjustments are conceivable. The 53-TET system for example, with $r = \frac{4}{9}$ and therefore approximately Pythagorean tuning, can be divided into three variable steps. Instead of a semitone of four *commas*¹⁵ and a whole tone of nine, the semitones are enlarged to five in favor of two of five whole tones. Note that the result of this would not deviate very much from the in equation 4 proposed tuning.

4.1.1 Turkish modes

An idea for further improvement would be to combine both two- and three-dimensional interpretations of 53-TET, or the three-limit Pythagorean and the five-limit just intonation instead. This would result in even more variables or steps, namely those used in Turkish modes (*makamlar*): *bakiye* (4 commas), *küçük mücennep* (5 commas), *büyük mücennep*

¹²This is called *three-limit*.

¹³According to renaissance counterpoint prescriptions all intervals considered consonant [Mann, 1987], whether perfect or imperfect, are now covered.

¹⁴That is, NOT the MIDI note number 69 is the reference.

¹⁵In contrast to 31-TET a part is called comma instead of diesis. This comma refers to the *syntonic* comma, which will be discussed later.

(8 commas), *tanini* (9 commas), and the augmented second *artuk ikili* (12 commas)¹⁶ [Signell, 1986 1977]. Muditulib could then be made very suitable for digitally synthesized reproduction of Turkish classical music or anything alike.

4.2 Creating useful *Tts* data

In contrast to *Ts* data, *Tts* data cannot be extracted from regular scores when it concerns diatonic music. *Tts* values shall then be obtained from *Ts* or even MIDI. This raises the problem of the *syntonic comma*, that is, the difference between the large and the small whole tone ($\frac{81}{80}$). This means that to fit the needs of a certain interval, another interval might be tuned too wide. In a dynamic tuning this comma can be replaced on-the-fly. In a more fixed tuning the comma will stick to its initial place and be rather present. For example, from this point of view Turkish modes are based on the placement of syntonic commas, giving each makam its very own character, based on some slightly wider and narrower intervals¹⁷. Only fifths and octaves are always tuned into perfection. One could possibly write a book about the placement of the syntonic comma. However, for this moment the present author prefers to skip such time-consuming research effort and focusses on two approaches, proposed in the next paragraphs. Again, there is a relatively simple approach and a more elaborate.

4.2.1 MIDI to *Tts*, user-defined

The simple approach is the user-defined key setting. The user defines the mode and the starting MIDI note. Currently two modes are available, one minor and one major, both displayed in table 2. Different modes can be created by moving the pattern to another reference MIDI note or, of course, by editing the source code or submitting a supported feature request¹⁸. For a piece

¹⁶The mentioned augmented second, that is, a perfect fourth (22 commas) minus two large semitones (5+5 commas), is clearly not unique here and can not be considered an extra variable. The only difference is that both Pythagorean (small) and five-limit (large) semitones appear. That makes a total of four different steps.

¹⁷An example of this is the *uşşak* makam, starting with the relatively small Pythagorean minor third (13 instead of 14 commas), although consisting of a small large whole tone (8) and a large semitone (5) from the tonic [Signell, 1986 1977].

¹⁸The pattern is best recognized by looking at the ‘difference’ column. The non-steps (places where no semitone or step occurs, e.g. ‘T/s’) are grouped just like the black keys on a keyboard.

in E minor one would usually choose ‘MIDI note 4’ as reference and ‘0’ (minor) as mode. These default patterns are carefully chosen to enable the best standard modulations from the reference key, without resulting in too many ‘misplaced’ commas¹⁹. How these patterns were actually chosen is not discussed here, for the sake of not going into detail of music theoretical considerations too much. Furthermore, more research on this topic would be desirable, for example comparing these considerations to those of how frets on a saz are placed.

Minor (0)			Major (1)		
Dif.	Total	C.	Dif.	Total	C.
s	-	0	s	-	0
s	s	5	T/s	T/s	4
T/s	T	9	s	T	9
s	Ts	14	s	Ts	14
t/s	Tt	17	t/s	Tt	17
s	Tts	22	s	Tts	22
T/s	TTt	26	T/s	TTt	26
s	TTts	31	s	TTts	31
s	TTtss	36	s	TTtss	36
T/s	TTTts	40	t/s	TTtts	39
s	TTTtss	45	s	TTTtss	44
t/s	TTTtts	48	T/s	TTTtts	48

Table 2: Tuning patterns for the MIDI keyboard: modes and modulation options from a reference tonic. Shown are the differences to the previous MIDI note and the total amount of distance to the reference in symbols and in commas.

4.2.2 A pattern matching approach, or the *hexachord analysis algorithm*

Another way of tuning is leaving this task to, again, a real-time controlling algorithm. For singers in the Middle Ages used Guido of Arezzo’s hexachord to choose their pitches [Grout and Palisca, 1988], the hexachord seems very suitable for on-the-fly tuning purposes. The next question is how the hexachord should then be tuned. Hermann von Helmholtz has been very helpful to answer this question for he explains how medieval singers related each note of the hexachord to a reference and what differences exist between major (on *Ut*) and minor (on *Re*) modes [Helmholtz, 1896, ch. 18]. The

¹⁹E.g. the fourth of 23 commas on the subdominant in minor mode (from *Tts* to *TTTtss*).

resulting conclusions are displayed in table 3. The *Re* and *Sol* are placed one comma lower in minor mode. However, for modulation purposes the major hexachord is placed one comma lower than the minor altogether. The translation from *Ts* to *Tts* is done by pattern matching²⁰. The choice between major and minor tuning of each individual hexachord is done by a tonic-finding algorithm.

	Ut	Re	Mi	Fa	Sol	La
Minor	0	8	17	22	30	39
	-	t	T	s	t	T
Major	-1	8	16	21	30	38
	-	T	t	s	T	t

Table 3: The tuning of the hexachord, displayed in commas.

5 Implementation

All the previously mentioned functionality is bundled into a C header file, called `muditulib.h`²¹. The functions will be explained in a reference manual at <http://muditulib.eu>.

In essence this library is relatively small and simple. The challenging part is probably the implementation, depending on the environment. An implementation for Pure Data is ready yet and consists of the header file, the C files written against the Pd-API to create a collection of separate Pd classes, a *Makefile* based on the template by H.-C. Steiner, along with help-files and examples. An example of the Pd-implementation is shown in figure 1.

6 Concluding remarks

The tuning approaches described here highly depend on implementation possibilities. For a low-level music production environment like Pure Data there is actually no problem, although this requires quite some background knowledge from the user, both of music and tuning theory. Most popular electronic music production platforms, however, are mainly based on the MIDI note system. Tuning workarounds making use of ‘pitch bend’ are familiar to the present author, though not a satisfying solution. Plans are to develop a file format other than MIDI. More about such can be expected

²⁰The pattern of the diatonic hexachord is T-T-s-T-T.

²¹<http://sourceforge.net/projects/muditulib/>

in the near future. Any suggestions about file formats or implementations and especially questions arising from implementation ambitions, are very welcome.

7 Acknowledgments

My thanks go out to Albert Gräf, Gerard van Wolferen, Pieter Suurmond, and Hans Timmermans for supporting my research within this field, to Horus and its initiator Marc Groenewegen for contributing to the Linux Audio spirit around Hilversum/Utrecht, to Miller Puckette for sharing a great piece of software called Pure Data, to IOhannes Zmölzig for offering a tutorial about extending the Pure Data core, to Hans-Christoph Steiner for the Makefile template as well as to all the cited researchers for their effort.

References

- J. Murray Barbour. 2004. *Tuning and Temperament*. Dover Publications, Inc., dover edition.
- Emilios Cambouropoulos. 2003. Pitch spelling: A computational model. *Music Perception*, 20(4).
- Elaine Chew and Yun-Ching Chen. 2005. Real-time pitch spelling using the spiral array. *Computer Music Journal*, 29(2).
- A. D. Fokker and Balth. van der Pol. 1942. *Harmonische Muziek*. Martinus Nijhoff, 's-Gravenhage.
- Donald J. Grout and Claude V. Palisca. 1988. *A History of Western Music*. W. W. Norton & Company, Inc., fourth edition.
- Hermann von Helmholtz. 1896. *Die Lehre Von Den Tonempfindungen Als Physiologische Grundlage Für Die Theorie Der Musik*. Druck und Verlag von Friedrich Vieweg und Sohn, Braunschweig, fifth edition. Replica.
- A. K. Honingh. 2006. *The Origin and Well-Formedness of Tonal Pitch Structures*. Ph.D. thesis, Universiteit van Amsterdam. <http://staff.science.uva.nl/~ahoningh/publicaties/proefschrift.pdf>.
- H. C. Longuet-Higgins and M. J. Steedman. 1971. On interpreting bach. <http://aitopics.org/sites/default/files/classic/Machine%20Intelligence%206/MI6-Ch15-LonguetHigginsSteedman.pdf>.
- Alfred Mann. 1987. *The Study of Fugue*. Dover Publications, Inc., New York, dover edition.
- David Meredith. 2003. Pitch spelling algorithms. In *Proceedings of the 5th Triennial ESCOM Conference*.
- Funs Seelen. 2014. A parametric dynamic tuning system for the diatonic scale / towards five-limit just intonation based on a hexachord analysis algorithm. *Berichte der Reihe Musikinformatik und Medientechnik*, (52).
- Karl L. Signell. 1986, 1977. *Makam - Modal Practice in Turkish Art Music*. Da Capo Press, New York.
- David Temperley. 2004. *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, Massachusetts, first paperback edition.