Eikosany: Microtonal Algorithmic Composition with R

2023-08-19

1 Algorithmic Composition

Overview of Methods (Nierhaus 2009)

- Markov Models / Stochastic
- Generative Grammars
- Transition Networks
- Chaos and Self-Similarity
- Genetic Algorithms
- Cellular Automata
- Artificial Neural Networks

My Main Compositional Focus

- Markov Models / Stochastic
- Pioneered by Iannis Xenakis (**Xenakis 1992**)
- Random chord progressions on microtonal harmonic structures

When Harry Met Iannis (2021) (Borasky 2021)

- Microtonal harmonic structure is Harry Partch's Tonality Diamond (**Partch 1979**)
- Tonality Diamond was an inspiration for Erv Wilson's Combination Product Sets

2 Musical Scales

Types of scales

- Standard “western” tuning - 12 equally-spaced tones / octave
  - abbreviated 12-TET (12 tone equal temperament) or 12-EDO (12 equal divisions of the octave)
- Alternative tuning - anything else

Alternative tunings

- scales from other cultures
- “just” scales - scales based on rational numbers
- scale repetition periods different from the octave
- scale repetition period divided into more than 12 tones
- combinations of the above!

Microtonal music

- Usually defined as an octave divided into more than 12 tones
- Common microtonal scales
    - 19-TET
    - 24-TET aka quarter tones
    - 31-TET

3 Erv Wilson (Narushima 2019)

Ervin Wilson (June 11, 1928 – December 8, 2016)

- Mexican/American (dual citizen)
- Prolific music theorist
- Developed keyboard layouts, scales and lattices
- Primarily known for theories of microtonal just scales

Combination Product Set Scales (Narushima 2019, chap. 6)

- Focus of the `eikosany` package
- Start with a set of N *harmonic factors*
- For even number of factors N, `choose` = N/2
- For an odd number, `choose` = either N/2 - 1/2 or N/2 + 1/2
    - Four factors: `choose` 2: six combinations
    - Five factors: `choose` 2 or 3: ten combinations
    - Six factors: `choose` 3: 20 combinations
    - Pascal’s Triangle is your friend

Making the Scale

- Take all the combinations of the factors with `choose` elements
- Take the products of the combinations
- Reduce the products to ratios in [1, 2)
    1. Divide all by smallest product
    2. Multiply or divide by powers of two to get values in [1, 2)
- Sort

Example: 1-3-5-7 Hexany

- Products of combinations: `1*3, 1*5, 1*7, 3*5, 3*7, 5*7`
- Raw ratios (divide by smallest product): `1 5/3 7/3 5 7 35/3`
- Reduce: `1 5/3 7/6 5/4 7/4 35/24`
- Sort: `1 7/6 5/4 35/24 5/3 7/4`

Using the package!

```r
remotes::install_github("AlgoCompSynth/eikosany", quiet = TRUE)
library(eikosany)
hexany_scale_table <- cps_scale_table(
  harmonics = c(1, 3, 5, 7),
  choose = 2
)
print(hexany_scale_table)
```

```
   note_name    ratio ratio_frac ratio_cents interval_cents degree
1:       1x3 1.000000          1      0.0000             NA      0
2:       1x7 1.166667        7/6    266.8709      266.87091      1
3:       3x5 1.250000        5/4    386.3137      119.44281      2
4:       5x7 1.458333      35/24    653.1846      266.87091      3
5:       1x5 1.666667        5/3    884.3587      231.17409      4
6:       3x7 1.750000        7/4    968.8259       84.46719      5
7:      1x3' 2.000000          2   1200.0000      231.17409      6
```

- The prime on the bottom `note_name` indicates the next octave

Cents??

- A logarithmic measure used by scale theorists
- 1 cent = 1/100th of a semitone
- 12 semitones = 1200 cents = 1 octave = ratio of 2/1

The Chord Table

- Like scales, chords are combinations of harmonic factors
- Currently only works for even number of factors
- For a chord table, we take choose + 1 combinations
    - Four factors: three combinations (four triads)
    - Six factors: four combinations (15 tetrads)

Harmonic and Subharmonic Chords

- Each chord has a harmonic and subharmonic form
- Roughly corresponds to major and minor chords
- Four factors: four harmonic triads and four subharmonic triads
- Six factors: 15 harmonic tetrads and 15 subharmonic tetrads
- In the chord table, the subharmonic ones have “/” in their names

The Hexany Chord Table

```r
hexany_chord_table <- cps_chord_table(hexany_scale_table)
print(hexany_chord_table)
```

```
      chord degrees chord_index is_subharm
1:    1:3:5   1:3:5           1          0
2: /1:/3:/5   0:2:4           1          1
3:    1:3:7   2:3:4           2          0
4: /1:/3:/7   0:1:5           2          1
5:    1:5:7   0:2:5           3          0
6: /1:/5:/7   1:3:4           3          1
7:    3:5:7   0:1:4           4          0
8: /3:/5:/7   2:3:5           4          1
```

4 Pseudo-Demo

The 1-3-5-7-9-11 Eikosany

- Six harmonic factors, choose 3

```r
eikosany_scale_table <- cps_scale_table(
  harmonics = c(1, 3, 5, 7, 9, 11),
  choose = 3
)
```

The Eikosany Scale Table

```r
print(eikosany_scale_table)
```

```
    note_name    ratio ratio_frac ratio_cents interval_cents degree
 1:     1x3x5 1.000000          1     0.00000             NA      0
 2:    5x9x11 1.031250      33/32    53.27294       53.27294      1
 3:     1x7x9 1.050000      21/20    84.46719       31.19425      2
 4:    1x3x11 1.100000      11/10   165.00423       80.53704      3
 5:     3x5x9 1.125000        9/8   203.91000       38.90577      4
 6:     1x5x7 1.166667        7/6   266.87091       62.96090      5
 7:    3x9x11 1.237500      99/80   368.91423      102.04332      6
 8:    1x7x11 1.283333      77/60   431.87513       62.96090      7
 9:     5x7x9 1.312500      21/16   470.78091       38.90577      8
10:    3x5x11 1.375000       11/8   551.31794       80.53704      9
11:     1x3x7 1.400000        7/5   582.51219       31.19425     10
12:    7x9x11 1.443750    231/160   635.78514       53.27294     11
13:     1x5x9 1.500000        3/2   701.95500       66.16987     12
14:     3x7x9 1.575000      63/40   786.42219       84.46719     13
15:    5x7x11 1.604167      77/48   818.18885       31.76665     14
16:    1x9x11 1.650000      33/20   866.95923       48.77038     15
17:     3x5x7 1.750000        7/4   968.82591      101.86668     16
18:     1x3x9 1.800000        9/5  1017.59629       48.77038     17
19:    1x5x11 1.833333       11/6  1049.36294       31.76665     18
20:    3x7x11 1.925000      77/40  1133.83013       84.46719     19
21:    1x3x5' 2.000000          2  1200.00000       66.16987     20
    note_name    ratio ratio_frac ratio_cents interval_cents degree
```

The Eikosany Chord Table

```r
print(eikosany_chord_table <-
  cps_chord_table(eikosany_scale_table)
)
```

```
           chord     degrees chord_index is_subharm
 1:      1:3:5:7   1:6:11:15           1          0
 2:  /1:/3:/5:/7   0:5:10:16           1          1
 3:      1:3:5:9  7:11:14:19           2          0
 4:  /1:/3:/5:/9   0:4:12:17           2          1
 5:     1:3:5:11   2:8:11:13           3          0
 6: /1:/3:/5:/11    0:3:9:18           3          1
 7:      1:3:7:9   1:9:14:18           4          0
 8:  /1:/3:/7:/9  2:10:13:17           4          1
 9:     1:3:7:11    1:4:8:12           5          0
10: /1:/3:/7:/11   3:7:10:19           5          1
11:     1:3:9:11   5:8:14:16           6          0
12: /1:/3:/9:/11   3:6:15:17           6          1
13:      1:5:7:9    3:6:9:19           7          0
14:  /1:/5:/7:/9    2:5:8:12           7          1
15:     1:5:7:11   4:6:13:17           8          0
16: /1:/5:/7:/11   5:7:14:18           8          1
17:     1:5:9:11 10:13:16:19           9          0
18: /1:/5:/9:/11  1:12:15:18           9          1
19:     1:7:9:11    0:4:9:16          10          0
20: /1:/7:/9:/11   2:7:11:15          10          1
21:      3:5:7:9   3:7:15:18          11          0
22:  /3:/5:/7:/9   4:8:13:16          11          1
```

What Does It Sound Like?

- To hear the scale:
  1. Get a synthesizer
  2. Map the keys of the synthesizer to the notes of the scale
  3. Play the keys
- But that’s kind of expensive and I’m in a hurry
- So I’ll emulate it in R!

First Step - Make a Keyboard Map

- Maps MIDI note numbers to frequencies
- MIDI note numbers range from 0 to 127
- MIDI note number 60 is middle C
- Our 20-note scale is note numbers 60:79

```r
# this is the whole keyboard
eikosany_map <- keyboard_map(eikosany_scale_table)

# extract scale from middle C
scale_map <- eikosany_map[
  note_number %in% 60:80,
  list(note_number, freq)
]
```

The Scale Map

```
print(scale_map)
```

```
    note_number     freq
 1:          60 261.6256
 2:          61 269.8014
 3:          62 274.7068
 4:          63 287.7881
 5:          64 294.3288
 6:          65 305.2298
 7:          66 323.7616
 8:          67 335.7528
 9:          68 343.3836
10:          69 359.7352
11:          70 366.2758
12:          71 377.7219
13:          72 392.4383
14:          73 412.0603
15:          74 419.6910
16:          75 431.6822
17:          76 457.8447
18:          77 470.9260
19:          78 479.6469
20:          79 503.6292
21:          80 523.2511
    note_number     freq
```

Second Step - Make Waves!

- uses `seewave` and `tuneR` (**Sueur 2018**)
- creates a *multisample* - a collection of `WAV` files
  - can be used in sample-based workflows
  - emulates multisamples captured by **1010music Blackbox**

Make Waves!

```r
scale_multisample(
  keyboard_map = scale_map,
  start_note_number = 60,
  end_note_number = 80,
  duration_sec = 2,
  output_directory = "./Eikosany-Scale-Multisample"
)
```

```
[1] "./Eikosany-Scale-Multisample"
```

(Optional) Play Files with a Media Player

- 40 seconds you’ll never get back

(Optional) Play Some Eikosany Chords

```r
eikosany_scale_table <- cps_scale_table()
render_cps_chords(eikosany_scale_table, "./eikosany_chords")
```

```
[1] "./eikosany_chords"
```

5 Roadmap

Current status

- Enough infrastructure to manually make music!
  - (which was my initial goal)
- Documentation - you’re looking at it

Current plan

1. Clean up documentation and do a first release (August)
    - Hoping to release an album / EP on Bandcamp
2. Open feature requests for triage
3. Add consonance analysis capability (**Sethares 2013**)
4. Remove the MIDI functionality!
    - MIDI is a great language for 4/4 time 12-EDO music
    - For microtonal algorithmic composition, not so much
    - The "other tools" in the Appendix are designed to deal with MIDI so I don't have to!

Where does all this stuff live?

- GitHub: **https://github.com/AlgoCompSynth/eikosany**
- Pkgdown: **https://algocompsynth.github.io/eikosany**
- These slides: **https://github.com/AlgoCompSynth/eikosany-slides**

Appendix - Microtonal Music Resources

PC / Mac / iOS Software

- **Scala**. Note: this is ***not*** the Scala multi-paradigm programming language that runs on the Java Virtual Machine. This is a specialized tool for working with musical scales.
- **ODDSound MTS-ESP**. This is a plugin for digital audio workstations (DAWs) that facilitates production of microtonal music. I own a copy and if you’re making microtonal electronic music, you should too. The Eikosany and other scales Erv Wilson developed all ship with MTS-ESP, so you don’t really need my R package to compose with them.
- **Entonal Studio**. Entonal Studio is a user interface package for microtonal composition. It can operate as a standalone application, a plugin host or a plugin. I own a copy of Entonal Studio and recommend it highly.

PC / Mac / iOS Software (continued)

- **Infinitone DMT**. From the **Infinitone DMT FAQ**:

  "Infinitone DMT is a DAW plugin and standalone that empowers musicians to easily use micro-tuning within their own workflow.

  …
  "As a plugin, Infinitone DMT is inserted in your DAW as a MIDI effect.

  …
  "The standalone can be used separately from a DAW, or it can be used in conjunction with a DAW by routing MIDI data from the DAW to the standalone (and back)."

PC / Mac / iOS Software (continued)

- **Universal Tuning Editor**. Universal Tuning Editor is an application for computing and visualizing microtonal scales and tunings, and includes tools to interface with hardware and software synthesizers.
- **Wilsonic**. This is a free app that runs on iOS devices. I don’t have any iOS devices so I’ve never used this.
  There is also a version of Wilsonic in development for use with ODDSound MTS-ESP. See **https://wilsonic.co/downloads/downloads-mts-esp/** for the details.
- **Surge XT**. Surge XT is an open source full featured software synthesizer. The Surge XT community has invested a significant level of effort into supporting alternate tuning systems.

  See the **Xenharmonic Wiki List of microtonal software plugins** for more ways of making microtonal music.

Websites

- **Kraig Grady’s Anaphoria Wilson Archive** Australian-American composer Kraig Grady studied with Erv Wilson for many years and has collected Wilson’s writings.
- **Sevish’s Scale Workshop**. This is a web-based tool for working with musical scales.
- **Leimma and Apotome**. These tools, by **Khyam Allami** and **Counterpoint**, are browser-based applications for creating microtonal scales and making generative music with them.

YouTube Playlists

- Surfing the Sonic Sky
- Combination Product Sets - Music
- Combination Product Sets - Theory

References

https://algocompsynth.bandcamp.com/album/when-harry-met-iannis

https://books.google.com/books?id=jaowAtnXsDQC