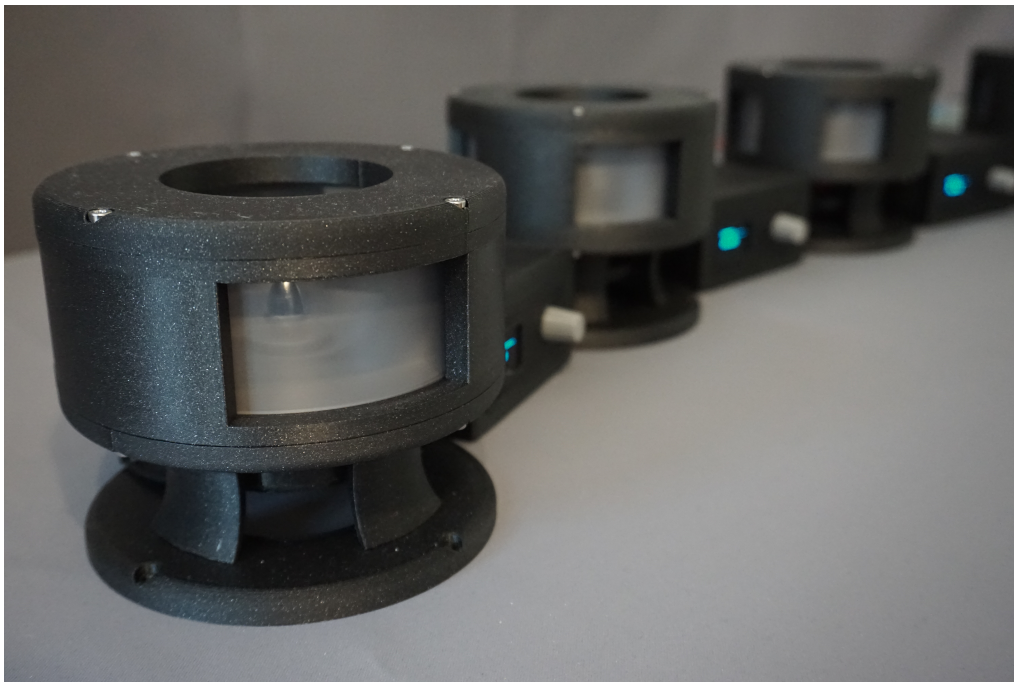


for sirens perhaps, and/or ...
(compact sets 4 - alarming)



michael winter (2024-2026)

Description (which can optionally be used/adapted as a program note)

This piece was written for a set of custom-designed, tunable air-raid sirens. It is one of a series of pieces, including *seeds and ledgers* (2023) and *compact sets 1–3* (2024), generated with an algorithm that maintains consonant groups of tones among any simultaneously sounding tones, but favors smaller steps in pitch space when one voice moves melodically.

By the time of completing this piece, I have formalized this process and described it extensively in a theoretical context.¹ The original idea of the algorithm was conceived of in order to try to minimize the glissandi necessary for a siren to move from one pitch to another. The solution to this minimization coincided with a reexamination of the traditional concept of ‘voice leading’: how individual melodic lines create and maintain harmonies in aggregate while sometimes modulating. In all the pieces that use the aforementioned algorithm, which I describe in more detail below, this musical question of voice leading is recontextualized in the phenomenological framework of *just-intonation in harmonic space*. In just-intonation, whole number ratios express the frequency relationship between pitches. Harmonic space is defined as a lattice where two tones are connected if they are related by one of a set of simple intervals. The resulting musical scales are untempered. They do not favor the accuracy of one interval over another such as with different well- and equal-temperaments, which prioritize and sacrifice the accuracy of different intervals for key cyclicness. While the algorithm was developed for an idiosyncrasy of sirens, the reason that this piece is preceded by the aforementioned pieces is that it took me approximately two years to design and develop a set of tunable sirens. Of course, had I known then what I know now, it would have taken a small fraction of the time.

Formally, a compact set in the context of these pieces is defined as a group of pitches such that each pitch is connected in harmonic space to some other pitch in the group, but not necessarily all of them. When one voice moves melodically, the algorithm will favor notes that move by a smaller step in pitch space within a voice while modulating/transitioning to another compact set among all the voices. Any individual part would be near impossible to play by itself. However, because compact sets are always maintained, each successive tone within a part can always be tuned via a relatively simple interval in harmonic space to a tone that is already sounding in one of the other instruments.

¹https://unboundedpress.org/pubs/Tom_Johnson_and_a_rational_theory_of_harmony.pdf

Performance Instructions

The piece should last at least 15 minutes long.

If played with sirens, the piece should start with long glissandi from as low and soft a pitch as possible into the first chord (lasting at least 2.5 minutes) and end by slowly walking each siren downward while maintaining simple intervals amongst all the sounding pitches until the sirens are very soft/barely audible. If not played with sirens, the glissando and ending walk downwards are optional.

The entire piece can be transposed by any amount and any individual pitch can be transposed by any number of octaves. The notated range was selected for the custom-designed sirens to be loud, but not too loud, at their highest/loudest point. To make the piece louder, the piece can be transposed up. To make the piece softer, the piece can be transposed down.

While the piece has four parts generated by an algorithm that satisfy the constraints described above, performers can make (primarily chromatic) melodies from notes among different parts. The melodies can also draw from notes that span several measures before and after the currently sounding chord. In this sense, each measure should be considered a base chord and the surrounding measures create a field from which one can build melodies. Once one builds a melody, it can be repeated several times.

Even if the piece is played without sirens and pitches can be transposed by octaves, the piece should have a general progression from high to low and loud to soft as it would be idiomatically with sirens.

Tuning and Notation

This piece explores an extended just-intonation with no fixed fundamental. Each written note indicates the closest pitch in twelve-tone equal temperament with a cents-deviation (100th of a tempered semitone) provided above. As explained in the description above, the note-to-note melodic movement may be very complex. However, each note can always be tuned via a relatively simple interval from a note that is currently sounding. For the purposes of the following explanation, the current note will be referred to as the *referencing note* and the pitch against which it can be tuned will be called the *reference pitch*.

A Roman numeral below a note indicates the part in which the reference pitch is. The Arabic superscript and the corresponding arrow indicate the exact interval between the referencing note and the reference pitch as follows. A Roman Numeral with a superscript of 3, 5, 7 followed by an up arrow indicates that the relationship between the referencing note and the reference pitch is a frequency ratio of a $3/2$, $5/4$, $7/4$, respectively, from the lower pitch to the higher one. A Roman Numeral with a superscript of 3, 5, 7 followed by a down arrow indicates that the pitch of the referencing note is a frequency ratio of a $4/3$, $8/5$, $8/7$, respectively, from the lower pitch to the higher one. That is, the down arrow is the inversion of an up arrow.

Performers may also choose to play pitches that are not notated but closely related to any of the written pitches, extending each chord/compact set to include intervals of $3/2$, $5/4$, $7/4$, $11/8$, $13/8$ (and their inverses, $4/3$, $8/5$, $8/7$, $16/13$, $16/11$) and thus also potentially expanding the harmonic space.

Siren Design and Codebase

The algorithm has undergone several iterations. For this piece, I completely redeveloped the code base using Python as a backend for a JavaScript frontend to generate the base sequence, transcribe the music, and play the sirens (via OSC messages that can also be used to synthesize the tones). The sirens themselves have also gone through several iterations of development. As I intend to continue this development, I have created a release in the code base pinned to this piece for posterity's sake. The code and siren design (including motor control) can be found in the following two git repositories respectively:

https://unboundedpress.org/code/mwinter/compact_sets

<https://unboundedpress.org/code/mwinter/sirens>

for sirens perhaps, and/or ...
(compact sets 4 - alarming)

michael winter (2024--2026)

17

System I: +17, -2, -47, -29, -12, +43, +24
II 7↓ IV 3↓ III 3↓ IV 3↑ IV 5↓ III 7↑ IV 7↑

System II: -14, -33, -41, -23, -43, +42
IV 5↑ IV 3↓ III 5↑ IV 3↑ I 7↑ IV 5↑

System III: -45, +36, -19, -27, -29, -10, -25, +10
II 7↑ II 7↑ II 5↓ I 3↑ II 5↓ I 3↑ I 5↑ II 7↑

System IV: 0, -31, -25, -14, -45
III 5↓ III 3↑ I 3↓ II 3↓

33

System I: +28, +8, +32, -19, -38, -21
II 5↑ III 5↓ IV 7↑ III 3↑ IV 7↑ IV 3↑

System II: -3, -19, -7, -23, -25
III 5↑ IV 3↓ III 3↓ IV 5↓ I 5↓

System III: -5, +40, -5, -21, -9, -40, +9
II 3↓ I 7↓ I 5↑ II 3↑ IV 3↓ I 3↓ IV 7↓

System IV: +24, +12, -17, -36, -7, -23
III 5↓ III 3↑ II 5↑ III 7↑ III 5↓ II 3↑

49

System I: -9, +26, +44, -23, -39
IV 5↓ II 5↑ III 3↑ IV 3↑ III 5↓

System II: +40, +28, +12, +46, -39, +22, +46
III 7↓ I 3↑ IV 3↑ I 3↑ IV 5↑ III 7↓ I 7↑

System III: +42, -41, +6, -10, +10, -6, +47
II 3↑ II 5↓ IV 7↓ I 5↓ IV 3↑ IV 5↑ II 3↑

System IV: +10, +30, -41, -25, +8, +44, +32
III 3↑ I 5↑ II 5↓ I 7↓ I 7↓ II 3↓ II 5↑

65

System I: -27, -31, +40, +20, +36
III 5↓ III 3↓ II 5↑ IV 7↑ III 3↓

System II: -43, +46, -43, +38, -43, -47, +24, +49, -31
III 3↓ IV 3↑ III 3↓ I 7↑ IV 3↑ IV 3↓ III 5↑ I 5↓ III 7↓

System III: -41, -29, -33, +38
I 3↓ I 3↓ I 3↓ IV 5↑

System IV: +44, +4, +44, -45, -49, +34, -49
II 3↓ I 7↓ II 5↑ II 3↓ II 3↓ I 3↓ III 5↓