The Well–Sequenced Synthesizer: a Series of Generative Sequencers

Luisa Pereira Hors
Interactive Telecommunications Program, New York University
721 Broadway, 4th floor
New York, NY 10003
luisa.ph@nyu.edu

ABSTRACT

The Well–Sequenced Synthesizer is a series of sequencers that create music in dialog with the user. Through the sequencers' physical interfaces, users can control music theory-based generative algorithms. This series –a work-in-progress– is composed by three sequencers at this time. The first one, called The Counterpointer, takes a melody input from the user and responds by generating voices based on the rules of eighteenth–century counterpoint. The second one is based on a recent treatise on harmony and counterpoint by music theorist Dmitri Tymoczco: El Ordenador lets users explore a set of features of tonality by constraining randomly generated music according to one or more of them. El Ordenador gives the user less control than The Counterpointer, but more than La Mecánica, the third sequencer in the series. La Mecánica plays back the sequences generated by El Ordenador using a punch-card reading music box mechanism. It makes the digital patterns visible and tactile, and links them back to the physical world.

Keywords

NIME, Generative music, Interaction Design, Physical Computing

1. INTRODUCTION

This project was inspired by J.S. Bach's The Well Tempered Clavier (1722), and Wendy Carlos' The Well Tempered Synthesizer (1969). Both works explore music through the lens of the new technologies of their time: Bach's preludes and fugues explore the possibilities of counterpoint and the well–tempered tuning system; Wendy Carlos' albums, under their irreverent covers, explore the programming and the timbric possibilities of analog synthesizers.

The Well-Sequenced Synthesizer is a series of sequencers. Taking advantage of current technology –digital representations of music, open source algorithms and programming frameworks, and widely available micro-controllers and digital fabrication tools– these sequencers create music in dialog with the user, who is given varying degrees of control over the system.

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

The sequencer's interfaces are physical: our intention is to bring the experience of interacting with music theory rules nearer that of playing an instrument than to that of reading a species counterpoint textbook. The language on the sequencers' interfaces evokes Bach´s time, their visual design, components and materials evoke Carlos´, their functionality and production process is only possible in ours.

2.THE COUNTERPOINTER

The Counterpointer is a step sequencer that creates second and third voices based on a melody entered by the user, and the rules of first species counterpoint.

A set of sliders, each ranging an octave, define an eight-note sequence. All notes are equal in duration. As the sequencer steps through them in a loop, the user sets each slider to a note, creating her melody: the Cantus Firmus (CF). Above the sliders lay two buttons labeled "compose" (Figure 1). When the user presses one of them (for example, the one labeled "second voice"), the system generates a higher voice based on counterpoint rules. Similarly, the "compose" button on the right triggers the generation of a voice lower than the user's CF. These melodies are represented in terms of degrees in a scale, not specific notes: the user can change the scale type (minor, major) and its tonic in real time by turning the potentiometers to the left of the sliders. Tempo can also be adjusted as the melodies play, and each voice can be switched on or off.

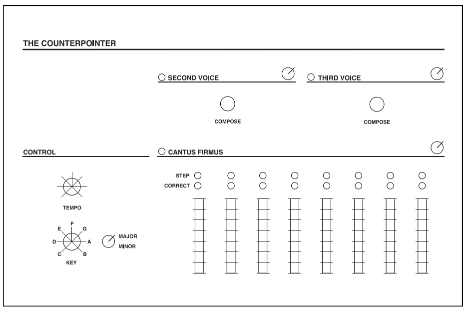

Figure 1. The Counterpointer's interface.

The algorithm that generates the second and third voices constructs new melodies by trying to add successive intervals and evaluating the results against a set of rules. These are:

1. The notes in the melody must be above those in the base.
2. The first interval must be an unison, an octave, or a perfect 5th over the CF.
3. The last interval must be an octave.

4. Vertical intervals must be consonant.
5. The range should not be larger than 16 steps.
6. There should be no tritone leaps.
7. There should be no ties (repeating a note).
8. There should be no hidden perfect intervals.
9. There should be no parallel 5ths or octaves.
10. There should bee at most three of the same interval in a row.

First species counterpoint also requires the Cantus Firmus to be valid:

1. It must begin and end on the tonic.
2. Its range must be larger or equal to seven steps.
3. From one note to the next there should be no tritones, and no augmented seconds.
4. There should be no more than two consecutive jumps.
5. Jumps should be smaller than ten steps.
6. The direction of the melody should change after a jump greater than a third.
7. There should not be two consecutive jumps in the same direction.
8. The melody must end by a step.

During early user testing it became clear that there was a tension between the strict application of rules and a fluidity of interaction. This posed a couple of design issues:

1. Enforcing Cantus Firmus correctness means that a user might spend a long time with the device before hearing any counterpoint at all.
2. Maintaining the counterpoint correct at all times limits the users' ability to experiment: if each time the user changes their melody the second and third voice adapt, she will lose the grounding of a repetitive melody to play against, and to try variations over.

We decided to prioritize expressiveness over the correctness of the counterpoint, and to encourage the breaking of rules, while making these departures explicit:

1. Cantus Firmus checking was disabled, meaning the generated sequences will often be an approximation to first species counterpoint, but not strictly correct. The next version of The Counterpointer will have a switch to turn CF checking on or off.
2. The voices generated by the system only change if the user explicitly requires it by pressing the "compose" button. This means that incorrect counterpoint is likely to happen as the user tweaks their melody once system voices have been composed. A strip of red LEDs placed over the sliders will indicate if there is a problem, and where –but the music will go on.

3. EL ORDENADOR

El Ordenador is based on the work of music theorist Dmitri Tymoczko, who has proposed an extended definition of tonality that includes a wide range of genres –his examples go from Bach to Miles Davis to Radiohead. In A Geometry of Music [1], a treatise on harmony and counterpoint, he presents a set of features that contribute to our experience of tonality. These are:

1. Conjunct melodic motion.
2. Acoustic consonance.
3. Harmonic consistency.
4. Limited macroharmony.
5. Centricity.

To prove that they do, he proposes an informal experiment: to constrain randomly generated notes according to these features, and listen if the music sounds increasingly tonal. El Ordenador implements the constraints in his experiment, and gives them on/off switches, allowing users to listen to the musical qualities yielded by different combinations.

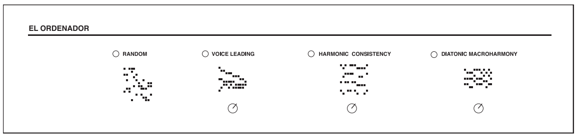

Figure 1. El Ordenador's interface.

With all switches set to off, El Ordenador plays a completely random chord sequence. With the "voice leading" switch on, the randomness is constrained: each note must move only by a scale step from chord to chord. The "harmonic consistency" switch forces chords to be stacks of a given interval (thirds, fourths, fifths, etc.). Finally, the "diatonic macroharmony" switch forces notes to be taken from the same diatonic scale. With all switches off, the sequence sounds chaotic; with all switches on, it sounds, if not particularly interesting, clearly harmonic and organized.

4. LA MECÁNICA

During the development of El Ordenador, we implemented a rudimentary notation for its chord progressions –white squares representing notes, scale position on the Y axis, time on the X axis. It was a utility, meant for verifying the algorithm, but it caught our attention. We noticed, for instance, that as we started to hear melodic lines when turning on the voice leading switch, we also started to see lines on the notation. Tymoczko's features seemed more immediately recognizable in these simple graphs than they had been in traditional notation.

It was compelling to watch the square patterns on the screen as we switched constraints on and off, but it also distracted us from the aural and tactile experience, drawing us into the screen and the interaction modes of the computer. We decided that the third sequencer in this series should emphasize these visual patterns while keeping the experience tactile.

La Mecánica is a mechanical sequencer: it uses a traditional music box mechanism to play back punch card scores generated by El Ordenador. The bamboo and acrylic box serves as a resonance box. Users can choose which chord progression to play, and have control over one variable: tempo. Their control is limited, but feels surprisingly expressive. One of the constraints from El Ordenador is determined by the device's mechanism: the music box is on a diatonic scale, so 'diatonic macroharmony' is always enforced. The fact that the interpretation of the scores isn't exact reminds us there is loss in the translation between the purely digital and the physical world.

5.ACKNOWLEDGEMENTS

The first species generating algorithm used by The Counterpointer was taken from the open source CFGen application, developed by Michael Rotondo.

The software for The Counterpointer, El Ordenador and for generating the music box punch cards were developed using Open Source framework processing.org; The Counterpointer and El Ordenador use the Open Source platform Arduino.

6.REFERENCES

1. D. Tymoczko. A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice. *Oxford University Press, New York, NY, 2011*.