

A Microtonal Tempo Canon Generator After Nancarrow and Jaffe

Nick Collins
sicklincoln.org
e-mail: nick@sicklincoln.org

Abstract

A composition system for tempo canons is described which implements mensural and acceleration canon as explored by Conlon Nancarrow, and sinusoidal oscillation canon as introduced by David Jaffe. The work has the capacity for microtonal melodies in up to eight voices, with substitutable modules for the synthesis and the algorithmic composition of the canon fundamental line. With a user interface to assist exploration, the system can be used as a research tool, or as a generative piece in its own right.

Key Words: tempo canon, time map, Nancarrow, Jaffe, generative music

1 Introduction

Whilst mensural canon has its origins in the musical games of fifteenth century Flemish composers like Johannes Ockeghem, the modern exploration of the form is associated with the life's work of Conlon Nancarrow (Gann 1995). Nancarrow introduced innovations like the acceleration canon, in which each voice linearly or exponentially accelerates over the course of playing the fundamental melody. He explored the important compositional factor of the convergence point, at which voices offset by delays and tracking different tempi reach the same point in the melody at a given instance in time.

Work over the last twenty five years in the representation of rhythm for computer music (Jaffe 1985, Desain and Honing 1992, Honing 2001) has introduced such concepts as time maps, which provide a natural model for convergence points. David Jaffe, in his celebrated Silicon Valley Breakdown (Jaffe 1982) used time map methods to set up sinusoidal tempo oscillation canons with convergence, where the tempo of each voice of the canon follows a sinusoidal variation.

In an effort to provide a platform for exploration of tempo canon and convergence, a tempo canon generator has been built in the SuperCollider 2 language (McCartney 1998). The spirit of this research is on the one hand as an analytical tool, in the vein of a reconstruction of Xenakis' GENDY3 program (Hoffman 2000), and on the other as an involved generative music system, to highlight the possibilities of controlled but massive combinatorial composition spaces (Collins 2002). For open ended research, the application has been built with modules

for the sound synthesis and the generation of the canon melody. Currently, the defaults are a delay line physical model of a string, and a recursive microtonal melody generator with many parameters.

Aside from tempo games, canons are very much alive as a musical form thanks to other mathematical inquiries into their properties (Tudor Vuza 1990, Andreatta, Agon and Amiot 2002).

2 Tempo Canon Representation

Time maps provide a natural mechanism for representing tempo canons (Figure 1). The score along the bottom of the graph can be thought of as the score of the fundamental line. This is revealed in performance time, at a rate determined by the changing tempo of a voice, the tempo being the inverse of the gradient of the time map curve. Each voice i has some function $v_i(s)$ which maps from score to audition time. We draw all voices on one tempo map, to show their inter relation. One can check the respective positions of all voices in the canon melody by solving $v_i(s) = T$ for some time T , and all i . Visually, one just looks at the intersection of the zero gradient line $y=T$ with the functions v_i . Any convergence point is simply the crossing of voice lines. A common point (S,T) means that two or more voices are at the same point in the score at the same performance time.

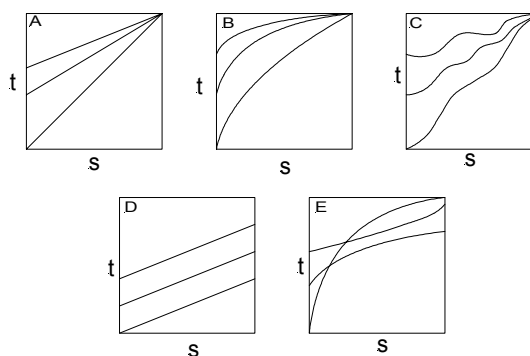


Figure 1 Time maps from score time s to performance time t for tempo canons in three voices. A, B, and C are strict mensural, acceleration and sinusoidal oscillation canons respectively, with a trio convergence point at the very end. D is a round, where all voices run at the same tempo but, because of delays in their entry times, can never catch one another. E is a more complex clash of accelerating and decelerating

lines, with two convergence points between two voices but no convergence point for all three.

The method of calculation for tempo canons in the program is to work relative to a primary voice. This is deemed to be voice one in the interface. The interface allows the selection of a single all voice convergence point, whether at the end of the piece, at the start, or outside the auditioned bounds. The all voice convergence point is determined in time with respect to the timeline of the primary voice. The function v_1 for each voice can be calculated from the canon melody and the user interface settings for the tempo map. Once the output time version of the melody is generated for that voice, it is lined up with respect to the convergence point (Figure 2). Finally, any user chosen delay can be added to force the voices out of alignment with the convergence point, and gain traditional rounds type canonic entry.

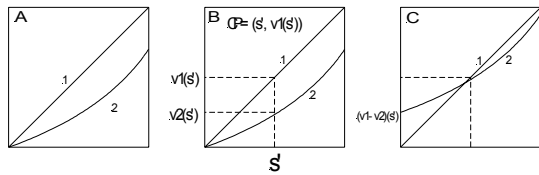


Figure 2 Calculating convergence points. The chosen convergence point is halfway through the score, at s' . A shows the unadjusted time maps for the voice lines. B makes explicit the difference in time of execution of score point s' for each voice. In C voice 2's entry is delayed to force the same performance time for score point s' .

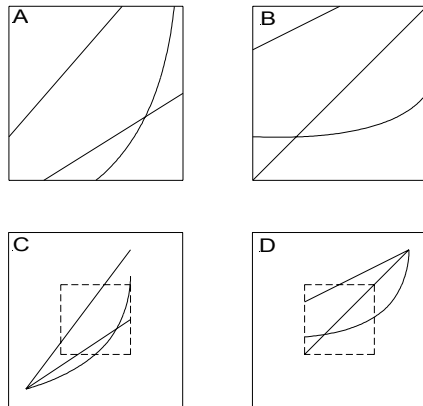


Figure 3 Convergence outside the given canon melody. The score (melody) of the auditioned canon is represented in C and D by the dotted box. A and B zoom in on the box to show what is actually performed, leading an observer to infer the existence of earlier or posterior convergence made explicit in C and D.

The case where we join a canon which is already in progress is very simply conceptually motivated by only looking at that range of the performance time for $endtime > t > starttime$. Figure 3 demonstrates how we

may audition only a subsection of a grander design, where the position of convergence points must be extrapolated outside the observed region. In the calculation of such schemes, one can add beats of silence to the event list of the canon melody for the positioning of convergence points.

3 An Implementation

Canonic Hill Loss is a SuperCollider 2 program and infinite length composition designed for the exploration of microtonal tempo canons, of standard mensural, acceleration and sinusoidal oscillation types. The user enters start and end tempi for each independent voice, as well as global delays, base frequencies for the voice, and parameters of the sinusoidal oscillation. There are options for selecting the position of the all voice convergence point that determines the form, as well as acceleration type, and the number of active voices. Save and load of canons as well as play back and write modes are supported, alongside controls for manual exploration, and a carefully prepared autocanons mode for infinite playback, aimed as a composition in it's own right.

In order to promote generality, and allow customisation, the generator for the fundamental canon melody, as well as the rendering engine for playback, are provided as separate classes. By substituting their own classes conforming to the appropriate interface, a third party can modify the canon generation system whilst keeping the tempo calculations and UI.

The melody generator provided as default tries to generate interesting monodies with glissandi and recursive ornamentation providing a variety of speeds. A set of outside time pitch materials are first created before the melodic line is put together. The finished line is the concatenation of distinct segments to make up the desired length. It would be fair to say that local structure up to about ten seconds area is accounted for, but that over the long term there is simple conjunction between segments of starting and ending conditions. Nancarrow's construction of melodic lines according to the placement of convergence points, and the through composition of canon melodies based on the growing harmonic structure as voices enter is not currently implemented. The user does have the capacity to audition different convergence point placements and could chase denser sections in that roundabout way. The scheme is still of sufficient scope to provide interesting melodies for the canonic treatments, and many parameters are available to the user to tweak the composition module. The canon melody is defined by an array of the form [duration, pitch, amplitude, timbre class] where pitch can itself be a pitch envelope for portamenti.

The support for microtones is for any equal tempered division, where the voice's base frequencies can be arbitrary or forced to conform to the tuning.

Stretched versions of melodic material used for ornamentation can be kept within the scale elements or allowed to diverge from the base tuning.

The synth module has as default an adaptation of a delay line string model by Staffan Liljegren. Organ and harmonica presets are also available.

To speed up data entry, and facilitate quick transformations of data arrays, SuperCollider's interpret facility for strings is used. This allows the compilation of code as you work, and one can create arrays of tempi very conveniently using legitimate SC code such as `Array.geom(7, 1, 1.04)` (creating a geometric series with ratio 1.04, starting at 1 beat per second, across the first seven voices). Further, rather than write a new class to replace the melody module, a melody in the correct format can be generated using any external code, and passed in via an interpreter variable. Interpret supports live coding in performance, as well as language based exploration of the canon space.

Aside from experimental exploration of tempo canons, the system has the capacity to run in 'infinite play' mode. Instead of random allocation of parameters, a lot of preparation and testing of combinations has gone into controlling parameter choices. As much as is feasible given the combinatorial spaces involved, this generative music composition has been restricted to felicitous parameter space regions. Undoubtedly, we are entering a world of proper software testing procedures for generative music works as they become sensorially complex. These issues are discussed in more detail in (Collins 2002).

Canonic Hill Loss is available freely under the GNU GPL from the author's web site above, with help files and many example canons.

4 Future Extensions

The main drawback of the program at the moment is the lack of a general graphical control over tempo trajectories, and the establishing of multiple and arbitrary many voice convergence points. This is somewhat beyond the GUI capabilities of SuperCollider 2, but the move to SC Server will allow a better attempt. An abortive attempt was made to draw a graphic of the melody using the Signal and SignalView classes, with the convergence point set using the selection tool.

Currently, one canon follows the completion of another in infinite play mode, with the termination of the last canon voice before the calculation of a new canon. Overlapping would move us more towards the advanced presentations of Nancarrow, for instance in Study 37. Conceptually, however, overlapping can be considered by concatenating vocal parts across time maps, as indicated in figure 4 on the right. Effects of 'waves of canon' can be heard within the system now if the melodic source is suitable. True double canon is not achieved by the current system, but would be a

straight forward addition. The time map diagrams for the union of voices would share a fundamental beatline, with multiple scores overlaid on the x axis depending on the voice.

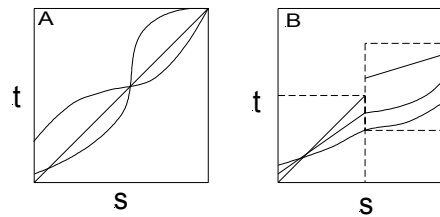


Figure 4 Complex convergence point structures. In A there are two all voice convergence points. This situation is only approximated at present by sinusoidal oscillation canons running in close proximity. B contains two dotted boxes, representing overlapping canons (the second begins whilst the first is still being completed by some voices). Note that this overlap could be modelled by a more complex score, concatenating event lists, and fluid time maps.

There are other advanced multitempo structures in music that would make good subjects for a generative music inquiry. Ligeti's experiments with simultaneous tempo by a single performer and tempo fugue come to mind, as demonstrated in his etudes for piano (Taylor, 1997). The connections between Ligeti and Nancarrow are very interesting in themselves, with Ligeti's discovery of Nancarrow's music later in his life, and pieces for mechanical instruments like barrel organ and player piano (Toop, 1999).

The explorations of an algorithmic composition system may be advanced by certain tools, as in the use of interactive genetic algorithm exploration of a search space. (Dahlstedt 2001) described a combined system for evolving both synthesis and composition parameters. GAParams, a SuperCollider library for interactive evolution, could be fruitfully combined with this research line into tempo canon (Collins 2003).

This author would be very interested to attempt an installation version of the canon generator, supporting multiple generators running at once for multiple canon effects. Visual analogues of the audio canons could also be constructed since the principles of tempo canon are about events in time, and not necessarily restricted to hearing. It would be trivial to output MIDI or OSC messages rather than synthesise the canons within SuperCollider, so an external instrument, like a MIDI piano, could be the target of the piece.

Finally, whilst sieves within equal tempered tuning systems are part of the current composition, more advanced tuning systems could still be applied, such as the Partch 43 note just intonation scale or independent tuning systems for each voice, an idea which I must credit to composer Fabrice Mogini.

5 Conclusions

Whilst Nancarrow often laboured for up to a year stamping his piano rolls, the computerised system described here takes seconds to calculate new tempo canons and synthesises them comfortably in realtime on a 400MHz Powerbook G4. That is certainly no claim for greater worth than Nancarrow's superlative compositions, but the advantages of such a system for exploration and adventure are clear.

A lot of care has gone into testing and fine tuning the system to produce canons in restricted classes deemed to be interesting, and thereby cutting out a lot of dross that the system would otherwise produce. Yet the system is also customisable enough that the user may modify it to their own exigencies. The building and use of the system has allowed an experimental attention to the nuances of tempo canon as a form that has aided the author in their appreciation of Nancarrow's pioneering work.

References

Andreatta, M., Agon, Carlos., and Amiot, E. 2002. "Tiling problems in music composition: Theory and Implementation." *Proceedings of the International Computer Music Conference*, Goteborg, Sweden. pp. 156-163.

Collins, N. 2002. "Infinite Length Pieces: A User's Guide." *Proceedings of MAXIS*, Sheffield, April, 2002.

Collins, N. 2003. "Experiments with a New Customisable Interactive Evolution Framework." *Organised Sound* 7:3. In press.

Dahlstedt, P. 2001. "Creating and Exploring Huge Parameter Spaces: Interactive Evolution as a Tool for Sound Generation." *Proceedings of the International Computer Music Conference*, Habana, Cuba. pp 235-242.

Desain, P. and Honing, H. 1992. "Tempo Curves Considered Harmful." in *Music, Mind and Machine: Studies in Computer Music*, Amsterdam: Thesis Publishers, 1992.

Jaffe, D. 1985. "Ensemble Timing in Computer Music." *Computer Music Journal*, 9:4, pp. 38-48.

Jaffe, D. 1982. *Silicon Valley Breakdown*, Available on CDs from WERGO and Well-Tempered Productions.

Gann, K. 1995. *The Music of Conlon Nancarrow*, Cambridge University Press, 1995

Hoffman, P. 2000. "The New GENDYN Program." *Computer Music Journal*, 24:2, pp. 31-38.

Honing, H., 2001. "From Time to Time: The Representation of Timing and Tempo." *Computer Music Journal*, 25:3, pp 50-61.

McCartney, J. 1998. "Continued Evolution of the SuperCollider Real Time Synthesis Environment." *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan.

Taylor, S.A. 1997. "Chopin, Pygmies, and Tempo Fugue: Ligeti's "Automne a Varsovie." *Music Theory Online* 3:3. <http://societymusictheory.org/mto/issues/mto.97.3.3/mto.97.3.3.taylor.html>

Toop, R. 1999. *Gyorgy Ligeti*, Phaedon Press.

Tudor Vuza, D. 1990. "Supplementary Sets and Regular Complementary Unending Canons (Part Four)." *Perspectives of New Music*, 31:1.

