International Conference on New Interfaces for Musical Expression

Canons for Conlon: Composing and Performing Multiple Tempi on the Web

Roger B. Dannenberg

Published on: Apr 29, 2021

ABSTRACT

In response to the 2020 pandemic, a new work was composed inspired by the limitations and challenges of performing over the network. Since synchronization is one of the big challenges, or perhaps something to be avoided due to network latency, this work explicitly calls for *desynchronization* in a controlled way, using metronomes running at different rates to take performers in and out of approximate synchronization. A special editor was developed to visualize the music because conventional editors do not support multiple continuously varying tempi.

Author Keywords

Network performance, telematic, metronome, conducting, composing interface

CCS Concepts

• **Applied computing** → **Arts and humanities** → **Sound and music computing**; • **Applied computing** → **Arts and humanities** → **Performing arts**;

Introduction

Music performance over networks, sometimes referred to as telematic music, has a long history and a variety of approaches, both technical and philosophical [1]. As a result of the pandemic of 2020, many musicians and music schools find themselves scrambling to leave conventional performance behind and form a workable and musically meaningful new approach in which musicians and audiences are not in physical proximity. The work “Canons for Conlon” was written for the Contemporary Ensemble at Carnegie Mellon University. The work necessarily avoids tight synchronization of multiple musicians since that is not possible using off-the-shelf teleconferencing tools such as Zoom. Instead, I decided to take the loss of synchronization as a “feature,” writing for performers that intentionally drift out of synchronization and back in. The form of the canon was chosen because it allows the audience to hear deliberate manipulation of synchronization when identical phrases are heard with delays, just as a conventional canon contains two or more parts at fixed offsets (or sometimes in other relationships).

Two interesting technologies were developed to enable these canons. First, the calculation of relative note timing when each part is following a different tempo trajectory is very complex and impractical to work out by hand. Therefore, I created a custom sequence editor that calculates all the parts and their timing immediately as

notes are entered and edited in one part. This compositional support can be applied to music for conventional live performance, recorded performance, and computer sequencer performance as well as network performance.

Secondly, I developed a web page to conduct performers and help them stay synchronized (or *desynchronized*) according to very carefully calculated tempo trajectories so that performances at least comes close to the intended timing. While this seemed absolutely necessary to realize these canons (in the presence of network latency or not), the metronomes also assist in performances of conventional music, especially with high network latencies.

The next section explains some of the rationale for the concert configuration and characterizes latency in network performances. Subsequent sections present more detail of the canon concept and brief descriptions of these two technical components.

Latency Issues

Latency is a well-known problem for network performances. In a metropolitan area and with good network connections, latency can be low enough for musicians to synchronize effectively if not as comfortably as in close physical proximity. In an ideal situation, audio capture and playback takes 5 to 10 ms at each end. Small network packets introduce significant overhead, so audio is typically sent in packets with around 5 ms of audio, adding another 10 ms to the round-trip latency. Packets are also buffered in network switches and routers, typically adding at least 10 to 20 ms of travel time each way. This gives an overall round-trip time of 40-70 ms. This range is achievable only if there is no jitter (variance) in the network transmission time. In reality, packets are sometimes delayed due to contention at routers. Additional samples must be buffered at the receiving end to allow continuous output when incoming packets are delayed. This can easily add 100 ms each way, but typical systems buffer fewer samples to reduce latency and suffer from occasional dropouts when packets are late. In practice, round-trip latencies of 60 to 120 ms are achievable, which is marginal for rhythmic performance. (Note that some network audio systems report latencies that are either one-way or do not consider all of the sources of delay from audio input to audio output. One must be skeptical of latency claims and perform some simple measurements.)

Although fairly low latency is achievable in the best case, it is also interesting to consider longer latencies, either because of longer distances or because of less optimized audio delivery. For example, systems for multiple performers often send

audio to a hub where audio is mixed and returned to performers. This can double latency because now there are *two* round trips across the network (although no audio conversion at the hub). System such as Zoom appear to use a server even when two clients are on the same local area network, and typical latencies with Zoom are around 300 ms. Latency is also increased by long distances. The round-trip network time across the US is around 100 ms, and intercontinental connections of course take longer, giving latencies of 200-300 ms or higher.

This work was designed for a concert based on Zoom. In retrospect, this was not a good choice for music performance due to Zoom's built-in policy of (often) favoring one voice over the rest, which gives very erratic behavior for polyphonic music. Alternatives include systems designed for music, including JackTrip, Sound Jack, JamKazam, Jitsi, and Jamulus. These were not explored due to a lack of time, expertise and equipment (student musicians were in isolation early in the pandemic, and some did not even have laptops). However, network latency can be high even with the best choices, and dealing with latency is still an area of active experimentation and exploration.

Of course, music can also be performed asynchronously, one part at a time, and mixed later rather than in a live synchronous manner. This overcomes synchronization problems due to network latency, but in the case of these poly-tempo canons, musicians need help to realize the complex timing. Thus, this investigation of metronomes is relevant to many situations. We will discuss later how these metronomes were used for more conventional music as well.

Related Work

Network performance is now an established field with an extensive literature, [2][3] and the effects of delay on performers have been studied [4][5].

Metronomes have been suggested before. Reid Oda's thesis [6] describes various network synchronization schemes and in particular synchronized audio beats instead of visual metronomes. Hupke, et al. describe an audio metronome "Rhythmic Synchronization Service" [7]. The author's Global Net Orchestra [8] used scrolling piano-roll displays to conduct 64 musicians in synchronized performance where latencies were variable and up to 1.3 s for measured *minimum* round-trip time. Other research has studied even higher latency where delays are synchronized to whole musical units of repetition [9][10].

Polytempo music has been explored by 20th Century composers including Cowell, Nancarrow and Ives and has inspired a number of interfaces and software tools in the Computer Music community. Charles Holbrow reviews some history and describes a tool for calculating sliding tempos [11]. Interfaces where computers assist in conducting polytempo music include Polytempo Network [12]. Notational and compositional support for polytempo music is rare. Sibelius (www.avid.com/music-writing-software) supports multiple tempos, but does not support precisely specified continuous tempo changes used in the present study and implemented in the editor described below.

The Canons

There are two three-voice canons forming the two movements of this work for flute, clarinet and bassoon. The first canon begins in octaves, but soon the flute begins to play faster while the bassoon plays slower. The players drift apart and counterpoint develops between them. At the half-way point (90 seconds), the instruments are back to playing the same tempo, but the flute continues to slow down while the bassoon continues to speed up. The instruments converge at the very end, still playing the same music in octaves. Aside from tempo and octave differences, instruments play the same pitches and rhythms throughout the piece.

The flute tempo places the flute ahead of the clarinet by a smoothly varying function of the time. The function is a raised cosine function that continuously increases from zero to maximum and back to zero over the course of the movement. The bassoon mirrors this behavior using the same raised cosine function to describe how far *behind* it should be at each time point in the performance.

The second canon is more ambitious. The flute speeds up continuously from beginning to end, the clarinet slows down from beginning to end, and the bassoon plays a fixed tempo in between. The clarinet part is an inversion of the flute and bassoon parts. Finally, each part is a musical palindrome that is the same whether played forward or backward, except for tempo changes.

The tempi in this movement change linearly over time, resulting in a parabolic mapping from time to beat, and the tempi are further constrained such that at the beginning, the flute part has exactly one half the tempo of the clarinet.

The Editor

To enable the composition of these canons, I created a sequence editor that automates the calculation and visualization of timing and pitch as notes are graphically entered and edited. This editor was not intended as production quality software since it was only made for the creation of one short piece of music. Nevertheless, quite a few features were added to make composition more practical, including the ability to stretch, shrink or shift regions of notes, divide note onsets equally over an interval of time, MIDI preview, and a simple Undo command.

Figure 1 shows a screen shot of the editor in use. It can be seen that notes on the left side (time zero) are synchronized, while on the right (about 45 seconds), notes are skewed. The zoom control at the bottom is used to zoom in on the horizontal (time) axis to allow better visualization and fine timing adjustments.

Most of the composition progressed left to right, focusing on the fastest part. As each note is entered, a copy appears in each of the other two parts. In the second movement, each entered note appears 6 times because each of the three parts is a palindrome. Thus, entering the first note in the flute part automatically creates the last note for the flute and the first and last notes for both the clarinet and bassoon. Notes that are automatically added in the slower parts come later and become context for new notes in the faster tempo part. Of course, it is possible to enter and revise notes at any location.

The editor was implemented from scratch in Serpent, a Python-like scripting language created by the Author for music processing [13].

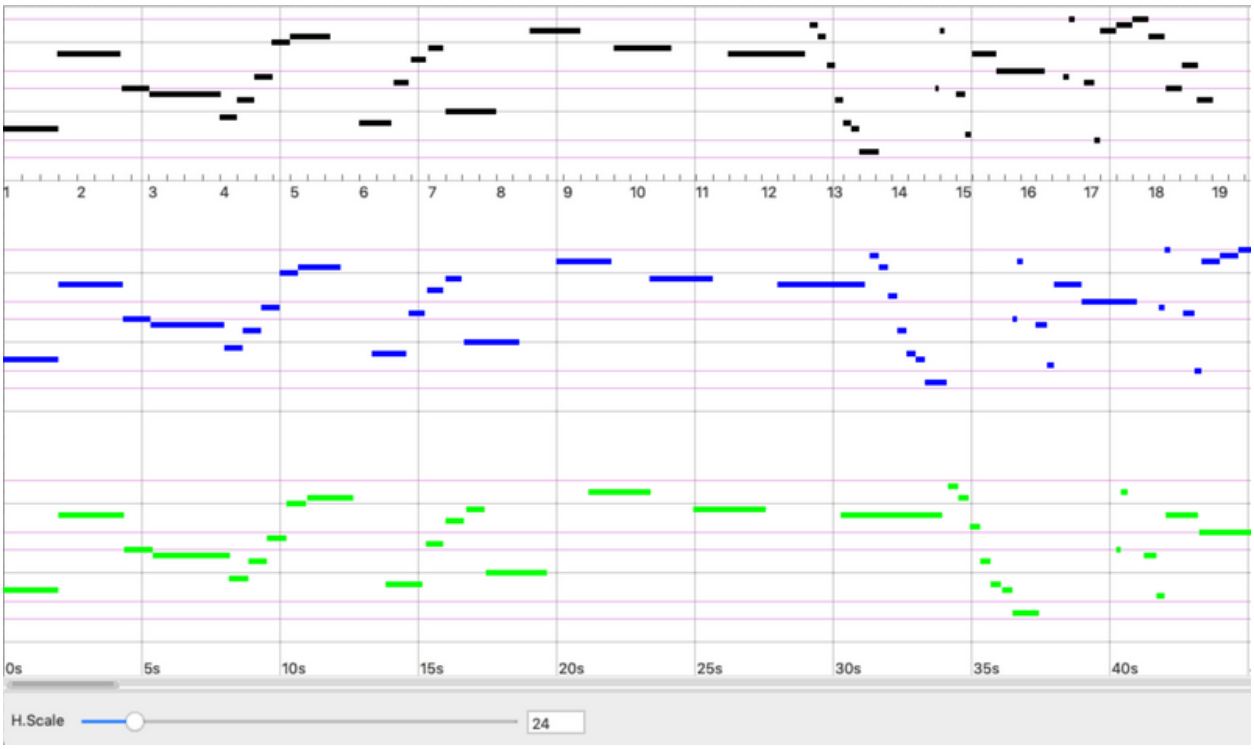

Figure 1. The canon editor. Parts 2 and 3 are copies of Part 1 with timing automatically mapped according to continuously varying tempi

The editor produces Standard MIDI files as output. The final output was imported into Finale and edited by hand to clean up the quantized rhythms, add dynamic markings and other indications, and generate parts for musicians.

The Metronomes

Performing the piece over the network (or anywhere) would be extremely difficult given the changing tempi. I created a simple metronome using p5.js (https://p5js.org) to display a circle moving in the familiar 4/4 conducting pattern and to generate clicks. Initially, players see the screen in Figure 2.

Figure 2. The web-based metronome.

The p5.js program makes several requests to the server to get the server time, and after obtaining a suitably low round-trip response time (ensuring that the timestamp is accurate), computes the offset between the local clock and the server clock. We assume that clock drift is negligible during a short performance and that even a few tens of milliseconds of error will be dwarfed by other latencies in the performance, so this synchronization scheme is quite simple.

Once metronomes are online and synchronized, they poll for status changes. (Polling was used for simplicity. It would be more efficient to use web sockets to push state changes to clients.) To start the performance, any player pushes a start button. This selects a start time about 10 seconds in the future, sends a command to the web site, and within a few seconds the plan is picked up by all metronomes. All metronomes now share synchronized clocks and share a known common start time. They can “conduct” each player locally without further interaction and with immunity from network latency. Figure 3 shows the interface on the first beat of measure 1. The conducting pattern illustrates the path of the ball over the course of each measure.

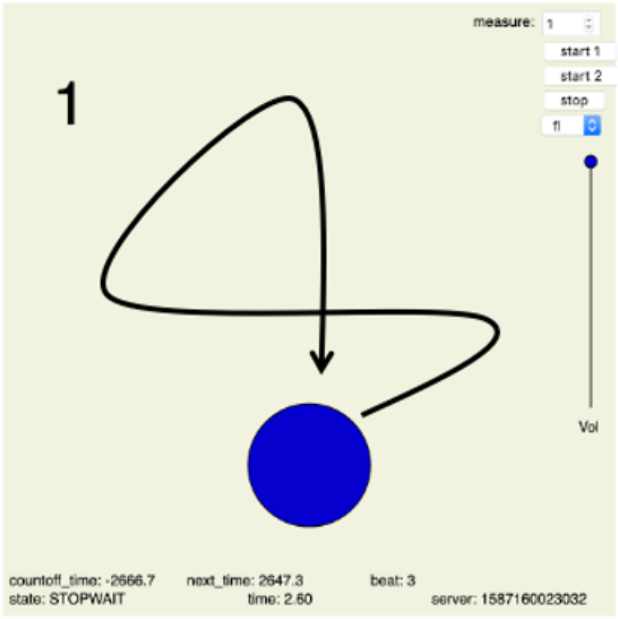

Figure 3. The web-based metronome on measure 1, beat 1, with an illustration of the animation path of the conducting ball.

The tempo depends on the instrument, so a drop-down box is used to select Flute (fl), Clarinet (cl) or Bassoon (bn). Based on the instrument and the movement (determined by “start 1” or “start 2” buttons), a different mapping is used from beats to times. These mappings are pre-computed and stored in tables. At run time, the metronomes compute the time of the next beat and interpolate smoothly over that time to generate the conducting animation.

The metronomes also allow any player to stop the performance, which is useful for rehearsals, and to start conducting at any measure. This last feature is a bit tricky when parts play at different tempi. When a player requests a starting point not at the beginning, the measure is converted to a time offset, the offset is shared through the server, and then each metronome converts the time to the nearest measure for each respective instrument. Each instrument receives a 4-beat “count-off,” but the count-offs will generally start at different times at different tempi so that instruments will be synchronized as intended.

Another possible application for these metronomes is in larger network performances where network connections might be intermittent. Given the use of the local clock, beat display, and measure numbers, it is possible that performers could lose and regain their network connection without losing a beat.

Results

The piece was performed in April, 2020 using Zoom as planned. The metronome system worked well and enabled performers to realize the poly-tempo composition as intended. In rehearsals for the same concert, another piece was found to be unplayable given the latencies of Zoom and the rhythmic nature of the piece. The performers and conductor asked for a similar metronome interface. This piece was also successfully performed with the interface, demonstrating that synchronized metronomes can assist with more conventional pieces performed in the presence of significant latency.

To further investigate timing behavior, two experienced musicians played some simple duets over Jitsi audio connections (290 ms round-trip end-to-end audio latency) with and without metronome support. Average tempo was estimated over 4 contiguous segments of the duet with durations from 9 to 14 beats by dividing the length in beats by the time to perform the segment. Not surprisingly, tempo was more stable and predictable with metronomes: With metronomes, tempo was essentially constant. Without metronomes, tempo variation from slowest segment to fastest segment was 9% at a nominal 68 beats per minute (BPM), 2% at 90 BPM, and 10% at 120 BPM. The nominal tempo was essentially perfect with metronomes, but without metronomes, tempo was about 10% too slow when attempting to play at 68 BPM, 1% fast at 90 BPM, and 9% too fast at 120 BPM (players listened to a reference metronome before starting in each case to establish the nominal tempo).

These tests show that good players can in fact play together in the face of high latency (290 ms), with or without metronome support, and tempo deviations might even be considered “expressive” rather than problematic. A better assessment comes from looking at synchronization. Estimating note onset times reliably is a tedious manual process, so we evaluated a selected set of 60 onsets rather than all onsets. To assess synchronization, we measured the mean absolute deviation of onsets after correcting for bias. (For example, if one player was always exactly 100 ms early, then we added 100 ms to all onset times to correct for this—essentially delaying the entire performance of the early player—before computing onset deviations. In this case, after removing the bias, the resulting mean absolute deviation would be zero).

With metronome, the mean absolute deviation was 93 ms. Without metronome, the mean absolute deviation was 201 ms. The errors were larger at the slow (68 BMP) and fast (120 BPM) tempos. Of course, these numbers are dependent on familiarity with the music, tempo, rhythmic complexity, and even performance tricks. For example, one player could take the lead and not even listen to the other player, letting the other

player follow, and this might lead to high synchrony in terms of mean absolute deviation after removing the bias of the latency from the leader to the follower. Nevertheless, there is strong support for our claim that *metronomes can enable performers to perform together even in the presence of significant latency*.

It should be mentioned that success in synchronization does not mean success in making music. Although this has not been studied carefully, Metronomes, with or without latency, seem to diminish the ability of musicians to respond to one another, and following a metronome seems to make it hard to also concentrate on expressing music. This was one of the inspirations for poly-tempo canons: turning the lack of a shared pulse and entrainment into a unique feature of the work. Hopefully, listeners appreciate drifting in and out of synchronization as an interesting compositional device rather than a musical deficiency.

Conclusions

There are many possible approaches to network performances. Most researchers have had the luxury of setting up special software and audio equipment and sometimes even special networking facilities. It was quite unexpected to be working with acoustic instrumentalists who were restricted to their homes and limited to devices (laptops, phones, and tablets) that happened to be at hand when the campus suddenly closed. In the end, the new metronome interfaces helped to perform both conventional metrical compositions and new canons based on multiple tempos. I hope that this work will raise some awareness of music technology and interfaces and that hearing these principles worked out musically will delight some listeners, just as canonical writing has fascinated composers, musicians and audiences for centuries.

ACKNOWLEDGMENTS

Thanks to performers Nicholas Evans, Alicia Maria Gutierrez Oviedo and Leah Stephens, conductor Daniel Evans, and composer Nancy Galbraith. Thanks to the School of Music for its support. These canons are inspired by the seminal work of Conlon Nancarrow. Thanks to an anonymous reviewer for suggesting the possibility of using metronomes to help musicians rejoin a network performance after suffering a network drop-out.

ETHICAL STANDARDS

Accepted principles of ethical and professional conduct have been followed, and there are no known conflicts of interest.

Citations

1. R. Mills. *Tele-Improvisation: Intercultural Interaction in the Online Global Music Jam Session*. Springer, 2019. ↩
2. A. Kapur, G. Wang, P. Davidson and P. Cook. Interactive Network Performance: A dream worth dreaming?, *Organized Sound*, 10, 2005. ↩
3. P. Oliveros, P., S. Weaver, M. Dresser, J. Pitcher, J. Braasch, and C. Chafe. Telematic Music: Six Perspectives, *Leonardo Music Journal,* 19, 2009. ↩
4. C. Chafe, J. Cáceres, M. Gurevich. Effect of temporal separation on synchronization in rhythmic performance. *Perception*, 39, 2010. ↩
5. E. Chew, A. Sawchuk, C. Tanoue and R. Zimmermann. Segmental Tempo Analysis of Performances in User-Centered Experiments in the Distributed Immersive Performance Project. In *Proceedings of the Sound and Music Computing Conference,* 2005. ↩
6. R. Oda. *Tools and Techniques for Rhythmic Synchronization in Networked Musical Performance.* Ph.D. Thesis, Princeton University, 2017. ↩
7. R. Hupke, L. Beyer, M. Nophut, S. Preihs, J. Peissig. A Rhythmic Synchronization Service for Music Performances over Distributed Networks. In *Fortschritte der Akustik – DAGA,* 2019. ↩
8. R. Dannenberg and T. Neuendorffer, “The Global Net Orchestra: A Large-Scale Internet Music Performance,” in *Proceedings of the Fourteenth Biennial Symposium on Arts and Technology*, New London, CT, February 2014, pp. 730 - 736. ↩
9. Nicolas Bouillot. nJam user experiments: Enabling remote musical interaction from milliseconds to seconds. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME 2007)*. 2007, pp. 142–147. ↩
10. M. Goto, R. Neyama, and Y. Muraoka. RMCP: Remote Music Control Protocol: Design and Applications, In *Proceedings of the 1997 International Computer Music Conference*, 1997, pp.446-449. ↩
11. C. Holbrow. RMCP: Tempic Integrator: A modern tool for composing with sliding polytempos as conceived by Henry Cowell in 1930, In AES 146st Convention, 2019. ↩

12. Philippe Kocher. Polytempo Network: A System for Technology-Assisted Conducting, In *Proceedings of the 11th Sound & Music Computing joint with the 40th International Computer Music Conference*, 2014. ↩

13. R. Dannenberg, “A Language for Interactive Audio Applications,” in *Proceedings of the 2002 International Computer Music Conference.* San Francisco: International Computer Music Association, (2002), pp. 509-15. ↩