

Conductor Follower

Bennett Brecht and Guy E. Garnett
Center for New Music and Audio Technologies (CNMAT)
Department of Music
University of California, Berkeley
1750 Arch Street, Berkeley, CA 94709
(510) 643-9990
brecht@cnmat.berkeley.edu , guy@cnmat.berkeley.edu

Abstract:

We present further enhancements to the conductor following system first defined in (Lee et al, 1992). The conductor follower is a real-time program for controlling synthesis and/or live processing under the dynamic control of a conductor. It is written in Max and runs on a Macintosh. The first part of this paper presents an overview of the system. The second section will focus on current solutions for problems of timing and temporal expression; these are the fundamental concerns of this performance paradigm.

Conductor Follower:

Conductor following, related gesture recognition, and score following, have been previously discussed by Dannenberg (Dannenberg 91), Vercoe (Vercoe 84), Keane (Keane 91), Chabot (Chabot 87), Subotnick (Subotnick 94), Mathews (Mathews 1980), Morita (Morita 1990), et. al. What differentiates this project from the others is our dedication to creating a follower controlled by standard conducting gestures with an inexpensive and unobtrusive instrument interface. The aim has been to develop a tool for synchronizing and controlling an electronic performance in conjunction with live ensemble performance. For the instrument interface we use continuous information from an infrared baton, "Lightning," developed by Don Buchla. Using this system, a conductor can control an electronically stored score (which can either contain synthesis or processing commands, via MIDI or ZIPI, or algorithms of any sort) with standard conducting gestures. The project has, since 1992, been in use in different forms of its development in live concert performances with various instruments including full symphony orchestra. The conductor follower offers control over dynamic scaling and tempo by allowing the program to monitor all parts of a conducting gesture (or 'curve'). It also offers a remarkable amount of flexibility in compositional possibilities by being able to execute and steer preset algorithmic "processes" that might normally defy standard music notation (done with a special "conducting track"). Rehearsing with human performers is facilitated by the addition of easy cuing to rehearsal marks and quick stop/start capability. There is a simple way to input scores via MIDI files, with some special interpretations encoded on one of the MIDI channels.

Timing Problems and Solutions:

As with other conducting or score-stepping programs, timing issues are crucially important. Changes of tempo, at both the micro and macro levels, and the need to follow pauses and other aspects of temporal phrasing are inherent in the performance process. We have pursued research and development on prediction of future conducting events and temporal spacing of present and future score events via gesture recognition by using three models. The first model is a broad-scale historical updating algorithm used in the conductor follower; this is extremely reliable but often not very smooth. We developed a second algorithm based on more detailed attributes of conducting curves in which tempo is determined at more points in the gesture giving flexibility on a much finer level. The third model uses Neural Networks. Since conducting curves vary in complicated ways with respect to articulation as well as tempo, we hope to use neural nets to adapt to these local variations. We have been encouraged by our preliminary results in getting the networks to predict the intended tempo and look forward to training them in a wider variety of circumstances.

Algorithm #1 is straightforward. The estimated beat period is simply the previous measured beat period. Just as is the case for ensembles, the first beat and non-gradual tempo changes are usually preceded by an anticipatory beat. Tempo is derived from the period and a clock outputs isochronous ticks to update a scheduler. The chief problem with this method is it does not change rapidly enough when the performance tempo changes substantially. Updating is only done at the onset of each new beat.

Algorithm #2 came about as the result of discoveries regarding inherent characteristics of every beat curve when monitored continuously. Using zero-crossings in velocity and acceleration, we can consistently mark six mappable points in a conducting beat 'curve', regardless of conductor and tempo. These six points and their time relation to "time zero" (beat onset) were then used in direct comparison to the same time segment of the previous period, creating " ΔT ." We then extrapolate a prediction of the next downbeat at each of these

markers based on that ΔT relationship. The results gave us a similar paradigm as Algorithm #1, but with more frequent updating. This algorithm enabled us to avoid gross over- and undershooting of the beat. However it exacerbated one of the other problems that plague tempo control systems: how much tempo flexibility is intentional and how much is unintentional? In certain kinds of music following the beat too precisely can be a definite disadvantage, e.g., sometimes a change in "lift" (the part of the beat rising from the lowest point) velocity does not mean a change in tempo but a different articulation (such as staccato or legato). To help solve this we have been pursuing a third algorithm.

Algorithm #3 uses a Neural Net trained on the relationship between marker points, time, and the probability of the next beat happening. Scaled running time for the current beat is fed into the first inlet, the six markers create a state of 0.0 (hasn't happened yet) or 1.0 (has happened) into the remaining inlets. The output is a single node which shows the rising % of beat completion, this can be equated with the "probability of the next beat happening." The output is smoothly rising with each inlet having a significant contribution to the result. Running time here serves as an interpolator between the markers and does an excellent job. A second net then takes this "% complete" number as a single input and remaps it onto the subdivisions required. This system provides an inherent degree of smoothness in the tempo response. The final problem we are tackling is to interpret gestures more globally; nets can be used to compare local differences in velocity in the context of other nearby velocities. This should allow us to interpret various beat types (legato, staccato, etc) without inadvertently affecting tempo.

References:

- Boulanger, Richard. 1990. "Conducting the MIDI Orchestra, Part 1: Interviews with Mathews, Vercoe and Dannenberg". Computer Music Journal. MIT Press, Summer 1990.
- Chabot, Xavier. 1987. "Performance with electronics : gesture interfaces and software toolkit" Dissertation, UCSD.
- Dannenberg, Roger and Bookstein, Kenneth. 1991. "Practical Aspects of a MIDI Conducting Program". ICMC Proceedings San Francisco: Computer Music Association.
- Dannenberg, Roger. 1984. "An Online Algorithm for Real Time Accompaniment". ICMC Proceedings, San Francisco: Computer Music Association.
- Dannenberg, Roger. 1989. "Real Time Scheduling and Computer Accompaniment". Current Directions in computer music research, MIT Press.
- Keane, David and Wood, Kevin. 1991. "The MIDI Baton III". ICMC Proceedings San Francisco: Computer Music Association.
- Lee, Michael, Garnett, Guy, and Wessel, David. 1992. "An Adaptive Conductor-Follower." ICMC Proceedings San Francisco: Computer Music Association.
- Mathews, Abbott. Winter 1980. "The Sequential Drum". Computer Music Journal, V4:4. MIT Press.
- Morita, Hideyuki and Watanabe, Hiroshi et. al. 1990. "Knowledge Information Processing in Conducting Computer Music Performer". ICMC Proceedings San Francisco: Computer Music Association.
- Subotnick, Morton. 1994. Personal communications with Mark Coniglio.
- Vercoe, Barry. 1984. "The Synthetic Performer in the Context of a live performance". ICMC Proceedings, San Francisco: Computer Music Association.