Firefly with me: distributed synchronization of musical agents

Kristian Nymoen, Arjun Chandra, and Jim Tørresen

A system where individual musical nodes are aware of others and adapt to reach a synchronous state provides a collaborative active music performance for smartphones.

Since the 1980s mobile music technologies have evolved from being just simple portable music players to versatile mobile phones that can be used as musical instruments, recording studios, music games, recommendation services and more. Mobile technologies allow people to consume or perform music anywhere, and challenge the traditional distinction between the performer creating the music and the perceiver receiving the music.[1] It is now possible to develop music technologies that provide users with a range of control possibilities that is much larger than the traditional play, pause, skip song and volume control, and at the same time more restricted than the demanding task of playing, for instance, a violin. We call these 'active music technologies.' A subset of such technologies involves 'collaborative active music,' meaning that a group of people come together to engage in a musical activity. The individual music from the mobile phones of different people may fuse together into one piece of music.

To facilitate collaborative active music performances, several challenges must be tackled. One is the synchronization between individual devices. While some people may be skilled enough to keep up with the beat of the rest of the group, others may require their mobile device to handle the synchronization task. Our European consortium has developed a framework for describing self-awareness in systems of interacting computing components, such as people interacting using mobile phones.[2] This framework is well suited for designing a system where individual musical nodes are aware of their own level of synchrony with the rest of the group, and adapt to reach a synchronous state.

To facilitate such musical interactions in social settings, people should be free from configuring network connections. In addition, they should be allowed to leave or enter as they please without causing the ongoing music to stop. We introduce a
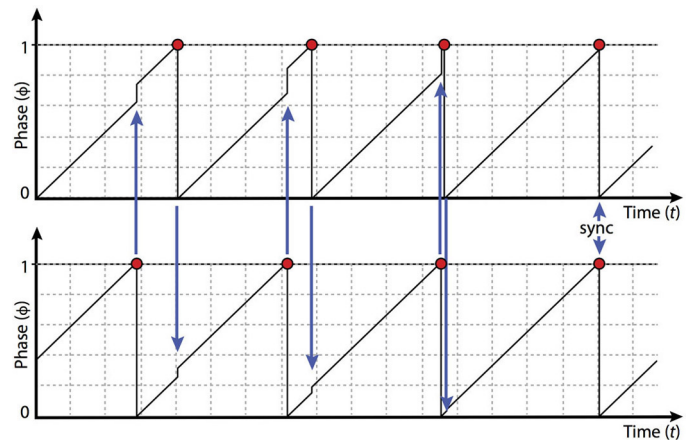
Figure 1. Illustration of the interaction between two pulse-coupled oscillators. Individual nodes are modelled as pulse-coupled oscillators that adjust their phases when they receive a fire event, such as when the phase (φ) of another node reaches 1.

system where individual musical nodes communicate solely through audio to achieve synchronization. The system is decentralized, meaning that there is no central controller to which the other nodes synchronize. Rather, the nodes influence each other equally, and eventually converge to a common musical beat. Our system is based on previous research inspired from synchronizing systems in nature, namely the synchronous flashing of certain species of firefly. Mirollo and Strogatz[3] showed how individual nodes reach a synchronous state if modelled as pulse-coupled oscillators that adjust their phases whenever they receive a fire event from another node. Each node contains an oscillator whose phase moves from 0 to 1 at a steady rate. Whenever the phase reaches maximum, the node outputs a pulse and the phase is reset to 0. Other oscillating nodes perceiving the fire event increase their phase by a small amount that depends on their own current phase (see Figure 1).

In our system, the individual musical nodes are modelled as pulse-coupled oscillators emitting a short impulsive tone on

each fire event. Since our current focus is not on harmonies or melodies, the tones are restricted to a pentatonic scale (C4, D4, E4, G4, A4), which ensures a degree of consonance between the nodes. The ability of the node to synchronize to other nodes is what has previously been termed 'protomusical behaviour,' that is, a behaviour that exhibits musical features, such as harmonic oscillations or rhythmic patterns, but lacking cultural realization as music.[4] The system has been implemented on iOS devices using the graphical programming environment Pure Data[5] and the iOS app MobMuPlat[6] (see Figure 2). Each node consists of four simple blocks: an oscillator, an onset detector, a phase update function and a synthesizer. The onset detector listens for fire events from other nodes received through the microphone. The phase update function then receives the fire events from the onset detector and causes a phase jump in the oscillator, and the synthesizer plays short, impulsive tones when the oscillator phase is equal to 1.

The use of audio as a communication channel implies a short time delay between the time when a node is firing and when the fire event is received by another node. The Mirollo and Strogatz model assumes that communication is done by infinitely short impulses without transmission delay. One possible solution to circumvent the problem of delays is to introduce a short time period after a node has fired in which it is prevented from making phase jumps, a so-called refractory period.[7,8] In the current implementation, a 50ms refractory period is used. A video demonstrating the system in action can be found on the Vimeo channel that has been set up for EPiCS project.[9] The software has also been made available for download for anyone who wants to try it out.[10]

In summary, we have developed a collaborative music performance system for mobile devices that operates by distributed phase synchronization of musical agents. Our next effort in this research is towards frequency synchronization of oscillators, which will enable adjustments of the musical tempo.

Author Information

Kristian Nymoen, Arjun Chandra, and Jim Tørresen
University of Oslo
Oslo, Norway

References
1. C. Small, **Musicking: The Meanings of Performing and Listening**, Wesleyan, 2011.
2. http://www.epics-project.eu/ European Commission project Engineering Proprioception in Computing Systems (EPiCS). Accessed 23 October 2013.
3. R. E. Mirollo and S. H. Strogatz, *Synchronization of pulse-coupled biological oscillators*, **SIAM J. Appl. Math. 50** (6), pp. 1645–1662, 1990.
4. I. Cross, *Music, cognition, culture, and evolution*, **Ann. New York Acad. Sci. 930** (1), pp. 28–42, 2001.
5. http://puredata.info/ Pure Data open source visual programming language. Accessed 23 October 2013.
6. http://www.mobmuplat.com/ MobMuPlat iOS application developed by D. Iglesia and published by Iglesia Intermedia. Accessed 23 October 2013.
7. Y. Kuramoto, *Collective synchronization of pulse-coupled oscillators and excitable units*, **Physica D: Nonlinear Phenom. 50** (1), pp. 15–30, 1991.
8. J. Klinglmayr and C. Bettstetter, *Self-organizing synchronization with inhibitory-coupled oscillators*, **ACM Trans. Auton. Adapt. Syst. 7**, p. 30, 2012.
9. http://vimeo.com/67205605 Video demonstrating the system in action. Accessed 23 October 2013.
10. http://fourms.uio.no/downloads/software/musicalfireflies/ Demonstration of the 'Musical Fireflies' distributed synchronization of musical nodes. Accessed 23 October 2013.

Figure 2. *The system running on three mobile devices.*