Open Sound Control: an enabling technology for musical networking

MATTHEW WRIGHT

Center for New Music and Audio Technology (CNMAT), University of California at Berkeley, 1750 Arch St., Berkeley, CA 94709, USA
Center for Computer Research in Music and Acoustics (CCRMA), Dept of Music, Stanford University, Stanford, CA 94305-8180, USA
E-mail: matt@{cnmat.Berkeley, ccrma.Stanford}.edu

Since telecommunication can never equal the richness of face-to-face interaction on its own terms, the most interesting examples of networked music go beyond the paradigm of musicians playing together in a virtual room. The Open Sound Control protocol has facilitated dozens of such innovative networked music projects. First the protocol itself is described, followed by some theoretical limits on communication latency and what they mean for music making. Then a representative list of some of the projects that take advantage of the protocol is presented, describing each project in terms of the paradigm of musical interaction that it provides.

1. INTRODUCTION

There is no doubt that computer networking has profoundly changed the industrialised world. The story of the exponential growth of the Internet has become a cliché on a par with comparisons of ever-improving computer processing power, storage capacity, and miniaturisation. More profoundly, perhaps, the Internet has become a dominant cultural factor: we are in the 'Information Age' and millions of people, on every continent, use e-mail, the Web, instant messaging, file sharing, etc., in their everyday lives.

The paper 'Beyond being there' (Hollan and Stornetta 1992) argues that the attempt to make electronic communication imitate face-to-face communication is ultimately wrong-headed and doomed to failure. Network delays, limited bandwidth, mediation by loudspeakers and video screens, and related factors guarantee that telecommunication will never be 'as good as' face-to-face communication on its own terms.[1] They reference studies showing that people tend choose face-to-face communication when available, that groups of people working in the same place tend to have more spontaneous interactions, etc. Therefore, they argue, the design goals should be to leverage the specific advantages of alternate forms of communication, with examples such as asynchrony, the potential for anonymity, and mechanisms to keep individuals from dominating a group discussion. Applying these insights to the music domain, I believe that networks are strictly a disadvantage when used simply as a transmission medium: they enable a hobbled form of musical contact over a distance, but can never be 'as good as' face-to-face music making on its own terms. The point isn't that physical proximity is inherently superior to technologically mediated communication, but that we should use computer networking to provide new and unique forms of musical communication and collaboration. As Hollan and Stornetta say, 'we must develop tools that people prefer to use even when they have the option of interacting as they have heretofore in physical proximity'.

What, then, are the potential advantages of computer networking for music? Certainly the most common use today is to distribute music via downloads and streaming. Although we always want minimal latency between selecting music and being able to hear it, we could view the (arbitrary amount of) time between when music is made available and when it is selected for download, i.e. the asynchrony, as a beneficial form of latency, one of the enormous benefits of network distribution. Other benefits include not worrying about physical distance, access to large collections, the ability to search, etc.

Music creation is a richer and more technically challenging domain than transfer of pre-recorded music. Many projects have explored the simple transmission of near-real-time musical data via networks (e.g. Young and Fujinaga 1999; Chafe, Wilson, Leistikow, Chisholm and Scavone 2000; Lazzaro and Wawrzynek 2001) Over long distances (e.g. intercontinental) these mechanisms enable musical collaboration that would otherwise be impossible, but with unavoidably long latencies (discussed below). Over medium distances (e.g. within a city) these mechanisms could provide for greater convenience, for example, taking a music lesson without having to get to the teacher's studio. I agree with Hollan and Stornetta that all such attempts to emulate 'being there' over a distance will provide less richness than face-to-face interaction. In some situations, of course, the less rich interaction will actually be an advantage, for example, being able to give a piano lesson in one's bathrobe, or to give simultaneous piano lessons to multiple students at the same time. And in many

---

[1] Even with best-case, high-bandwidth, audiovisual teleconferencing, there remain fundamental issues such as not being able to see what is the object of the other person's gaze.

Organised Sound **10**(3): 193–200 © 2005 Cambridge University Press. Printed in the United Kingdom. doi:10.1017/S1355771805000932

situations the less rich interaction is a small price to pay for the network's advantages with respect to time management, travel expense, availability, etc. But for the most part, I believe that this kind of networked music will always be an inferior substitute to music making in person.

How, then, can networked music get 'beyond being there?' I believe that the key is to take advantage of computation in each of the networked machines. Only when each computer is doing something interesting does a network of computers behave like a network of computers instead of unreliable microphone cables with built-in delay lines.

Given a system composed of networked computers, the architectural questions are what each computer's role will be and how they will communicate. The communication is the key element; it should be reliable, accurately timed, sufficiently general in data content and semantics, as simple as possible (but no simpler, as Albert Einstein famously said), as fast as possible, and, ideally, an accepted standard for ease of interoperability. The Open Sound Control ('OSC') protocol was designed to meet these goals.

2. OPEN SOUND CONTROL

Adrian Freed and I developed Open Sound Control, 'a protocol for communication among computers, sound synthesizers, and other multimedia devices that is optimised for modern networking technology', at CNMAT in 1997 (Wright and Freed 1997). We leveraged previous work at CNMAT by Michael Lee and Guy Garnett in experiments with early Internet wide area links: frame relay and ISDN. Roberto Morales used a predecessor to OSC to communicate between a Macintosh running Max and a Sun computer running Prolog in an interactive performance. OSC's design was also informed by CNMAT's work on the ZIPI protocol, especially the Music Parameter Description Language (McMillen, Wessel and Wright 1994).

OSC addressed our need for a network protocol usable for interactive computer music that could run over existing high-speed network technologies such as Ethernet; it was obvious that attempting to develop computer networking technologies within the music community (McMillen, Simon and Wright 1994) could never compete with the computer industry as a whole. Therefore OSC is a 'transport-independent' network protocol, meaning that OSC is only a binary message format, and that data in the OSC format can be carried by any general-purpose network technology. Today most OSC implementations use the main Internet protocols (UDP and TCP/IP) via Ethernet or wireless network connections, but there is also a serial port implementation (Wilson, Gurevich, Verplank and Stang 2003) for microcontrollers. There are also benefits to using OSC even within a single application (Wright, Freed, Lee, Madden and Momeni 2001).

The other motivation in our design of OSC was a desire for generality in the meaning of the messages. MIDI's model of notes, channels, and continuous controllers (Moore 1988) was not adequate to represent, organise or name the parameters of our additive synthesizer (Freed 1995). We therefore chose to use symbolic parameter names rather than parameter numbers, accepting slightly larger message sizes in exchange for expressive, potentially self-documenting names. Likewise, we chose to allow arbitrary organisation of parameters and messages into hierarchical 'address spaces', so that, for example, a message to set the frequency of the fourteenth oscillator in the third additive synthesis voice might be named '/voices/3/osc/14/freq'. The arguments to OSC messages can be strings, 32-bit floating point or integer numbers, arbitrary 'blobs' of binary data, or any of a dozen optional data types.

Part of what makes OSC 'open' is that it comes with no standard set of messages every synthesizer must implement, no preconceptions of what parameters should be available or how they should be organised. Each implementer of OSC can and must decide which parameters to make accessible, what to name them, and how to organise them in a tree structure. This form of openness has led to great creativity among OSC implementations, supporting idiosyncratic, creative software and hardware. This is both a blessing and a curse, because OSC's openness means that it also supports superficial uses of OSC, confusing names, etc. OSC is not the kind of tool that attempts to enforce 'correct' usage.

An important next step for OSC will be an ongoing mechanism to standardise 'schemas', which are OSC address spaces with a formal description of the semantics of each message. This will allow, for example, a standard set of message names, units, etc., for certain kinds of applications, so that multiple OSC implementations with the same function will be controllable the same way. Until this happens, the main users of OSC will continue to be relatively technically proficient programmers with the ability and desire to implement their own parameter mappings.

OSC is 'open' in many other ways. The protocol itself is open, not a secret proprietary format; the specification is available online (Wright 2002). CNMAT's software implementations of OSC are also open in the sense that the source code is available online without charge; we released our first collection of OSC software tools, the 'OSC Kit', in 1998 (Wright 1998) and continue to make our OSC software available from the OSC Web page.

There are now about forty implementations of OSC, including the following:

- Computer programming languages: C, Java, Javascript, Objective C, Perl, PHP, Python, Ruby (Scheme (Lisp)), Smalltalk

Table. Latency limits (in milliseconds) for various musical tasks, and the corresponding distances travelled by sound and light in that amount of time.

Task	Acceptable latency (ms)	Distance sound travels (m)	Distance light travels (km)
Traditional 'real-time' latency limit for real-time interactive music	10	3.43	2,998
Maintaining tempo in ensemble playing	20	6.86	5,996
Playing 'together' for chamber music	50	17.15	14,990

- Web graphics/animation systems: Director, Flash
- Interactive sound synthesis and processing languages: Bidule, Chuck, Common Music CPS, Intakt, Max/MSP, Open Sound World, Pd, SuperCollider, Reaktor, Traktor
- Sensor/gesture capture hardware: EtherSense, Gluion, IpSonLab Kroonde, Lemur, Smart Controller, Teabox, Toaster
- Idiosyncratic control-message-generating software: EyesWeb, Picker, SonART, SpinOSC

Some of these implementations will be described below. The OSC home page (http://cnmat.berkeley.edu/OSC) contains links to all of these implementations, the specification, software downloads, publications, and a list of OSC's application areas.

3. PHYSICAL DISTANCE AND NETWORK LATENCY

Latency, defined as time delay between the sending and receiving of a message, is unavoidable. Even with perfect networking, e.g., a dedicated direct electrical connection between two points, the theoretical maximum message speed is the speed of light, approximately 300 kilometres per millisecond. In practice, computer networking is slower than this, both because of additional network delays from routers, hubs, etc., and because the computers themselves may have substantial delays to receive and process messages, buffer digital audio, etc.

The physics of relativity tells us that simultaneity is relative. Events that seem to occur at the same time in one location may in fact occur at different times to other observers; the 'light cones' of Einstein-Minkowski space-time (Feynman 1970; Pickover 1999) are a beautiful way to visualise this phenomenon, which holds for networked music just as for relativistic physics.[2]

A perennial question is how much latency is acceptable for music. The answer depends on the specific musical task in question. For two people trying to clap a simple interlocking pattern together, round-trip delays greater than 20 ms produce 'significant deceleration' (Chafe, Gurevich, Leslie and Tyan 2004). A recent review(Lago and Kon 2004) reports that human variation when attempting to tap a steady beat is only around 4ms, but that asynchronies of up to 50 ms between supposedly simultaneous notes are common in chamber music. Table 1 summarises these results and indicates the maximum allowable distance for acoustic and (perfect) electronic transmission.

Jitter is defined as the variation of latency; it is musically much more harmful than a fixed delay because it removes fine temporal detail from a performance. Musicians can learn to adapt to a fixed latency (the classic example being pipe organists), but it is impossible to adapt to a randomly varying latency. Psychoacoustic experiments on temporal auditory acuity indicate that we can perceive a timbral change when the relative onset times of pairs of notes differ by as little as 1 ms (Ronken 1970; Henning and Gaskell 1981). This means that a jitter of 1 ms is enough to cause random changes of timbre in some situations. OSC addresses the problem of jitter by providing time-tags indicating when messages are to take effect; this allows the receiver to eliminate jitter by delaying the messages that happen to arrive early, at the expense of increasing the average latency.

Given that the potential forms of musical interaction depend on latency and therefore on physical distance (as shown in table 1), the following examples of OSC's use will be organised into three categories:

- Wide-area networks
- Local networks
- Networks of software running in a single computer

These are not arbitrary classifications, but reflect the realities of the latencies inherent in various physical distances.

4. OSC APPLICATIONS IN WIDE-AREA NETWORKS

To the best of my knowledge, the first OSC-based wide-area network project was the 1997 piece *Points of Presence* by the seminal computer networking band

---

[2] See, for example, http://physics.syr.edu/courses/modules/LIGHTCONE/minkowski.html or http://www.davidbodanis.com/books/emc2/notes/relativity/lightcones/main.html for visual explanations of the light cone model.

The Hub. This project was 'technically and artistically a failure' (through no fault of OSC) according to composer/technologist Chris Brown, but Brown continued this work and was successful the following year with *Invention #5* (Brown and Bischoff 2004; Brown and Bischoff 2005). These projects used OSC simply as a mechanism for packaging MIDI messages across the Internet, which was certainly not what we had in mind when we invented the protocol. However, this showed us that the protocol and its implementations were sufficiently general and useful that people outside CNMAT could adopt it starting with simple uses.

Georg Hajdu's *Quintet.net* (Hajdu 2005) uses OSC to provide a virtual environment in which five performers (or groups of performers) play together under the control of a conductor. Sound sources include triggering of banks of samples; each player's notes sound at each site. There is a mechanism for real-time music notation, both so that players can see what other players play and so that the conductor can send pre-composed notation to the performers. Quintet.net premiered in 2000 and has since been used for several large projects connecting players in Europe and the USA, including a Munich biennale opera in which this author performed. One aspect of *Quintet.net* that goes 'beyond being there' is that the musical information exchanged between players is strictly notation and synthesised sound, not acoustic sound. So although some performers played MIDI guitars and keyboards while others played flutes and violins through pitch-trackers, all of the sound heard across the network came from the sound world of the samples selected for the piece.

In Randall Packer's, Steve Bradley's and John Young's 'collaborative intermedia work' *Telemusic #1* (Young 2001), visitors to a website interact with Flash controls that affect sound synthesis in a single physical location. The resulting sound is streamed back to the Web users via RealAudio. There is a clear distinction between the event that happens in physical space and the remote, possibly very casual contributions of the Web participants, which would be hard to achieve in person. In addition to being far away, the Web participants as a whole have less 'commitment' to the event; in fact, some of the remote participation came from the Web link usage of people who did not even know they were affecting the music. This certainly is a form of musical collaboration that could not take place in person.

Computer networking is now commonly used to create ephemeral online communities; for example, it is possible to play online chess against a human opponent at any time. I am not aware of any such project based on OSC, but other musical examples include *Auracle* (Freeman, Ramakrishnan, Varnik, Nehaus, Burk and Birchfield 2004) and *FMOL* (Jorda 2002).

5. OSC APPLICATIONS IN LOCAL-AREA NETWORKS

In some local-area network applications of OSC, users interact explicitly with the networked computers. In each of these cases, all of the participants are actually physically present in the same room, and the network is used to create rich interaction in the space rather than to attempt to bring people together across space. At ICMC 2000 in Berlin, a network of about twelve Macintoshes running SuperCollider synthesised sound and changed each others' parameters via OSC, inspired by David Tudor's composition 'Rainforest'. The *Meta-Orchestra* project (Impett and Bongers 2001) is a large local-area network that uses OSC. *Simulus* (http://listen.to/simulus) is a Melbourne-based improvising electroacoustic ensemble that uses OSC over WiFi to synchronise clock and tempo information between SuperCollider and AudioMulch in their live performances (Bencina 2003). *PhopCrop* (http://www.xdv.org/cgi-bin/twiki/view/Xdv/PhopCrop) is a prototype system in which multiple users create and manipulate objects in a shared virtual space governed by laws of 'pseudophysics'; each object has both a graphical and sonic representation. *Grenzenlose Freiheit* (http://www.grenzenlosefreiheit.de) is an interactive sound installation using OSC with wirelessly networked PDAs as a sound control interface for the audience.

In the remaining local-area applications in this section, OSC-networked computers handle subtasks of a larger system, even if the network *per se* is not apparent in the final product. Not only does the combination of computers provide more processing power, but it allows for *heterogeneous* systems containing different types of computers, for example, a Macintosh for sound synthesis and a Linux PC to interface with sensors. (In fact, the original need that OSC met at CNMAT was to control our additive synthesizer, running on a Silicon Graphics (SGI) workstation, from a Macintosh running Max.) Although this may seem a mere implementation detail and aesthetically irrelevant, it is important because this kind of local networking allows for more complex systems to be built faster and with less effort, thereby encouraging increased technical and artistic experimentation.

In the *Tgarden* project (Ryan and Salter 2003; Wei, Visell and MacIntyre 2003), visitors in a space collectively affect the synthesised sound indirectly through physical interaction with sensor-equipped objects such as special clothing and large balls. Three different projects at the University of Illinois at Urbana-Champaign (Garnett, Jonnalagadda, Elezovic, Johnson and Small 2001; Goudeseune, Garnett and Johnson 2001; Garnett, Choi, Johnson and Subramanian 2002) are based on systems consisting of real-time 3D spatial tracking of a physical object, processed by one processor that sends OSC to a

Macintosh running Max/MSP for sound synthesis and processing. *Listening Post* (Hansen and Rubin 2002) is a networked multimedia art installation based on representing conversations in Internet chat rooms on a large number of video monitors and also with sonification via a ten-channel speaker system; it was displayed at the Whitney museum of American Art. A local network of four computers handles text display, text-to-speech, sound synthesis, and coordination of all these elements; all of the components of the system communicate with OSC.

6. SECOND-GENERATION OSC IMPLEMENTATIONS

OSC has now reached 'critical mass'; it is easy enough to implement and there are enough implementations with enough functionality that it is now often easier to add OSC to a system to take advantage of existing OSC-based tools than to re-implement the functionality of those tools. This has created a sort of ecology of OSC-based software and hardware, because people know they can make something that solves a small problem, and people will use OSC to connect it to, e.g., their favourite interactive sound synthesis environment. Here the form of the computer networking is not so important as the *social* network of OSC developers who support each other to implement these tools.

OSC is now the protocol of choice for general-purpose hardware sensor interface boxes for music, because the protocol is established and easy to use, and the bandwidth and latency of the networks that can carry OSC are much better than the MIDI alternative. Examples include IRCAM's EtherSense (Fléty, Leroy, Ravarini and Bevilacqua 2004), Sukandar Kartadinata's Gluion (http://www.glui.de/prod/gluion.html), La Kitchen's Kroonde and Toaster (http://www.la-kitchen.fr/kitchenlab/kitchenlab.html), Angelo Fraietta's Smart Controller (Fraietta 2005), and the Teabox (Allison and Place 2004). In all of these cases, the developers worked only on the problem of gestural input, trusting that by producing OSC messages, their products would be usable for a variety of applications.

Developers of 'alternate' musical controllers often implement OSC for similar reasons, so that once they have solved the mechanical and electrical problems of building a device, they can just output OSC and do (or leave their users to do) the remaining work of mapping and sound generation in a general-purpose music programming/synthesis environment. The Lemur (http://www.jazzmutant.com) is a commercially available LCD screen touch-panel capable of sensing multiple fingers at once. The MATRIX ('Multipurpose Array of Tactile Rods for Interactive expression') (Overholt 2001; Overholt 2002) is a 12 × 12 array of spring-loaded rods with continuous displacement sensing, designed to be performed with the hands. The Graphonic (Overholt 2004) consists of a large writing surface (currently a plate of plexiglass) which both takes input from real-time pen movement and outputs sound by being vibrated by a tactile sound transducer. The Gyrotyre (Sinyor and Wanderley 2005) is a hand-held spinning wheel equipped with numerous sensors to detect rate of spinning, orientation, wobbling, etc. Both the SoniMime (Fox and Carlile 2005) and the Beat Boxing gloves (Lugo and Jack 2005) consist of accelerometers mounted on the hands, but with differing metaphors and sound mappings. OROBORO (Carlile and Hartmann 2005) is a two-person controller implementing a 'haptic mirror' in which each performer's 'actuated' hand receives force feedback from the other performer's 'sensed' hand.

The addition of sensors to acoustic musical instruments is yet another domain in which OSC eases project development and opens up many options for gesture mapping and sound processing. Examples include detecting fret position and pluck direction on a sitar (Kapur, Davidson, Cook, Driessen and Schloss 2004), and the addition of buttons, force-sensitive resistors, and loudspeakers to a tuba (Cáceres, Mysore and Treviño 2005).

In the same spirit, OSC has also enabled people to write idiosyncratic software that has obvious musical potential but does not contain any built-in mapping to sound. Not only does this allow programmers to implement their novel ideas without writing sound synthesis code, it also encourages users to develop their own ways of mapping the data to sound. *EyesWeb* (Camurri, Hashimoto, Ricchetti, Ricci, Suzuki, Trocca and Volpe 2000; Camurri, Mazzarino and Volpe 2003) is a camera-based motion detection system including a software library for recognising expressive gestures; it has been used with OSC for video analysis projects including conducting (Kolesnik and Wanderley 2004) and a therapeutic immersive environment for autistic children (Timmermans, Wolferen, Newland and Kunath 2004). Woon Yeo's *SonART* (Yeo, Berger and Lee 2004) is an OSC-enabled image layering program similar to PhotoShop; users can send OSC messages to *SonART* to control placement of images, transparency, etc., and also *SonART* sends OSC messages containing information such as the colour values of the pixel currently under the pointer. IXI Software's *Picker* is similar, generating OSC messages based on the colour values of four positions on the screen as recorded and streaming video are blended and mixed with still images. IXI software's SpinOSC (Magnusson 2005) implements a virtual world of rotating objects with controllable location, size, rotation speed, etc., that generates OSC messages based on the states of the objects. PTL (Henry 2004) is a sequencer for graphical scores; the user draws curves and other shapes on a time axis and also specifies the mapping from these to OSC messages.

Version 3 of the SuperCollider real-time audio synthesis programming language (McCartney 2002) has a client/server architecture, in which the high-level interpreted programming language communicates to a separate low-level DSP program with OSC. This has allowed the creation of alternate interfaces to the sound synthesis server, such as the graphical, Java-based SCREAM (Leahy 2004).

7. CONCLUSIONS AND FUTURE DIRECTIONS

OSC is clearly useful for a wide variety of networking applications. Because OSC allows each implementer to design an address space instead of coming with an *a priori* set of parameter names, it is easily adaptable to situations never envisioned by the designers. Use of OSC over wide-area networks ranges from simple transmission of performance information to allow people to play together at a distance, to more sophisticated and/or innovative forms of long-distance interaction. In local-area networks, some uses of OSC involve explicit interaction with multiple networked computers, while other uses result in a single system whose components (be they individual computers, programs, or even software modules) use OSC for ease of interconnectivity. We can expect to see more projects using OSC, more implementations, and more completeness in existing implementations.

By the time this article is published, CNMAT will have unveiled a new OSC website that can be maintained collaboratively by the entire OSC community. This should encourage the creation of improved documentation and the consolidation of OSC code and other developer resources, as well as provide a forum in which it is easy to announce and document projects, software, hardware, publications and performances related to OSC.

The future of the OSC protocol itself will be determined by a collection of working groups drawn from the OSC developers' community. Topics to be addressed include the following:

- Recommended practices for clock synchronisation and/or network jitter attenuation without clock synchronisation
- A standardised query system
- Tunnelling of OSC within XML, MIDI, 8-bit serial, etc.
- Defined identifications for OSC, such as official IANA port numbers and ZeroConf protocol names
- Persistent storage of OSC data
- Development of a formal specification for OSC schemas (address spaces with their semantics) and the establishment of a collection of generally useful schemas.

ACKNOWLEDGEMENTS

Thanks go to Chris Chafe, Adrian Freed, Margaret Schedel, David Wessel and Michael Zbyszynski for their help and support, and to my anonymous reviewers for their insightful critical feedback.

REFERENCES

Allison, J., and Place, T. 2004. Teabox: a sensor data interface system. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 699–701. Miami, FL: ICMA.

Bencina, R. 2003. PortAudio and media synchronisation. *Proc. of the 2003 Australasian Computer Music Conf.*, pp. 13–20. Perth, Australia.

Brown, C., and Bischoff, J. 2004. Indigenous to the Net: early network music bands in the San Francisco Bay area. http://crossfade.walkerart.org/brownbischoff

Brown, C., and Bischoff, J. 2005. Computer Network Music Bands: a history of the League of Automatic Music Composers and The Hub. In A. Chandler and N. Neumark (eds.) *At a Distance: Precursors to Art and Activism on the Internet*. Cambridge, MA: MIT Press.

Cáceres, J. P., Mysore, G. J., and Treviño, J. 2005. SCUBA: The Self-Contained Unified Bass Augmenter. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 38–41. Vancouver, Canada.

Camurri, A., Hashimoto, S., Ricchetti, M., Ricci, A., Suzuki, K., Trocca, R., and Volpe, G. 2000. EyesWeb: toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal* **24**(1): 57–69.

Camurri, A., Mazzarino, B., and Volpe, G. 2003. Analysis of expressive gestures in human movement: the EyesWeb expressive gesture processing library. *Proc. of the 2003 XIV Colloquium on Musical Informatics.* Firenze, Italy.

Carlile, J., and Hartmann, B. 2005. OROBORO: a collaborative controller with interpersonal haptic feedback. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 250–1.Vancouver, Canada.

Chafe, C., Gurevich, M., Leslie, G., and Tyan, S. 2004. Effect of time delay on ensemble accuracy. *Proc. of the 2004 Int. Symp. on Musical Acoustics.* Nara, Japan.

Chafe, C., Wilson, S., Leistikow, R., Chisholm, D., and Scavone, G. 2000. A simplified approach to high quality music and sound over IP. *Proc. of the 2000 COST G-6 Conf. on Digital Audio Effects (DAFX-00).* Verona, Italy.

Feynman, R. P. 1970. *Feynman Lectures On Physics.* Reading, MA: Addison Wesley Longman.

Fléty, E., Leroy, N., Ravarini, J.-C., and Bevilacqua, F. 2004. Versatile sensor acquisition system utilizing Network Technology. *Proc. of the 2004 Int. Conf. on New Interfaces for Musical Expression*, pp. 157–60. Hamamatsu, Japan.

Fox, J., and Carlile, J. 2005. SoniMime: movement sonification for real-time timbre shaping. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 242–3. Vancouver, Canada.

Fraietta, A. 2005. The Smart Controller Workbench. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 46–9. Vancouver, Canada.

Freed, A. 1995. Bring your own control additive synthesis. *Proc. of the 1995 Int. Computer Music Conf.*, pp. 303–6. Banff, Canada: ICMA.

Freeman, J., Ramakrishnan, C., Varnik, K., Nehaus, M., Burk, P., and Birchfield, D. 2004. Adaptive high-level classification of vocal gestures within a networked sound instrument. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 668–71. Miami, FL: ICMA.

Garnett, G. E., Choi, K., Johnson, T., and Subramanian, V. 2002. VirtualScore: exploring music in an immersive virtual environment. *Proc. of the 2002 Symp. on Sensing and Input for Media-Centric Systems (SIMS)*, pp. 19–23. Santa Barbara, CA.

Garnett, G. E., Jonnalagadda, M., Elezovic, I., Johnson, T., and Small, K. 2001. Technological advances for conducting a virtual ensemble. *Proc. of the 2001 Int. Computer Music Conf.*, pp. 167–9. Habana, Cuba.

Goudeseune, C., Garnett, G., and Johnson, T. 2001. Resonant processing of instrumental sound controlled by spatial position. In *Proc. of the 2001 CHI '01 Workshop on New Interfaces for Musical Expression (NIME'01).* Seattle, WA: ACM SIGCHI.

Hajdu, G. 2005. Quintet.net: an environment for composing and performing music on the Internet. *Leonardo* **38**(1): 23–30.

Hansen, M., and Rubin, B. 2002. Listening Post: giving voice to online communication. In *Proc. of the 2002 Int. Conf. on Auditory Display.* Kyoto, Japan.

Henning, G. B., and Gaskell, H. 1981. Monaural phase sensitivity measured with Ronken's paradigm. *Journal of the Acoustical Society of America* **70**: 1,669–73.

Henry, D. 2004. PTL, a new sequencer dedicated to graphical scores. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 738–41. Miami, FL: ICMA.

Hollan, J., and Stornetta, S. 1992. Beyond being there. *Proc. of the 1992 ACM Conf. on Human Factors in Computing Systems*, pp. 119–25.

Impett, J., and Bongers, B. 2001. Hypermusic and the Sighting of Sound – a Nomadic Studio Report. *Proc. of the 2001 Int. Computer Music Conf.*, pp. 459–62. Habana, Cuba: ICMA.

Jorda, S. 2002. FMOL: toward user-friendly, sophisticated new musical instruments. *Computer Music Journal* **26**(3): 23–39.

Kapur, A., Davidson, P., Cook, P., Driessen, P., and Schloss, W. A. 2004. Digitizing North Indian performance. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 556–63. Miami, FL: ICMA.

Kolesnik, P., and Wanderley, M. 2004. Recognition, analysis and performance with expressive conducting gestures. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 572–5. Miami, FL: ICMA.

Lago, N. P., and Kon, F. 2004. The quest for low latency. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 33–6. Miami, FL: ICMA.

Lazzaro, J., and Wawrzynek, J. 2001. A case for network musical performance. *Proc. of the 2001 Int. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pp. 157–66. Port Jefferson, NY.

Leahy, M. 2004. SCREAM – SuperCollider Resource for Electro-Acoustic Music. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 79–81. Miami, FL: ICMA.

Lugo, R., and Jack, D. 2005. Beat Boxing: expressive control for electronic music performance and musical applications. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 246–7. Vancouver, Canada.

Magnusson, T. 2005. ixi Software: the interface as instrument. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 212–15. Vancouver, Canada.

McCartney, J. 2002. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal* **26**(4): 61–8.

McMillen, K., Simon, D., and Wright, M. 1994. A summary of the ZIPI network. *Computer Music Journal* **18**(4): 74–80.

McMillen, K., Wessel, D., and Wright, M. 1994. The ZIPI music parameter description language. *Computer Music Journal* **18**(4): 52–73.

Moore, F. R. 1988. The dysfunctions of MIDI. *Computer Music Journal* **12**(1): 19–28.

Overholt, D. 2001. The MATRIX: a novel controller for musical expression. In *Proc. of the 2001 CHI '01 Workshop on New Interfaces for Musical Expression (NIME'01).* Seattle, WA: ACM SIGCHI.

Overholt, D. 2002. Musical mapping and synthesis for the MATRIX interface. *Proc. of the 2002 Symp. on Sensing and Input for Media-Centric Systems (SIMS)*, pp. 7–10. Santa Barbara, CA.

Overholt, D. 2004. The Sonic Scanner and the Graphonic Interface. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 189–92. Miami, FL: ICMA.

Pickover, C. A. 1999. *Time: A Traveler's Guide.* Oxford: Oxford University Press.

Ronken, D. 1970. Monaural detection of a phase difference between clicks. *Journal of the Acoustical Society of America* **47**: 1,091–9.

Ryan, J., and Salter, C. 2003. TGarden: wearable instruments and augmented physicality. *Proc. of the 2003 Int. Conf. on New Interfaces for Musical Expression*, pp. 87–90. Montreal.

Sinyor, E., and Wanderley, M. M. 2005. Gyrotyre: A dynamic hand-held computer-music controller based on a spinning wheel. *Proc. of the 2005 Int. Conf. on New Interfaces for Musical Expression*, pp. 42–5. Vancouver, Canada.

Timmermans, H., Wolferen, G. V., Newland, P., and Kunath, S. 2004. MEDIATE: key sonic developments in an interactive installation for children with autism. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 201–4. Miami, FL: ICMA.

Wei, S. X., Visell, Y., and MacIntyre, B. 2003. TGarden Media Choreography System. Atlanta, GA, Graphics, Visualization and Usability Center, Georgia Institute of Technology. http://www.gvu.gatech.edu/people/sha.xinwei/topologicalmedia/papers/Sha-MacIntyre02.pdf

Wilson, S., Gurevich, M., Verplank, B., and Stang, P. 2003. Microcontrollers in Music HCI Instruction: reflections on our switch to the Atmel AVR Platform. *Proc. of the 2003 Int. Conf. on New Interfaces for Musical Expression*, pp. 24–9. Montreal.

Wright, M. 1998. Implementation and performance issues with OpenSound Control. *Proc. of the 1998 Int. Computer Music Conf.*, pp. 224–7. Ann Arbor, Michigan: ICMA.

Wright, M. 2002. OpenSound Control Specification. http://www.cnmat.berkeley.edu/OSC/OSC-spec.html

Wright, M., and Freed, A. 1997. Open Sound Control: a new protocol for communicating with sound synthesizers. *Proc. of the 1997 Int. Computer Music Conf.*, pp. 101–4. Thessaloniki, Hellas: ICMA.

Wright, M., Freed, A., Lee, A., Madden, T., and Momeni, A. 2001. Managing complexity with explicit mapping of gestures to sound control with OSC. *Proc. of the 2001 Int. Computer Music Conf.*, pp. 314–17. Habana, Cuba.

Yeo, W. S., Berger, J., and Lee, Z. 2004. SonART: A framework for data sonification, visualization and networked multimedia applications. *Proc. of the 2004 Int. Computer Music Conf.*, pp. 180–4. Miami, FL: ICMA.

Young, J. P. 2001. Using the Web for live interactive music. *Proc. of the 2001 Int. Computer Music Conf.*, pp. 302–5. Habana, Cuba.

Young, J. P., and Fujinaga, I. 1999. Piano master classes via the Internet. *Proc. of the 1999 Int. Computer Music Conf.*, pp. 135–7. Beijing, China: ICMA.