

2001: Problems and Prospects for Intimate Musical Control of Computers

David Wessel and Matthew Wright

1 Introduction

When asked what musical instrument they play, there are not many computer music practitioners who would respond spontaneously with “I play the computer.” Why not? In this report we examine the problems associated with the notion of the computer as musical instrument and the prospects for their solution.

Here at the onset it would be useful to consider some of the special features that computer technology brings to musical instrumentation. Most traditional acoustic instruments such as strings, woodwinds, brass, and percussion place the performer in direct contact with the physical sound production mechanism. Strings are plucked or bowed, tubes are blown, and surfaces are struck. Here the performer’s gesture plays a direct role in exciting the acoustic mechanism. With the piano and organ the connection between gesture and sound is mediated by a mechanical linkage and in some modern organs by an electrical connection. But the relation between the gesture and the acoustic event remains pretty much in what one might call a *one gesture to one acoustic event* paradigm.

When sensors are used to capture gestures and a computing element is used to generate the sound, a staggering range of possibilities become available. Sadly but understandably, the electronic music instrument industry with its insistence on standard keyboard controllers maintains the traditional paradigm. Musical instruments and their gestural interfaces make their way into common use or not for a variety of reasons most of which are social in character. These more sociological aspects

D. Wessel

Center for New Music and Audio Technologies, Department of Music (CNMAT),
University of California, Berkeley, Berkeley, CA, USA

M. Wright (✉)

Center for Computer Research in Music and Acoustics (CCRMA),
Stanford University, Stanford, CA, USA
e-mail: matt@ccrma.stanford.edu

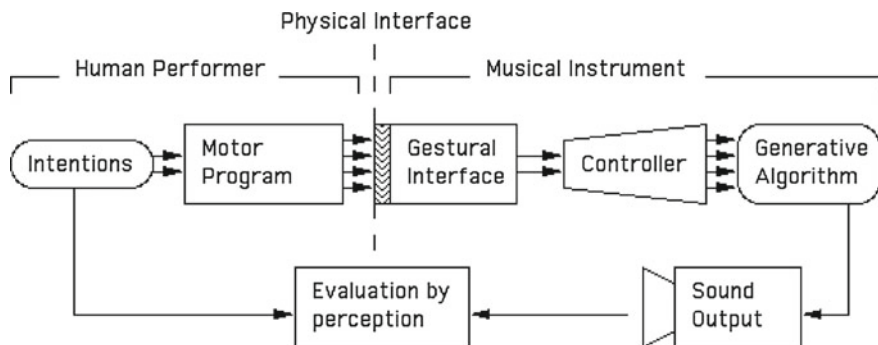


Fig. 1 Conceptual framework for NIME research and development

like the development or not of a repertoire for the instrument are beyond the scope of this paper. Here we will concentrate on factors such as ease of use, potential for development of skill, reactive behavior, and coherence of the cognitive model for control.

In Fig. 1 we provide a conceptual framework for our controller research and development. Our human performer has intentions to produce a certain musical result. These intentions are communicated to the body’s sensorimotor system (“motor program”). Parameters are sensed from the body at the gestural interface. These parameters are then passed to controller software that conditions, tracks, and maps them to the algorithms that generate the musical material. Admittedly this diagram is schematic and incomplete. One aspect that is not well captured by it is the way in which performers’ intentions are elaborated upon by discovery of new possibilities afforded by the instrument. Experimental and otherwise exploratory intentions are certainly dear to the authors. We find that this albeit schematic framework allows us to view the roles of human motor learning, controller mapping, and generative software as an overall adaptive system (Lee and Wessel 1992).

Unlike the *one gesture to one acoustic event* paradigm our framework allows for generative algorithms to produce complex musical structures consisting of many events. One of our central metaphors for musical control is that of *driving* or *flying* about in a space of musical processes. Gestures move through time as do the musical processes.

2 Low Entry Fee with No Ceiling on Virtuosity

Getting started with a computer-based instrument should be relatively easy but this early stage ease-of-use should not stand in the way of the continued development of musical expressivity. Most of the traditional acoustical musical instruments are not

easy to play at first but do afford the development of a high degree of musicality. On the other hand many of the simple-to-use computer interfaces proposed for musical control seem, after even a brief period of use, to have a toy-like character and do not invite continued musical evolution.

Is the low entry fee with no ceiling on virtuosity an impossible dream? We think not and argue that a high degree of control intimacy can be attained with compelling control metaphors, reactive low latency variance systems, and proper treatment of gestures that are continuous functions of time. With the potential for control intimacy assured by the instrument, musicians will be much more inclined to the continued development of performance skill and personal style.

3 Latency Requirements for Control Intimacy

Few practitioners of live performance computer music would deny that low latency is essential. Just how low is the subject of considerable debate. We place the acceptable upper bound on the computer's audible reaction to gesture at 10 milliseconds (ms) and the systems described in this paper provide for measured (Freed et al. 1997) latencies nearer 7 ms.

Low variation of latency is critical and we argue that the range of variation should not exceed 1 ms. Grace-note-generated timbres as in flams can be controlled by percussionists with temporal precision of less than 1 ms. This is accomplished by controlling the relative distance of the sticks from head during the stroke. Timbral changes in the flams begin to become audible when the variations in the time between the grace note and the primary note exceed 1 ms. Psychoacoustic experiments on temporal auditory acuity provide striking evidence for this criterion (Ronken 1970; Henning and Gaskell 1981). As we will argue below, prospects for the solution to the latency variation problem can be resolved by using time tags or by treating gestures as continuous signals tightly synchronized with the audio I/O stream.

4 Discrete Event Versus Continuous Control

MIDI is a discrete event protocol. MIDI events turn notes on and off and update changes in controller values. MIDI events are almost never synchronized with digital audio samples. Furthermore, MIDI provides no mechanism for atomic updates. Chords are always arpeggios and even when MIDI events are time tagged at the input of a synthesizer they arrive as a sequence. Moore (Moore 1988), McMillen (McMillen 1994), and Wright (Wright 1994) provide numerous examples of the dysfunction of MIDI. Much of this dysfunction is addressed by the Open Sound Control (OSC) protocol described below.

Many musical gestures are continuous functions of time and should be treated as such, for example, the position along the string of a finger on a violinist's left hand. The new generation of software synthesis systems such as Max/MSP, SuperCollider, PD, and Open Sound World (OSW) provide for multi-rate signal processing. In these programming environments it is quite natural to treat gestures with a sample-synchronous signal processing approach. CNMAT's connectivity processor (Freed et al. 2000) described below provides a mechanism for getting continuous gestures into the computer in a manner that is very tightly synchronized with the audio sample stream. With this system we demonstrate a significant increase in control intimacy.

5 Open Sound Control (OSC)

Open Sound Control is a discrete event protocol for communication among controllers, computers, sound synthesizers, and other multimedia devices that is optimized for modern networking technology. Entities within a system are addressed individually by an open-ended URL-style symbolic naming scheme that includes a powerful pattern matching language to specify multiple recipients of a single message. We provide high-resolution time tags and a mechanism for specifying groups of messages whose effects are to occur simultaneously. Time tags allow one to implement a scheduling discipline (Dannenberg 1989) that reduces jitter by trading it for latency.

OSC's use of symbolic names simplifies controller mapping and its hierarchical name space helps in the management of complexity. There is also a mechanism for dynamically querying an OSC system to find out its capabilities and documentation of its features. OSC has been integrated into Max/MSP (by Matthew Wright), Csound (by Stefan Kersten and Nicola Bernardini) SuperCollider (by James McCartney), and OSW (by Amar Chaudhury). It has been used in a variety of contexts involving controllers.¹

6 A Programmable Connectivity Processor

The conventional approach for communicating gesture and sound to real-time performance systems is to combine a microcontroller or DSP chip with A/D, D/A converters and a network interface such as a MIDI serial controller. We have developed an alternative, more flexible approach that supports scalable implementations from a few channels of audio and gestures to hundreds of channels.

Our new system to address computer music and audio connectivity problems is based on integrating all digital functions on a single field programmable gate array (FPGA). All functions are determined by compiling high-level hardware

¹See <http://www.cnmatic.berkeley.edu/OSC> for more detail and downloadable OSC software.

descriptions (in VHDL) into FPGA configurations. This approach allows the considerable investment in developing the interface logic to each peripheral to be easily leveraged on a wide variety of FPGA's from different vendors and of different sizes. Since FPGA's are now available in sizes greater than a million gates, entire DSP and microcontrollers can also be integrated if required.

We have developed and tested VHDL descriptions for processing serial audio data for the SSI, S/PDIF, AES/EBU, AES-3, and ADAT industry standards. For gestures that are continuous we sample at submultiples of the audio rate and have VHDL modules for multichannel 8-bit, 12-bit and 16-bit A/D converters. We also provide modules for multiple MIDI input and output streams. Although such descriptions have been developed for proprietary systems, this library of modules represents the first complete, independent suite available in VHDL.

This suite makes possible some unusual cross-codings such as embedding gestural data in audio streams, increasing temporal precision by exploiting isochronous data paths in the control processor. We sample continuous gestural signals at a submultiple of the audio sampling rate, multiplex the channels, and represent them as audio input signals. This allows us to get gestural signals into our software with the same low latencies as audio input, and guarantees that gestural and audio input signals will be tightly synchronized.

A novel module of particular importance in portable computer-music performance systems implements Fast Ethernet from the hardware layer up through IP to the UDP protocol of TCP/IP. Because of the importance of Internet performance, Fast Ethernet implementations are extremely reliable and finely tuned on all modern operating systems.

A key feature of the connectivity processor is the analog subsystem for continuous gesture acquisition. We currently provide for 32 channels of analog-to-digital conversion. Voltage ranges of the converters are selectable as are the sampling rates which are constrained to be integer divisions of the audio rate and the appropriate antialiasing filter cutoff frequencies.

Our system combines VHDL connectivity modules that multiplex 8 channels of bidirectional audio, MIDI, S/PDIF and transduced gestures into UDP packets which are exchanged with a portable computer using new, customized ASIO drivers in Max/MSP (Avizienis et al. 2000).

7 Musical Control Structures for Standard Gestural Controllers

Throughout history, people have adapted whatever objects were in their environment into musical instruments. The computer industry has invested significant resources in creating broadly available, low cost gestural controllers without any musical application in mind; thoughtful adaptation of these controllers for music is a fruitful yet overlooked route.

We find the latest incarnations of the venerable digitizing tablet (a.k.a. “artist’s tablet”) very interesting for musical control. Tablets offer accurate and fast absolute position sensing of cordless devices in three dimensions. Additionally, pressure, orientation, tilt, and rotation estimates are available. The tablet we use allows for simultaneous sensing of two devices, usually one in each hand. This rich, multidimensional control information can be mapped to musical parameters in a variety of interesting ways.

The most direct kind of mapping associates a single synthesis parameter with each control dimension, for example, vertical position controlling loudness, horizontal position controlling pitch, etc. This kind of mapping proved to be musically unsatisfying, exhibiting the toy-like characteristic that does not allow for the development of virtuosity.

More interesting interfaces define regions of the tablet associated with particular behaviors. For example, one region might consist of a grid providing access to a large palette of musical material, while other regions represent musical processes that can operate on selected musical material. Repeating rhythmic cycles can be represented graphically on a region of the tablet, and sonic events can be placed at particular time points within the cycle (Wright and Wessel 1998).

We have created software in the Max/MSP environment that we use to develop control structures for the two-handed digitizing tablet. Examples include navigation in timbre space, multidimensional synthesis control, note stream synthesis, and emulations of the gestures of strumming, plucking, and bowing strings. We also developed an interactive musical installation that uses two joystick controllers.

8 Some Custom Controllers

At CNMAT we have developed applications for variety of custom controllers; these include Don Buchla’s Thunder and Lightning,² Tactex controllers,³ Force Sensing Resistor (FSR) technology (Rovan et al. 1997), and Piezo electric sensors in conjunction with percussion (Campion 1999). We currently have research projects underway that exploit a key feature of the previously described connectivity processor—namely the synchronization of control signals with the audio I/O stream. These projects include an organ keyboard with continuous sensing of each key position (Freed and Avizienis 2000), a variety of micro-accelerometer projects, and new FSR devices.

In addition we have developed new sensor systems for multidimensional string motion (Freed and Isvan 2000) and have made some advances in extracting control signals from vocal sounds.

²<http://www.buchla.com>.

³The Tactex company is now out of business; see (Jones 2001) and <https://web.archive.org/web/20100317144002/http://www.tactex.com> for more information about their then-revolutionary MTC Express multi-touch controller product.

9 Metaphors for Musical Control

As suggested in the introduction, metaphors for control are central to our research agenda. We have found the work of George Lakoff and his collaborators (Lakoff and Johnson 1999; Lakoff and Nunez 2000) on embodied cognition to be particularly applicable. They argue that abstract concepts like time and space and even the loftier concepts of mathematics are grounded in sensorimotor experience. We now present some of the metaphors that have inspired the development of our controller software.

9.1 *Drag and Drop*

The drag and drop metaphor is well known to users of the Apple Macintosh. An object is selected, picked up, and dropped upon a process. This is a natural application of Lakoff's movement and container metaphors. Our drag and drop system has been extended to the problem of the control of musical processes using the pen and tablet interface. Musical material is selected and then dropped onto a musical process.

One of the most critical features of any musical control system is a silencer, a mechanism that allows the performer to gracefully shut down a musical process. To this end we have the performer place the pen on the process and using a circular motion like the traditional copy editor's cursive delete sign the process is silenced at a rhythmically appropriate point in time. Other interfaces use the "eraser" end of the pen to silence processes.

9.2 *Scrubbing and Its Variants*

Sinusoidal models allow arbitrary time-scale manipulation without any change in pitch or spectral shape. We have built "scrubbing" interfaces for the tablet in which one dimension of the pen's position on the tablet maps to the time index of a sinusoidal model. Moving the pen gradually from left to right at the appropriate rate results in a resynthesis with the original temporal behavior, but any other gesture results in an alteration of the original. This interface allows a performer to play more arbitrary musical material, while preserving the fine continuous structure of the original input sounds. This kind of interface has been used in live performance contexts with classical Indo-Pakistani singing (Wessel et al. 1998), trombone samples with expressive glissandi, and saxophone quartet material.

Other dimensions of the tablet sensing data, e.g., pressure, tilt, and vertical position, can be mapped to synthesis parameters such as loudness and spectral shape.

9.3 *Dipping*

In the “dipping” metaphor the computer constantly generates musical material via a musical process, but this material is silent by default. The performer controls the volume of each process, e.g., using a poly-point touch-sensitive interface with the pressure in each region mapped to the volume of a corresponding process. Other gestural parameters control other parameters of the musical processes.

An advantage of this metaphor is that each musical event can be precisely timed, regardless of the latency or jitter of the gestural interface. Once a given process is made audible, its rhythm is not dependent on the performer in an event-by-event way.

This kind of interface is most satisfying when there are multiple simultaneous musical processes of different timbres, allowing the performer to orchestrate the result in real-time by selecting which processes will be heard.

Acknowledgements Special thanks to Rimas Avizienis, Adrian Freed, and Takahiko Suzuki for their work on the connectivity processor and to Gibson Guitar, DIMI, and the France Berkeley Fund for their generous support.

Unsolved Problems and Continuing Prospects for Intimate Musical Control of Computers

Matthew Wright

When we wrote this article in 2001 (and expanded it into a fuller article the following year (Wessel and Wright 2002)) there seemed to be a small community of people making music interactively with computer-based instruments of their own design, and as far as I could tell all of them were friends with David Wessel. We jumped on the opportunity to share our thoughts on goals, difficulties, and techniques for achieving a quality of intimacy that we strongly desired in our own instruments and found lacking in most of the triggering-based instruments then readily available. We were excited that the esteemed ACM SIGCHI was to some degree turning its attention to questions of musical interaction, and welcomed the opportunity to shape this discussion by injecting our own values and methods. We had no idea this small fringe gathering would develop into the NIME conference and community as it now exists.

Today I question the relationship between NIME and mainstream CHI. Although computer music has always benefitted from and indeed depended on advances in EE and computer science generally not made with music in mind, it does not seem that the concerns of NIME have had much influence on CHI in particular. In the other direction, much of the apparent influence of CHI on NIME has been shoe-horning the inherently creative and exploratory artistic process of designing musical instruments into the objectively quantifiable mindset of evaluating task performance.

Today the minimum viable NIME paper seems to be a slightly novel combination of technologies that is demonstrably good at something, with only the best making theoretical contributions to guide the exploration of the infinite-dimensional space of possible instruments with ethical consideration of the potential of the human performer. Perhaps if we could better capture and theorize the experiences of skilled practitioners as contributions to knowledge, then themes of performance, expertise, experience, artistry, and control could become more prominent to the betterment of both communities.

Low entry fee with no ceiling on virtuosity, aka Low Floor High Ceiling (a phrase popularized by Seymour Papert that reached us via Brian Harvey), is perhaps the most-cited idea in this paper, though of course it is much easier to demonstrate the floor than the ceiling. We disparaged low-ceiling NIMEs as “toy-like;” I now regret this phrase and prefer to frame the question as how to design high-ceiling toys. Toys and their enclosing adaptive systems can harm or help the continued musical evolution of kids of all ages, inviting and rewarding human motor learning in many ways. Gurevich et al. rightly question the importance of an instrument’s inherent properties in light of people’s seemingly inexhaustible creativity in adapting their music making to even the simplest of NIMEs (Gurevich et al. 2010). Nevertheless, I would still argue that their beeping button box’s and other excellent toys’ remarkably high ceilings (e.g., variety of musical results) can be predicted and partially attributed to low latency/jitter and musicians’ drive to apply continuous control even to an instrument that presents a discrete-only interface.

Latency raises the floor and jitter lowers the ceiling; indeed these have been a major theme of NIME, questions that are today almost required of proposed NIMEs and their component technologies. A few papers focus exclusively on these issues. We arrived at our 10 ± 1 ms criterion through subjective personal experiences using instruments with various measured latencies and at our flam timbre justification through thought experiment; to my knowledge the literature still lacks better empirical justification for these engineering constraints. In the context of graphical objects following a user’s finger on a touch screen, 1ms total latency is clearly⁴ much better than 10ms (Jota et al. 2013). Perceptual and cognitive studies of various musical activities and results under various latency/jitter conditions could also result in better models for how skilled performers adapt and possibly shed light on development of musicality in general.

Wessel’s SLABS⁵ represent the fullest embodiment of this paper’s goals and ideas, with extremely low latency, OSC (whose huge impact on the NIME community is addressed in the commentaries to the chapter “2003: Open Sound Control: State of the Art 2003”), a representation of gestural data as audio signals, a direct descendent of the programmable connectivity processor isochronously scanning a poly-point touch-sensitive gestural interface, and the “dipping” metaphor in its most satisfying form. Many NIMEs centering on a digitizing tablet (Zbyszynski et al. 2007) have

⁴The demonstration video <http://www.youtube.com/watch?v=vOvQCPLkPt4> is compelling.

⁵http://cnmat.berkeley.edu/user/david_wessel/blog/2009/01/15/slabs_arrays_pressure_sensitive_touch_pads.

joined my continuing work, which includes recent experiments spatially arranging sound grains in 3D according to analyzed acoustical properties, then flying about this space, still a rich and powerful metaphor. Scrubbing remains vital to my tablet interfaces, but the analysis portion of sinusoidal modeling is still difficult and likely to produce unsatisfactory results without extensive fussing, hampering widespread adoption of this technique. Granular scrubbing seems more promising, with the high dimensional control space of window size, rate, etc. both a challenge and opportunity.

Understandably, the number of cases of developing a deep, longstanding, and fruitful connection to a NIME is far less than the number of NIMEs that have been proposed. One could argue that modular synthesis' current popularity, with many performers developing intimate relationships with their personal instruments, is because they "treat gestures with a sample-synchronous signal processing approach"; unfortunately it is still very common for NIMEs to sample or delay control signals much more slowly and erratically, to our detriment. I would like to see and perform additional research into these control signals (e.g., their spectral and phase properties) in sophisticated musical contexts.

The Sempiternal Quest for Intimacy

Sergi Jordà

2001 was the year the first NIME took place. A small workshop inside the huge CHI conference in Seattle, I was lucky enough to be one of the first dozen NIME participants ever. There, I had the honor to meet some of the pioneers and gurus of the burgeoning "live computer music world" (for the lack of a better term, "NIME" not yet existing before the workshop): Max Mathews, Bill Verplank, Perry Cook, Joe Paradiso were all there, giving presentations that would enlighten and guide the younger ones such as myself. If I remember correctly, David Wessel did not attend the workshop—I would have to wait one or two more years for meeting him in person and becoming good friends—and Matt Wright gave their presentation.

Wessel and Wright's paper introduces one of the concepts that would come to recur most frequently in our NIME community: that of "low entry fee with no ceiling on virtuosity." The paper starts also with one sentence, that while less popular, still constitutes one of my favourite NIME quotes; one, which I subsequently often borrowed and even used for starting my Phd thesis: "When asked what musical instruments they play, there are not many computer music practitioners who would respond spontaneously with 'I play the computer.'" But the paper encompasses much more than some popular NIME quotes; it does, in fact, contain almost too much. It ranges from conceptual topics, such as control intimacy and its importance for attaining a hypothetical instrumental virtuosity, to fully technical solutions such as the OSC protocol, which had been specifically designed for addressing the low latency and the high resolution requirements previously identified by the authors as being essential for attaining the aforementioned intimate control. On top of this, it introduces

some musical interfaces and controllers which were then being designed, revamped or customised at CNMAT, and it concludes with a discussion on some of the control metaphors that inspired the development of these interfaces. If that wasn't enough, some essential knowledge (then novel and original) is dropped here and there, such as the empirically observed toy-like behaviour of "one-to-one mappings" (although the authors do not employ that term), and their unsuitability for the development of virtuosity.

Considering its generous loquacity, it is very likely that by today's standards, a paper like this would have been rejected for publication: "lack of clear objective and structure, and lack of evidence and proof for most of the authors' statements." While its overwhelming flood of ideas was tamed for the extended and matured journal version published a year later (Wessel and Wright 2002), and which probably constitutes a more cohesive reading, I would like to stress the fact that urgency to communicate and effervescence, were indeed the trademarks of most of the papers presented at this first NIME workshop in 2001. Much knowledge and know-how had been accumulated and was pressurized, eager to be shared with a sympathetic audience! But this specific paper, together with Perry Cook's "Principles for Designing Computer Music Controllers" (Cook 2001), which was also presented at the 2001 workshop, clearly stand as two of the early pillars or compendiums of NIME knowledge. Together with two even earlier seminal publications by Pressing (Pressing 1990) and Ryan (Ryan 1992), they still constitute the first four introductory must-read papers in any NIME-related course I have ever organized!

In 2001, NIME-knowledge was necessarily based on personal experience and empiricism. This type of knowledge is not easily replicable, not always demonstrable, not very scientific in sum, but it is not less valid for these reasons. Fifteen years later, the NIME community has matured and a growing number of university courses are being taught (Jordà and Mealla 2014). Yet, creating DMIs is still, in many respects, very similar to creating music. It involves a great deal and variety of know-how and technicalities, while at the same time, as in music, there are no inviolable laws. Indeed, in his contemporary aforementioned paper (Cook 2001), Cook sceptically and romantically affirmed: "musical interface construction proceeds as more art than science, and possibly this is the only way it can be done?" Because I tend to agree with him, I believe we should never forget the previous successes and the failures of experienced practitioners. In that sense, it comes as no surprise that the first tentative and informal—but still very valid—NIME design and conceptual frameworks were proposed by experienced digital luthiers, performers and educators, and David Wessel, regretfully not longer with us, definitely excelled in all these categories.

References

- Avizienis, R., Freed, A., Suzuki, T., & Wessel, D. (2000). Scalable connectivity processor for computer music performance systems. In *Proceedings of the International Computer Music Conference* (pp. 523–526).
- Campion, E. (1999). Playback. Musical composition. Paris: CNMAT and IRCAM. Retrieved from http://www.edmundcampion.com/project_playback/.
- Cook, P. R. (2001). Principles for designing computer music controllers. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Seattle, WA.
- Dannenbergh, R. (1989). Real-time scheduling and computer accompaniment. In M. V. Mathews & J. R. Pierce (Eds.), *Current Directions in Computer Music Research* (pp. 225–261). Cambridge, MA: MIT Press.
- Freed, A., & Avizienis, R. (2000). A new music keyboard featuring continuous key-position sensing and high-speed communication options. In *Proceedings of the International Computer Music Conference* (pp. 515–516) San Francisco, CA.
- Freed, A., & Isvan, O. (2000). Musical applications of new, multi-axis guitar string sensors. In *Proceedings of the International Computer Music Conference* (pp. 543–546) San Francisco, CA.
- Freed, A., Avizienis, R., Suzuki, T., & Wessel, D. (2000). *Scalable connectivity processor for computer music performance systems*. Presented at International Computer Music Conference. Berlin, Germany.
- Freed, A., Chaudhary, A., & Davila, B. (1997). Operating systems latency measurement and analysis for sound synthesis and processing applications. In *Proceedings of the 1997 International Computer Music Conference* (pp. 479–481), San Francisco, CA.
- Gurevich, M., Stapelton, P., & Marquez-Borbon, A. (2010). Style and constraint in electronic music instruments. *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp. 106–111). Sydney: Australia.
- Henning, G. B., & Gaskell, H. (1981). Monaural phase sensitivity measured with Ronken's paradigm. *Journal of the Acoustical Society of America*, 70(6), 1669–1673.
- Jordà, S., & Mealla, S. (2014). A methodological framework for teaching, evaluating and informing NIME design with a focus on expressiveness and mapping. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, London, UK.
- Jota, R., Ng, A., Dietz, P., & Wigdor, D. (2013). How fast is fast enough? A study of the effects of latency in direct-touch pointing tasks. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 2291–2300). Paris, France.
- Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. New York: Basic Books.
- Lakoff, G., & Núñez, R. E. (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being*. New York: Basic Books.
- Lee, M. A., & Wessel, D. (1992). Connectionist models for real-time control of synthesis and compositional algorithms. In *Proceedings of the 1992 International Computer Music Conference* (pp. 277–280), San Francisco, CA.
- McMillen, K. (1994). ZIPI - Origins and Motivations. *Computer Music Journal*, 18(4), 47–51.
- Moore, R. F. (1988). The dysfunctions of MIDI. *Computer Music Journal*, 12(1), 19–28.
- Pressing, J. (1990). Cybernetic issues in interactive performance systems. *Computer Music Journal*, 14(1), 12–25.
- Ronken, D. (1970). Monaural detection of a phase difference between clicks. *Journal of the Acoustical Society of America*, 47(4), 1091–1099.
- Rovan, J. B., Wanderley, M. M., Dubnov, S., & Depalle, P. (1997). Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In A. Camurri (Ed.), *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop* (pp. 68–73). Genoa: Associazione di Informatica Musicale Italiana.
- Ryan, J. (1992). Effort and expression. In *Proceedings of the 1992 International Computer Music Conference* (pp. 414–416), San Francisco, CA.

- Wessel, D., Wright, M., & Khan, S. A. (1998). Preparation for improvised performance in collaboration with a khyal singer. In *Proceedings of the 1998 International Computer Music Conference* (pp. 497–503), Ann Arbor, MI.
- Wessel, D., & Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3), 11–22.
- Wright, M., & Wessel, D. (1998). An improvisation environment for generating rhythmic structures based on north Indian ‘tal’ patterns. In *Proceedings of the 1998 International Computer Music Conference* (pp. 125–128), Ann Arbor, MI.
- Wright, M. (1994). A comparison of MIDI and ZIPI. *Computer Music Journal*, 18(4), 86–91.
- Zbyszynski, M., Wright, M., Momeni, A., & Cullen, D. (2007). Ten years of tablet musical interfaces at CNMAT. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, New York, NY.

A NIME Reader

Fifteen Years of New Interfaces for Musical Expression

Jensenijs, A.R.; Lyons, M.J. (Eds.)

2017, XLII, 485 p. 220 illus., 154 illus. in color.,

Hardcover

ISBN: 978-3-319-47213-3