The Dual-Analog Gamepad as a Practical Platform for Live Electronics Instrument and Interface Design

Christopher Ariza
MIT, Music and Theater Arts
4-246, 77 Massachusetts Ave.
Cambridge, MA 02139
ariza@mit.edu

ABSTRACT
This paper demonstrates the practical benefits and performance opportunities of using the dual-analog gamepad as a controller for real-time live electronics. Numerous diverse instruments and interfaces, as well as detailed control mappings, are described. Approaches to instrument and preset switching are also presented. While all of the instrument implementations presented are made available through the Martingale Pd library, resources for other synthesis languages are also described.

Keywords
Controllers, live electronics, dual-analog, gamepad, joystick, computer music, instrument, interface

1. INTRODUCTION
While countless innovative computer music controllers and musical interfaces have been newly built or repurposed from existing hardware, many of these devices end up being used only by a small group of performers or deployed in highly idiosyncratic or non-portable ways. Often this is the result of the controller being dependent on particular software or the skills of a particular performer; or the controller being prohibitively expensive or lacking enough controls or expressive range for enduring musical interest. While the resulting diversity of controllers is a source of great richness and creativity in the design of new instruments, musical interfaces, and music, in some cases this diversity prohibits the productive reuse and refinement of practical and portable implementations.

Game controllers have been re-purposed for music-making for decades: the Mattel Power Glove, first released in 1989, generated much interest in the early 1990s with the "glove" Max object [13]. Contemporary practitioners of live electronics continue to effectively use off-the-shelf joysticks, gamepads, and other hand-held controllers (e.g., the Nintendo Wii and Microsoft Kinect). The dual-analog gamepad, however, offers practical advantages over the many alternatives. While not a new hardware design, a multiplicity of mappings and interface designs, as demonstrated here, offers a new approach to design plurality with a common controller.

While novel controllers have attracted significant research interest, the dual-analog has rarely been isolated as offering particular advantages. Griffiths describes the use of a dual-analog controller for live-coding, illustrating different mappings for Betablocker and Al Jazari, and emphasizing the value of using ring menus to select programming elements [4]. The focus here, however, is not on live-coding control, but on direct performance and control of real-time instruments.

This paper demonstrates the practical benefits and performance opportunities of the dual-analog gamepad. In addition to describing the important features of this controller, numerous diverse designs for instruments, implemented in Pd and made available in the open-source Martingale Pd library, will be demonstrated. The diversity of compelling and expressive instrument designs is evidence of the musical opportunities of this controller.

NIME'12, May 21-23, 2012, University of Michigan, Ann Arbor.

2. THE DUAL-ANALOG CONTROLLER
The term "dual-analog gamepad" is here used to describe a number of similar gamepads, each distinctive in featuring two small, thumb-operated joysticks (sometimes called thumbsticks). A typical dual-analog gamepad is illustrated in Figure 1 and Figure 2.

Figure 1. Top view of a dual-analog controller.

Figure 2. Side view of a dual-analog controller.

The first dual-analog controller, called the DualAnalog (later the DualShock), was first released in 1997 as a revision to the original Sony PlayStation control pad. The original PlayStation control pad, while including the two sets of four top and four back buttons, lacked the pair of joysticks. Since the original DualAnalog, a number of variants have been released by many manufacturers, with numerous models still widely available. Logitech (as well as many other manufacturers) has released a number of USB-based models, such as the Dual Action Game Pad, Gamepad F310, and others. Supplied with a USB interface, these controllers can be easily plugged into a computer. Widely available and mass-produced, prices for these controllers are around $20.

For a small, cheap, and compact controller, the dual analog offers access to a large number of controls. The two joysticks provide four continuous parameters with self-centering positions, eight discrete button controls (sometimes labeled buttons 1 through 8), and (on most models), a five-position (four-button) centering directional-pad (d-pad, or hat-switch) control. Most models feature two additional smaller top buttons, often labeled "select" and "start" (here called buttons 9 and 10.) Thus, 18 independent controls are available.

The dual-analog controller is a good value: obtaining 14 buttons and 2 joysticks in a USB-ready package, such as is offered by the dual analog, is impossible by other means at a similar price. For example, similar small joysticks, purchased as individual components for use in custom-designed circuits (perhaps connected to a USB microcontroller) cost between $2 and $4 each. While highly configurable software for creating touch-screen interfaces is available at low cost (e.g., TouchOSC [hexler.net/software/touchosc]), such controllers offer no tactile response similar to the feel of a self-centering joystick or a physical button, and are nearly impossible to be played without actually looking at the touch-screen.

While experience is not required, many dual-analog operators will already be familiar with the dual-analog hardware, thus offering an opportunity to "leverage expert technique" [1] in gaming toward expressive musical control. Further, USB gamepads do not have the complications, expense, and other problems associated with "custom" music controllers [2]. Finally, while ergonomic, these controllers require effort to master the simultaneous control of two joysticks and many buttons; this effort, for many, is a desirable feature, as "effort is closely related to expression in the playing of instruments" [9].

3. HARDWARE CHARACTERISTICS AND MAPPING ARCHETYPES

A number of detailed mappings of the dual-analog controller will be presented in the subsequent section, implementing a wide variety of instruments, interfaces, and control paradigms. Before these details, a few common characteristics of the dual-analog hardware, and mappings suggested by this hardware, are presented.

The ability to control four continuous parameters simultaneously and orthogonally via two thumb-joysticks offers great expressive flexibility. As will be shown subsequently, the most expressive and "playable" musical parameters are assigned to the joysticks. For each X and Y axis, at least 256 discrete values (8 bits) are delivered from most hardware. This is twice the range of common MIDI continuous controllers, which typical deliver only 128 discrete values (7 bits of data, due to the use of 1 bit per octet to distinguish command messages). As the joysticks are self-centering, in some mappings only half of the 256-value range is used per parameter, with the center position being zero, and extreme positions providing two different 128-value ranges.

Perhaps the most common mapping of the joystick Y axis is for amplitude control. For example, in the implementations described below, the center position is set to a floating-point value scalar of 0.7; all the way down is 0.0, all the way up is 1.25. Floating-point values are then taken to an exponent of 4 to provide appropriate logarithmic loudness control. Such a mapping provides a resting position base-level below 1.0, a wide range below this value for smooth fading in and out, and an additional range above 1.0 for accents.

A common mapping of the joystick X axis is for frequency control, such as adjusting the cutoff frequency of a low-pass filter. Again, the center position is set to a default value (such as 4,000 Hz); different ranges and scaling factors can be used for above and below this center position, such as scaling the right range from 4,000 to 18,000 Hz, and the left range from 4,000 to 200 Hz. Exponents can be applied to achieve approximately equal distance-increment-per-octave movement.

As previously suggested, having a default center position within a range of values for a joystick axis is not always desirable. For example, a joystick might be used to provide the magnitude of modulation, or the amount of a signal sent to an additional processor. In these cases, the center position can represent zero, and two different parameters can be controlled on a single axis with each direction away from the center. For example, with an X-axis control, movement to the right might scale the amount of amplitude modulation from 0 to 1, while movement to the left might scale the amount of frequency modulation from 0 to 1. Such dual-mappings on the same axis assert that the two controls are independent and can only be applied one at a time.

The four main top and four back buttons (buttons 1 through 8) provide eight easily accessible triggers, capable of rapid repetition and simultaneous access by fingers of both hands. (These buttons will be referred to as the "eight main" buttons.) Common usage of these buttons is for triggering primary events, with an implied ordering from low to high possible by the suggested sequence from top to back. A significant shortcoming of these buttons when used as event or sample triggers is the lack of velocity sensitivity. This can be mitigated to some degree through careful use of joystick-controlled parameters.

The d-pad control can be considered as a type of discrete joystick: while it offers four directional button controls (conventionally labeled left, right, up, and down), releasing any button before pressing another sends a specific discrete message, a center-position trigger, that cannot be otherwise transmitted. Common usage of the d-pad is for simply selecting temporary controls, such as routing a signal to a processor or (when combined with an appropriate visual display) navigating a grid of option selections.

4. INSTRUMENT AND INTERFACE DESIGNS

In order to demonstrate the diversity of instruments and interfaces available with the dual analog, a brief overview of eleven different instrument designs, broken down into six categories, will be described. For representative instruments, a table of mappings is presented. Each instrument features numerous dynamic performance parameters exposed through the dual-analog controller. The other parameters that are needed to define the instrument, but are not exposed with the controller, make up a "preset," or a collection of static parameters, potentially including value ranges for the dynamic, real-time parameters. Methods of switching between presets and instruments are discussed.

All of the instruments presented are based on existing implementations available using Pure Data (Pd) [6] and the Martingale Pd library (code.google.com/p/martingale). These instruments were developed over many years for solo and ensemble live-electronics performance, with the explicit goal of offering expressive, easy to learn, and "durable" performance tools. Durability here refers to designs that resist failure; that is, instruments that work every time, that can be played consistently, and that can reliably be played by touch, even in the dark. The names of all abstractions in this library are proceeded by an "mg" to distinguish the Martingale library namespace.

4.1 Polyphonic Triggered Samples

From the earliest drum machines, such as the Linn Drum, to modern controllers and instruments in the tradition of the Akai MPC series, the presentation of an instrument as a bank of samples that can be independently triggered is common. While percussive samples are often employed for such instruments, other types of samples, such as ambient sounds or synthetic textures, can be used as well. An important design decision for such instruments is whether, once triggered, the sound continues to the end of the sample, or ends as soon as the key-press is released.

The mgSynthBuffer8 is a dual-analog implementation of a similar instrument. Here, the eight main buttons each trigger an independent sample. As is common with the instruments presented here, Y1 controls amplitude, X2 controls the cutoff frequency of a low-pass filter, and Y2 controls the resonance of the low-pass filter. Joystick control of such simple but expressive parameters permits altering amplitude envelopes, producing tremolo-like effects, modifying timbre, and even performing wah-wah-like effects. Additionally, the d-pad is used as a toggle to send the signal to various delay and reverb processors. The parameters for this instrument are summarized in Table 1.

Table 1. Mappings for a polyphonic sample-playback instrument (mgSynthBuffer8)

Joysticks	X1	Not connected
	Y1	Amplitude
	X2	Low-pass filter cutoff frequency
	Y2	Low-pass filter resonance
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Sample 1
	2	Sample 2
	3	Sample 3
	4	Sample 4
Back buttons	5	Sample 5
	6	Sample 6
	7	Sample 7
	8	Sample 8

4.2 Polyphonic Pitched Synthesizers and Noise Generators

Triggering pitched synthesized tones, such as might be done from a conventional piano-style keyboard, is another common instrumental need. The pitched material might be based on common waveforms (such as saw-tooth wave tables) or various types of noise.

The mgSynthSaw is a polyphonic subtractive synthesizer. Each of the eight main buttons trigger a trio of slightly detuned oscillators. The combined tone is then mixed and processed with a low-pass filter (again using X2 for cutoff frequency and Y2 for resonance). The pitches of an eight-note scale are mapped to each of the main buttons; the tuning of this scale is encoded in the preset. The d-pad left and right buttons here are used to shift the scale up or down an octave, extending the range of the eight-note scale. The d-pad up button returns the scale to its initial octave. An independent ADSR envelope is applied to each voice; the parameters for this envelope are encoded in the preset and are not exposed in the controller. Additionally, the depth of vibrato (low-frequency frequency modulation) is made available on the X1 axis. The parameters for this instrument are summarized in Table 2.

Table 2. Mappings for a polyphonic pitched instrument (mgSynthSaw)

Joysticks	X1	Vibrato depth
	Y1	Amplitude
	X2	Low-pass filter cutoff frequency
	Y2	Low-pass filter resonance
D-pad	L	Octave down
	R	Octave up
	U	Initial octave
	D	Reverb
Top buttons	1	Scale step 1
	2	Scale step 2
	3	Scale step 3
	4	Scale step 4
Back buttons	5	Scale step 5
	6	Scale step 6
	7	Scale step 7
	8	Scale step 8

Numerous related instruments can be created with a similar design. For example, mgSynthNoiseFilter uses, instead of oscillators, a bank of eight different types of noise. Some of these noise-types are band-filtered, others are various approaches to low-frequency noise. For this instrument, X1 and X2 control the cutoff frequencies of a high-pass and a low-pass filter, respectively. Used in tandem, the two controls provide an intuitive way to dynamically shape the width and range of a pass-band applied to broad-spectrum noise timbres. The parameters for this instrument are summarized in Table 3.

Table 3. Mappings for a polyphonic noise instrument (mgSynthNoiseFilter)

Joysticks	X1	High-pass filter cutoff frequency
	Y1	Amplitude
	X2	Low-pass filter cutoff frequency
	Y2	Amplitude modulation depth
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Noise source 1
	2	Noise source 2
	3	Noise source 3
	4	Noise source 4
Back buttons	5	Noise source 5
	6	Noise source 6
	7	Noise source 7
	8	Noise source 8

4.3 Polyphonic Looping Samples

The remaining instruments offer more unconventional interfaces and instrument models. One example is an instrument made from samples in a looping buffer, where main buttons simply open and close an envelope, and expressive nuance is achieved by varying the rate of looping (from reverse [-1], to slow [.2], to fast [20]), the size of the window looped (from the whole sample to a small sample window), and the overall amplitude.

The mgSynthBufferLoop8 instrument implements this approach. Here, each of the main eight buttons turn on or off a different looping sample. The Y2 axis controls the rate of looping (with 1 at the center position) and the X2 axis controls the window size (where the center position is the complete window). X1 controls the low-pass filter cutoff frequency, and Y1 serves the now familiar role of amplitude control. The parameters for this instrument are summarized in Table 4.

Table 4. Mappings for a polyphonic looping sample instrument (mgSynthBufferLoop8)

Joysticks	X1	Low-pass filter cutoff frequency
	Y1	Amplitude
	X2	Sample window end boundary
	Y2	Sample loop rate
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Sample 1
	2	Sample 2
	3	Sample 3
	4	Sample 4
Back buttons	5	Sample 5
	6	Sample 6
	7	Sample 7
	8	Sample 8

4.4 Pulse Stream Gates

The previous instruments support a direct, real-time control of individual events. An alternative approach is to control larger-level gestures or event clusters. Such designs require more data to be encoded in the preset, but tend to require from the performer less rigorous timing and nuanced performance of the discrete controls. One type of more gesture-based interface, here called a pulse-stream gate, is specialized for triggering and controlling percussive layers. Main buttons open and close envelopes, exposing streams of pulses.

The pulse stream gates implemented as mgSynthBufferPulse8 and mgSynthBandpassNoisePulse use each of the main eight buttons to select different sound sources. Rather than triggering a specific sound, each button opens an envelope on a sound source that is automatically and repeatedly triggered at a constant rate. The rate of triggering for each of the eight sources is different, and is based on a preset-defined multiple of a high-rate metronome. The layering of multiple sound sources at different rates is made significantly more interesting by using, for the triggering of each event, a probabilistic determination. Based on the away-from-center position of X1, the probability of triggering each pulse-determined event is scaled from one (every pulse) to zero (no pulses). Thus, through the selection of sound sources and X1 modulation, a variety of rhythmic presentations is possible. Additional timbral control with a low-pass filter is provided with X2 and Y2. Various sound sources might implement this model; mgSynthBufferPulse8 and mgSynthBandpassNoisePulse differ in the use of fixed samples and band-pass filtered noise, respectively. Table 5 summarizes the parameters of mgSynthBufferPulse8.

Table 5. Mappings for a sample-based, pulse-stream gate instrument (mgSynthBufferPulse8)

Joysticks	X1	Event trigger probability
	Y1	Amplitude
	X2	Low-pass filter cutoff frequency
	Y2	Low-pass filter resonance
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Sample 1
	2	Sample 2
	3	Sample 3
	4	Sample 4
Back buttons	5	Sample 5
	6	Sample 6
	7	Sample 7
	8	Sample 8

4.5 Duophonic Nonstandard Synthesizers

Whereas, as demonstrated previously, the availability of eight independent buttons suggests the polyphonic control of independent sound sources, monophonic-style control offers valuable expressive opportunities. For example, monophonic pitch or timbre selection offers portamenti or glissandi between tones. While the eight main buttons of the dual analog might offer a wider range of monophonic selections, here the top and bottom sets of four buttons are assigned to two monophonic voices, creating a duophonic instrument.

Two instrument models are offered as examples of nonstandard duophonic instruments. The mgSynthGranular-Sample is an instrument using sample-based granular synthesis [8]. Each of the banks of four buttons (half of the main eight buttons) is used to control sounds derived from a single sample. Each button, based on a preset specification, defines a time range within the assigned sample. Thus, each button can select a different range within the sound source. The selected sound source is used by eight or more simultaneous layers, using probabilistic parameter variation to create typical granular density. Expressive variation in the granular sound source is provided by X1, X2, and Y2, which can be used to dynamically vary the grain playback rate, the grain window size (or duration), and the grain density (muting grain events with a probabilistic function). Y1 again functions as amplitude control. Granular synthesis typically requires definition of many parameters; reducing these parameters for practical, real-time control is a signifcant challenge, but one that promotes the identification of the most expressive and musical features of the synthesis system. As shown here and elsewhere, the dual-analog, in forcing the isolation of the most expressive parameters, encourages the design of more playble and durable instruments. The parameters for this instrument are summarized in Table 6.

Table 6. Mappings for a duoponic granular instrument (mgSynthGranularSample)

Joysticks	X1	Grain playback rate
	Y1	Amplitude
	X2	Grain window size
	Y2	Probabilistic grain density
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Sample 1 window range 1
	2	Sample 1 window range 2
	3	Sample 1 window range 3
	4	Sample 1 window range 4
Back buttons	5	Sample 2 window range 1
	6	Sample 2 window range 2
	7	Sample 2 window range 3
	8	Sample 2 window range 4

The mgSynthStochWave instrument employs the dynamic generation of probabilistic square-segment wave tables. Based loosely on the stochastic synthesis techniques of Xenakis [14, 10], this instrument uses a single wave table for each of two voices. The waveform in the table is drawn based on probabilistic parameters directly controlled by X2 and Y2: the waveform segment size (or the size of each horizontal segment with a randomly selected amplitude) and the waveform redraw rate (or the frequency with which new segments are drawn), respectively. Each monophonic voice is assigned four pitches. X1 here serves as the cutoff frequency of a low-pass filter, and Y1 controls amplitude. While based heavily on randomness, providing real-time control of the ranges of this randomness create an expressive and musical instrument. The parameters for this instrument are summarized in Table 7.

Table 7. Mappings for a duophonic stochastic synthesizer (mgSynthStochWave)

Joysticks	X1	Low-pass filter cutoff frequency
	Y1	Amplitude
	X2	Probabilistic waveform segment size
	Y2	Waveform segment redraw rate
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Voice 1 pitch 1
	2	Voice 1 pitch 2
	3	Voice 1 pitch 3
	4	Voice 1 pitch 4
Back buttons	5	Voice 2 pitch 1
	6	Voice 2 pitch 2
	7	Voice 2 pitch 3
	8	Voice 2 pitch 4

4.6 Duophonic Phrase Stream Gates

A final interface archetype extends the gesture-level control of the pulse stream gate to patterns of both rhythm and pitch control, and similarly employs a duophonic, two-voice model. Rather than only encoding the pulse-rate and sound source for each gesture, these instruments encode complete pitch and rhythm patterns, and assign a unique, looping pattern to each of the four buttons for each voice. Pressing a button thus presents different looped phrases; through dynamically varying button selections, hybrid phrases are created and recombined. A variety of monophonic instruments can be used the synthesis the voices.

The mgSynthSawSequenceDuo, mgSynthPafSequenceDuo, and mgSynthBufferSequenceDuo all employ this approach with different sound sources and real-time controls. The mgSynthPafSequenceDuo instrument, for example, uses a phase-aligned formant (PAF) synthesis instrument [7], where X2 and Y2 control the center frequency and pulse width, respectively, of the synthesizer voice. X1 controls low-pass filter cutoff frequency and Y1 controls amplitude. The parameters for this instrument are summarized in Table 8; similar instruments have related mappings. While instruments such as these provide a meta-control of musical phrases, rather than direct control of musical events, the continuous parameters made available on the joysticks provide the performer with the opportunity of invigorating these gestures and phrases, providing additional nuance, phrasing, and contrast.

Table 8. Mappings for a duophonic phrase-stream instrument (mgSynthPafSequenceDuo)

Joysticks	X1	Low-pass filter cutoff frequency
	Y1	Amplitude
	X2	PAF center frequency
	Y2	PAF pulse width
D-pad	L	Slow delay
	R	Fast delay
	U	Not connected
	D	Reverb
Top buttons	1	Voice 1 pattern 1
	2	Voice 1 pattern 2
	3	Voice 1 pattern 3
	4	Voice 1 pattern 4
Back buttons	5	Voice 2 pattern 1
	6	Voice 2 pattern 2
	7	Voice 2 pattern 3
	8	Voice 2 pattern 4

5. INSTRUMENT AND PRESET SWITCHING

The models described above are presented as singular, static instruments. Such an approach may be effective for limited musical contexts, but performers will benefit from the ability to quickly switch instruments and presets during live performance. The approach embraced here is to provide a broad sonic palette, not through the use of one complex instrument, but through the quick switching of numerous simple instruments.

Throughout the Martingale library, dual-analog instruments are switched using buttons 9 and 10 (the “select” and “start” buttons), selecting an active instrument on a bidirectional circular array or ring menu of all instruments. As many as eight different dual-analog instruments are simultaneously loaded and made active. A Pd abstraction routes the control data: values from the dual-analog are routed to each of the instruments, one at a time, based on a value that is incremented or decremented by buttons 9 and 10, wrapping around from, for example, 8 to 1 or 1 to 8. This type of control, similar to the ring menu selector used in many computer games and demonstrated in Griffiths [4], provides quick access to each instrument.

The 9 and 10 buttons can also be used to advance presets for individual instruments. Martingale implements this by triggering a change in present when both buttons 9 and 10 are pressed simultaneously. Presets are incremented within a set range, and mapped around to the first preset after exceeding the range of all presets. Thus, a second ring menu (albeit monodirectional) is used for preset control.

By exposing both instrument and preset switching directly on the dual-analog controller, a vast grid of sound and control options is available from a single piece of hardware. This approach permits a performer to call up a large range of sound sources without recourse to the computer trackpad, mouse, or keyboard. While visual feedback regarding the current instrument and/or preset is beneficial, it is not required. Such a model lets the performer play a single controller away from the computer, without even looking at the computer or the controller, providing a compelling musical performance while avoiding many of the common criticisms of live-electronics performance. Dating from early performances of The Hub, audience members have been baffled by the disparity between the sounds of live electronics and the static, screen-focused performance of many musicians; Gresham-Lancaster, for example, relates how audience members said they “looked like a bunch of air traffic controllers” [3]. Trueman similarly describes how a common complaint “leveled at laptop performers” is that “they were all just checking their email” [12]. As Griffiths states, “use of a gamepad allows the performer to leave the confines of the screen and keyboard, and means the computer is less of a central point compared to other laptop performances” [4]. Leaving the screen and keyboard is an advantage common to many controllers; the dual-analog, however, offers a pleasing and compact visual presentation, letting the audience clearly see the performer playing a physical instrument.

6. LANGUAGE AND PLATFORM SUPPORT

For most operating systems, the USB interface of the dual analog presents itself as a standard human interface device (HID). Various computer-music languages need only expose the serial data-stream of the controller to begin mapping data to musical applications.

The Martingale Pd library includes, in addition to all of the instruments described above, Pd abstractions for numerous variations of dual-analog controllers and specific platform requirements. For MacOS and Linux users, the Pd "hid" object, created by Hans-Christoph Steiner (at.or.at/hans/pd/hid.html) and distributed as part of Pd-extended, performs correctly for a variety of dual-analog controllers. Actual values returned by the controllers may be in different formats or value ranges. Martingale manages this diversity by creating, for each controller type, abstractions that return values and triggers in uniform and normalized ranges.

Use of dual-analog controllers in Pd on Windows is presently less stable, due to the lack of a robust HID interface. Good solutions are available through alternative "joystick" and "hidin" objects (distributed with Martingale and Pd-extended).

Use of dual-analog controllers in Max/MSP can be easily achieved through the "hi" (human interface) object. SuperCollider provides a similar "HIDDeviceService" object; additional, more specialized game-pad interfaces are also available for SuperCollider (e.g., OSCGamePad at oscgamepad. sourceforge.net). Other computer-music languages are likely to provide similar interfaces. If no native support is available, HID to MIDI or OSC translators, such as junXion (www.steim.org/steim/ junxion_v4.html), may be used.

7. CONCLUSION

The diversity of instruments for which the dual analog can serve as a controller is made clear from the previous examples. All of these instruments are presently available in the Martingale library. The goal of this paper, however, is not just to supply example mappings and interface designs, but to illustrate the benefits of designing instruments for a common, inexpensive, and practical hardware controller, and encourage others to also design instruments and interfaces for this controller. Further, embracing parameter reduction and isolation, as this controller demands, promotes more playable and durable instruments. Dual-analog controllers are literally everywhere; providing interfaces and instruments for these controllers offers many opportunities.

For example, in introducing a new synthesis technique or synthesis language, developers can offer simple interfaces using the dual-analog. Better then a static recording or a GUI-based demonstration, performers using a dual-analog controller can quickly and easily explore the parameters of a new synthesis design.

For students of live electronics, finding a point of entry can be difficult. Instead of trying to explore performance with a keyboard and a mouse, or using an expensive controller that might require extensive and time consuming custom mappings, students can begin immediately making expressive gestures with the dual-analog controller and existing instruments. Providing numerous contrasting designs (such as those provided in Martingale) exposes students to a variety of synthesis techniques and mapping strategies. As the dual-analog controller is likely to be familiar and inviting, such instruments "invite exploration and intuitive approaches to performance ... [to] draw the performer into a kind of performance flow" [12]. Such an approach means that, for these instruments, programmability will not be a curse [1] and the trap of "constant technical improvement means eternal unfamiliarity with the instrument" [5] can be avoided.

Finally, extensive practice on a single type of controller, even with diverse instrument models, lets performers gain a type mastery, avoiding the problem that "the performer doesn't really know how to play his or her instrument, which he or she may just have finished building" [5]. While not esoteric or unusual, dual-analog instruments offer "interfaces that demand practice and force the players to 'break a sweat'" [11].

The dual-analog controller does not met every instrumental or control need: velocity-sensitive buttons are desirable, as are sensors such as accelerometers (which some models do include). However, for around $20, the dual-analog offers easy access to a number of controls rich with potential for designing engaging real-time live-electronics instruments.

8. REFERENCES

[1] Cook, P. 2001. "Principles for designing computer music controllers." In *Proceedings of the Conference on New Interfaces for Musical Expression*.

[2] Fiebrink, R. and G. Wang, P. Cook. 2007. "Don't Forget the Laptop: Using Native Input Capabilities for Expressive Musical Control." In *Proceedings of the Conference on New Interfaces for Musical Expression*, pp. 164–167.

[3] Gresham-Lancaster, S. 1998. "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music." *Leonardo Music Journal* 8:39–44.

[4] Griffiths, D. 2007. "Game Pad Live Coding Performance." In J. Birringer, T. Dumke and K. Nicolai, eds. *Die Welt als virtuelles Environment*. Dresden: TMA Helleraue.

[5] Perkis, T. 2009. "Some Notes on My Electronic Improvisation Practices." In R. T. Dean, ed. *The Oxford Handbook of Computer Music*. Oxford: Oxford University Press, pp. 161–166.

[6] Puckette, M. 2002. "Max at 17." *Computer Music Journal* 26(4):31–43.

[7] Puckette, M. 2007. *The Theory and Technique of Electronic Music*. Hackensack, New Jersey: World Scientific. Available online at http://www.crca.ucsd.edu/~msp/techniques/latest/book-html.

[8] Roads, C. 2002. *Microsound*. Cambridge, Massachusetts: MIT Press.

[9] Ryan, J. 1991. "Some Remarks on Musical Instrument Design at STEIM." *Contemporary Music Review* 6(1):3–17.

[10] Serra, M. 1993. "Stochastic Composition and Stochastic Timbre: GENDY3 by Iannis Xenakis." *Perspectives of New Music* 31(1):236–257.

[11] Smallwood, S. and D. Trueman, P. R. Cook, G. Wang. 2008. "Composing for Laptop Orchestra." *Computer Music Journal* 32(1):9–25.

[12] Trueman, D. 2007. "Why a Laptop Orchestra?" *Organised Sound* 12(2):171–179.

[13] Winkler, T. 1998. *Composing Interactive Music*. Cambridge: MIT Press.

[14] Xenakis, I. 1992. *Formalized Music: Thought and Mathematics in Music*. Indiana: Indiana University Press.