Ha Dou Ken Music: Mapping a joysticks as a musical controller

Gabriel Lopes Rocha , João Teixeira Araújo , Flávio Luiz Schiavoni

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei - UFSJ
São João del-Rei - MG

gbr.cdc@gmail.com, teixeira.araujo@gmail.com, fls@ufsj.edu.br

Abstract. *The structure of a digital musical instrument (DMI) can be splitted up in three parts: interface, mapping and synthesizer. For DMI's, in which sound synthesis is done via software, the interaction interface serves to capture the performer's gestures, which can be mapped under various techniques to different sounds. In this work, we bring videogame controls as an interface for musical interaction. Due to its great presence in popular culture and its ease of access, even people who are not in the habit of playing electronic games possibly interacted with this kind of interface once in a lifetime. Thus, gestures like pressing a sequence of buttons, pressing them simultaneously or sliding your fingers through the control can be mapped for musical creation. This work aims the elaboration of a strategy in which several gestures captured by the interface can influence one or several parameters of the sound synthesis, making a mapping denominated many to many. Buttons combinations used to perform game actions that are common in fighting games, like Street Fighter, were mapped to the synthesizer to create a music. Experiments show that this mapping is capable of influencing the musical expression of a DMI making it closer to an acoustic instrument.*

1 Introduction

The current state of the technology allows relatively inexpensive access to Personal Computers with some reasonable processing capacity. This facilitates the creation of live musical performances through the control of sound synthesis by computers in real time. Researches in the field of Computer Music led to the development of several models of sound synthesis for instrument creation. Such models can be classified between physical models and signal models [1].

Those of the first category are based on the physical and acoustic events related to the sound production of a given instrument [2]. Through the analysis of these instruments it is possible to develop a system of equations to simulate this instrument in a realistic way. Signal models, on the other hand, rely on the spectral representation of the sound and signal processing structures to produce the desired sound [3]. Such models enable the creation of sounds that do not resemble those of existing acoustic instruments.

However, the standard input interfaces used in desktop or laptop computers can be not adequate when searching for expressive control of the synthesis parameters of a digital instrument. Such interfaces were developed from an ergonomic approach and are focused in traditional tasks like typing or drawing. Their goal is to be simple, easy to learn and objective. On the other hand, the role of creativity has great importance in the development of controllers for musical interaction [1]. These controllers are more expressive but also are projected to be used by an expert skilled performer. The learning process for using these interfaces can require a certain amount of effort that can be prohibitive for lay musicians interested into explore digital instruments. This is an important characteristic of a digital musical instrument because it is possible to make a correlation between effort and expressiveness[4].

Beyond the interface and the synthesizer, a digital musical instrument also has a kind of hidden layer also known as mapping. For digital musical instruments, mapping[5, 6, 7] consists of the task of making the connection between control parameters, related to the sensors present at the interface, and synthesizer parameters, related to the chosen sound synthesis method. There are several possible approaches to mapping. Commonly it is designed in a simplistic way, mostly with one-to-one connections between parameters. This practice is contraindicated because it tends to generate an instrument with poor expressive abilities. If we think the expressiveness of an acoustic instruments, since they are governed by physical laws, control methods are often interconnected in a complex way with sound results. A researcher might take this into account in the mapping process when developing a new digital instrument, specially if the goal is not to copy an existing acoustic instrument. The chosen strategy has great impact in achieving an expressiveness close to that of an acoustic instruments.

In our work, we did not aimed to create a new musical interface, but we reused one developed for another context. We used a video game controller, specifically the DualAnalog model originally developed by Sony company for its Playstation console, as a musical interface. This choice had three main reasons. The first is the ease of access at low prices, whether in specialty stores, the Internet or popular markets. The second is the state of ubiquity given to this technology by the strong presence of video games and electronic games in popular culture. Most people already have some experience with this interface or similar, even if not strongly attached into video games. The third is an emotional and aesthetic question. This is an interface that brings to many gamers a lot of good memories. In the authors' experience in using this interface in performance, people tend to be impressed when they see a video game controller being used for musical purposes.

Our interface can be classified as alternate controller, since it is not based on any existing acoustic instrument. This makes the task of mapping more difficult because the possible relationships between control and synthesis parameters are numerous and not obvious. To overcome this difficulty, in this work we suggest to analyze how these interactions are performed in electronic games, besides using them as inspiration for our mapping. In our DMI we have selected the case of arcade-style fighting games such as “Street Fighter” and “The King of Fighters” and the way special attacks are performed on them.

2 Related Works

Matthew Blessing and Edgar Berdahl [8] developed, at Louisiana State University, a control similar to the one presented in this work. Authors used as input interface an Arduino Micro with five levers, where the X and Y position of each lever determines the changes in range and pitch of the sound. As a synthesizer, this work uses a Raspberry Pi running a PureData patch. In addition, the instrument was mounted in an enclosure with input interface, sound synthesizer, mapping and sound box.

In the music department of Stanford University, SookYoung Won, Humane Chan and Jeremy Liu [9] developed an instrument consisting of aligned PVC tubes containing photovoltaic resistors at the bottom of each tube. This project uses a microprocessor as a data processing unit, which transforms the signal sent by the photovoltaic resistors, when triggered by the incidence of light, into a MIDI signal. However, the data coming from the photovoltaic resistors are only considered as notes, varying initially, from note 60 to 72 and then maintaining the MIDI velocity value at 64 as the default for all notes. The project also has the possibility of varying scale, which they call “high” and “low”, being able to play MIDI notes from 48 to 60, and from 72 to 84. After the signal is transformed into a MIDI signal it is sent to the PureData patch, which contains the sound synthesis, and then the sound is generated.

Christopher Ariza [10] also explores the DualAnalog model as a control possibility for musical interaction. He discusses the characteristics of this control, its possibilities and performance opportunities. Practical uses for the button set and mappings applied to the pair of joysticks are presented, such as the use of continuous axes to send MIDI messages and to control the cutoff frequency of a low-pass filter, which are also used in our work. For Ariza, the pair of joystcks carries the most expressive possibility in this type of control. He presented in his work a set of detailed mappings that implement 11 different instruments divided into 6 categories, each with its characteristics and applications. Such implementations are done using a Pd library called Martingale Pd, but the possibility of using other software is discussed. It even features a way to quickly switch between different presets. In this way, the artist will have access to a wide range of instruments for his performance.

Several papers explore the possibility to use the control used on the Nintendo Wii, equipped with motion sensors that allow the capture of more complex gestures, as musical interface [11] [12].

3 The joystick as a DMI

To present the construction of our musical interface, we divided our instrument in three parts and we are presenting it part by part: The interface, the synthesizer and the mapping strategies. Certainly, the presented project reflects several development decisions that can be totally different in a similar project.

3.1 The Interface

Among the various video game controls currently on the market, the popular model of the Sony’s PlayStation, called DualAnalog (later Dualshock). Many variants of its first version were launched by various companies like the Double Shock B-Max Controller, which includes a Universal Serial Bus (USB) connection to be connected to a Personal Computer (PC), used in this work. This type of control can be found easily at low prices and offers a number of attractions if compared with similar interfaces. It is practically impossible to achieve this amount of buttons and analogue levers available for mapping at such a low cost if the option were to build such a circuit electronically.

About the control, it is worth mentioning that it can be used in two modes, with analog mode on or off, selected with the “Analog button”. When the analog mode is disabled, the continuous axes assigned to the levers are not used. Thus, the left joystick controls the discrete axis of the directional buttons and the right joystick can be used to trigger the right-hand buttons. This configuration is interesting because it allows two ways to execute the same command, even if the number of available commands is somewhat limited.

As can be seen in Figure 1, this control has

- (1 to 4) four front buttons (right hand - popularly known as “triangle”, “x”, “square” and “circle”);
- (5 to 8) four upper buttons (“L1”, “L2”, “R1” and “R2”);
- (9 to 12) four directional buttons (left hand - also known as D-pad);
- (13) a start button;
- (14) a select button;
- (15 and 16) two analog joysticks;
- (17 and 18) two buttons pressing the levers;
- (19) an analog mode (on/off) button.

Currently, there are APIs in the Linux operating system that allow interaction with events triggered by video game controls. There is even a Webaudio library that runs this function in a web browser, the gamepad API. In this work, specifically regarding the implementation, the Linux library “joystick.h” was used to interact with this device.

The joystick.h library, available for Linux, allows the programmer to interact with the controller interface. It has a structure that encapsulates the information, whether

Figure 1: Joystick controls used in this work

Table 1: Control outputs and possible values

Buttons	Value
1 to 4 (front)	0 / 1
5 to 8 (upper)	0 / 1
13 - Start	0 / 1
14 - Select	0 / 1
15 - Left Analog	0 / 1
16 - Right Analog	0 / 1
19 - Analog	-
Axes	Value
9 to 10 - D-pad X	-32767 / 0 / +32767
11 to 12 - D-pad Y	-32767 / 0 / +32767
15 - Left Analog X	-32767 to +32767
15 - Left Analog Y	-32767 to +32767
16 - Right Analog X	-32767 to +32767
16 - Right Analog Y	-32767 to +32767

sent by the buttons or sent from the movement of the analog joysticks. In this way, the value of a button can be discrete, meaning only if it has been pressed, released or continuous, indicating the amount of movement of the joystick. It is noteworthy that the directional buttons, although discrete, control two axes, being one horizontal and one vertical, as well as the two joysticks. The events captured by this library is also timestamped and it is possible to capture the time in which each these events occurred. All the buttons and axes with their respective outputs are presented in Table 1.

3.2 The Synthesizer

Two options were considered when defining the synthesizer of our experiment: to use an existing synthesizers or to develop a new synthesizer in an sound programming environment like Pure Data. In this work, it was decided to use a MIDI protocol interface to connect with existing synthesizers, allowing to connect the a type of mapping in different synthesis engines, making the process of creating a instrument more free and interesting.

A MIDI synthesizer has a set of specific and predefined inputs as well as a well-defined communication protocol based on Notes and Dynamics (velocity). Although we find it sufficient for our initial proposal, we understand that this gives us a sparse set of synthesis parameters to work on mapping.

We have a wide variety of compatible synthesizers available for Linux, which makes it possible to explore several results with great ease. In this work, we used the set of synthesizers available in a tool called LMMS - The Linux Multimedia Studio. The LMMS, presented in Figure 2, is a multi-platform digital audio workstation that was widely used in our research.

Apart from a large number of interesting synths, the LMMS has a default MIDI interface to all synths, listed in the table 2, with several different parameters. Thus, it would be possible to use MIDI using control parameters, extrapolating the MIDI note events.

In these synthesizers it was possible to control the pitch, intensity and duration of the note. We can also work with LFO (Low Frequency Oscillator) and ADSR (Attack, Sustain, Decay, Release) envelope, which are common in several synthesizers. As for pitch control is possible to variate it over time in a way that make possible the execution of a glissando. With this, we maintain the advantage of being able to switch easily from synthesizer in order to explore new results.

To integrate the LMMS' synths in our project, we used the ALSA (Advanced Linux Sound Architecture)

Figure 2: Example of LMMS Synthesizer.

Table 2: Parameters accepted by the synthesizers

Parameter	Value
MIDI NOTE	
Note	0 to 127
Velocity	0 to 127
Envelope	
Attack	0 to 127
Duration	0 to 127
Sustain	0 to 127
Release	0 to 127
LFO	
Attack	0 to 127
Velocity	0 to 127
Filter	
Frequency	0 to 127
Resonance	0 to 127
General	
Volume	0 to 127
Pitch	0 to 127

MIDI API. Using this API, we were able to translate the events we captured from the control into MIDI protocol events. We build our prototypes in this way due to the great popularity and simplicity of handling of this protocol, which until today is widely used in the context of musical production.

3.3 1-to-1 mapping

Under the first formulated implementation, a simpler mapping was used to test the control possibilities. As mentioned previously, a 1-to-1 mapping is not indicated because it does not contribute to the search for expressiveness in the interface. However, this implementation served as a starting point for our research. Among those who tested it, the simple fact of turning a video game controller into a musical interface served as a great draw.

In this mapping we use the 8 main buttons, four front and four upper to trigger MIDI notes, organized in an octave, as musical notes. The D-pad serves to determine the speed parameter, where the value 80 corresponds to the neutral position and keeping it pressed under any of the directions can change it. The resulting values increase from the bottom to the top, from left to right. In addition, the left joystick can be used in the same way. The right lever is used to control the volume. Rotating it clockwise will increase it and counterclockwise will decrease it. The buttons pressed when pressing the joysticks serve to raise, with the left, and descend, with the right, an octave.

The start button serves as the Panic Button, it sends a NOTE_OFF for all possible notes to be used in case an error occurs and some note is lost. This implementation is recurrent in all mappings. The table 3 shows this mapping organization.

Table 3: 1-to-1 mapping with main buttons sending Notes and the D-Pad controlling Dynamics

Buttons	Mapped to
Button 1	NOTE 60
Button 2	NOTE 62
Button 3	NOTE 64
Button 4	NOTE 65
Button 5	NOTE 67
Button 6	NOTE 69
Button 7	NOTE 71
Button 8	NOTE 72
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad Left	Velocity 50
D-pad Right	Velocity 100
D-pad Up	Velocity 127
D-pad Down	Velocity 30
D-pad Neutral	Velocity 80
Left Analog	Equals D-pad
Right Analog	Volume

3.4 1-to-many mapping

The 1-to-many mapping grants a macro-level control of the sound event, though it fails to give more detailed control over the parameters composing it [13]. In order to test such characteristic, we implemented the idea of using the D-pad or the left analogue not only to control the dynamics, but to change several characteristics of the resulting sound by pressing the buttons. This time, the use of diagonals was included to expand the possibilities. We started using the Envelope and LFO features present in the synthesizers using the MIDI control API. In the neutral position the sounds have short duration. As the left analog is moved clockwise, the sounds begin to acquire longer duration, lift and strength. The last three positions add the LFO oscillation. Finally, the rest of the control continues with the same functions as the previous mapping. This organization is shown on the table 4.

Table 4: 1-to-many mapping, with the right analogue speed controlling, Envelope and LFO

Buttons	Mapped to
Button 1	NOTE 60
Button 2	NOTE 62
Button 3	NOTE 64
Button 4	NOTE 65
Button 5	NOTE 67
Button 6	NOTE 69
Button 7	NOTE 71
Button 8	NOTE 72
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad	Velocity, Envelope and LFO variation
Left Analog	Equals D-pad
Right Analog	Volume

3.5 Many-to-1 mapping

Advancing a bit more in mapping styles, the idea was to use some “many-to-1” mappings to extend control possibilities. For this purpose, instead of just using the 8 main buttons to trigger MIDI notes, it was also implemented the possibility of making combinations between them.

In total, 16 combination possibilities were created with the top four buttons, 4 with one button pressed, 6 with two buttons pressed, 4 with three buttons pressed, 1 with four buttons pressed and one with no button pressed. We use these buttons as a form of control in combination with others.

Thus, for each of the four buttons on the right side of the control, we have 16 note possibilities totaling 64 achievable notes. From the greater number of degrees of freedom achieved now, we have the possibility to also add functions to change the ADSR and the LFO.

Then, the right lever was used, which used to control only the volume. The characteristic and parameter are selected according to the combination of the buttons pressed. When the buttons are released, the vertical position of the lever determines the value of the parameter. Thus, our instrument ends up having different forms of configuration, although in this version of mapping, the control of individual parameters still leaves the task a bit complex.

For greater expressiveness, the left lever, which previously had the same effect as the directional ones, starts to control parameters such as the cutoff frequency and the resonance of a low pass filter, under its vertical and horizontal axes, respectively. The D-pad here continues to determine the velocity of the notes, however, to facilitate control, which has now become more complex, simply press one of the directional keys once to effect the change, in an on/off behavior. To reach the neutral position in this way, it is necessary to press the active position again. All the buttons and axes definitions are show on the table 5.

Table 5: Many-to-1 mapping, with the upper buttons selecting notes and parameters

Buttons	Mapped to
Button 1 to 4(frontal)	NOTE activation
Button 4 to 8(upper)	NOTE/Parameter selection
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad	Velocity control
Left Analog X	Filter frequency
Left Analog Y	Filter resonance
Right Analog Y	Parameter value
Right Analog(Rotation)	Volume

3.6 Mapping many to many

In our final mapping, we have sought to take inspiration on commands used in fighting games such as Street Fighter. We thus seek to exploit greater expressive capabilities of video game control with the capture of complex movements. With this we can also explore the affective memory of people whose games were part of their childhood and adolescence.

Here the controls become somewhat more simplified than in the previous mapping because we abandoned the idea of using the upper buttons to select notes and parameters, transferring this function to the directional buttons. When making combos by sliding your fingers through the D-pad, in the classic style of Arcade, a number of parameters can be modified in the sound event to be executed, be it in pitch, intensity, duration, or parameters of ADSR, LFO and filters. We therefore classify this mapping as a “many-to-many”.

Aiming for greater similarity to the controls used in the game, the eight main buttons were used in a similar way to the first mapping. The directional buttons, when just pressed and held, change the sound in a simple way.

The “hold forward” action in games means moving toward your opponent (assuming you’re on the left side of the screen, which is common for player 1 at the beginning of the fight). This results in stronger sounds with short

ADSR envelope and without the LFO to symbolize simple blows.

Holding the back button would mean defending or retreating your character. We chose to vary the message channel in order to activate another synthesizer with percussive sounds for that case. If any of these buttons is pressed twice in a row, it activates the LFO in a way that the sounds will be executed in a quick succession. In the game those commands would make the characters to dash.

Up, which causes the character to jump, means to go up an octave while down, which would be to crouch, means to go down an octave. These commands cam be used in the same way described above with the same effect. The diagonals can be use to achieve the results of pressing the combination of directions, like up and forward at the same time. For the more complex commands, an analysis was performed under the list of possible combos of the game Street Fighter IV. In this way, the most common commands used in the game were grouped in the form in which each is responsible for generating a specific effect.

Instead of thinking on isolated parameters we started thinking on abstract concepts, like force and energy, to determine the effects generated in each sound event. That is, the longer the command, the more intense and long-lasting the result will be.

If we look at the list of commands it is easy to see that although they vary from character to character, there is a pattern between them. Among the most common are the half-circle and quarter-circle that can be performed backward or forward. Such commands require the player to move the joystick or slide the finger across the D-pad to complete these patterns. Some commands require the player to hold the joystick in one direction and move in the opposite direction quickly. There are more powerful attacks called Ultra Combos that often require the above moves to be made in sequence.

For this mapping we added the execution of glissando. Holding the joystick back and then moving it forward will result in a crescent glissando. This movement can be performed in succession to increase duration. If the movement is performed from bottom to up it will have the same effect. Such movements performed in the opposite direction will result in an decreasing glissando.

The quarter circle results in the execution of chords that vary according to the button pressed at the end and whether it was executed clockwise or counterclockwise. If executed twice then it increases the dynamics and duration of the envelope.

A half-circle will perform these chords in increasing glissando. Just as before, running them in sequence will increase their duration. To perform a decreasing glissando, it is necessary to finish the combo on the D-pad with the direction opposite to the one half circle was executed.

All commands always end by pressing one of the main buttons. Which of these has been pressed will determine the pitch of the resulting sound. To vary the octave in this way is done pressing the D-pad or moving the joystick down or up before the combos. An alternative is to use the buttons on the left and right joysticks to vary the octave in a fixed way. The organization of the last mapping is illustrated in the table 6.

Table 6: Many-to-many mapping using video game combos

Buttons	Mapped to
Button 1 to 8	NOTE activation
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad	Combo entry
Left Analog	equals D-pad
Right Analog	Volume

4 Discussions

Despite all the possibilities described in our mapping implementations, it is possible to argue that DualAnalog and our project and implementation does not bring great advances in relation to the standard computer entries, like keyboard and mouse, in relation to musical expressiveness. The set of discrete buttons provided by the control are as limited as a mouse button since they do not give information about the applied force and give just the number of the clicked button. On the other hand the way the controller and the arrangement of the buttons on it can be handled top allow a faster and intuitive access to them for those used to this ind of interfaces.

In our opinion, these interfaces have some advantage if compared to a keyboard and a mouse that is the set of continuous axes. The pair of joysticks allows the simultaneous control of 4 continuous axes, which can only be done with two axes through a mouse. However, this control can only be done from a self-centered input value, giving always a relative value and not an absolute value. For this reason, it is impossible to leave the joysticks stopped at a specific position and a continuous movement is necessary to deal with this interface. In fact, this is not a complex interface that requires training to be used satisfactorily.

Returning to the original context of this interface, we will analyze the interaction in an electronic game between the player and the system. To beat the game, one will need not only an understanding of the challenges posed, but also how to use the controls made available efficiently. So, even if the joystick itself is a simple interface, mastering its use in a particular game may require some training. This is related to the way in which we choose to map commands and can influence interface performance.

The video game control we use also has some drawbacks. Analogs are too sensitive to perform a more precise movement. For the choice of mappings some physical impediments must be taken into account: it is not possible to manipulate the D-pad at the same time that the left analogue is used; it is not possible to reach the 4 front buttons at the same time the right analogue is used and the simultaneous use of many buttons at the same time requires training.

5 Final Considerations

In this paper we presented the possibility of using video game control for musical interactions focused on the different mappings to a synthesizer. More specifically, we worked with a variation of the DualAnalog control developed by Sony that has a USB input to be connected to personal computers. The connection between electronic games and music comes from the beginning of the video game era [14]. We believe that this is a point that can be explored in the creation of musical performances.

The task of developing mappings, taken as the focus of our research, brings up a unlimited range of possibilities. This also has its negative side, since it is easy to get lost in such a vast horizon. Finding mappings with strong semantic meanings is not a trivial task, especially in the case where there is no acoustic instrument to be taken as a reference [6]. So, for our more complex mapping we made the choice to base ourselves on the universe of electronic games.

Our research aims not only to create a digital musical instrument, but the elaboration of alternative controls to be used in the general artistic context. We chose to use MIDI protocol and compatible synthesizers because of their great popularity and due to the ease of prototyping. However, we believe that the implementation of a proprietary synthesizer brings greater control over sound and increases the number of synthesis parameters to be considered. Still, we have had good results with the choices we made. We were focused on the mapping and we believe that all these proposed mappings can be be used with other synthesizers.

The various mapping modalities presented, with their respective examples, serve as a basis for future research. As we have argued, there are numerous possibilities for mapping and each can result in a digital musical instrument with very different characteristics.

The choice of using MIDI synthesizers is justified, but can be discussed. MIDI is a Curse, is still here, and we still use it. But Open Sound Control (OSC) became a great option too[15].

6 Acknowledgments

Authors would like to thanks to all ALICE - Arts Lab in Interfaces, Computers, and Else members that made this research and development possible. The authors would like also to thank the support of the funding agencies PROPE, CNPq and FAPEMIG.

References

[1] Marcelo M Wanderley and Philippe Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632–644, 2004.

[2] Vesa Välimäki and Tapio Takala. Virtual musical instruments—natural sound using physical models. *Organised Sound*, 1(2):75–86, 1996.

[3] Gianpaolo Borin, Giovanni De Poli, and Augusto Sarti. *Musical signal synthesis*. Swets & Zeitlinger BV, 1997.

[4] Joel Ryan. Effort and expression. In *Proceedings of the International Computer Music Conference*, pages 414–414. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 1992.

[5] Andy Hunt, Marcelo M Wanderley, and Ross Kirk. Towards a model for instrumental mapping in expert musical interaction. In *ICMC*. Citeseer, 2000.

[6] Andy Hunt, Marcelo M Wanderley, and Matthew Paradis. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32(4):429–440, 2003.

[7] Andy Hunt and Ross Kirk. Mapping strategies for musical performance. *Trends in gestural control of music*, 21(2000):231–258, 2000.

[8] Matthew Blessing and Edgar Berdahl. The joystyx: A quartet of embedded acoustic instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 271–274, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.

[9] Sook Young Won, Humane Chan, and Jeremy Liu. Light pipes: A light controlled midi instrument. In *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 209–210. National University of Singapore, 2004.

[10] Christopher Ariza. The dual-analog gamepad as a practical platform for live electronics instrument and interface design. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.

[11] Jace Miller and Tracy Hammond. Wiiolin : a virtual instrument using the wii remote. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 497–500, Sydney, Australia, 2010.

[12] Elaine L. Wong, Wilson Y. F. Yuen, and Clifford S. T. Choy. Designing wii controller: A powerful musical instrument in an interactive music performance system. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '08, pages 82–87, New York, NY, USA, 2008. ACM.

[13] Joseph Butch Rovan, Marcelo M Wanderley, Shlomo Dubnov, and Philippe Depalle. Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop*, pages 68–73. Genoa: Associazione di Informatica Musicale Italiana, October, 1997.

[14] Ben Olson. Transforming 8-bit video games into musical interfaces via reverse engineering and augmentation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 73–77, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[15] Perry R. Cook. Re-designing principles for computer music controllers : a case study of squeezevox maggie. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 218–221, Pittsburgh, PA, United States, 2009.