Towards more effective mapping strategies for digital musical instruments

Patrick MCGLYNN
Music Department, NUI Maynooth
Maynooth,
Co. Kildare, Ireland
patrick.j.mcglynn@nuim.ie

Abstract

This paper contends that the design of digital musical instruments for live performance and composition has been hindered by the tendency to create novel applications which fail to offer musicians access to the more perceptually-significant aspects of electronic music.

Therefore, as a means of addressing this problem, this paper promotes the establishment of a more intelligent approach to the construction of digital musical instruments: one that is informed by relevant studies in human-computer interaction, cognitive psychology and product design.

Keywords

Mapping, digital musical instruments, real-time performance, Csound, controllers

1 Introduction

Recent commercial interest in the gestural control of home entertainment and portable computer systems has led to the rapid development of affordable and elegant systems for human computer interaction. The computer music community is responding to these advances with enthusiasm, producing a steady stream of musical applications which make use of the Apple iPad[1], Nintendo Wii Remote[2], Microsoft Kinect[3], and similar devices.

One trait shared by all of these interfaces is their tendency to employ *implicit communication* – a term coined by Italian cognitive scientist Cristiano Castlefranchi to describe interactions which exploit "perceptual patterns of usual behavior and their recognition" [1]. Examples of implicitly understood actions are the 'swipe' and 'pinch' gestures common to Apple iOS devices (which are analogous to page-turning and shrinking/expanding respectively). One potentially-destructive side-effect of these intuitive interfaces is the misconception that all applications should adhere to this simplistic approach – a paradigm whose limitations are especially destructive when it comes to applications for musical performance.

The expressive range and musical potential of these music applications is, being extremely fair, varied. Without an informed approach to designing these musical performance systems, developers haphazardly juxtapose musical functions in the hope of providing an instantly-gratifying musical experience. There exists an urgent need to discuss design issues which can potentially separate 'serious' electronic musical endeavors from the ever-growing selection of 'novelty' music applications.

2 Designing a digital musical instrument

In their book *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*, Miranda & Wanderley deconstruct the process of designing an electronic performance system into five distinct steps:

1. Decide upon the gestures which will control it
2. Define the gesture capture strategies which work best
3. Define the accompanying synthesis algorithms / music software
4. Map the sensor outputs to the music control
5. Decide on the feedback modalities available, apart from the sound itself (visual, tactile, kinesthetic, etc.) [2]

Depending on the circumstances, these questions will often be dealt with in a different order, with the available technology or musical goal providing the answer to several of them before the design process even begins. Assuming that every possible situation will have its peculiarities and idiosyncrasies, a general guide to assist designers in selecting

[1] http://www.apple.com/ipad/
[2] http://www.nintendo.com/wii/console/controllers
[3] http://www.xbox.com/kinect/

the best possible gestures and mapping strategies would be a valuable complement to this approach.

3 Mapping

In a digital musical instrument (DMI), mapping describes the manner in which data gathered by the input device(s) is related to the system's musical parameters. The importance of selecting or devising an appropriate mapping scheme cannot be understated – effective and elegant systems can lead to "a more holistic performance exploration of the parameter space" [3] and essentially define the "essence" of a DMI [2].

This is not to say that a performance system should necessarily be overly simplistic or immediately accessible. In the study of Human Computer Interaction (HCI), it has been suggested that "an efficiency-focused approach to interaction may no longer suffice: it needs to be complemented by knowledge on the aesthetic aspects of the user experience" [4]. In a musical context, an expressive interface design must accommodate the capacity to practise, learn, make mistakes, and develop skill.

Literature devoted specifically to the study of mapping schemes is sparsely available for a number of reasons – the theoretically limitless combinations of devices and musical goals that a musician might seek to accommodate render the discussion of general mapping principles difficult and of limited use.

Therefore, a more detailed vocabulary which enables musicians to assess their own situations is essential.

3.1 Mapping in digital musical instruments

Musical mapping schemes are generally classified according to the number of parameters over which the user can exert control at once. The most commonly-used terms are 'convergent mapping' and 'divergent mapping'. Convergent mapping employs a number of devices to control a single parameter ('many-to-one') whereas devices which use divergent mapping operate several parameters at once ('one-to-many'). It has been suggested that human operators expect such complex schemes and ultimately find analytical 'one-to-one' interactions more rewarding and intuitive [3]. Most acoustic musical instruments can be thought of as combining elements of both of these schemes.

3.2 Mapping in product design

Outside of a musical context, mapping schemes for human-technology interaction are more efficiency-focused and hence easier to discuss. In *The Design of Future Things*, Donald A. Norman encourages designers to utilize what he refers to as 'natural mappings' wherever possible (citing the oft-inconsistent positioning of hobs and their controls on a cooker as an example). In this context, it is preferable that controls should be laid out "in a manner spatially analogous to the layout of the devices they control" and that the principle can be extended to "numerous other domains" including sound [1]. With this consideration in mind, it is surprising how many supposedly-intuitive musical performance systems opt for the most convenient or visually-appealing layout for their controls, rather than considering the perception of the user.

In the same volume, Norman provides a summary of the essential design considerations discussed. His 'rules of interaction' state that interactive technology should:

1. Provide rich, complex, and natural signals
2. Be predictable
3. Provide a good conceptual model
4. Make the output understandable
5. Provide continual awareness, without annoyance
6. Exploit natural mappings to make interaction understandable and effective

It should be stressed that these considerations are clearly intended for functional applications which can be effectively used almost instantly - a description which cannot reasonably accommodate the law of diminishing returns that we associate with successful musical endeavors. However, they do provide a model of simplicity and efficiency which can be useful to bear in mind while working on more complex multimedia environments.

4 Towards systematic mapping

Adopting a methodical approach towards identifying and classifying the types of data generated by a particular device allows the interface designer to assess its suitability for various musical tasks in a logical, efficient manner.

4.1 Classifying performance data according to complexity

This high-level approach to mapping separates performance data into three distinct groups in ascending order of complexity:

A. Raw data (on/off, fader positions, X/Y/Z co-ordinates, etc.)
B. Symbolic / semiotic data (predefined actions associated with various postures, themselves represented by different combinations of the raw data)
C. Gestural data (predefined actions associated with dynamic movement)

An alternative way of phrasing this concept would be to think of group A as simple data, group B as elements of that data being placed in the context of one another to create more complex cues, and group C as the resulting cues being placed in the context of one another. Groups B and C can thus be thought of as constructing both the gestural 'vocabulary' and 'grammar', respectively, and play a crucial role in defining the usability and character of a given performance system. Future publications from this project will focus intently on the development of effective schemes to populate and manipulate these groups.

Input options classified according to these varying degrees of complexity can subsequently be allocated to different musical tasks, depending on the sophistication of control deemed necessary.

4.2 Degrees of freedom

In order to populate groups B and C as defined above, a system for assembling more complex commands from the raw data is required. By listing the available sensors and/or triggers of an input device and noting their inherent degrees of freedom a comprehensive 'toolbox' of available data can be defined.

Devices which offer one degree of freedom include buttons, switches, faders and dials. While the latter two examples can provide more detailed data than simple on/off controls (0-127 in MIDI, for example) they are still incapable of representing more than one piece of information at a time. Devices which offer two degrees of freedom include touch-sensitive keyboards (sending both note on/off and a velocity value) and simple X/Y pads (horizontal and vertical co-ordinates).

One must be careful not to confuse the terms 'degrees of freedom' with 'dimensions' – while the two terms are often used interchangeably they describe different aspects of a device [5]. An X/Y pad is typically referred-to as a two-dimensional surface and assumed to have two corresponding degrees of freedom. However a true 2D surface in fact provides three degrees of freedom – the X and Y co-ordinates of an object and the rotation of that object on the Z-plane (the Reactable, developed within the Music Technology Group at the Universidad Pompeu Fabra, implements Z-plane rotation as a central control device [6]). Add to this the possibility of multitouch, or placing multiple objects upon the plane, and the possible array of data to be obtained expands rapidly.

4.3 Augmenting simple control data

While not strictly provided by the device itself, the introduction of computer intelligence in the gathering of this data allows us to introduce a number of subtle factors which can expand the complexity of even the most basic input devices.

One example is an *'InUse'* variable which becomes true whenever a control has been accessed by the performer. By simply comparing the current value of a controller to a value stored one frame/sample ago, we can infer whether or not the state of a device has changed. A MIDI fader using this technique now provides two degrees of freedom – fader position (0-127) and *'isCurrentlyBeingMoved'*, or equivalent (0-1).

By lengthening the comparison times, we can also determine if the fader has been moved since *n* – this technique can be employed, for example, to terminate a musical event associated with a fader if it has not been interacted with for a certain period of time (analogous to the art of 'spinning plates', where elements require a certain amount of stimulation or energy input to survive).

Further to this, another variable can be added to keep track of the amount or intensity of user interaction with a device. This can take the form of a 'counter' which increases every time a change is detected in the value/state of the device (and perhaps decreases over time if the device is idle). An example of this exact technique is outlined below in section 5.

4.4 Combining simple control data

Using combinations of simple input data is a simple and efficient way to expand the number of options available to a user – the most familiar

example being the 'shift' key common to QWERTY keyboards which changes accompanying keystrokes to upper-case.

The computer mouse, as described by Buxton's '3-state model of graphical input', provides a more advanced example [7]. While the mouse prompts simple X/Y movements, these are interpreted differently depending upon which of the aforementioned three states the user has selected – state 0 is inactive (mouse is out-of-range or away from surface), state 1 is for pointer-movement, and state 2 is for the dragging and moving of objects and is invoked when the user holds down the mouse button. Needless to say, modern mouse, touchpad and trackball devices have greatly expanded this range through extra buttons and gesture recognizers.

However, caution should be advised when accommodating multiple layers of functionality within a single device – this increases the cognitive load upon the user and can compromise the building-up of associations required for intuitive and skilled performance [8].

5 An example application

A mapping experiment was conducted in order to examine the viability of the classifications as outlined in 4.1. The goal of the experiment was to replace the control surface of a hardware synthesiser with a different interface and demonstrate, via the application of the ideas outlined above, how alternate mapping schemes can extrapolate the functionality of a digital musical instrument from a performance perspective.

The process can be split into three parts – examining the original device, replicating the functionality of the device in a software model, and extending control of the model to a new interface.

5.1 Drone Lab V2

Drone Lab V2 is a four-voice analog synthesizer and effects processor by Casper Electronics[4]. It was designed to facilitate the creation of "dense, pulsing drones" by allowing the user to individually de-tune the bank of oscillators. The resulting phase-cancellation creates rhythmic textures which can be exaggerated and emphasised using the built-in filters and distortion effect.

This synthesiser was chosen for several reasons – the most pertinent being the lack of a predefined technique for controlling and 'playing' its noise-based output. The absence of any performance conventions facilitates the objective analysis of exactly how useful our classifications can be when designing an interface for an innovative or experimental performance system.

Figure 1: The original hardware version of Drone Lab V2

5.2 Csound implementation

In order to experiment with different control schemes, the synthesis model was implemented in the Csound audio programming environment. Both the signal flow chart and the comprehensive sound examples provided on the Casper Electronics website allowed for the construction of a software emulator which duplicates quite closely the output of the original Drone Lab V2.

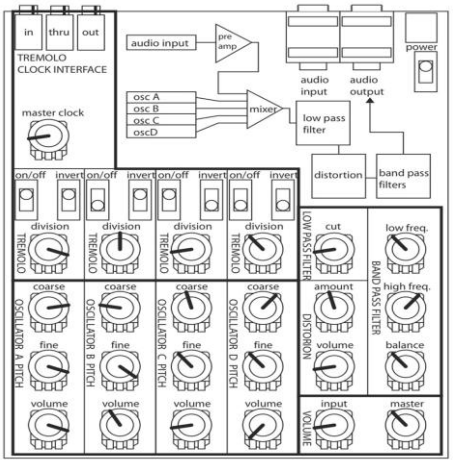

Figure 2: Open-source plans for Drone Lab V2[5]

[4] http://casperelectronics.com/finished-pieces/drone-lab/drone-lab-v2/

[5] http://casperelectronics.com/images/finishedpieces/droner/V2/KitLayoutLabels.jpg

One important issue that must be highlighted is the reduced functionality of the software implementation. While the general behaviour and audio output of the synthesiser are quite close to the original, the new GUI-based interface limits the user to manipulating a single parameter at a time via the mouse. User precision is also hindered by the lack of any tactile feedback and the need to rely exclusively on the visual display in order to discern the state of the various parameters.

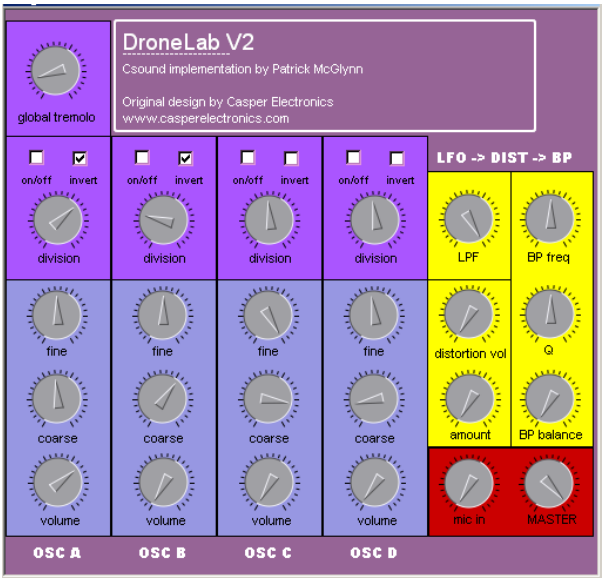

Figure 3: Csound GUI built using the QuteCsound frontend[6]

This problem is not unique to this project by any means – it could be argued that the tendency of software-based instruments to rely heavily on GUI-based controls is one of the main contributors to a lack of clearly-defined performance practice, not to mention the difficulty encountered by accomplished musicians when trying to develop an intuitive sense of these instruments.

5.3 Wii Remote control

The Nintendo Wii Remote is a motion sensing game controller which can function independently from the Wii console itself. Its ability to communicate wirelessly through the Bluetooth protocol and three-axis accelerometer has made the Wii Remote an extremely popular tool in the computer music community. Most of the major audio programming languages feature some level of support for the device and several dedicated interface-management programs (such as OSCulator[7] and DarwiinRemote[8]) allow the conversion of Wii Remote data into other useful formats such as MIDI or OSC.

For this project, the Windows-based program GlovePIE[9] was used to receive data from the Wii Remote and convert it into values readable by Csound (sent via OSC messages). One function was created for each type of performance data (as outlined in this paper) in order to illustrate the practical benefits of a systematic approach to parameter-mapping.

5.4 Mapping gestural data to instrument parameters using the group system

The ‘A’ button along with the plus and minus buttons on the Wii Remote were used to turn on/off the various oscillators and switch between them respectively. This is an example of *raw data* (group A) being used as a simple selection and triggering system.

GlovePIE provides access to ‘roll’ and ‘pitch’ variables which are derived from the angular velocity of the Wii Remote’s X and Y axes respectively. These were mapped to simultaneously control the frequency and volume of the oscillators. While these are both *raw data* / group A attributes, their combined values determine the overall behaviour of a single oscillator and accordingly allow the user to associate certain postures with the sound they produce. As such, the two values used in this mapping scheme depend upon each other and together represent an example of *symbolic / semiotic data* (group B).

While these mappings provide adequate access to the parameters concerned, they do not necessarily alter the way the instrument is played. The distortion volume and amount were mapped using a more complex setup which changed the behaviour of the sound considerably.

Using techniques described in section 4.3, a function was set up which continually checked if a certain threshold was exceeded by the combined acceleration of the Wii Remote’s three axes. If the overall movement of the user was violent enough to exceed this value, a global variable called *agitation* was augmented. When the movement was less pronounced, the *agitation* value would gradually decrease.

[6] http://qutecsound.sourceforge.net/

[7] http://www.osculator.net/

[8] http://sourceforge.net/projects/darwiin-remote/

[9] http://glovepie.org/glovepie.php

Mapping the *agitation* value to the distortion effect created a very effective control metaphor – users could easily associate the violent shaking or agitation of the Wii Remote with the resulting disintegration of clarity in the audio output, perhaps due to associations with real-world instruments which exhibit similar behaviour when shook vigorously (certain percussion instruments and electric guitar, for example). As it analyses complex cues in the context of previous actions, this final mapping can be placed within group C – *gestural data.*

6 Conclusion

Taking inventory of the data generated by a controller interface is an essential part of assessing its suitability for a specific musical task. However, one can easily underestimate the interdependence of certain variables and hence proceed to design a strictly functional device with no distinct characteristics other than to respond to various switches and faders (albeit virtual ones).

By categorising controller data according to *how* it may be used, as opposed to *where* it is coming from, we can avoid simply replicating the behaviour of physical controllers, escape unnecessary performance paradigms, and move towards the development of more complex, elegant and satisfying interactive performance systems.

7 Acknowledgements

The author would like to express his sincere gratitude towards Dr. Victor Lazzarini and Dr. Gordon Delap for their endless enthusiasm, support and inspiration throughout this research. Also special thanks to Dr. Charles Markham for his invaluable help approaching the field of gestural control.

This research project is funded by the John and Pat Hume Scholarship programme at NUI Maynooth.

References

[1] Norman, Donald A. *The Design of Future Things.* Basic Books, New York, 2007.

[2] Miranda, Eduardo Reck, and Marcelo M. Wanderley. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard.* Pap/Com. A-R Editions, 2006.

[3] Hunt, Andy. *Radical User Interfaces for Real-time Musical Control.* PhD thesis, University of York, 1999.

[4] Djajadiningrat, J.P., Matthews, B., Stienstra, M. 'Easy doesn't do it: skill and expression in tangible dynamics'. In *Personal and Ubiquitous Computing,* vol. 11, no. 8, pp. 657-676, 2007.

[5] MacKenzie, I.S., Motor behaviour models for human-computer interaction. In J. M. Carroll (Ed.) *HCI models, theories, and frameworks: Toward a multidisciplinary science,* 27-54. Morgan Kaufmann, San Francisco, 2003.

[6] Jordá, S., Kaltenbrunner, M., Geiger, G., and Bencina, R. 'The Reactable*'. In *Proceedings of the International Computer Music Conference (ICM05),* San Francisco, 2005.

[7] Buxton, W. A. S. 'A three-state model of graphical input'. In *Proceedings of INTERACT '90,* pp. 449-456. Elsevier Science, Amsterdam, 1990.

[8] Jordá, S. *Digital Lutherie: Crafting musical computers for new musics*. PhD thesis, UPF, Barcelona, 2005.