Experiencing Mappings on a Music Controller: Case Ableton Push

Ilkka Kuivanen

University of Tampere
School of Information Sciences
M.Sc. Thesis
Supervisor: Markku Turunen
April 2016

University of Tampere
School of Information Sciences
Human-Technology Interaction / Interaction Design and Research
Ilkka Kuivanen: Experiencing Mappings on a Music Controller: Case Ableton Push
M.Sc. Thesis, 62 + 9 pages
April 2016

During the past two decades, computers have taken a major role in composition and production of modern music. Various different devices are used to control modern DAWs (digital audio workstations). This diversity of controllers gives users a wide range of options to interact with a computer during the composition and creation. A deeper integration blurs boundaries between a software and controller and creates more complex modes of interaction. This study focuses on a connection between a musical controller and computer through the concept of mapping.

The aim of the study was to understand, how mappings of a contemporary software controller can be conceptualised and how they affect user experience. The material for this study was collected through an online-survey, user-sessions and semi-structured interviews for users of Ableton Live software and Push controller. Collected data show that users consider the mappings as an important part of the initial usability of the controller, but there is no clear correlation with the user experience.

The results stress the importance of avoiding idiosyncrasies in the mappings and reveal the versatile approaches between users. Users with previous experience from the software tend to have established use patterns that might affect the deployment of a new controller. Mappings can be considered to have an essential role in these patterns.

Keywords and terms: DAW, Controller, Mappings

Forewords

I would like to thank my Thesis supervisor Markku Turunen and researchers Matti Luhtala and Salim Haniff for giving me ideas and guidance. I am thankful to Ableton for using their time and resources for the Thesis. A special thanks goes to Johannes Russ from Ableton for helping me greatly from start to finish. Many thanks to Sam "Stray" Hurley from nativeKontrol for giving his valuable comments. Lastly, I would like to thank all the participants who made the sessions possible and everyone who participated in this work.

Contents

1. Introduction

The role of a computer in modern music production is essential. This is largely due to rapid development of technology in the 1990's (Dean, 2009). The affordability and availability of digital music production equipment enabled the use of relatively expensive tools in computer-mediated environments (Hewitt, 2011). In case of computer-created music, the development of technological features can be seen more than just digitalised signal processing. Moreover, the development has profoundly enabled a progress towards expanding interaction between computer and the performer. In fact, one of the mainline areas on computer usage in modern music is production and performing with the computer itself (Keith, 2010).

The relationship between musicians and computers has evolved into a field of possibilities, where they can interact in various contexts. Today, musical activities can be performed with different types of computer instruments and controllers. Commercial products range from relative modest MIDI-keyboards to complex digital audio workstation (DAW) integrated interfaces. Different approaches can be taken towards the controller paradigm due to diversity of technologies and purposes of use. One way to examine and analyse the interaction between a user and controller is to focus on how control elements are connected to sound parameters and how these connections are experienced in creative contexts. Instruments are commonly associated with gestural interface that is directly connected to a sound (Dean, 2009). Computer-based controllers function differently in terms of the need for fixed connection between the gesture and sound source. The rules for communication between the interfaces are defined in the relationships between the sonic objects and the controlling actions. These bindings of various objects are called mappings.

The use of the term mapping varies depending on the research. In this work, observations from different studies and findings from research-based controllers are viewed in wide context. A major influence for this work comes from a research presented in NIME (New Interfaces for Musical Expression) publication. A large proportion of the NIME articles focus on musical interfaces with some level of technical or artistic novelty (Johnston, 2011). However, the research tends to focus on results rather than methodology (Kiefer et. al. 2008). The potential of using human-computer interaction techniques to study experiences of performers has been recognised by several authors, however the approach has been to use mostly quantitative techniques that tend to disregard user experiences that are harder to measure (Johnston, 2011). The essential element traced from the researches in NIME is to approach musical controllers as tools for expression and then combine common human-technology interaction methodologies to understand technology from user's perspective.

This study focuses on finding the essential aspects of mappings in virtual devices that affects user experience. Three research questions were formed: 1) How can the concept of mappings be used in the context of commercial music controllers. 2) How does mapping affect user experience and 3) how common research methods work in contexts of interaction and music.

The selected controller for further study was Ableton Push, a dedicated digital audio workstation controller for Live software. Ableton's products were selected based on Live's reputation as being one of the most common digital audio workstation on the market (Music Radar, 2014). Push was considered as an applicable subject for two reasons. First, the range of functionalities enables various purposes for use. The controller can be used in multiple music composition and production related scenarios which sets the mapping of devices into wider context. Second, users with previous experience Live are familiar with the software and its virtual devices. The controller can be then considered as an alternative interface for controlling already familiar software objects. The importance of the study topic can be addressed by noting that in the field of music controller research it is rare that co-existing interfaces are studied based on their structure in mappings. Yet, the design and technical structure of mappings define how the virtual parameters are controlled. It is considered that by approaching the subject holistically from the users' perspective the relation between mappings and user experience is more inclusive and explanatory than it would be when only parts of systems are analysed.

The study is based on common research methods in the field of human-computer interaction. The study consisted a survey, user-sessions and interviews. An online survey for Push users was conducted between February 27th and March 22th, 2015. The survey was shared in several discussion boards, Facebook groups and email messages. The target group was Push users globally. The survey aimed to find out how users experience the controller and what kind of issues they see in Push's mappings. The general goal for the survey was to gain contextual and informative data from Push users. The user-sessions and interviews were run between February and March 2015 for five Live users at participants' home studios in the cities of Tampere and Turku, Finland.

This work is divided into three main sections. First, the focus is on the concept of a controller and mappings. The studied controller Push is also briefly presented. Chapters two to four focus on limiting the research scope and identifying the key areas of interest. In addition, relevant aspects of the theoretical concept mapping are presented. Second, chapters from five to seven present the conducted research and results. Selected research methods are presented and their use in context of the study is argued. Chapters six covers the conducted survey by first describing the survey structure and then moving on the results and findings. Chapter seven focus on sessions and interviews by presenting the key findings as well as including comments from an independent script developer. Lastly, chapters eight and nine conclude the work in form of discussion and conclusion.

2. Controllers and Interaction

This chapter focuses on defining a musical controller and presenting broad taxonomies for various usage types and patterns. Observations from different studies and findings from research-based controllers are viewed and discussed. The emphasis is on describing the field of controllers and clarifying the research area.

2.1. Characteristics of a Musical Controller

To understand the functions and roles of a musical controller it is preferable to examine controllers in context of use. Musical controllers are typically used to reach creative and artistic aspirations. In this paradigm, the concept of a computer-based controller is closely linked to the concept of a computer-based instrument. Characteristics of computer instruments are often established on precepts of the respective acoustic instruments. The traditional depiction of instruments is based on features, such as timbral characteristics and activation-sonification model. However, it is neither beneficial nor descriptive to use a similar approach to a computer-based instrument (Dean, 2009). The field of computer-based instruments becomes obscure if the classification criteria used in traditional instruments is applied. The most notable difference between traditional and computer-based instruments relates to signal creation, processing and sound parameter alterations. On more conceptual level, there is an apparent distance between gestures that enable creation of musical structures and the actual explicit synthesis operations (Dean, 2009).

Hereinafter, a computer-based music controller is defined using a computer-based musical instrument as a reference. Chadabe (2002) has defined a computer-based musical instrument to include the following components: a gestural interface (controller), data paths and an audio system that outputs sound. Wessel et al. (2002) also added a conceptual model that determines interaction between controller and software. When defining a computer-based controller, it is reasonable to exclude the system output. Therefore, it is presumed that the audio is generated or modified based on (some of) the actions on the controller, but the controller does not define the actual properties or the structure of the output device.

Typically, there is no bi-directional movement between the controller and generated audio. Nevertheless, the controller and the audio interface can be physically built into the same device, and the important notion then is the requirement for the computer to perform the synthesis and signal processing. While controllers do not directly create sound, they enable the creation through different control actions. An acoustic instrument generally combines gestural interface with a sound production unit (Miranda & Wanderley, 2006). In contrast, separating the control from the sound synthesis has been evident since the early development of electronic music devices (Dean, 2009). The concept of modern computer-based controller can be built on the modularity aspect of an electronic instrument. By confining the module of sound synthesis and signal processing from computer-based instrument, the result is a musical controller. The field of different computer-based control environments is relatively unorganised compared to the world of traditional instruments. In order to define the realm of controllers more profoundly, it is beneficial to differentiate controllers based on selected criteria.

2.2. Classification of Controllers

The field of computer music controllers is extremely rich. Controllers are built for diverse sets of tasks: from expressive interactive and pervasive performance to intent studio engineering. Controllers vary, for example, in their input modalities, purpose, physical features, level of augmentation and technologies; including protocols, physical sensors and software. Creating definite taxonomies for these controllers is rather constrained due to the nature of music creation itself. First, there is no inherited need to use a controller for all the enabled functions, or even to use it for the purpose of the initial design. Second, personal routines that transform into creative workflows tend to be idiosyncratic. These routines rarely follow absolute external rules or pre-determined conventions. Controllers cannot be classified in orthodox tradition because of the lack of strict tradition and legacy. However, by comparing controllers to traditional instruments, it is possible to reveal some important traits that expand the concept of controller.

Traditional instruments are typically well classified (Kvifte, 1989). Some studies have categorised controllers based on their gestural resemblance to existing instruments (Miranda & Wanderley, 2006). While traditional instruments usually have a defined relation between sound source and required action, controllers do not necessarily possess similar level of simplicity. The disjunction between interface and function disposes typical affordances related to traditional instrument interfaces (Dean, 2009). Today, the gestural event is considered to be more abstract as the link, or mapping, between action and sound generation becomes unfixed. Therefore, it can be argued whether it is beneficial to categorise controllers and instruments based on their resemblance on existing instruments. Another type of categorisation of commercial instruments could increase the flexibility of conceptions related to the functions of various controllers. The purpose of the following technology-based and user-centred classifications is to maintain simplicity, freedom and flexibility.

2.3. Categories Based on Mapping Technology

A practical way to separate controllers is to define the type of technological framework that enables and determines the characteristics of the communication between the physical controller and the software running on a computer. In the case of consumer products, controllers can be roughly divided into two groups: general controllers and controllers that have a distinct mapping medium integrating the controller with the software. General controllers often use traditional communication standard such as MIDI. In such controllers, the basis for mapping is rather elementary and relies on assigning controls to sound objects. On the contrary, integration based controller operates over a specific protocol or a mapping software. Examples of such implementations are Novation's Automap (Novation, 2015) and Native Instrument's Native Kontrol Standard (Native Instruments, 2015b). However, controllers using closed systems may still support third party software.

MIDI standard itself is focused on clarifying the interpretation of messages with general MIDI (GM) specification. However, the structure of the current market and state of technology has an influence across hardware and software. The choices that hardware and software companies make eventually engender other products, devices and components in the field. The reason for generic controllers being widely available is most likely due to the tradition of modularity in production environment paradigm and strong foothold of MIDI as a go-to communication standard. MIDI has turned to be compatible communication protocol for various tasks because of it's pervasive representation of musical data (Dean, 2009; Wessel & Wright, 2002). An interesting question is if digital environments will eventually move away from this paradigm towards closed and integrated systems. Some views from software and hardware developers are collected in section 2.10 to give insight into issues on combining software and hardware.

2.4. Categorisation Based on Context of Use

In addition to technological sorting, it is possible to examine controllers based on their general purpose in music production. To simplify, the use of the controllers has been divided based on their context of use and user groups. A central viewpoint regarding this work is the separation of controllers manufactured and sold commercially, and controllers created in the field of experiment and research. This division gives two rather vague groups: professionals and amateur musicians who use commercially available controllers, and researchers and enthusiasts who create sometimes-unique pieces of technology. These categories are not explicit and they might overlap in situations where musicians are creating their own instruments.

Two main categories are presented based on their use: 1) composing and performance oriented controllers and 2) production and engineering targeted controllers. Controllers in the first category are mostly used for generating, modifying and triggering sounds based on gestural input. The conceptual workflow focuses on musical aspects such as notation, arrangement, sound design or mixing. Typical examples of such controllers are MIDI-keyboards, multi-effect devices with touch control, foot pedals or mixing decks used by DJs. In the second category, the workflow is not necessarily focused on performance-oriented spontaneous sound triggering and rapid alternation of relatively different musical ideas, but instead detailed and deliberate adjustments of sonic artefacts. Such controllers could be used as motorised mixing surfaces for digital audio workstation or offline duties related to post-production of audio. These categorisations are not explicit and the purpose is only to describe variance in functional approach to controllers. Nonetheless, it is important to understand the technological framework and purpose of use when the mappings of any controller are studied. It can be argued that especially the context of use highly affects the nature of interaction and the way mappings are structured and designed. A task that requires detailed, restrained and total control of given parameter has fundamental difference with a task that embodies improvisation, practice and even desired randomness. It is considered that, to some extent, the performed tasks define the essence of the controller in tandem with the initial design.

The way controllers are used depends on the gestures supported by the interface. The term gesture can be used to describe human actions in various ways. According to Miranda et al. (2006), gestures are considered to be any human action used to generate sound. Different research approaches and viewpoints might put gestures in wider context. Although the characteristics of an input method are an important part of the performed action, it is considered less important as the studied spectrum of consumer controllers (and the selected controller Push) follows the paradigm of tactile interfaces, that relies on touching the knobs, faders and buttons of the physical interface.

2.5. Design Approaches

The design approaches for computer-based instruments and controllers is an active and well-discussed topic. Analysis of instrument design strategies is sized out of the scope of this study. However, understanding design approaches for controllers is essential when concentrating on the problem space around human-technology interaction and musical controller mappings. The interface design is tightly related to technology and artistic creation. Conception and design requires technological knowledge and understanding the musician's culture (Jordà, 2001). Computer-based instruments should be simple enough to begin playing, but still be open for exploration and include the possibility for mastery (Newton & Marshall, 2011). As with musical instruments, the nature of musical controllers seems to include inherited appeal of progression and mastery. To benefit from this setting, the design must include degrees of difficulty in learning. The ideal learning curve for instrument usage would initially be low and gradually increase over time (Holland et al., 2013).

Closely related to the concept of mastery is the aim for deeper stimulation. Machover (2002) outlines several design issues relating to the necessity of controllers; one of the relevant questions is how to stimulate rather than placate. That is, how to guide users beyond apparent features and enable pursuit for rich, expressive and meaningful experiences. The aim should be in creating instruments that feel fresh and alive rather than arbitrary and contrived.

The aforementioned relatively abstract view stems from the creative side of the interface development and research. Even if designers describe the interfaces as usable, it does not necessarily imply that the design is axiomatic in terms of musical expression (Poepel, 2005). It is a common belief that instruments that are harder to play are also more sophisticated in terms of expressivity (Jordà, 2001). Therefore, it can be argued that, to some extent, these endeavours for creating controllers and instruments that enable virtuosity are dispersed and based on subjective view. In fact, Poepel (2005) hypothesises that the reason for this is the rarity of the evaluation methods. It can only be speculated how much of the given commercial product's design and development cycle includes evaluation of the actual learnability and capabilities that foster the process of mastery. However, creating commonalities that support both ease of use and virtuosity is not a simple design task. Perhaps the most favourable and obvious answer for this is to recycle and reform already well-established designs. This aligns with D'Arcangelo's (2001) suggestion to use sampling as a creative activity and use it as an approach to interface design.

As noted earlier, the design of musical interfaces relates to understanding of technology and musician culture. Musical interfaces can be partially responsible for shaping the future of music, even in case of computer-based instruments (Jordà, 2001). Computer based instruments and controllers should not be designed from idiosyncratic point of view, which is often the case in research-based work (Wanderley & Orio, 2002). The design should not be based on specific tradition either, even though the knowledge of existing approaches, personal beliefs and intuition give designers a desirable building ground. Being a part of the field helps the designer to create easy to use, sophisticated and expressive instruments that are evolved from previous efforts and designs (Jordà, 2001).

2.6. Difference Between Assigning and Interaction

A long-term issue in digital realm is accessing various parameters of software device with a limited number of physical controllers. One solution is to make the controllers assignable through a process called "learning", which means creating fixed wirings between received input messages and software parameters. Traditional MIDI controllers, with assignable knobs and faders, have several drawbacks from an interactional point of view. First, parameters exist in different physical and virtual spaces and the user has to make additional conversions between the spaces. Second, users need to remember the connection between the controller and the parameter (Kobayashi & Akamatsu, 2005). Third, the conventions of MIDI-mappings vary depending on the software or hardware.

	Interface design	Data-path	Conceptual Approach
Assign based	Applicable	General protocols, limited	Assign to function
Integration based	Premeditated	General / custom protocols, extendable	Assimilate

Table 1. Distinctions between assigning and integration.

By comparing assigning- and integration-based controllers to the definition of musical computer-based controllers presented earlier, the distinction between the two can be made more evident by generalising some of the characteristics (summarised in Table 1). Whether the computer-based controllers designed to function with specific software and to use specific data-path structures are more coherent on the conceptual level, is not clear. However, even the current integration-based controllers cannot provide fully adaptive tactile feedback between a control-unit and a software parameter. This problem of binding the physical controller to the sound parameter has been a question of concern every physical controller (Kobayashi & Akamatsu, 2005). When this connection is made in full duplex, the relation has been described as instrument's consciousness, i.e., the resonance of the instrument during the sound production process. Although bi-directional mappings (haptic or force feedback) could exist within the MIDI technology, a more profound connection in digital realm would require online analysis of the sound itself and its delivery back to the controller interface. This type of interaction has been rarely undertaken in controllers (Jordà, 2001).

2.7. Controller or Instrument

In this study, a controller is defined to have interface, data-path and conceptual model. This model has no unambiguous place for bi-directional interaction that would eventually enable afore mentioned instrument's consciousness that strongly binds with the live performance and expression paradigms. Chabade (2002) approaches the controllability of an instrument by creating a taxonomic line between deterministic and nondeterministic functioning. A deterministic instrument is predictable in its output based on input by the performer. Every detail is precisely controlled and no additional information is exchanged between the instrument and user when actions are performed. On the contrary, a nondeterministic instrument includes small amount of unpredictable information that stimulate user while the all the macro-music aspects remain unaffected. Depending on the level of unpredictable behaviour, talented users can benefit from nondeterministic instruments in their performance. Widely implemented illustration of non-deterministic functioning in the field of software-based instruments is "humanising" the performance. This is most often made by adding adjustable level of randomness into certain musical variables such as note's duration, velocity, or timing.

The capability of altering the degree of non-deterministic functioning can be seen merely as an operational description. The existential nature of the controller may depend more on teleological approach instead of distinct analysis of jitter between gestural input and musical output or probabilistic variance in expression. In this study the term controller is intermingled with idea of an instrument. To understand user experience in the context of computer-music, it is reasonable to attach cultural and social commonalities of an instrument to a controller, even if the instrumentality of a controller is not universally agreed.

2.8. Learning a Controller

Wessel and Wright (2002) have stated that starting with computer-based instrument should be relatively easy, although it should not restrict or limit the continued development. This assumption aligns with the concept of an ideal human-technology interaction (Holland et al., 2013). To some degree this can be considered to be opposite to traditional instruments, which are not generally easy to play at first. In contrast, Hunt et al. (2000) argue that complex mappings should not be learned instantly. One of the most relevant differences relates to the composition of gestures. Typically, gestures used in consumer controllers follow keyboard-engineer paradigm based on early electronic instruments, i.e. controlling through keys, knobs, sliders and buttons. The conceptual learning space of mappings can be more abstract in computer-based instruments than in traditional acoustic instruments.

Controllers do not necessarily embody distinctive gestural approach as when learning completely new acoustic instrument with radically different input modalities and gestures. This might be explained by the technological and practical conventions of computers and computer music. Magnusson and Mendieta (2007) state that digital music instrument and interface building is more aware of ergonomics. However, the irregular relation between mappings of gestures and sound might affect creation of masterly interfaces, even if they were easy to learn. Perhaps the mappings would require generative capabilities, as Hunt et al. (2000) suggest.

In a study by Magnusson and Mendieta (2007) it was found that musicians who had experience with traditional instruments, expressed the need for limited yet expressive software based instruments. Benefits of certain mapping related limitations are more easily understood in the context of acoustic instruments. However, it can be questioned if computer-based instruments or controllers truly benefit from arbitrary restrictions in mappings. More over, the expressiveness, creativeness and learnability of a controller should be evaluated in a holistic manner, separately from traditional instruments. Computer-based controllers impose partly different physical and psychological challenges in learning than traditional instruments. It can also be hypothesised if a correlative relation between the gestural difficulty and cognitive load of mappings exists. That is, the more difficult the gesture, the more discernible mapping is needed.

2.9. Expressiveness

The mere existence of an instrument does not ensure expressivity. The ability to control the sound source enables expression within that medium of control. Consequently, an important notion is that control does not equate with expression (Dobrian & Koppelman, 2006). Dobrian and Koppelman (2006) define the basic need for controller to accurately capture the data provided by the interface. In addition, the correspondence between input data and output sound should be intuitive for performer and the audience. In conclusion, the expressivity is related to the transparency of the interaction between a performer and the interface.

The expressivity also requires learnable and repeatable control of the sound. The control actions should be finely detailed and intimate, yet capable of creating complex and dramatic changes. To be engaging to the performer, the system needs to have appropriate level of difficulty (Holland et al., 2013), i.e. an efficient learning curve. Such diversity in the control often leads to systems that are difficult for beginners, still supporting long-term engagement. Nielsen (1996) has described the situation to be a trade-off between ease of learning and long-term flexibility. Wallis et al. (2013) have proposed design heuristics for musical instruments and their impact on long-term engagement. In addition to aspiration for long-term engagement, they propose a design that also includes incrementality. The progression in difficulty should be gradual to enable persistence within the given activity. The so-called "flow state" can be used as an example such persistence. The flow state itself is enabled by the transparency of interaction. In conclusion, the transparency of control and long-term engagement of an instrument increases the expressivity and the gradual learning gravitates towards the phenomena commonly described as flow state.

2.10. Developer's Views on Software-specific Controllers

In October 2014 Computer Music (2015) magazine surveyed six different software and hardware developers and manufactures if software-specific MIDI controllers are a good idea. The overall approach was positive towards software-specific controllers. However, there are differences in views on how many different software a controller should support and how the integration should be implemented (Computer Music, 2015). It should be noted that the answers most likely reflect the business and market strategies of the represented company.

Two different approaches to the nature of interface between hardware and software can be extracted from the answers. First, software and hardware are seen as one entity that can be developed simultaneously, in a two-way manner. The representative of Ableton pointed out the possibility for deeper integration and optimisation due to simultaneous development of software and hardware (Computer Music, 2015). In addition, in this scenario the developer has access to software functionalities not available for third party developers. This idea of tandem-like development of software and hardware at the same time got support from Native Instruments, whose approach was to consider hardware and software as one unit. This approach emphasises the top-down role of the manufacturer as the interaction is designed in holistic manner.

Second approach is focused on opening the middle ground of hardware and software by emphasising open protocols for communication between software and control devices. This approach stresses the importance of integration, but also focuses on diversity of controllers. Leo Nathorst-Böös, the CEO of Propellerheads Software, states they want Remote protocol (a protocol developed by Propellerheads Software) to be open so the controller market can thrive and be innovative (Computer Music, 2015). This strategy can be viewed as a service structure for controller designers, as the Propellerheads Software controls the protocol in question and they focus on offering a technical platform and commercial infrastructure for controller developers (Propellerhead Software, 2005). Also, it should be pointed out that the afore-mentioned diversity of controllers most likely relates to Propellerheads' own software, as Remote protocol is not universal protocol to be used across all audio workstations.

Manufactures like Novation and Akai Professional have developed controllers for Ableton Live software that are mediated by Ableton's Python based Remote Scripts (see section 3.6). Their extensive support for controllers using Remote Script is evident in the list of natively supported control surfaces (Ableton, 2015d). Remote Scripts are "*devices containing one or more controls that are automatically assigned to parameters in Ableton Live, allowing tactile control of the software*" (Ableton, 2015h). The difference between the views from Novation and Akai

Professional is on their approach towards focusing on single software. Novation considers that trying to reach the level of functional integration with different software would end up with unnecessary features and disrupted workflow. In contrast, Daniel Gill from Akai Professional states that it is important that controllers work with different software. (Computer Music, 2015.)

Concepts like open protocol and remote scripting distinctly differs from the holistic approach, especially from the interaction design point of view. Some companies, such as Ableton, pursue both directions. Dave Cross, and independent controller builder predicts that software developers will focus more on their own hardware to enhance workflows, although it will, to some degree, decrease customers' freedom to choose (Computer Music, 2015). However, all of the six manufacturers have positive views on software-specific controllers, despite the different opinions on hardware and software integration.

3. Case: Ableton Live Devices and Push Mappings

The premise for this study was to examine the interaction between a well-known DAW (digital audio workstation) and its dedicated controller. The selected environment for further analysis is Live software (Figure 1) and the dedicated controller Push (first generation), both developed by Ableton. Like any modern DAW, Live can perform various production related tasks. In context of the study, only a small portion of the functionalities of Live is covered. The specific interest of the software functionality is the control of virtual devices and how it is implemented in the Push interface. The study was conducted on Live version 9.1.7.

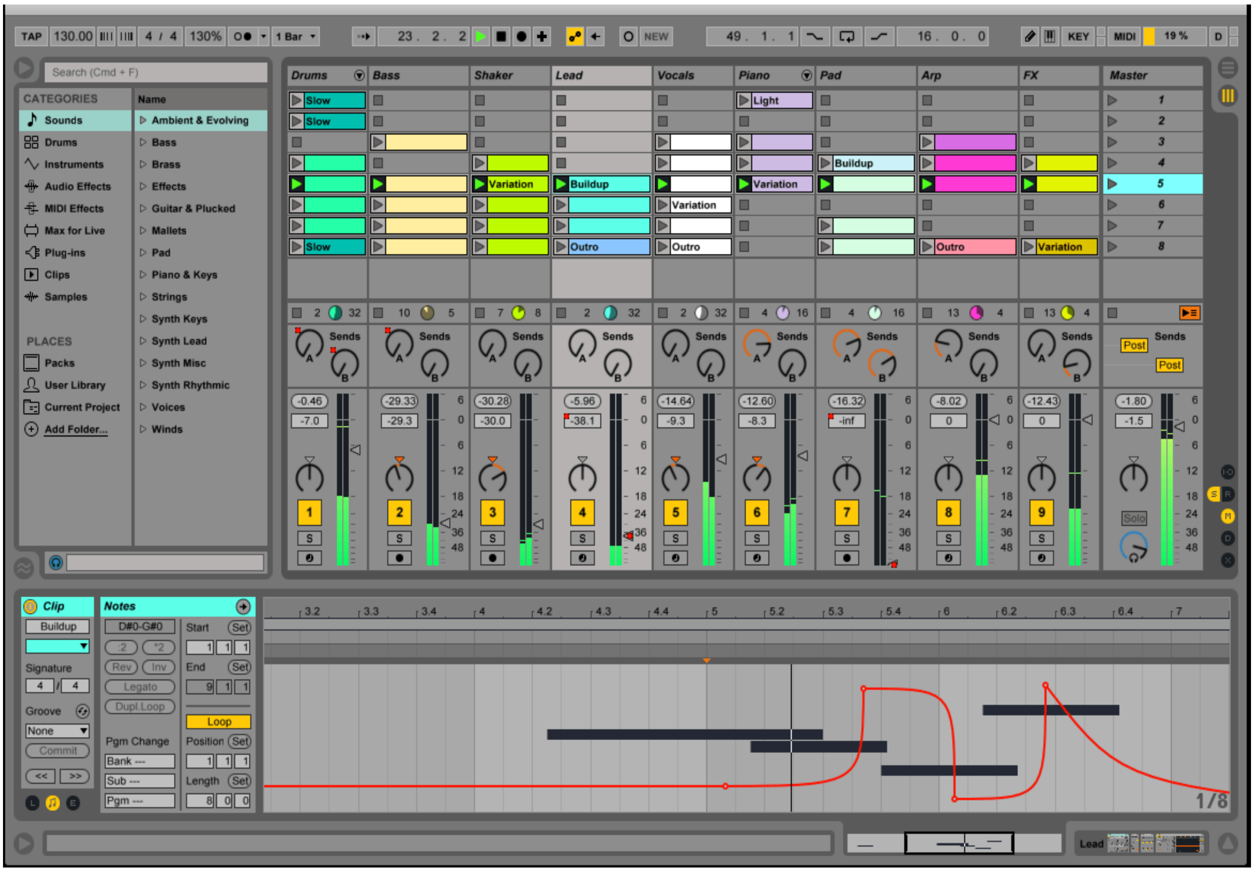


Figure 1. Live 9 in Session View (Ableton, 2015i).

3.1. Live

Ableton Live is a cross-platform music production software released in 2001. Today, Live is one of the most popular commercial DAWs on the market (Music Radar, 2014). Live has kept the same fundamental visual structure since the pre-release version from the 2000 (Figure 1, Figure 2). One of the central features in Live is the non-linear clip-based composition mode called session view (Figure 1), in addition to the traditional linear sequencer. Another unique aspect is the capability to build nested objects (racks) that can have assigned macro controls for parameters (section 3.5).

Figure 2. A pre-release version of first version of Live (November 2000) (Henke, 2015).

3.2. Push

Push (Figure 3) was released to the public in March 2013 and marks the first hardware device released by Ableton (Ableton, 2013). According to the company, the controller allows control of melody and harmony, beats, sounds, and song structure (Ableton, 2015g). The main functionalities of the controller are based on software, as the controller does not process audio signals. Upon the release, Push was reviewed positively in music related publications and websites (Attack Magazine, 2015; Rothwell, 2013).

Figure 3. First generation Push (Ableton, 2015f)**.**

Based on the categories presented in section 2.4, Push can be considered to be an integrated composing and performance oriented controller. The interface elements are fairly common in music controller domain. The input is given via encoders and rubber buttons, 64 velocity sensitive pads and a touch strip. Most notable difference in design, compared to other commercial products, is the high amount of pads for performing, triggering and sequencing.

Push’s interface is divided into different sections (Figure 4). Primary focus of this study is in sections *display/encoder*, *selection control* and *state control*. *Display/encoder* section enables altering the software parameters and virtual devices in Live. Selection and state controls enable selecting virtual device or devices of the current track as well as work for navigation in other contexts.

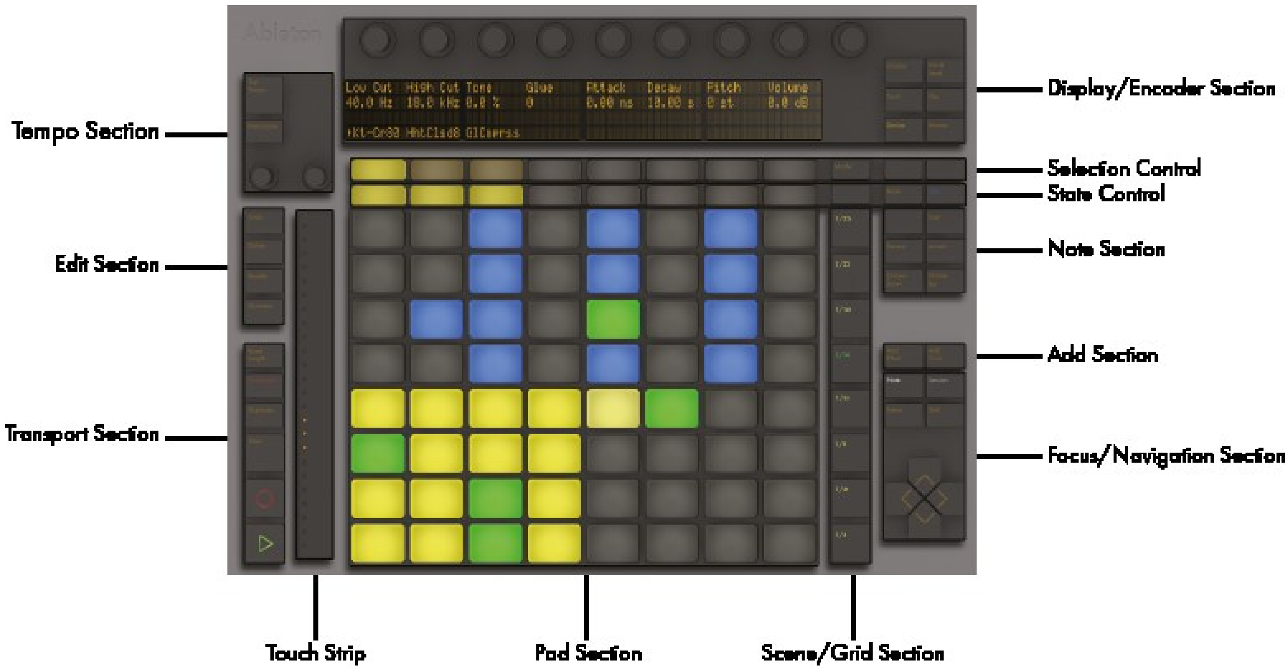


Figure 4. Push interface divided into sections (Ableton, 2015g)**.**

3.3. Devices

A *device* is Live's concept for virtual signal processing object that can be an effect (audio or MIDI) or an instrument (Ableton, 2015j). The range of virtual instruments and effects can be extended to external plugins in VST or AU format. In this study, the primary interest is in Live's native devices, more specifically a virtual analogue synthesizer called Analog. All of the Live's native devices have a dedicated user interface in software and pre-defined parameter mappings in Push. For the time being, third party plugins can be used via Push's interface, but does not have pre-defined parameter mappings.

A device can be inserted into the audio or MIDI track either from Live or Push. Any device can be controlled simultaneously from both interfaces and the values are visually updated in real time. Live's native devices are pre-mapped to the encoders and cannot be modified from default interfaces. An example structure of the parameter mapping can be seen from the Figure 5, where five devices (named *KeyPiano*, *AtFlter*, *EQ Eight*, *SmplDlay* and *Glue*) are loaded into a track. The currently selected device Auto Filter (*AtFlter*) is indicated with a caret in front of the device name. The first row from the top displays parameters for each device, and the second row corresponding values for each parameter.

Figure 5. An example of parameter mappings on Push (Ableton, 2015f)**.**

3.4. Analog

Analog, released in 2007 along with the version Live 7, is one of the several native devices of Live (Ableton, 2015a). Analog is a VA (virtual analogue) synthesizer developed by Ableton and Applied Acoustic Systems (Ableton, 2015b). The graphical user interface (Figure 6) consists two parts: the *display* and the *shell*.

Figure 6. Screenshot of Analog's user interface (Ableton, 2015c)**.**

The shell contains the most important parameters for each section, whereas the display updates additional controls according to the selected shell (Ableton, 2015b). The signal flow of Analog is presented in Figure 7. The structure is fairly common in subtractive synthesis. A notable similarity can be seen in the order of shell sections and signal flow. Signal originates from the oscillators, which are positioned on the left side in user interface. Filter sections are positioned in the middle and amplifiers on the right side, above the display element. Low-frequency oscillators (LFO's) and global controls are located on the right side of the user interface. Although visually aligned diagrams and graphical objects might seem trivial, the important notion is that the user interface visually aligns with signal flow by positioning the elements in logical order. Each group of additional parameters, such as filter envelopes are accessed from the display in Analog.

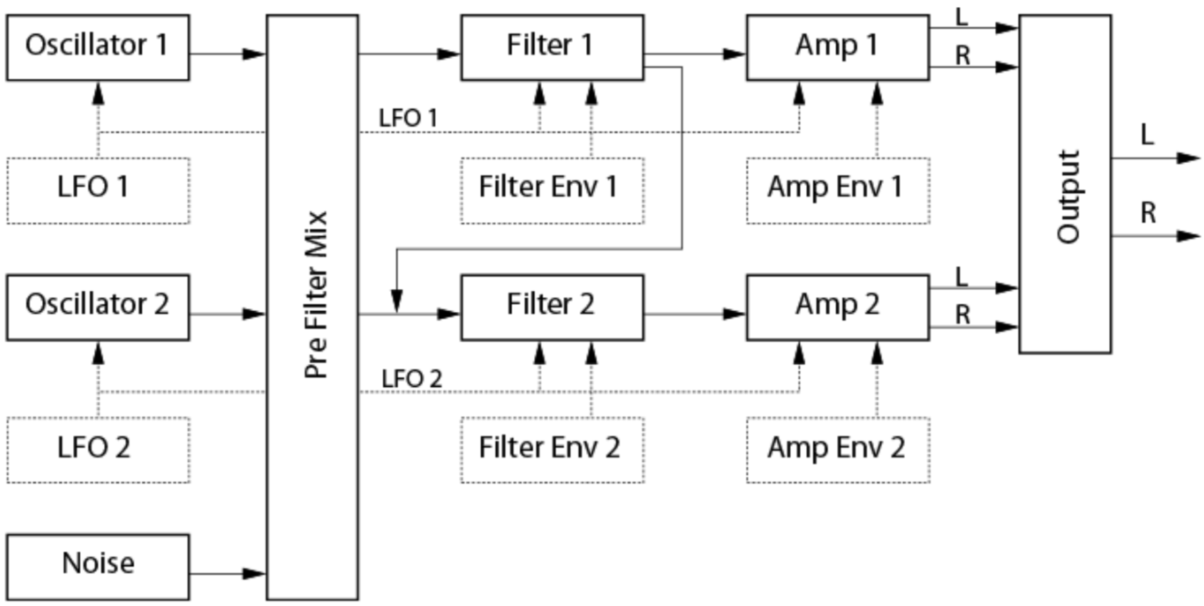


Figure 7. Diagram of Analog's signal flow (Ableton, 2015b)**.**

In this study, a great interest regarding the use of Analog is how users experience the differences between 1) user interface elements in Live, 2) the user interface of the controller and 3) the actual signal flow. In other words, the user may require up to three different conceptual models when using both Analog interfaces (Live and Push) simultaneously.

3.5. Racks and Macros

An important concept in Live is the grouping of effect and instrument elements into racks that can be controlled by eight macro controls. Although this study does not directly focus on Live's racks, it is important to understand how the concept of macro control is used across the system. A rack can contain virtually unlimited amount of devices. Devices are located in one or more *chains* that process the signal parallel to each other (Figure 8). The input of a rack is either a MIDI or audio signal, which is then forwarded to the chains. The output signal of a rack is the sum of the signals from the chains. A chain can be considered as a processing path for the signal that goes through devices.

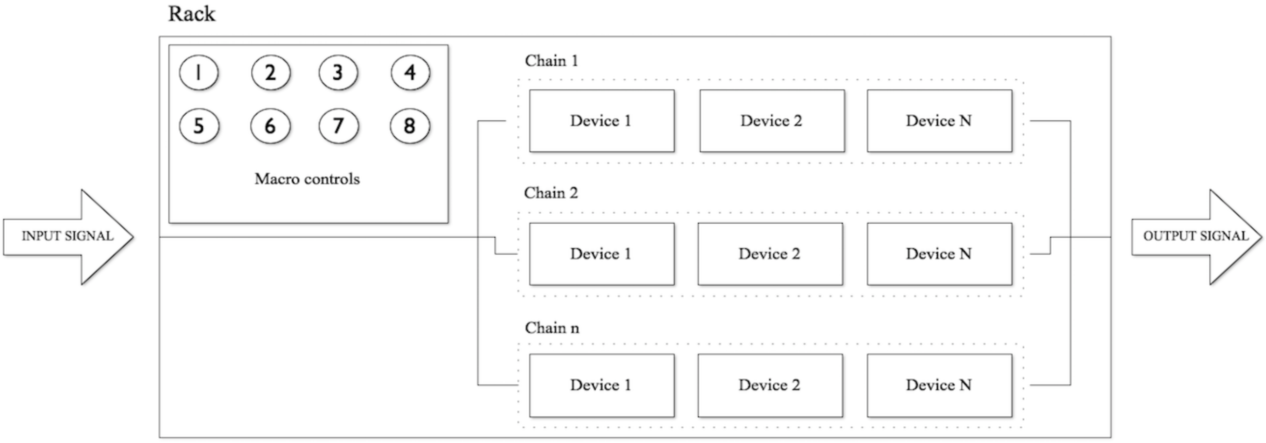


Figure 8. An illustration of rack structure.

Each rack has eight available macro controls (Figure 9) that have adjustable value ranges. Devices in the chain can be mapped to the macro controls of the parenting rack, i.e. the rack that contains the devices. The relationship between a parameter and a macro control can be considered as many to one, i.e. a single parameter can be mapped only to one macro control, but one macro control can adjust one or more parameters in any of the chains inside the rack. Therefore, macro controls can be considered as rack specific. Racks can also be nested. In such case, the rack works as any other device in the chain and the parameter mappings are always directed to the closest enclosing rack. Macro controls itself can be mapped into a physical controller either by creating fixed MIDI mappings or using control surfaces (see section 3.6).

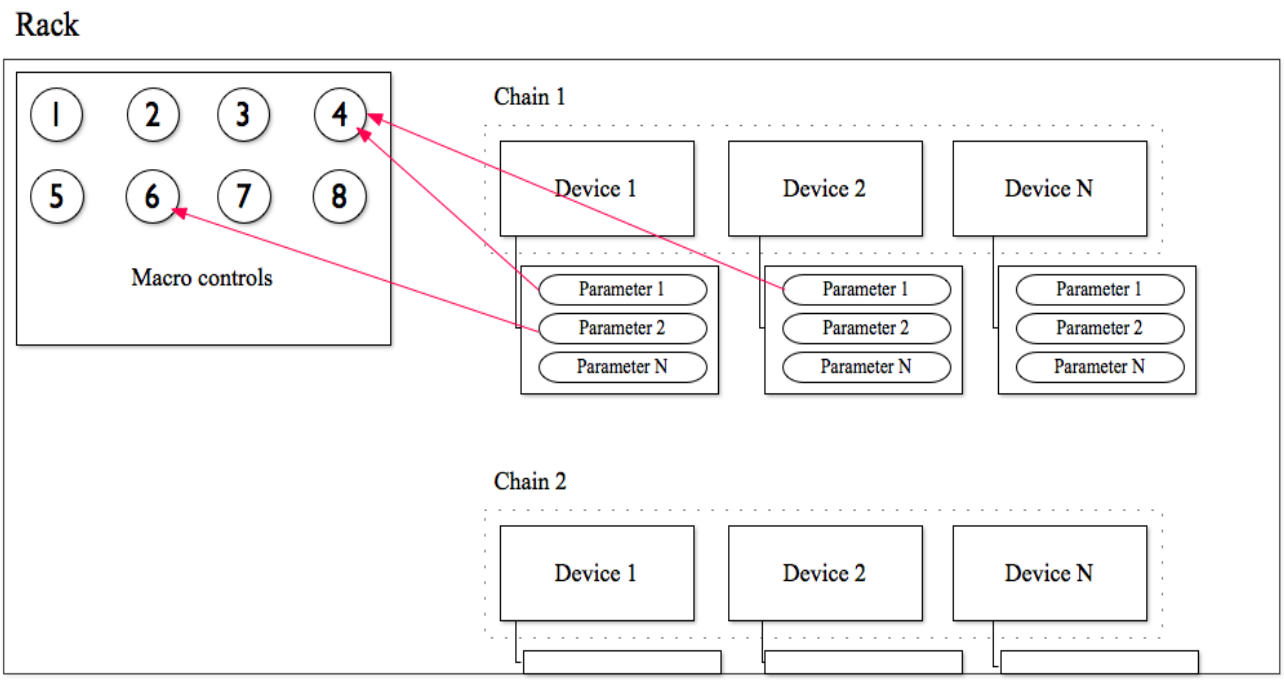


Figure 9. Device parameters can be mapped to rack's macro controls.

An example of a rack in Live is presented in Figure 10. In this example, the audio effect rack contains three chains (named “Chain1”, “Chain2” and “Chain3”). The selected chain (Chain3, indicated by turquoise colour) contains one device called Utility. From this instance of Utility, the parameters *Gain* and *Panorama* (indicated by green squares) are mapped to the macro controls 1 and 2, respectively.

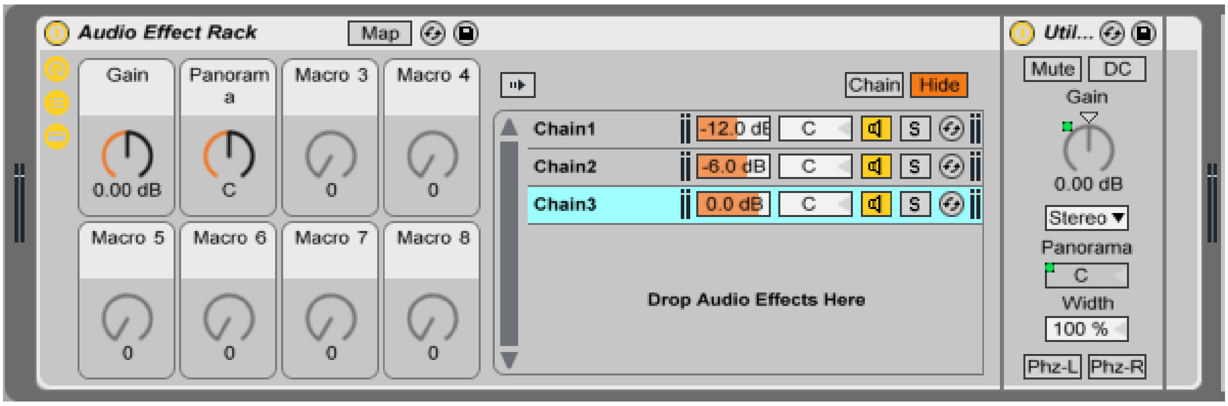


Figure 10. A screenshot of audio effect rack (Ableton, 2015c)**.**

3.6. Control Surfaces and Remote Scripts

Ableton defines Control Surfaces as “*devices containing one or more controls that are automatically assigned to parameters in Ableton Live*” (Ableton, 2015h). A key benefit of using Control Surfaces is to avoid the need for separate MIDI mapping for each parameter. The function that automatically adjusts controls to parameters is called Instant Mappings (Ableton, 2016a). Live includes a set of Control Surfaces for various controllers (Figure 11). The way each Control Surface function with the controller is defined in MIDI Remote Scripts, which are coded in Python language. The source codes for Remote Scripts are not publicly available (Bayle, 2014). However, users have a possibility to create their own Instant Mappings by editing a dedicated user configuration text file if the controller does not have a built-in Control Surface.

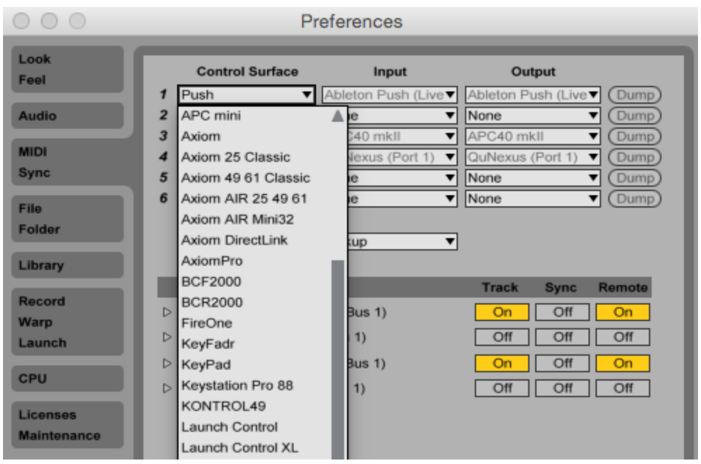


Figure 11. Selection of Control Surfaces in Live's preferences (Ableton, 2015c)**.**

Instant mappings reassign parameters to controllers when the virtual device is changed. For example, if a track contains two devices (Analog and Auto Filter) and the selection is on Analog, the controller is then assigned to Analog parameters. When the selection is switched to Auto Filter the controller interface is reassigned off from Analog to Auto Filter’s parameters. The currently controlled device is indicated with a blue-hand icon in device’s title bar (Figure 6). Users cannot modify the structure or grouping of Instant Mappings for different devices in Control Surface. The controls are assigned to predefined parameters in built-in devices. However, users have the possibility to manually override automatic parameter assigns and create fixed mappings to a parameter. The downside is that any manual assignment removes the mapped controller from automatically adjusting to other devices.

4. Mapping Strategies

Musical instrument can reflect complex relationship between the performer and the instrument. In digital realm, the connection between properties and parameters of sound synthesis and the interface used to control the sound objects is specious (Keislar, 2014). When the gesture and action are separated from the sound source, the connection between them has to be made by the system designer or user. This activity is generally described as mapping (Wanderley & Malloch, 2014). Acoustic and electronic instruments can be defined to have four parts: gestural interface, data-path, audio system and a conceptual model (Wessel et al., 2002). In this realm, computer-based instruments have greater structural and functional flexibility, as the control surface is independent from the sound source (Chadabe, 2002; Dean, 2009; Hunt et al., 2003). The space in-between can contain one or more layers that map controller events to the sound parameters. Resulting events can be enormously more complex than in traditional instrument. The way parameters are mapped directly affects how the musical events are concretised. Altering the mappings can change qualities such as controllability and expressivity of an instrument (Rovan et al., 1997).

The concept of mapping is used in the field of musical instrument research since the 1990's (Wanderley & Malloch, 2014) and is considered as an important research topic (Miranda & Wanderley, 2006). Some have defined mapping as the correspondence between control parameter and sound synthesis variable (Hunt et al., 2003; Miranda & Wanderley, 2006). Bencina et al. (2008) have defined mapping as control signals to specific sound generation strategies through body gestures and modulation. In the case of laptop orchestra (Fiebrink et al., 2007), the mapping refers to assigning musical notes to computer keyboard. Although the exact meaning of the term might depend on the context, the concept of mapping is firmly anchored in between a performer, musical interface and sonic output.

The mapping has also a significant role outside of the research field. The functionalities of a given commercial product is often reviewed and critiqued based on the technical capabilities and use in context. Understandably, the term mapping is rarely covered as discrete and analysed concept as in scientific field. However, this conventionality does not imply that concept of mapping is considered too abstract and therefore unimportant or peculiar to regular users of controllers. In contrast, the functionality and usability of controllers is continuously issued through message boards and online communities and often mapping is in key position in these discourses.

4.1. Dimensions of Mappings

Depth of mappings can be derived into different categories. Tanaka (2010) divides single input event into three types: binary, basic and expressive. In this scheme, binary mapping refers to activation of a sound. Basic mapping is a fixed parameter that affects the articulation of the sound. Expressiveness is considered to be a continuously varying parameter that reacts to the gesture. (Tanaka, 2010.) A given input can be filtered into these categories by analysing the sensor input in time. Using single sensor input as a source forka. different mappings can be described as one-to-many –mapping. To extend further, categories of this model can also be used to describe the mappings' functional purpose in a virtual device. For example, switching the on/off -state of a device might be considered as binary and expressive mapping. One-to-many mappings, such as macro controls in some cases, may include all these levels.

In addition to the depth, it is beneficial to understand the relationship between the sound and the controlling gesture. Miranda and Wanderley (2006) have categorised the relations between control parameter and sound parameter into four groups:

- One-to-one. One sound parameter is controlled by one gestural parameter.
- One-to-many. One gestural parameter affects two or more sound parameters.
- Many-to-one. One sound parameter is controlled by two or more gestural parameters.
- Many-to-many. Multiple gestural parameters control multiple sound parameters. (Miranda & Wanderley, 2006.)

A common example of controller's one-to-one mapping is a knob that is mapped to single interface element such as gain. Controllers entail the possibility for multiple input gestures to affect the same sound parameter. Adversely, it is possible to control multiple parameters with only one gesture.

4.2. Control Action

Control action is a more technical frame for previously defined musical gesture. Jensenius and Nymoen (2009) define the term control action as musician's act to create or modify sound. Control actions can be divided into two parts: one-dimensional and multidimensional. Author uses sustain pedal as an example of one-dimensional control action. There is no measurement of how much the foot touches the pedal or the weight of the foot. Therefore these variables have no effect on the sound and are considered irrelevant. In case these variables have an effect, this control action will be multidimensional.

Despite the initial usability of this model, it lacks necessary detail needed in software controlling. All of the control actions do not necessarily affect the sound directly. For example, control action may function interoperationally within the virtual device by switching binary values of sub-devices that have no immediate reaction to currently produced sound. Therefore, it is necessary to dismiss the need to alter sound when the relevancy of control action is evaluated. More accurate description is that control action needs to operate on at least single parameter of the device to be considered relevant. When operating in the context of virtual instrument control, it is also beneficial to widen the concept of sound parameters to include all the possible features of the system. For example, if a controller is designed to recognise velocity and aftertouch, but latter is not mapped to any virtual parameter, the control action itself should not be regarded as multidimensional. Moreover, if the controller's aftertouch is mapped to different virtual instrument that is currently being played it should not be dismissed as irrelevant despite the lack of direct effect on the sound. In such case, the connection between an action and sound parameter still exists in other conditions.

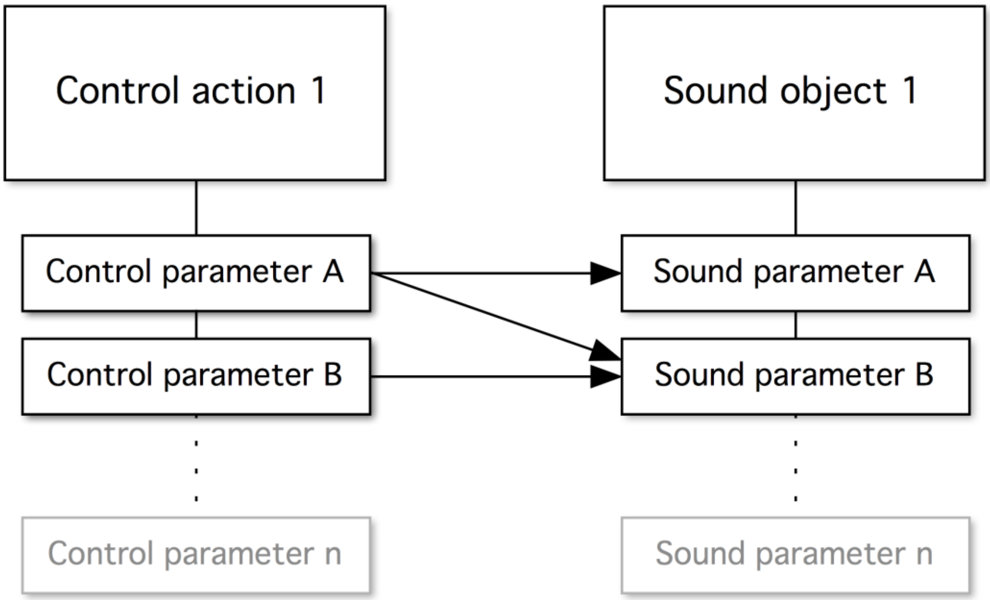


Figure 12. The relationship between control action and sound object (modified from Nymoen & Jensenius, 2009).

The control action and sound object may also be detailed by including different layers of parameters that identify the relationship between the control action and sound object on sub levels (Figure 12). Jensenius and Nymoen (2009) note that the perceptual relationship between action and object would be one-to one, although in parameter level it would be considered as many-to-many mapping. Kellum and Crevoisier (2009) covered the concept of activation conditions that are described as a definition of set of conditions for given action to trigger, for example the amount of fingers on laptop's multi-touch mouse pad. In theory, this could be used as additional control parameter or separate action. Consequently, activation conditions can also be set to be physical thresholds, for example velocity in a keyboard. Activation conditions have arguably a significant role in physical interfaces. Virtual parameter becomes concretised when the control is embedded in physical object. The way physical controller feels and responses to touch contributes to the experience of controlling sound.

4.3. Sensitivity of the System

Chadabe (2002) describes the sensitivity of the system using the amount of variables available. Fewer variables indicate more overall power for each variable. In case of two variables, only one variable can alter 50% of the whole system. Nevertheless, the system with 100 variables is arguably more sensitive due to the number of ways it can be altered.

To follow this observation, Chabade adds power and hierarchy to the concept of sensitivity, that is, grouping variables and creating hierarchical objects of which power is determined by how many parameters it contains. The ideal structure is creative rather than strict and formal, which could form unnecessary deep hierarchies. At the same time, constructing hierarchies might become harder and less useful when the system contains more independent variables (Chadabe, 2002). In addition, well-adjusted sensitiveness alone neither quarantines usability nor expressiveness of the system. Importantly, the criteria of how parameters are selected have a greater influence over sensitivity. This can be considered as an issue relating to initial design of the system. Another interesting question is how to design and align sensitivity in different sub-systems that work in parallel. An example of such case is where two virtual devices are used consecutively through the same interface. In case of controllers designed for specific software, it is rarely the case that physical controller can or should include all the functions of the software. The sensitivity of a controller system should be evaluated based on sensitivity of the sub-systems instead of the system as a whole.

4.4. Progressive Mappings and Affordances

Some studies approach instrument learning through simplifying the sensitivity of the system (Johnson et al., 2011). One of such systems is called MuSense, which is designed to help novice violinists. McDermot et al. (2013) present an idea of a growing instrument that gradually enables more actions. The idea that user can discard restrictions and make the instrument more difficult is called layered affordances. Macro controls of the device are a practical example of layered affordance. Adaptive affordance is when the system itself defines the required skill level for each step and autonomously prunes restrictions during the progress. As McDermot et al. (2013) note, similar concept is often used in games, when complexity and difficulties arise as the game progresses. Syntorial (2016) is one example of such game in musical context. The idea is to learn elements of subtractive synthesis by ear and introduce more synthesiser variables over time.

The concept of progressivity and movement has also been covered in the subject of neural networks that inter-operationally link events within the system (Kerlleñevich et al., 2011). Abstract and evolving nature of the explained system moves towards interactive music and distances itself from controller-computer paradigm, where actions are defined and traceable. At the same time it might make the concept of mappings unusable when the connection between control actions and sound operations becomes unstable (Chadabe, 2002). To use mappings effectively, they must be relatively static in control space. In context of deterministic controllers, the position and connection should remain the same, even though they would be displayed in different contexts. Using layered or adaptive affordances in mapping design would not break this coherence and therefore could be used as a tool to guide new users to internalise system more effectively. Such features are rarely implemented in commercial computer controllers or hardware devices.

4.5. Linking Mappings to User Interface Evaluation

The user interface evaluation of music controllers can benefit from previously presented concepts of mappings in several ways. For example, the desired control actions can be used to define suitable input modalities. At the same time, it can be questioned if the available sonic parameters are understandable and if the links between actions and parameters are obvious for the user. In the case of software controllers, the network of mappings is considered as a model for parallel user interface. In this paradigm, some of the user interface evaluation methods can be used to find out how users experience different interfaces.

The complexity of the controller's mappings and user interface might appear as a steep learning curve. An area of great interest is how users approach multidimensional mappings. Also, how much of the mappings should be customisable is relevant in modern controller discourse. In some cases, the sensitivity of the controller and software might differ. Therefore, a relevant question is how much sensitivity can vary between the systems before it comes noticeable and what kind of effects it has on user experience.

5. Research Methods

This chapter focuses on research methods in the field of musical controllers and the methods used in this study.

5.1. Common Research Strategies

The field of human-technology interaction has established research methods regarding usability and user experience. Although the human-technology interaction literature substantially covers evaluation of input devices (Wanderley & Orio, 2002), usability or user experience evaluation techniques are not widely adopted in the field of musical controller and interface research (Holland et al., 2013). Kiefer and others suggest its because the studies focus on results rather than methodology (Kiefer et al., 2008). However, human-computer interaction (HCI) techniques are recognised as potential method for studying user experiences in musical interfaces (Johnston, 2011).

In the field of human-technology interaction, collecting performance data in form of pre-defined tasks is a traditional method to get information from users. Subsequently, musical tasks are common in interface evaluation and musical controller research (Gelineck & Serafin, 2009; Wanderley & Orio, 2002). Wanderley and Orio favour musical tasks, such as performing phrases with different contours or playing different scales (Wanderley & Orio, 2002). Although the tasks give reasonable ground to construct benchmarks for instruments, the performance-oriented tasks might become a problem in situations when the system functions beyond the traditional instrument paradigm. A formal task study also imposes a requirement that specific and described goals can be reached (Barnum, 2011). Others have proposed to move away from the idea of using only task-based evaluation in context of musical controllers because it neglects user's experiences and the interpretations of the system (Johnston, 2011). According to McDermott et al. (2013), the direction should be taken towards experience design in context of music interaction.

A common interest is to use usability study to understand how to adjust design model according to user's model. Traditionally, the ways given system can be interpreted is divided into three models: design model, user's model and system image (Norman, 2013). Often the underlying design goal is to design a system that supports single interpretation. In contrast, Sengers and Gaver present the idea of considering multiple co-existing interpretations that avoid the risk of reducing the possibilities from niche users (Sengers & Gaver, 2006). By allowing different interpretations, approaches and uses of controller to co-exist will give the researcher more accurate depiction of the use. McDermott et al. (2013) stress the importance of specifying goals to conduct a productive research. In most interaction studies, the typical goal is to eliminate unnecessary difficulties, but in some cases it is more important to identify if the ease of use is not the priority. In such cases, a long-term study is preferred, although there are no specific and optimized methods for evaluating interfaces for long-term engagement (Wallis et al., 2013). However, it is common that the participants have a genuine interest towards the subject.

5.2. Defining Research Type

This study borrows practices from strategies and approaches that are familiar in different types of qualitative research. The research includes elements from research strategies such as case studies, ethnography, phenomenography and exploratory research. All of these qualitative methods are widely applied in human sciences as well as in the field of human-technology interaction. Metsämuuronen (2009) describes the nature of case study by questioning what can be learned from the specific case. The data is paradoxically “strong in reality”, it allows generalisations and is usually gravitated towards action (Cohen et al., 2007).

Lazar et al. (2010) define ethnography to be based on complexity of human practices that require in-depth and engaged study. The amount of cases is usually relatively small. The purpose is to embody the functional parts of the social behaviour (Metsämuuronen, 2009) and understand that the way people describe their actions is not necessarily reliable and accurate (Lazar et al., 2010). In context of human-technology interaction, the goal in ethnography is to understand how technology is used and how the design influences the use of the system (Lazar et al., 2010).

Phenomenography is a description of the perceptions of the subjects that may vary on differences in age, experiences and knowledge (Metsämuuronen, 2009). The focus is on how humans comprehend and understand the given phenomena (Metsämuuronen, 2009). Lazar et al. (2010) define exploratory research approach as allowing collecting data on unspecified problem.

5.3. Selected Research Methods

Research methods were selected in order to gather different types of data and ensure the validity of collection methods. To increase external validity of the study, it was considered important to use multiple methods to collect data. One of the general requirements was to gather data from users of Live.

The methods were selected based on their practicality, validity and conventionality. The research strategies included an online survey, sessions with tasks and interviews. A set of criteria was used to set goals for the selected methods:

1) The phenomena of mapping needed to be understood and placed in wider context. An open online-survey was selected to reveal user tendencies and trends, and to gather general data about users intentions and experiences from Live and Push.

2) To understand how users behave and work around with the controller mappings in real life situation. To address the issue, recorded user sessions were conducted to gather information that is not necessarily given out verbally by the user in other situations, such as interviews.

3) To know more about users and their experiences, a semi-structured interview was chosen as a method of choice. The validity of chosen methods was considered suitable based on traditional human-technology interaction practices and previously conducted studies by other researchers in the area of musical interfaces (Johnston, 2011).

The most essential terms were defined before any data collection was done to ensure content validity across different methods. However, during the initial discussions with the users some concerns arose with the term “mapping” due to multiple interpretations of the concept. Therefore, the scientific use of the term was considered to be too abstract and ambiguous to be used without clarifications. In addition, the user experience of the controller may be affected by several factors, such as the controller’s physical appearance, functionalities, usability as well as other user experience constructs. The effect mappings have on user experiences could not be evaluated based only on the data collected from previously conducted mapping studies. In this setting, the effects mappings may have on user experience are only directional and partial. Before proceeding with the research and data collection, it was clarified that the holistic evaluation of the controller’s user experience is out of the research scope of this study.

5.4. Data Collection Strategies

The majority of the database inquiries were made in ACM-digital library. Other notable instances for reference searches were Nime (nime.org), Computer Music Journal (mitpressjournals.org/cmj), the University of Tampere library and online web-search engines. Material related to Live and Push was gathered from Ableton website (ableton.com), Ableton user forums and online manual for Live and Push. A range of printed and digital product reviews and tutorials were helpful for finding more information on the subject. Information was also collected directly from the users, musicians and various professionals in the relevant fields as discussions and consultations. For further processing, the collected source material, information and data was divided into five main groups: literature, consultations, discussion boards, survey, sessions and interviews.

A range of challenges arose when the process of finding suitable participants for the test session started. First, it was considered hard to find Push users in Finland. With the help of Ableton, some Live users in Finland were contacted via email and inquired about their interest towards the study. In addition to the email inquiries, a range of social media platforms was used to find candidates. The absence of participants in reasonable distance shifted the session focus on to users who are not necessarily familiar with Push, but have at least some experience from Live.

The central challenges confronted during the study concerned acquiring participants, forming a functional setting for the sessions and deciding the kind of information to be collected. Finding the suitable participants for the sessions and interviews turned out to be more difficult than expected. The original idea was to find active Push users who have experience in Push and Live. Suitable users were searched from discussion boards, by using social media and direct email contacting. Along the process of recruitment, it turned out that the research setting has to be changed due to absence of suitable participants in range of reasonable distance. Concurrently, the need for experience from Push was discarded as restricting criteria. The aim of the research shifted towards any Live users who are familiar with the software and understands the essential concepts.

Another challenge was the selection of suitable methods. Two criteria for the methods were set. First, they should be able to be conducted with limited resources. Second, the methods are relevant in the field of controller and instrument research. The limitations of resources related to the inexperience in research conduction and time available for collecting and analysing data. In addition, it was considered financially problematic to travel outside of Finland to study users.

6. Survey – Push's Mappings

An online survey for Push users was conducted between February 27th and March 22th, 2015. The link for the survey was shared in public and posted in several discussion boards, Facebook groups and email messages. The target group was Push users globally. Survey was created and operated on E-lomake platform (e-lomake.fi), which is a browser-based service for conducting and collecting survey data. The survey got 42 answers in total.

6.1. Survey Design

The main goal for survey was to gain contextual and informative data from Push users to support other research methods. Lazar et al. (2010) describes surveys as useful for getting overview of a user population and stresses the general acceptance of survey as a research method. The strategy was to form compact set of relevant questions while keeping the answering time relatively short (under 10 minutes). The survey and the user-sessions were designed to trail each other to emphasise coherence of collected information. The structure of the survey and user-sessions were also tested at the same time to increase consistency. Comments were collected also from other students, researchers and Ableton employees. Before opening the survey for public it was sent to three people unfamiliar with the topic to exclude unnecessary convolution. Also, one test filling with real user was done during the test session. During the testing, some technical issues arose due to web browsers that displayed incorrect font. It was concluded that the problems were platform dependent and result of incorrect behaviour of survey service. All the major web-browsers were tested and no other problems arose. Survey was published with the knowledge of possible compatibility issues. As a result one user reported disturbing font in the survey.

The survey was targeted for all active Push users globally. The people of interest were electronic music amateurs and professionals, who have access to Push in their regular musical activities. In more practical terms, the target group was defined as a group of users who either own and use Push in their home studios, or have access to a private or shared studio environment where Push is used. However, the concept of "Push users" was not explicitly defined on survey's introduction or covering notes because it was relatively hard to predefine reasonable usage amounts that define a Push user. Predefined choices for usage are rather arbitrary as the usage can vary from intense daily use (professional) to few hours per month (amateur). Better way to capture variance of time and usage would be diary studies, as suggested by Lazar et al. (2010). Also, users' own estimation and averaging time of usage was considered non-beneficial due to expected variance in personal answering criteria. In sum, the underlying goal was to induce all users who consider themselves as Push users, despite their level of engagement with the device. Typical demographic classifications, such as age or sex were not in the area of interest.

6.2. Survey Questions

Survey questions (see appendix) were divided into six sub-themes:

1. General information (2 sections)
2. Push and Computer (1 section)
3. Push's Mappings – Ergonomic quality (EQ) (2 sections)
4. Experiences – Hedonic quality (HQ) (1 section)
5. Impressions – Appeal (1 section)
6. Open Answers (1 section)

General information section focused on information about the user's background and experience with music production. Users were asked to evaluate their experiences with Live and Push. Also, it was considered important to know if users consider Live as their primary digital audio workstation. By analysing the information given in this section users could be roughly divided into different groups based on their experience and usage strategies.

Second sub-theme was the relationship between Push and computer. Users were asked to place their distribution of usage in time into 10-point scale, where number 10 represents Push and 1 represents computer. The purpose of this question was to find out what kind of connections can be found between person's usage patterns and experiences with mappings.

Third sub-theme was Push's mappings. Users were asked to evaluate experiences when controlling devices and effects using Push. Answering was based on evaluating verbal anchors in level of agreement. For example, users were asked to evaluate how familiar they are with the mappings using five-point scale from *strongly agree* to *strongly disagree*. Verbal anchors used followed the structure of research conducted by Hassenzahl et al. (2000) on subjectively perceived ergonomic and hedonic quality. Ergonomic quality (EQ) includes quality elements traditionally associated with usability. In this context, third sub-theme focused specifically on ergonomic quality of mappings. In addition, one question focused on the preference control source (Push or computer) for device and effect control. This was also considered to reflect ergonomic preference of the user during Push-focused session.

Fourth sub-theme focused on hedonic quality (HQ) of Push's mappings. Again, a five-point scale was used. The difference between EQ and HQ is that the latter has no direct relation to the tasks, functionalities, features or capabilities of the evaluated subject. HQ anchors are selected on the basis of Push-experience from the quality aspect.

Fifth sub-theme focused on appeal of Push as whole. In the study by Hassenzahl and others the term appeal focused holistic judgement of the system. In this survey, the goal was to find out if the ergonomic and hedonic quality effects appeal. If the system is appealing, the users may experience enjoyment during use (Hassenzahl et al., 2000).

Sixth sub-theme was an open answer. Users were asked about their opinions on Push's current mappings and related problems, if any. There was also a possibility to send feedback about the survey.

6.3. Survey Distribution and Response Rate

The survey got 42 answers in total. The response rate per source site was not analysed, however the engagement towards the survey was most noticeable in Facebook and dedicated Ableton discussion board (Ableton User Forums, 2016). The feedback and attitude towards survey was mostly positive throughout the public forums. Comments from survey's feedback section showed general interest towards the subject matter. Especially in Ableton discussion board the research topic was considered important.

Some survey guidelines (Schonlau et al., 2002) emphasise the importance of password protecting Web surveys. In this research, the survey was open for everyone without login. This strategy was based on presumption that not requiring login or ID check is a trade-off between the validity of the results and amount of answers web-surveys usually receives, especially when the target population is relatively narrow.

The survey was expected to reveal general information about the phenomena of mappings and the effect it has on user experience. It was assumed that no quantitative analysis could be made from the collected data, because the amount of answers was expected to be relatively low. This assumption also affected the design of the survey.

6.4. Overview of the Correspondents

Among the correspondents, the average time with music in general ranged between 2 - 35 years, and the years using Live ranged between 1 - 11 years. The average correspondent had been making music for 15 years and been using Live for five years. There is no distinct relation between the Live usage and active years in music. More experienced users prefer another software than Live as their primary digital audio workstation. In most cases, users allocated different aspects of the music creation workflow to different software. However, this notion is not statistically generalizable but merely describes the usage patterns in this survey. When asked about current production equipment, more than 80% of the users mentioned another MIDI-controllers than Push. However, 69% used Push as their primary controller. Therefore, it can be estimated that roughly over half of the correspondents have additional MIDI-controllers at their disposal, but they still consider Push as their primary controller. It should be noted that these estimates are simply illustrative based on open questions.

6.5. Relational Effects of Live Usage

The collected data was analysed by calculating inter-item Pearson product-moment correlation coefficient for each attribute value. As a roughly constructed guideline, the correlation was considered to be meaningful within the context of the survey when surpassing 0.36

(Metsämuuronen, 2009). Interestingly, relatively high negative correlation (-0,53) was found between years of Live usage (x-axis) and ergonomic quality (y-axis) of Push's mappings (Figure 13). Negative correlation (-0,55) was also found between Live usage and hedonic quality (HQ). This indicates that, according to this survey, people with more experience in Live are more critical towards the quality of Push's mappings.

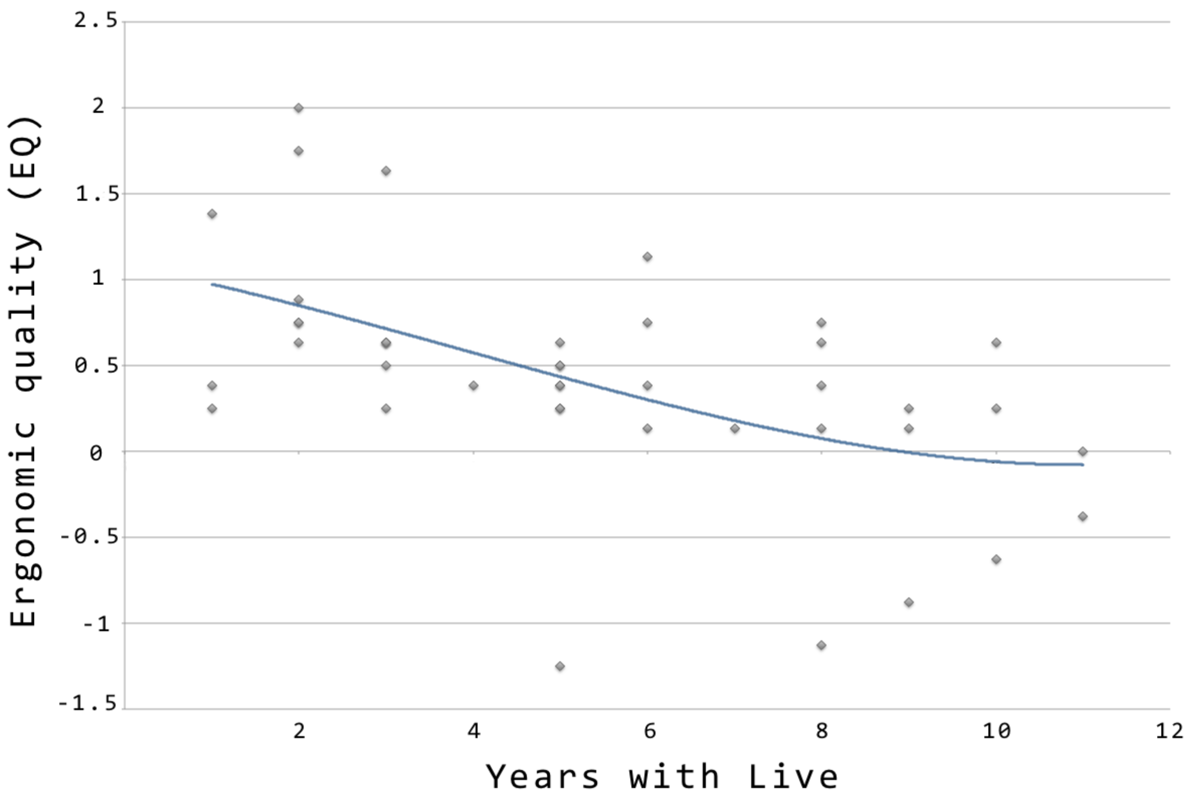


Figure 13. The correlation between Years with Live and Ergonomic quality (EQ) of Push's mappings. Regression model: polynomial order 3.

Years of Live usage affected negatively on evaluation of Push's appeal, although the correlation (-0,38) is not that strong. When users estimated their skills with Push to be above average they also tended to give lower scores on ergonomic quality than users who evaluated their skills to be rather modest. Users with higher skill levels and more experience may be more prepared to criticise functional and aesthetic features. Other possible reasons might be related to the divergence between user's expectations emerged from using Live and actual design of Push.

6.6. Relational Effects of Using Push

Users estimated their distribution of time between the controller and the computer in typical Push focused session. The time users spent on Push was estimated on the scale of 1-10. The more time spent on Push the better scores were given to hedonic quality and appeal. However, there were no correlation between time spent on Push and ergonomic quality of mappings. In sum, the appeal of

Push positively correlates with the time spent with Push, but it has no correlation on how mappings are experienced. The results show that users who experience the design and aesthetics of Push positively will spend more time with it. However, users tend to evaluate the controllability and functions separately from the form factor and external attributes of the controller. From this distinction it can be concluded that for some users a controller with strong physical appeal may compensate deficiencies in functional design.

Based on the data, the ergonomic quality of mappings and appeal has no direct relationship. The tendency to modify device environment has no relation to ergonomic quality of mappings either. The effect of mappings can be seen, when user has prior experience with Live, but mappings have little or no correlation between any quality aspects of Push. The results indicate that the process of evaluating mappings is most likely intermingled with various qualitative aspects of the controller, such as form factor of the enclosure, haptic response of interface elements and general appeal.

6.7. Open Answer and Comments

On the survey, participants commented the pros and cons of Push's mappings. Analysis revealed three main themes that were considered problematic by the users: mapping, navigation and operations.

Several comments mentioned problems regarding the logic of parameter grouping, the order of parameters and the sense of unpredictability and randomness. Needs for re-assigning and re-grouping destination parameters was also mentioned. The coherence of the parameter grouping was also criticised because of the distances between parameters. Mapping-related problems in the users' comments are summarised in Table 2.

Mapping problem area	**Sub-problem**
Illogical grouping of parameters	Lack of predictability
	Wrong order
Fixed state, not being able to modify	Parameter missing
	Not able to re-group or arrange parameters
	Not able to combine parameters
	Not able to change assigned destination parameter
	Not able to manipulate multiple parameters from different tracks simultaneously
No preferred range for parameters	

Table 2. Summary of mapping-related problems.

Problems in system navigation overlap with issues in mappings. However, it was considered necessary to distinguish problems in use of parameters and issues in navigating between them. The problems related to navigation are listed in the Table 3. Some users considered the abbreviations problematic. They were often described as cryptic or confusing, especially in case of third party plugins. The layout of parameters was also considered problematic when user navigated within the system. Some of the navigational issues related to the screen size and the limited capabilities of alphanumeric display. A multi-coloured screen with higher resolution could dispose some of the problems related to the representation of the data. Users need structured and effortless navigation within the system that displays information in an easily understandable form. These notions can be argued to apply even if the controller does not have a dedicated screen. Even relatively modest controllers still need clear structure to present data and inform user about the state of the system.

Navigational problem area	**Sub-problem**
Abbrevations	Not able to rename parameters
	Obscure
Virtual space	Locating self
	Unnecessary moves
	Dubious scrolling
Physical screen	Text-based
	Size
	No colours

Table 3. Summary of problems related to navigation.

Third problem area covers issues related to operations within Push. These issues are not solely related to previously explained mapping and navigation, but are still considered relevant. One of the central problems regarding production environment was the need for additional controller in certain situations. This was mostly due to inability to lock certain parameters in virtual space. Users also issued a need for “virtual controllers” that could run simultaneously and enable different “states” for Push. Some users commented that the actual problem was going back to the computer if the desired function was not in reach of Push’s functionality.

Only two comments focused on hardware aspect of the interface. Using rotary encoder to control toggle settings in software was considered problematic. One user considered velocity sensitivity and aftertouch predictability on pads as an issue. A separate template editor was proposed as a solution for the problems with software parameter mappings. Altering states of different types and modes of control with the same physical control is a common problem with no easy solution. A dedicated row of buttons for switching functions could help the *display/encoder* section to adapt to different situations.

6.8. Importance of Individual Preferences

The focus of the data is leaning more towards the controller's interface than to data-path and conceptual model. This is most likely a result of how the research was conducted, but also of the demography of users in this study. The data show that 48% of participants would rather do modifications into existing system than rely on default configurations. Yet, it is unclear how many of the users would modify control scripts that represents data-path layer in this case. The preference to modify controllers is evident. The formulation of survey questions was too constricted to go further in analysing intentions and goals in modifying controller.

Quite surprisingly, none of the studied factors had correlation with users' tendency to do modifications. This might simply indicate that the predisposition to modify existing hardware and software does not have generalizable influence to the subjective evaluation of the controller in use. Other explanation could be that influence could be detected, but the granularity of answers indicates more complex patterns behind the concept of customisation of musical controller.

7. Sessions and Interviews

The user-sessions and interviews were run between February and March 2015 for five Live users at participants' home studios. The sessions were conducted in the cities of Tampere and Turku, Finland. The purpose was to construct informal and conventional environment for the user and minimise artificial factors, i.e. the participants are familiar with their production and composing environment. In laboratory setting the challenge is to simulate real-world usage and represent the actual physical environment where tasks would be performed (Oztoprak & Erbug, 2006; TecEd, 2015). The purpose for organising sessions "on the field" as opposed to the laboratory was to get users involved with the tasks and minimise external factors that might affect thinking processes. During the preparations of the sessions, one of the participants proposed the idea of joining another session due to practicalities. Therefore, this participant performed the session outside of the regular setting.

All of the five user sessions were conducted by following a pre-determined structure. However, the first session was considered to be a test session, where the structure, tasks and flow of the session were tested and confirmed. The user session had four main parts: preparation, tasks, questionnaire and interview. Preparation included mounting of recording equipment and initialisation of Live and Push. Task-phase included four different tasks conducted with Push and Live. After completing the tasks participants were asked to fill the online questionnaire. Sessions ended with a semi-constructed interview (see appendix). The questions were formed based on estimation of the most essential qualitative aspects of controller usage. The structure of the interviews included the following sub themes: general, mappings, physical interface, synthesis, and interface conflicts. If the participants had any specific problems during the tasks, the issues were discussed during the interview. Each participant had slightly different set of questions to match observations and comments from tasks and to minimise repetition.

7.1. Session Participants

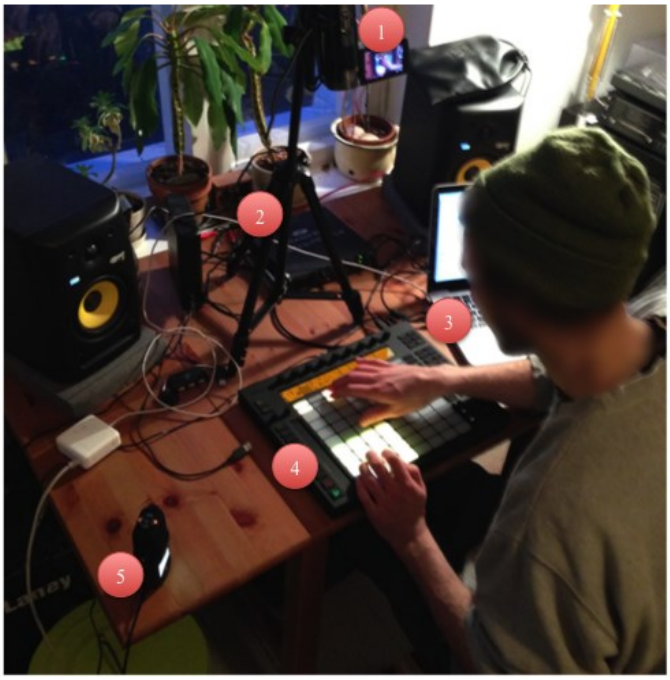
In order to apply study results to target group it was necessary to focus on participant selection criteria. As an example of used criteria in controller research, Gelineck and Serafin (2009) required that users make their own music, they have officially released music and they fit into overall category of electronic musicians. In this study, the users for the sessions were chosen based on their familiarity with Live and experience with musical controllers. To construct comparable test settings, it was made sure that users knew how to navigate in Live. One user had prior experience with Push, but was not a current user. The time making music varied from 7 to 15 years and the experience with Live from two to eight years. All of the participants stated that Live is their primary digital audio workstation and rated their skills from one to two on a five-degree scale (five being an expert user). The summary of participants' profiles can be found from the Table 4.

Participant	Years making music	Years with Live	Skills with Live (1-5)(self-estimation)
A	10	8	2
B	15	5	2
C	10	3	2
D	14	8	1
E	7	2	1

Table 4. Summary of session participants.

7.2. User-Sessions Preparations

Feasibility of the test setting was assured by confirming that users had a copy of Live 9 Suite version 9.1.7 installed. In addition, it was made sure that users have enough physical space for Push to be located into reasonable distance from computer and speakers (i.e. user's typical working place) and that the recording equipment have enough space to be set up properly. During the preparation it was made sure that Live's audio and midi settings are configured accordingly and Push is working properly. The buffer size was set to maximum of 128 samples to ensure appropriate responsiveness of the controller. Video camera was positioned over the controller to record user's hand movements and Push's display. Although the video camera also recorded audio, a more adjustable handheld recorder was placed near the test setting to record discussion and comments during the sessions. An example of the test setting is presented in the Figure 14 and the list of equipment used during the sessions can be found from the Table 5.



1. Video camera
2. Tripod
3. Computer
4. Push
5. Portable recorder

Figure 14. An example of a test setting.

Users were given general information about the research and informed about the structure of the session. All participants signed a consent form. Users were also verbally told that they could stop the test in any moment, in case any discomfort arises during the session. Similar to the study by Gelineck and Serafin (2009), the session started when users were asked to get familiar with Push. They were given approximately 20 minutes to explore and get a "feel" of the controller. The purpose of exploring was to give enough time to conceptualise the tactile reaction of the buttons and knobs before starting the actual tasks.

Article	**Supplied by the session conductor**	**Supplied by the participant**
Push	X	
Computer with Live 9 installed	X	X
Screen recording software	X	X
Portable recorder	X	
Video camera	X	
Tripod	X	
Additional MIDI-controller		X
Cables, data-storage, batteries, etc.	X	

Table 5. A list of session equipment and items.

7.3. Structure of Tasks

The users were instructed to perform a set of pre-defined tasks during the session (Table 6). A task-based approach was chosen based on previous studies and observations on how to measure musical interfaces (Gelineck & Serafin, 2009; Hunt et al., 2003; Johnston, 2011; Wanderley & Orio, 2002). Wanderley and Orio (2002) consider tasks as benchmarks that allows to compare different musical interfaces. However, in this study, the primary idea of the task was to direct users to engage themselves with Analog in more detail and evaluate users by enforcing them towards similar activities.

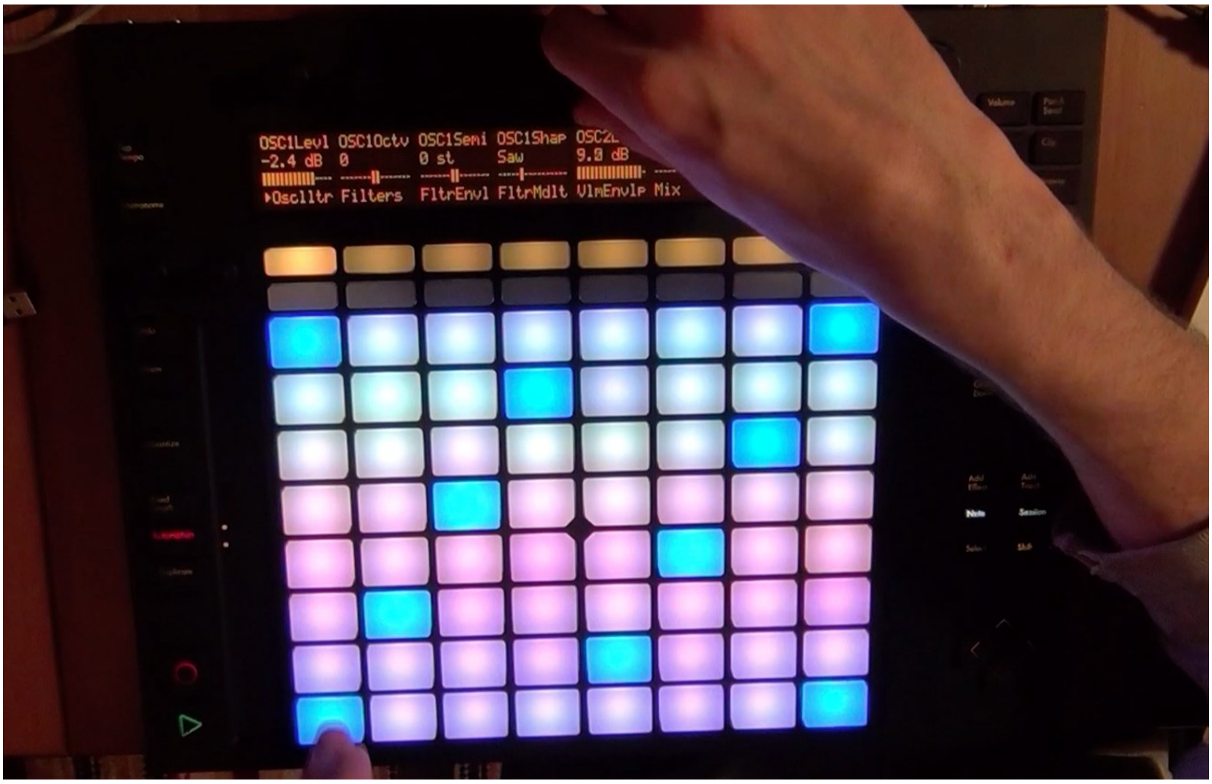


Figure 15. A still shot from a video recording. A participant is performing tasks.

Each participant concluded four different tasks (Figure 15). Tasks one and two focused on using only Push. Tasks three and four focused on using Live by traditional computer input and output modalities (keyboard, mouse and screen). Prior to tasks, users were given the option to do a backing track for following tasks. Backing track was suggested to be a simple drum loop that would play on the background during the tasks in order to simulate a real situation, where the composing and sound design starts from unfixed position and user can decide the starting point. Alternatively, it was suggested that a metronome could be used for rhythmic reference. Then participants were instructed how to navigate between devices, how to browse and swap devices and presets, and how to navigate within selected device's parameters. The purpose of this introduction was to make sure that participant knew how to add device, add effect and select different instrument before proceeding with the tasks. In addition, the functionalities of buttons *Devic*e, *Browse* and arrow keys were explained briefly. After the introduction the participants were verbally given a set of tasks (Table 6).

Task	Task Description	Time Reserved (minutes)
1	On Push, program any type of sound using Analog. The sound can be, for example a bass, lead or pad. If possible, please speak out loud what you are doing and thinking.	15
2	On Push, add any type of audio effect to the previously made Analog track. If possible, please speak out loud what you are doing and thinking.	10
3	On computer, program any type of sound using Analog. The sound can be, for example a bass, lead or pad. Alternatively, you can continue to edit the sound created in the first task. If possible, please speak out loud what you are doing and thinking.	15
4	On computer, add any type of audio effect to the previously made Analog track. Alternatively, you can continue to edit the effect created in the second task. If possible, please speak out loud what you are doing and thinking.	10

Table 6. Summary of the tasks.

Task 1. Program a sound using Live's device Analog with Push's interface. The sound could fit into any common category, such as bass, lead or pad. To minimise the possibility of confusion in sound design stage, the participants were told that the pre-set should be changed if it included any other devices. The programming of the sound was limited to Push and participants were instructed not look at the computer screen during the completion of the task.

Task 2. **Add an audio effect to the sound.** The effect could have been any of the Live's native effects. In traditional testing manner the participants were encouraged to think out loud and describe what they were doing during the task (Nielsen, 2012). Participants were again instructed not to use computer while adjusting the parameters of the effect.

Task 3. **Program a sound using Analog's graphical user interface instead of Push's interface.** In case the participant wanted to continue the sound created in the first task, they were allowed to do so. Participants were allowed to use external MIDI keyboard for note triggering during the sound design.

Task 4. **Add an audio effect to the sound using the graphical user interface.** Similar to the previous task, participants could continue working with an effect selected in the second task.

The primary goal of the tasks was to see and analyse how users behave with the controller. By using the same construct and structure in tasks, it was expected that the level of common usage-patterns between the users could be determined. A set of questions were formed to assist further analysis of the material collected from the tasks:

- How users experience the mappings of the parameters?
- What aspects of mappings are relevant to the new users of the controller?
- What kind of generalisations can be made about the users' behaviour and the use of mappings?

7.4. General Findings from the Sessions and Interviews

The general observation from the sessions was the positive and relaxed attitude towards using Push. None of the users displayed any visible signs of frustration or anxiety during the sessions. All users performed each task as they were designed within the reserved timeframe. Users expressed general interest towards the upcoming activities.

All of the five users had not used Push earlier except participant C, who no longer uses Push, but still considers Ableton as a main DAW. During the introduction phase participants asked general question regarding navigation, selecting and location of functions they assumed would be accessible from the controller. The users also made realisations that helped to achieve their goals, such as understanding the folder structure while browsing. After loading Drum Rack on the track, a common strategy was to start playing pads to trigger audible sounds. Only one participant asked how to adjust the length of the sample before getting familiar with the pads. The questions asked in introduction phase were about recording a pattern, note repeat, pattern length, quantization, and moving an event in a sequence.

Two somewhat distinct approaches towards Push can be conducted from the users' behaviour during the tasks. In the first approach, users focused on familiarising and understanding the device. The focus was on sensing the feel and responsiveness of pads and knobs, as well as occasionally asking confirming questions about functions and navigation. The second approach was to start working immediately and ask detailed questions how various specific actions can be performed, for example how to nudge recorded MIDI notes. The most relevant differences between users were time spent per tasks and the type of sounds they wanted to create. None of the users stopped working on tasks, even when they confronted insurmountable challenges, such as not finding a desired parameter or not understanding the functioning of the parameter.

The differences between the described approaches reflect mostly users' personal preferences and are not likely derived from the controller's characteristics. Yet, it is very likely that the physical appearance of the device affected users' receptiveness. In such cases, the design may consequently alter users' approach strategies. Users described the Push's physical features as inviting, beautiful and impressive. Such positive remarks might have led users to soften their negative comments and observations at later stages. In addition, only one of the session participants had used Push earlier. It is likely that the first impression on Push changes over time if the controller is used actively.

Users had similarities in several areas regarding the task completion. First, the time spent on tasks was noticeably longer on Push than on a computer. Second, users focused on altering different parameters depending on which interface they were using. For example, two of the users (B and D) spent majority of the time on Push on sub-page called *Osclltr* (oscillator). In Live, oscillator related parameters got significantly less attention. For most users, tasks started altering the same parameters such as *F1 Freq* (filter 1 cut-off), *F1Rsnanc* (filter 1 resonance), and *OSC1Shap* (oscillator 1 wave shape). Changing focus to different parameters was considerably faster when using Live. Similar behaviour was present in other sub-banks of parameters. When using Push for tasks, users tended to stay longer on a given parameter page and modify adjacent parameters, but on graphical user interface the equivalent parameters were not altered.

Another difference between tasks performed with computer and Push was the frequency of using melodic reference during the task. On Push, all of the users played different notes or chords between cycles of parameter changes, and parameters were rarely adjusted with both hands. Similar behaviour did not exist on tasks that were performed with computer. Only one user wanted to use a MIDI-keyboard for note triggering when using the computer.

Based on the interviews conducted after the sessions, the general approach towards Push was positive. Users were curious about the interface and interested in using the device. One participant expected Push to be more complicated and difficult to operate and considered the responsiveness of the device to be better than expected. No one of the interviewed considered themselves as experts in editing or modifying synth sounds. All participants preferred the idea of having a pre-set as a starting point for the sound creation. Participants considered themselves familiar with the main operational principles and components in a subtractive synthesiser, but none of the participants had used Analog extensively prior to the sessions.

When participants were asked about their preferred method for learning a new controller the most common answer was trial and error. All of the participants stressed the importance of practical use and familiarising themselves with the controller. Other main sources for learning were user communities, tutorials and manuals. One participant stated that new controllers might lead to new approaches in music creation. It was considered important that in the learning phase the default mappings are clear and logical.

7.5. Controlling the Parameters

The overall controllability of Push was considered good. Controlling the parameters via interface was considered logical and fluent, although most users stated that the overall range of functionalities was not clear.

The participants had mixed views on abbreviations in Analog and the selected audio effects. On the other hand, the abbreviations were considered to be logical and straightforward, but in some cases they seemed to be unnecessary complex or confusing. The familiarity of specific Live device was considered to create confusion in controlling when the desired parameter was known to be based on the visual location in graphical user interface, but the specific naming of the parameter was unknown.

> "The most common ones are easy to understand, but some of them were quite exotic." – A comment on abbreviations used in Analog.

The input modalities of Push were considered to meet the standards. Using knobs instead of sliders was thought to be a good choice. The most often requested feature was the fine-tuning of parameter values. The ideas for improvements included different modifier buttons for parameter control and for fine-tuning or returning to the default value. One participant preferred less sensitive stepped encoders as oppose to un-stepped endless encoders. Two participants considered the encoders to be too "sensitive" for detailed control. The ability to lock parameters into their position (i.e. they would stay on the screen in every sub-page) was thought to be not necessary, although it would help adjusting and designing sound in some contexts.

7.6. Working with Analog and Audio Devices

Some of the participants considered the structure of the signal flow to be unclear from the Push interface. On the other hand, the grouping of parameters was considered logical when compared to the GUI of Analog. Two of the participants noticed the parameters on the device's main page were so called best-of parameters. It was added that the selection of Analog's main parameters were sufficient and work well for quick editing on the device. Rest of the participants commented that they either skipped the main page or did not notice any of the main page parameters.

> "I like it as it is. It's clear." – A comment on switching parameter banks.

The participants were inquired if the parameter mapping corresponded with the signal flow. One user considered that the controller's interface enforces users to re-learn Analog as a device. Rest of the users stated that the order of parameters makes sense and the grouping is in most cases logical. One participant stated that there was a problem setting the polyphony to mono. Switching between groups of parameters was thought to be functional.

7.7. Impressions

In most cases, Push was expected to be more difficult to use and operate than it eventually was. From broader perspective, Push was described to be interesting, inviting and "good looking". The use of colours was described to bring depth to the interface, although it also adds complexity.
Users were asked about the possible benefits Push would bring to their music creation workflow.

The considered benefits were:

- Sketching new ideas
- Create demos
- Create backing tracks
- Get inspired
- Get away from computer screen
- Tactile feedback

Most of the benefits focused on simplifying the workflow in early stages of creation. Consequently, the participants doubted that it seems unlikely that they would use only Push through the song creation process. The interface was considered unable to deliver required definition and detail for intensive tasks such as using EQ. None of the participants were willing to replace a traditional keyboard with Push or use it extensively as a primary controller.

> "Based on my short experience (with Push), I would like to see what it could offer to me. Trying out new things is refreshing." – A comment from a participant.

The participants were asked if the impression of Push matches the impression of Live. One of the participants stated that Push can be seen as a method to access Live features and it can be compared to any other input methods, such as mouse or keyboard. However, it is more streamlined and therefore Live and Push can be considered to be different things. Another difference mentioned related to the disparity in workflows that traditional computer and controllers pose. On the contrary, one of the participants considered the similarity in concepts and the form factor of the controller to follow the conceptual framework of Live.

> "For me, Live and Push are different things, because I am not accustomed to use controllers."
> – A comment from a participant.

Four months after the interviews, the participants were asked if their impression towards Push and Analog were changed. None of the participants had used Push after the sessions, but on average the views on the controller were changed to be more positive. In addition, Analog was also considered to be slightly better than in previous interview.

7.8. Interruptions in Flow and Tendency to Modify

The participants were asked what kinds of things end their flow state during the music creation. The most common reason was a technical problem that is usually caused by the inability to understand how the interaction between a controller and software is designed. Rest of the issues related to the level of controllability and visual distraction. It was considered problematic if the controller's mappings were limited or difficult to adjust, for example selecting the input and output sources of a track.

> "I have noticed that if I don't find the right sound easily it will break my flow. Also, it is necessary that the controller works seamlessly with the DAW, so I don't have to troubleshoot during the session." – A comment from a participant.

The participants evaluated their tendency to modify Live's behaviour between the software and the controller on 1 to 10 scale. The average estimation between the participants was 5. Most of the participants stated that they prefer to get familiar with the default mappings of a controller rather than focus on modifying the controller or software. The only exception was live performing situations where more customised control was considered better.

> "I have never been too interested in adjusting controllers to work with the software so that I could use the all of the possible functions." – A comment from a participant.

When the participants were asked if they create their own racks in Live, the tendency to modify the devices was more recurrent when compared to the preference of modifying general controller and software interaction. Two of the correspondents stated they prefer to learn the system as it is rather than modify it. The most common workflow between the participants was to combine devices into racks to switch on and off multiple devices at the same time. Saving certain combinations of devices as pre-sets for future use were found useful, the main benefit being the time saved in configuration of device settings.

> "Racks work great for layering synth sounds." – A comment from a participant.

Overall, users' tendency to modify varied significantly. In most cases the desired modifications were relatively non-technical and required only basic knowledge of Ableton Live and it's features. Participants were asked if they use any control surfaces with their controllers. Only two of the five participants used control surfaces for their controllers. The main benefit was the Instant Mapping of devices. None of the users had modified Remote Scripts. Technical background did not seem to have clear connection to users' tendency to modify. It might be possible that in creative context technicality or technical knowledge itself is not a significant driving factor. It is reasonable to expect that user's, who are more accustomed to technical frameworks, are also keen to modify systems. It seems, however, that in context of computer music, user behaviour is a bit more complex.

7.9. Unity Mismatch in Analog

Interfaces of virtual devices in Live and Push differ significantly. Once the graphical user interface of the device in Live is translated into text-based interface on Push, then certain type of information is lost. Even the sensitivity between these devices differs drastically. Approximately 38% of the Analog's adjustable parameters that are visible in Live can be controlled via Push's interface. There are also logical differences that might become a problem. For example, the control for oscillator 2 detune is only available in "best of" controls on the main tab. Only two of the participants noticed that the same parameters were not available in both interfaces.

7.10. Comments from nativeKontrol

Third party developers offer custom Remote Scripts to use with Push. An example of such script is PXT series developed by nativeKontrol. PXT series is a collection of scripts that changes Push's default functionalities and behaviour (nativeKontrol, 2015). Sam "Stray" Hurley, the developer at nativeKontrol, was interviewed (17th of April, 2015) for the process of creating PXT Live, one of the three available Push extension scripts, to get a developers' perspective on the subject matter. According to Hurley, the design strategy behind PXT Live was to utilize Push's controls as in default Push script, or in a more sensible and intuitive manner. Some of the PXT-Live features, such as polyphonic sequencing, have later become a feature of the original Push through the official updates.

As described in the chapter two, most of the software dependent controllers rely on specific data-path structure. One of the downsides in hardware and software integration is, as Hurley points out, the lack of standardized communication mechanisms. The result is interoperability between the controllers and software from different developers. Hurley mentions preset switching as an example of non-interchangeable operation. If software and virtual instruments would operate over standardized communication interfaces, the controllers could be developed to work more sophistically across software. "*The controllers should be relatively simplistic, adhere to a standard and be usable in a variety of software via scripts and plugins specific to the software. As an example, the Mackie Control is useable in a wide range of software via a relatively simple protocol. I would like to have a standard developed that would utilize and expand upon a similar protocol*" states Hurley.

Regarding concept of progressive mappings (section 4.4), Hurley considers racks and macros as an example of simplified device control. The complexity of the devices is hidden and the macros provide the control to the most important parameters; "*The majority of scripts for controllers work quite well for controlling rack macros and also allow you to control devices inside the rack.*" he says. Regarding the input modalities, Hurley considers Push to have appropriate variety of control. However, additional touch stripe or X-Y -pad would increase the controllability in his opinion. On the subject of creating matching digital and physical interfaces, Hurley focuses on the distinction between general-purpose and dedicated instrument controller. Hurley notes that in the case of general-purpose controller, it does not make sense to design interface to match specific instrument. To the question if Push is an instrument, Hurley's answer is: "*No, I think it's a controller that makes Live feel more like an instrument.*"

8. Discussion

The study framework was concluded from and based on the academic field of electronic controller research and design. The research included user's evaluation of the controller, user-tests and interviews. The field of human-technology interaction has established research methods for measuring user-experience and usability of software and devices, but it can be argued that the routines have inadequacies when measuring these concepts in creative context. This study was a novel experiment to see how the common methods work in musical context.

8.1. Experiencing New Interfaces

According to the survey and session participants, users considered Push to be pleasing and impressive. In sessions, Push received positive feedback about construction and design. Overall criticism related to the functionality and ergonomic quality of the device. According to the survey, users who have previous experience with Live tend to be more critical in these areas. This observation can be explained based on the presumption that experienced users have generally more structured and solidified working habits, which consciously or unconsciously define the criteria for new objects. In some cases, the previous experience with different controllers and software creates expectations of how things should work. In general, the adaption of a new device should benefit the user in practice. However, in some cases users specially expect divergence to their workflows, as one of the participants noted. Users might experience new ways of working as refreshing or even a desired quiddity. If the features and functions of a controller are otherwise acceptable, the deviant design might be received as positive variety From the developers' standpoint, these issues are supposedly less important when the controller is designed from scratch and without strictly defined framework, such as existing software interface. In the case of Live, the software interface has existed considerably longer than the dedicated controller, Push. In conjunction, Live's interface has enabled various working methods to propagate into different workflows, which are more likely to be stabilized among the more experienced users. It can be generalised that the likeliness of adapting a new controller interface depends, among other things, on user's previous experience, the degree of openness towards new interfaces and the considered improvements that the new interface would bring into personal workflows. In conclusion, adapting a new interface is more critical process for experienced users than for beginners.

8.2. Degree of System Sensitivity

Traditionally, the connection between software and controller follows the system paradigm of master and slave, where the software is the master and the controller is a slave. Although the connection is not necessarily unidirectional in terms of transferred data, it is important to notice that a pure controller does not process the signal or generate logical operations or functionality that are not specified in the software. Yet, a controller can have interface-dependent capabilities that are not possible to perform using traditional input methods such as mouse and keyboard. However, virtually no controller includes all the functionalities of the software, especially not the capabilities of a modern digital audio workstation (DAW). Even in relatively modest non-DAW master system's it is relevant to pay attention to the degree of sensitivity. If the virtual parameter can have any value from zero to one hundred in two decimal accuracy, it would require the physical controller of the parameter to have the same accuracy in sensitivity. It can be argued that it is not necessary nor of purpose to level sensitivities in different interfaces, however, the divergence between the sensitivities should be designed with consideration. The system's sub components should be in-line in terms of sensitivity and serve the general purpose of the controller. If the user cannot control the virtual device with the desired accuracy, the user might consider the controller to be insufficient for a specific task. In sum, the sensitivity of a controller should match the initial purpose and be consistently accurate between sub-systems.

As noted previously, virtual devices in Push differs in sensitivity when compared to Live. According to Ableton, majority of the general differences between the controller and the software are due to design choices and the planned functionality and purpose of the controller (Ableton, 2015f; Rothwell, 2013). It is reasonable to exclude obvious design choices and consider them as macro-sensitivity. Analogously, the sensitivity in sub-systems can be considered as micro-sensitivity. For example, in the case of Analog, the fact that some of the parameters are not included in Push's interface is clearly a micro-level difference. According to the results, it is still questionable if and how the issues in micro-level sensitivity affect the user experience. It might be that in a relatively complex system the minor differences are not as noticeable, but instead the users are more sensitive to the consistency between the sub-systems. This could explain why users gave positive remarks on controllability even though they could not control the desired parameters. It is possible that users focus more on macro-level sensitivity, i.e. the design and functionalities, and evaluate the system based on more general observations of the controller.

8.3. Interfaces

The sessions revealed an interesting phenomenon common to all users. The approach towards the same virtual device (Analog) was different depending on the interface. Users spent more time on individual parameters on physical interface than in graphical user interface. In addition, users tended to focus on different parameters depending on the interface. The reason might be behind the appeal of the physical interface that leads to additional experimenting. Usually, users tend to react positively to the possibility to physically control parameters. The reaction may be commonplace, but it might be that the longer dwelling times are not necessarily a sign of an immersive interface. Instead, the user might simply spend more time on the interface looking for the right parameters.

One of the common issues in controller interfaces is finding the balance between available interface elements that controls a wide range of virtual parameters. For example, knobs are not ideal for toggle settings, and non-automated faders do not work well when the same fader is used for different parameters. Still, endless encoders are a good compromise for controlling various parameters with the same element. Next step could be adding additional mechanics to the components or adapting 3D-touch based solutions.

8.4. Using Test-approach Might Hinder Real Usage Patterns

Conducting a session-based research has several downsides in context of music creation. The most obvious issue is the factitious nature of the setting. Creative activities can be highly idiosyncratic. It is very unlikely that the same test setting contributes to each individual's personal preferences. The unfamiliarity of the controller was considered a possible bias during sessions. Participants were often very keen to spend more time on Push compared to Live. One of the reasons might be the novelty of the controller. This might have led the participants to spend more time on physically touching and feeling the controller responsiveness. It is virtually impossible to separate genuine enjoyment and curiosity from task-driven need to follow given orders. Another major problem in the research was recruitment of users with limited resources. It could have been better to focus on more general controllers to reach more users.

8.5. Users Differ

The term mapping is used in different contexts depending the purpose (Bencina et al., 2008; Fiebrink et al., 2007; Hunt et al., 2003; Miranda & Wanderley, 2006). The benefit of the term is its flexible nature. Mapping can be used to describe the relation between various objects as long as it comprises the fundamental components. In context of the relationship between a musician and a musical controller, the concept of mapping works best when there is a sufficient degree of freedom. According to the study results, computer musicians are not equally interested in the communication methods between the interfaces. Even though the process of creating computer music is strongly linked to natural sciences such as mathematics and computer science, it seems that the people working in digital environments are not necessarily inclined to modify the inner workings of the systems they use. Perhaps the reason is that in creative contexts the actual activity of creating is self-prioritised much higher than the capability to enhance the system. For some users, the learning of the default workings of a system is already a compulsory impediment. Based on these observations, it might be justified to focus on mappings more as a creative element rather than a purely technical feature and user-dependent activity. On the contrary, Nishino (2011) observed that computer music programming is a major domain of expert end-user programming. Nishino describes computer musicians as users who have stronger expertise in music domain than in computing. However, end-user programmers should be considered as expert in their own right instead of musicians that are deficient programmers (Blackwell & Collins, 2005).

8.6. Deeper Integration or Open Mapping Environments?

When the controller and software are integrated, the user is presented to two interfaces. A notable question is how to assure the users from different backgrounds experience appropriate level of freedom in mappings, when they start the process of adapting the whole system. Should mappings be freely customisable or does the system benefit from holistic and closed design of mappings? Consequently, do the users benefit from the ecosystem paradigm?

Digital systems and environments differ greatly when they are compared in their approach towards mapping. As Sam Hurley from nativeKontrol pointed out, there is a lack of standardized communication mechanisms. Today, several hardware companies are pushing forward their own independent mapping schemes for virtual instruments. It can be questioned if it is beneficial to move towards closed mapping systems, as they rely on specific hardware or software. On the contrary, a uniform and documented protocol for communication between controllers and software would arguably improve the user's ability to modify the interaction between controllers and software and would increase the interoperability between systems from different manufacturers. Users are, as customers, more constrained to invest time and money into products that are not inter-operational, but have similar features.

According to the results, users had problems in predicting control positions, order and resolution. In addition, the lack of possibilities to edit and customize mappings was considered problematic. The survey revealed that 48% of the users would like to do modifications to the system instead of relying on default configurations. Yet, the current market of controllers lacks the encompassing technology in area of customising and editing mappings in software instruments and effects. In some cases the MIDI messages sent by the controller can be edited in dedicated software, for example MPK mini MIDI Editor (AkaiPro, 2016), QuNexus Editor (Keith McMillen Instruments, 2013) or Controller Editor (Native Instruments, 2015a). Some software focuses on editing the actual Live remote scripts (Isotonik Studios, 2016). However, the underlying problem is that seemingly none of the current commercial controllers and editors support open communication layer that would go beyond rudimentary and limited message or parameter editing.

Bayle (2014) aptly describes Live's Remote Control Scripts as the brains of a controller. It means that the functionality and logic is stored in data-path layer interface that translates the messages sent from the controller. Yet, in most cases the logic is hidden from the user. Even though the installed base of both virtual instruments and controllers can be considered relatively large, the absence of modern and well-deployed communication frameworks in commercial products makes the community based development of software and controller interaction slow and difficult.

8.7. End-user Development and Mappings

The community of Push users have made a significant effort on redesigning the mappings of Push (Ableton User Forums, 2016). By decompiling pyc-files that construct Remote Script the end-users are able to modify mappings of parameters as well as other aspects of Push's functionality. This aligns well with Burnett's and Scaffidi's (2015) definition of end-user development wherein end-users design and customise user interface and functionality: the activity is based on the knowledge of the domain and context, and the users' needs. End-user development can also be an ideal scenario within the domain of programming, because of the close connection to the real life situations (Nishino, 2011). The customised mappings tend to solve concrete problems or add relatively small features that the users think are missing from the original design. The act of mapping customisation is as form of end-user development that redefines requirements to support common objectives.

The customisation of Push's communication with Live is possible because of the semi-open technological framework. Conceptually the focus is on data-path where the messages from and to the controller are interpreted and passed along. Interestingly, users exploit the possibility to alter this communication to make the end product more usable and satisfying. Remote Scripts could be fully open for end-user development in terms of documentation and resources. Another possible way to approach the customisation would be a dedicated editor environment that capsules logical operations into easier to understand components, similar to Automator (Apple, 2016) in OS X.

8.8. Development of Push During the Research

Live and Push received several updates during the research (Ableton, 2016b). Most notably, a next version of the controller Push 2 was introduced in November 2015 (Ableton, 2015e). Push 2 included several new features and changes in design. New higher resolution RGB display amended some of the mapping related problems presented in section 6.7. However, older version of Push was also updated in Live version 9.6. The update included changes for parameter mappings in form of dynamic parameter banks. The feature disposes unnecessary parameters that are not needed based on the state of the other related parameters. In case of Analog, the feature arguably improves the clarity of parameter groupings and consequently adds missing parameters that were not previously visible in Push's interface. Still, some of the problems related to abbreviations still exist.

In version 9.6, the parameter names include the full path to parameter. For example, "FEG1STme" refers to filter 1's envelope generator sustain time. In average, three to four characters of total eight are used to describe parent location of the parameter. It can be argued if discarding the path information from the parameter name could decrease the time spent on recalling the target of abbreviated parameter. All of the eight characters could be used solely for the parameter. However, this could lead to navigational issues such as locating self when the user browses banks further and the parent bank information disappears. It can be argued that the issues with abbreviations are strongly connected to the available screen resolution, concerning all text-based interfaces.

8.9. Future Work

The mappings of a musical controller are a central part of the interface design and the logic of the controller's functions. Almost any physical interface presents an interesting research topic in the field of human-technology interaction. In case of computer music, it would be interesting to study if the concepts like mapping could be used in other physical interfaces or non-musical contexts. The conducted research focused on only one strategy of mapping a virtual device for a controller. No alternative mappings on Analog were tested. It would of interest to know how different mappings of a virtual instrument affect the users, and how they are experienced in relation to each other. The difference could be measured within one controller with different mappings or with same mapping strategy with different interfaces. To go further with analysis, it would be beneficial to store logs of parameter changes and study if the amount of parameter changes correlate with measured user experience. Other fruitful direction for research could be studying user communities and their activities related to the customisation of controllers and methods to modify existing technologies.

9. Conclusion

The premise for this work was to study the interaction between user, musical controller and computer. The range of controllers available for computer music creation is diverse. The concept of software controller relies on communication between physical controller and software. The term mapping may be used differently depending on the context, but in most cases it used to describe how the communication between interfaces is structured and what kind of strategies and protocols it follows.

Commercially available controllers present different solutions for managing the communication between interfaces. One strategy is to rely on general protocols such as MIDI. Other strategies include combining software and hardware into one unit that has more than one user interface. Ableton's Live software and Push controller is a great example of combined software and hardware. Push was selected as a subject for further study of the concept of mapping in contemporary context.

The aim of the study was to use the concept of mappings together with simple user experience metrics to find possible connections between them. The research was conducted using tested and tried methods, such as survey, sessions and interviews. The study revealed no distinct or defined connection between mappings and user-experience. It is very likely that the mappings have an important role on how the interface is experienced, but based on the conducted research the complex construction of user experience on a musical controller requires more profound research methods.

Conducting a user experience research in musical context is challenging because of the complex nature of creative activities. User behaviour and reasoning behind the actions are not easily discovered and using typical metrics on interface evaluations do not necessarily reveal how users experience them. Musicians approach controllers in various ways and use different strategies to learn and work with them. Generally, the need for tuning the environment varies between users. In some cases users prefers modifying only some aspects of the controller, but would like to keep other parts intact.

Push was considered to be a well functional controller with capable features. It was interesting to notice how Ableton's design goals were in line with users' views on the controller. The participants had used Live prior to the test setting, which helped users to accustom Push's concepts. The results from the sessions were modest in terms of how much can be stated from the controller's functionality. When the studied controller is new for the users it may preferable to simplify research space into more easily analysable form.

Other important theme that grew along the research was the lack of standardised solution for mapping commercial controllers. The diversity of user-base and the idiosyncratic structure of production environments are based on environments that communicate together. It can be argued that the closed integrations between software and controllers limit users capabilities to come up with interesting ideas. Ableton's concept of Remote Scripts works relatively well as a mapping medium and it supports various controllers. If the mapping medium would be fully open, it could enable users and controllers to work more intelligently and creatively with the software.

References

Ableton. (2013). Live 9 and Push are here. Retrieved September 20, 2015, from https://www.ableton.com/en/press/press-archive/press-archive-release-9-push/

Ableton. (2015a). Ableton Archive. Introducing Ableton Live 7, New Instruments and Ableton Suite. Retrieved September 20, 2015, from https://www.ableton.com/en/pages/2007/live_7_announcement/

Ableton. (2015b). Live Instrument Reference. Retrieved September 20, 2015, from https://www.ableton.com/en/manual/live-instrument-reference/#analog

Ableton [Computer software]. (2015c). Live version 9.1.7. Retrieved from http://www.ableton.com/

Ableton. (2015d). Natively-supported Control Surfaces in Live 9. Retrieved December 30, 2015, from https://www.ableton.com/en/help/article/live-9-native-supported-control-surfaces/

Ableton. (2015e). Press Resources. Retrieved September 20, 2015, from https://www.ableton.com/en/press/

Ableton. (2015f). Push. Retrieved September 20, 2015, from https://www.ableton.com/en/push/

Ableton. (2015g). Using Push. Retrieved September 20, 2015, from https://www.ableton.com/en/manual/using-push/

Ableton. (2015h). What are Control Surfaces and how do I use them? Retrieved September 20, 2015, from https://www.ableton.com/en/help/article/control-surfaces/

Ableton. (2015i). What is Live? Retrieved September 20, 2015, from https://www.ableton.com/en/live/

Ableton. (2015j). Working with Instruments and Effects. Retrieved September 20, 2015, from https://www.ableton.com/en/manual/working-with-instruments-and-effects/

Ableton. (2016a). Getting Started 3: Setting Up Your MIDI Controller. Retrieved January 7, 2016, from https://www.ableton.com/en/help/article/getting-started-3-setting-your-midi-controller/

Ableton. (2016b). Live 9 Release Notes. Retrieved January 10, 2016, from https://www.ableton.com/en/help/article/live-9-release-notes/

Ableton User Forums. (2016). Ubermap. Retrieved January 10, 2016, from https://forum.ableton.com/viewtopic.php?f=55&t=201849

AkaiPro. (2016). MPK mini mkII. Retrieved January 7, 2016, from http://www.akaipro.com/product/mpk-mini-mkii#overview

Apple. (2016). Introduction to Automator Programming Guide. Retrieved January 10, 2016, from https://developer.apple.com/library/mac/documentation/AppleApplications/Conceptual/AutomatorConcepts/Automator.html

Attack Magazine. (2015). Ableton Push. Retrieved September 20, 2015, from http://www.attackmagazine.com/reviews/gear-software/ableton-push/1/

Barnum, C. M. (2011). *Usability testing essentials: ready, set...test!* Elsevier Inc.

Bayle, J. (2014). Ableton Live 9 MIDI Remote Scripts. Retrieved September 20, 2015, from http://julienbayle.net/ableton-live-9-midi-remote-scripts/

Bencina, R., Wilde, D., & Langley, S. (2008). Gesture ≈ sound experiments: process and mappings. *Proceedings of the 8th International Conference on New Interfaces for Musical Expression (NIME08)*, 197–202.

Blackwell, A., & Collins, N. (2005). The programming language as a musical instrument. *Proceedings of PPIG05 (Psychology of Programming Interest Group)*, (June), 1–11.

Burnett, M. M., & Scaffidi, C. (2015). End-User Development. Retrieved September 20, 2015, from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/end-user-development

Chadabe, J. (2002). The limitations of mapping as a structural descriptive in electronic instruments. *Proceedings of the 2002 Conference on New Interfaces in Musical Expression*, 1–5.

Cohen, L., Manion, L., & Morrison, K. (2007). *Research methods in education* (6th edition). Routledge.

Computer Music. (2015). Are software-specific MIDI controllers a good idea? Retrieved September 20, 2015, from http://www.musicradar.com/news/tech/are-software-specific-midi-controllers-a-good-idea-608537

D'Arcangelo, G. (2001). Creating contexts of creativity: musical composition with modular components. *Proceedings of the 2001 Conference on New Interfaces in Musical Expression*, 37–41.

Dean, R. (2009). *The Oxford Handbook of Computer Music*. Oxford University Press Inc.

Dobrian, C., & Koppelman, D. (2006). The E in NIME: Musical Expression with New Computer Interfaces. *Proceedings of the 2006 Conference on New Interfaces in Musical Expression*, 277–282.

Fiebrink, R., Wang, G., & Cook, P. R. (2007). Don ' t Forget the Laptop : Using Native Input Capabilities for Expressive Musical Control. *Proceedings of the 2007 Conference on New Interfaces in Musical Expression*, 164–167.

Gelineck, S., & Serafin, S. (2009). A quantitative evaluation of the differences between knobs and sliders. *Proceedings of the 2009 Conference on New Interfaces in Musical Expression*, 13–18.

Hassenzahl, M., Platz, A., Burmester, M., & Lehner, K. (2000). Hedonic and ergonomic quality aspects determine a software's appeal. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI 00*, *2*(1), 201–208.

Henke, R. (2015). Ableton Live. Retrieved March 19, 2016, from http://roberthenke.com/technology/ableton_live.html

Hewitt, M. (2011). *Harmony for Computer Musicians*. Course Technology.

Holland, S., Wilkie, K., Mulholland, K., & Seago, A. (Eds.). (2013). *Music and Human- Computer Interaction*. London: Springer.

Hunt, A. D., Paradis, M., & Janhan Sousa, W. (2003). The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, volume 32 issue 4, 429-440.

Hunt, A., Wanderley, M. M., Kirk, R., Pompidou, I. C., & France, P. (2000). Towards a Model for Instrumental Mapping in Expert Musical Interaction. *Proceedings of the International Computer Music Conference. (ICMC'2000).*

Isotonik Studios. (2016). Preditor Version 1.2 is now Available. Retrieved February 27, 2016, from http://isotonikstudios.com/sigabort/isotonik-preditor/

Johnson, R., Linden, J. van der, & Rogers, Y. (2011). Prototyping for the Wild : reflections on the design of MuSense a lightweight wearable practice aid for the violin.

Johnston, A. (2011). Beyond Evaluation: Linking Practice and Theory in New Musical Interface Design. *Proceedings of the 2011 Conference on New Interfaces in Musical Expression*, (30 May - 1 June 2011), 280–283.

Jordà, S. (2001). New Musical Interfaces and New Music-making Paradigms. *Proceedings of the 2001 Conference on New Interfaces in Musical Expression*, 46–50.

Keislar, D. (2014). About This Issue. *Computer Music Journal*, *38* (3), 1–2.

Keith McMillen Instruments. (2013). QuNexus Editor Manual. Retrieved April 1, 2016, from https://files.keithmcmillen.com/downloads/qunexus/QuNexus_FullManual_1.1.1.pdf

Keith, S. (2010). Bridging the Gap: Thoughts on Computer Music and Contemporary (Popular) Electronic Music. *Proceedings of the 2010 ACMC.*

Kellum, G., & Crevoisier, A. (2009). A Flexible Mapping Editor for Multi-touch Musical Instruments. *Proceedings of the 2009 Conference on New Interfaces in Musical Expression*, 242–245.

Kerlleñevich, H., Eguía, M. C., & Riera, P. E. (2011). An Open Source Interface based on Biological Neural Networks for Interactive Music Performance. *Proceedings of the 2011 Conference on New Interfaces in Musical Expression*, *2* (June), 331–336.

Kiefer, C., Collins, N., & Fitzpatrick, G. (2008). HCI Methodology For Evaluating Musical Controllers: A Case Study. *Proceedings of the 2008 Conference on New Interfaces in Musical Expression,* 87–90.

Kobayashi, S., & Akamatsu, M. (2005). Spinner : A Simple Approach to Reconfigurable User Interfaces. *Proceedings of the 2005 Conference on New Interfaces in Musical Expression*, 208–211.

Kvifte, T. (1989). Instruments and the Electronic Age. Solum Forlag.

Lazar, J., Feng, J. H., & Hochheiser, H. (2010). *Research Methods in Human-Computer Interaction*. *Research Methods in HumanComputer Interaction*.

Machover, T. (2002). Instruments , Interactivity , and Inevitability. *Proceedings of the 2002 Conference on New Interfaces in Musical Expression*.

Magnusson, T., & Mendieta, E. (2007). The acoustic, the digital and the body: A survey on musical instruments. *Proceedings of the 2007 Conference on New Interfaces in Musical Expression*, 94–99.

McDermott, J., Gifford, T., Bouwer, A., & Wagy, M. (2013). Should Music Interaction Be Easy? In *Music and Human-Computer Interaction* (pp. 29–48).

Metsämuuronen, J. (2009). *Tutkimuksen tekemisen perusteet ihmistieteissä*. International Methelp.

Miranda, E. R., & Wanderley, M. (2006). *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard (Computer Music and Digital Audio Series)*. A-R Editions Inc.

Music Radar. (2014). The 19 best DAW software apps in the world today. Retrieved September 20, 2015, from http://www.musicradar.com/tuition/tech/the-19-best-daw-software-apps-in-the-world-today-238905

Native Instruments. (2015a). Controller Editor Manual.

Native Instruments. (2015b). Introducing Native Kontrol Standard. Retrieved December 30, 2015, from http://www.native-instruments.com/en/products/komplete/keyboards/komplete-kontrol-s-series/nks/

nativeKontrol. (2015). nativeKONTROL PXT Series for Ableton Push. Retrieved September 20, 2015, from http://www.nativekontrol.com/PXT_Series.html

Newton, D., & Marshall, M. (2011). Examining How Musicians Create Augmented Musical Instruments. *Proceedings of the 2011 Conference on New Interfaces in Musical Expression*, (June), 155–160.

Nielsen, J. (1996). The Anti-Mac Interface. Retrieved October 31, 2015, from http://www.nngroup.com/articles/anti-mac-interface/

Nielsen, J. (2012). Thinking Aloud: The #1 Usability Tool. Retrieved September 20, 2015, from http://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/

Nishino, H. (2011). Cognitive Issues in Computer Music Programming. *Proceedings of the 2011 Conference on New Interfaces in Musical Expression,* (June), 499–502.

Norman, D. a. (2013). *The Design of Everyday Things: Revised and Expanded Edition*. MIT Press.

Novation. (2015). Automap. Retrieved December 30, 2015, from http://us.novationmusic.com/software/automap#

Nymoen, K., & Jensenius, A. R. (2009). A discussion of multidimensional mapping in Nymophone2. *Proceedings of the 2009 Conference on New Interfaces in Musical Expression*, 94–97.

Oztoprak, A., & Erbug, C. (2006). Field versus laboratory usability testing: A first Comparison. *Developments in Human Factors in Transportation, Design, and Evaluation*, 205–212.

Poepel, C. (2005). On interface expressivity: A player based study. *Proceedings of the 2005 Conference on New Interfaces in Musical Expression,*, 228–231.

Propellerheads Software. (2005). Propellerhead Software Opens ReMote Protocol to Developers of Control Surfaces. Retrieved April 4, 2016, from https://www.propellerheads.se/press/37-propellerhead-software-opens-remote-protocol-to-developers-of-control-surfaces

Rothwell, N. (2013). Ableton Live 9 & Push. *Sound On Sound (March)*, 22–41.

Rovan, J. B., Wanderley, M. M., Dubnov, S., & Depalle, P. (1997). Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In *KANSEI - The Technology of Emotion*.

Schonlau, M., Fricker, R. D., & Elliot, M. N. (2002). Guidelines for Designing and Implementing. *Conducting Research Surveys via E-Mail and the Web*, (1997), 41–53. Retrieved from http://www.rand.org/pubs/monograph_reports/MR1480.html

Sengers, P., & Gaver, B. (2006). Staying open to interpretation: engaging multiple meanings in design and evaluation. *Proceedings of the 6th Conference on Designing …*, 99–108.

Syntorial. 2016. A new way to learn synthesis. Retrieved April 3, 2016, from http://www.syntorial.com/

Tanaka, A. (2010). Mapping Out Instruments, Affordances, and Mobiles. *Proceedings of the 2010 Conference on New Interfaces in Musical Expression*, 15–18.

TecEd, I. (2015). Field Usability Testing. Retrieved March 19, 2016, from http://teced.com/services/usability-testing-and-evaluation/field-usability-testing/

Wallis, I., Ingalls, T., Campana, E., & Vuong, C. (2013). Amateur Musicians, Long-Term Engagement, and HCI. In *Music and Human-Computer Interaction* (pp. 49–66).

Wanderley, M. M., & Malloch, J. (2014). Editors' Notes. *Computer Music Journal*, *38*(3), 4–5.

Wanderley, M. M., & Orio, N. (2002). Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, *26*(3), 62–76.

Wessel, D., & Wright, M. (2002). Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*, *26*(3), 11–22.

Wessel, D., Wright, M., & Schott, J. (2002). Intimate Musical Control of Computers with a variety of Controllers and Gesture Mapping Metaphors. *Proceedings of the 2002 Conference on New Interfaces in Musical Expression*, 1–3.

Appendix 1: Survey

This survey is aimed for **Ableton Push users**. The survey is a part of a master's thesis conducted at the University of Tampere, Finland. No additional information is collected. All the data is collected, stored and analysed for research purposes only. All the answers are anonymous, if email address not given.

This survey is not organised by Ableton. All the questions regarding their products or services should be directed to their respected website: www.ableton.com. If you have any questions regarding the survey, please send an email to Kuivanen.Ilkka.O@student.uta.fi

In this survey, you will be enquired about your background in music production; environment, software and hardware usage and your experiences with Ableton Live and Push softwares. Filling in the survey takes approximately **10 minutes**. Please answer all questions as accurately as you can, and remember: YOUR answers is the most important. Have fun answering!

Please NOTE: This survey has been tested with latest versions of Chrome, Firefox and Safari running on OS X. Please try another browser if the layout is unreadable.

Please NOTE: If you have used Push only for few times try to answers based on those sessions.

User Survey - Push

General information

* How many years have you been making music (acoustic or electronic)?

* How many years have you used Ableton Live? ?

	Yes	No
Is Live your primary DAW?	○	○

	Yes	No
Is Push your primary controller?	○	○

Evaluate your skills with Live:
- Beginner
- Familiar
- Fluent
- Expert
- Certified trainer

Evaluate your skills with Push:
- Beginner
- Familiar
- Proficient
- Expert
- Certified trainer

Next >>

Page 1 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

General information

* Briefly describe your current music production setup. Write down the most used hardware (f.e controllers or synths) and software (DAWs) that you are using at the moment.

Select the answer which best describes you.

- I like to do modifications to find out ways to improve my workflow even if it breaks (down)/modifies factory settings.
- I rather learn the built-in functionality so I can learn how the software or hardware was designed to work.

<< Previous Next >>

Page 2 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

Push / Computer

Try to remember your typical Push focused sessions and roughly estimate how much do you spend time on Push compared to computer during the session. Do not count sessions that are not focused on using Push. In other words, if your setup consists multiple different devices and software, count only those sessions where Push is used.

On the following question:
Number 10 refers using only Push during the session.
Number 1 refers using mostly computer during the session.

	10	9	8	7	6	5	4	3	2	1
The time spent on Push / computer is approximately	○	○	○	○	○	○	○	○	○	○

<< Previous Next >>

Page 3 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

Push's mappings

In this study, mapping refers to the connection between a button or a knob to a corresponding software parameter. Push includes various different functions, such as adding clips, adjusting volumes and changing tempo.

This study is focused on **mappings of devices and audio effects**. An example of mapping of an effect in Push is the way how Auto Filter's parameters are organised in Push's screen (Figure 1). In this example, the most left knob is altering the cut-off frequency, next one controls resonance and so on.

Figure 1. Auto Filter mapped on Push's screen (Ableton, 2015).

Evaluate your experiences when using Push's to control devices and audio effects. Consider the quality of Push's mappings that are related to device and audio effect controlling. Please mark your answers below.

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Comprehensible					
Supporting					
Complex					
Predictable					
Clear					
Shady					
Controllable					
Familiar					

<< Previous Next >>

Page 4 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

Push's mappings - 2

When using Push, do you prefer to control your devices and audio effects directly from the Push or by using the computer?

- I prefer using Push
- I use both, mostly Push
- I use both, mostly Computer
- I prefer using computer

<< Previous Next >>

Page 5 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

Experiences

This time think about your overall experiences with Push. You can also add your own comments on later pages.
Please mark your answer to reflect how you feel about Push.

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Interesting	○	○	○	○	○
Ordinary	○	○	○	○	○
Exiciting	○	○	○	○	○
Standard	○	○	○	○	○
Impressive	○	○	○	○	○
Original	○	○	○	○	○
Innovative	○	○	○	○	○

<< Previous Next >>

Page 6 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

Impressions

Please evaluate your impressions of Push. Please mark your answer to reflect how you feel about Push.

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Pleasant	○	○	○	○	○
Good	○	○	○	○	○
Aesthetic	○	○	○	○	○
Inviting	○	○	○	○	○
Attractive	○	○	○	○	○
Sympathetic	○	○	○	○	○
Motivating	○	○	○	○	○
Desirable	○	○	○	○	○

<< Previous Next >>

Page 7 / 10

User Survey - Push

Open answers

What is good in Push's current mapping of devices and audio effects?

What kind of problems do you see with Push's mappings? What would make your experience of controlling the devices and effects better?

<< Previous Next >>

Page 8 / 10

User Survey - Push

Thank You!

Thank you for participating.

Your answer is highly appreciated. It helps me to do the research.
If you have any comments or feedback, feel free to write it down below.
If you have a specific question, please email: Kuivanen.Ilkka.O@student.uta.fi

Kind Regards,

Ilkka Kuivanen

Feedback:

Your email (optional):

<< Previous Next >>

Go to next page to submit the survey.

Page 9 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

Form is going to be submitted.

Proceed

<< Previous Finish (Send all Data)

Page 10 / 10

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

User Survey - Push

The data has been successfully saved!

Järjestelmänä Eduix E-lomake 3.1, www.e-lomake.fi

Appendix 2: Interview Questions

Theme	Question
General	How did you feel when you used Push for creating and modifying sound?
	What do you think about controlling the parameters? Did you understand what was displayed on the screen all the time?
	Do you think that modifying the sound with Push is intuitive?
Mapping	What do you think about the abbreviations of the parameters?
	Did you notice the “best-of” parameters?
	What do you think about the parameter groupings? Are they logical? Would you change the order of banks or specific parameters? For example, would you like to change one of the “best of” parameters?
	Would you like to “lock” one of the parameters, so it would stay to its position?
	Can you think of any other ways to select banks or parameters?
	Did you notice any difference between the amount of total parameters when you used Push and when you used Live?
	What did you like about the mappings of device parameters?
	What would you change in the mappings of device parameters?
Physical interface:	What do you think about the knobs? Evaluate how well they worked when you adjusted the parameter?
	What do you think about the resolution of the knobs? Was it detailed enough?
	Can you think of any extra features or functions that the knobs could do?
	Is there anything that you would change in the behaviour of knobs? For example, would you like the knobs to be more analogous to the parameter that is being altered?
Synthesis	Are you familiar with subtractive synthesis? Do you prefer to make your own patches or do you prefer to use presets?
	Have you used Analog before? What do you think of Analog?
	Did the mapping of parameters follow the signal chain of the Analog?
	Were you satisfied with the available controls to edit your patch?
Interface conflicts	Did you felt a need to go to the computer during the tasks? If so, describe what were you doing?
	What kind of image do you have on working with computer and Push at the same time?

	Do you think that Push is a separate entity?
	Which one, computer or Push, would you be more involved and feel more creative?
Closing questions	How would you evaluate your performance on the task?
	How would you benefit from Push?
	Would you like to have your own Push?
	Is Push an instrument?
	Would Push replace your keyboard or other controller?
	What do you think about the tasks when researching a controller?
	What do you think about the whole session?
Follow Up questions after the interview	Think about those situations where your “flow state” is disrupted. What controller related reasons you can think of? The question concerns all controllers.
	Do you think that Push matches the mental image of Live? Are they different things? If so, why?
	On scale 1 to 10, how interested you are to modify the interaction between Live and a controller.
	Do you create, use or modify device effects or racks in Live? What kind of benefits do they offer you?
	Do you use control surfaces? Do you think that control surfaces deepen the level of interaction?
	Have you modified Live’s remote scripts, for example writing your own Python code?
	Have you used Push after the session? Has your impression of Push changed in any way?
	Have you used Analog after the session? Has your impression of Analog changed in any way?
	What are the best strategies for learning a new controller?
	If you have any other comments, feel free to write it down.