Wilson, Thomas R

The Gestural Control of Audio Processing

THE GESTURAL CONTROL OF AUDIO PROCESSING

Thomas Robert Wilson

A thesis submitted to The University of Huddersfield in fulfillment of the requirements for the degree of Master of Science by Research

The University of Huddersfield

January 2015

Copyright Statement

Abstract

Gesture enabled devices have become so ubiquitous in recent years that commands such as 'pinch to zoom-in on an image' are part of most people's gestural vocabulary. Despite this, gestural interfaces have been used sparingly within the audio industry. The aim of this research project is to evaluate the effectiveness of a gestural interface for the control of audio processing. In particular, the ability of a gestural system to streamline workflow and rationalise the number of control parameters, thus reducing the complexity of Human Computer Interaction (HCI). A literature review of gestural technology explores the ways in which it can improve HCI, before focussing on areas of implementation in audio systems. Case studies of previous research projects were conducted to evaluate the benefits and pitfalls of gestural control over audio. The findings from these studies concluded that the scope of this project should be limited to two-dimensional gestural control. An elicitation of gestural preferences was performed to identify expert-user's gestural associations. This data was used to compile a taxonomy of gestures and their most widely-intuitive parameter mappings. A novel interface was then produced using a popular tablet-computer. This facilitated the control of equalisation, compression and gating. Objective testing determined the performance of the gestural interface in comparison to traditional WIMP (Windows, Icons, Menus, Pointer) techniques, thus producing a benchmark for the system under test. Further testing is carried out to observe the effects of graphic user interfaces (GUIs) in a gestural system, in particular the suitability of skeuomorphic (knobs and faders) designs in modern DAWs (Digital Audio Workstations). A novel visualisation method, deemed more suitable for gestural interaction, is proposed and tested. Semantic descriptors are explored as a means of further improving the speed and usability of gestural interfaces, through the simultaneous control of multiple parameters. This rationalisation of control moves towards the implementation of gestural shortcuts and 'continuous pre-sets'.

Table of Contents

Thesis Body Word Count - 24422

List of Figures

List of Tables

Dedication.

This Thesis is dedicated to the memory of Ronald Kershaw. With who's final advice to "Keep your nose to the grindstone" was a constant inspiration for the continuing pursuit of this research project.

Acknowledgements.

I'd like to thank Steve Fenton for his supervision and guidance throughout the year. His repeated encouragement and tuition furthered the horizons of this project and my academic career, especially in convincing me to publish AES papers as additional work. Berlin 2014 will not soon be forgotten.

Thanks to the Music Technology Students & Staff of Huddersfield University and Blackpool Sixth Form for volunteering to take part in numerous interface tests. I couldn't have done it without you all.

And finally, thanks to my parents & family, for supporting and understanding my desire to 'stay in school'.

1 - Introduction

A gesture can be broadly defined as a communication through movement (Nielson et el, 2008). The human endeavour to communicate fluently using gestures has produced incredibly elaborate sign languages that facilitate the non-verbal articulation of endless emotions, observations and ideas. Conversation between people that are experienced in sign language can seem as effortless as it is instantaneous. They display a level of fluidity and complexity that current methods of Human Computer Interaction (HCI) are, as yet, unable to achieve. Notably, A start-up company called 'Motionsavvy' hope to change this by developing a Leap-Motion based translator between sign language and speech (Steinmetz, 2014). The technological challenge lies in developing a system that can interpret and process the intricacies of complex hand gestures. If a gesture recognition system could capture and process all the subtle nuances of a hand gesture, then interaction with software could be vastly improved. Workflow would be streamlined and usability heightened. The ultimate goal of HCI development is to remove as many boundaries between the user and the computer as possible, making communication with a digital system fluent and intuitive. The intrinsic characteristics of gestural control, such as the ability to execute them without a GUI (Graphical User Interface), can offer improvements over 'soft buttons' (Bragdon et al, 2011). This project evaluates the suitability of a gestural interface for the control of real-time audio processing, with the intent of removing some of the boundaries between the mix engineer and the audio data stream. Thus enabling a more immersive mixing experience. The project aims to conclude whether a gestural approach could rationalise the mix environment by reducing the complexity often present in WIMP (Windows, Icons, Menus, Pointer) (Hinckley & Wigdor, 2011) GUIs.

1.1 - The Historical Development and Popularity of Gestural Control

Technological advances have allowed the proliferation of gestural HCI platforms. There are two main branches of gestural control, two-dimensional and three-dimensional. These have primarily remained separated by differing gesture recognition techniques, *touchscreen* and *freeform* (Dan Saffer, 2009).

1.1.1 - Two-Dimensional

Two-dimensional gesture recognition systems typically employ a touch surface to gather x-y coordinate information about the user's movements. The development of multi-touch sensitive screens provided a platform to implement numerous two-dimensional gestures. Bob Boie produced some of the earliest of these systems at Bell Labs in the early 80s (Dan Saffer, 2009). This innovation opened the doors to more intricate gestural control, where users were no longer confined to the WIMP 'point and click' environment or single touch 'pen gestures' (Long et al, 2000). Bill Buxton, a computer science professor at the University of Toronto, was a pioneer of the developing technology and began working with multi-touch surfaces in 1984 (Buxton, 2014a). Buxton was also a music enthusiast and advocates the use of gestural interfaces for the control of audio, praising early products such as the Roland CF-10 'digital fader' touch-sensitive MIDI interface for offering more fit-for-purpose methods of sound manipulation (Buxton, 2014b).

Interest in multi-touch was significantly popularised by Jeff Hann, demonstrating his work producing cheap, scalable touch surfaces at a TED convention in 2006 (Heller, 2011). His design also included pressure sensitivity, which is a feature that remains unimplemented in modern touch screen devices (Park and Nieto, 2013). This technology exhibited a practical 'third dimension' of gestural data that could be detected on a platform formerly limited to two dimensions. Mainstream exposure of multi-touch was achieved when Apple were granted a patent for a method which determined touch-commands by applying heuristics in 2009 (Apple Inc, 2009).

At present, Multi-touch gestural recognition is incorporated into the vast-majority of modern touchscreen devices such as smartphones and tablets (Yap, 2010). This increased popularity has made the 'pinch to zoom', 'rotate to pan' and 'swipe to turn page' a common feature of most peoples gestural vocabulary. The technology is now so ubiquitous that it could be argued modern-day users have a tacit and instinctive understanding of touch-screen interaction.

1.1.2 - Three-Dimensional (Freeform)

One of the first appearances of three-dimensional (or *freeform*) gestural control was the invention of the Theremin in 1928. The Theremin is a device that enables a user to control the pitch and volume of a synthesizer with the movement of their hands in relation to two antennas (Hammond, 2000).

The gaming industry has proved to be an area that has proliferated gestural control. Devices such as the X-box kinect and Nintendo Wii have proved very popular in consumer markets (BBC, 2011). They employ a combination of accelerometers and infrared (IR) sensors to detect the movement and relative position of a user, with the hope of giving them a more immersive game-playing experience. *Freeform* gestures have received criticism in the past for the lack of haptic feedback or tactile controls making the interface feel unnatural, but advances in technology are starting to make it more practical in areas other than gaming (Elgan, 2014). In particular, the Leap Motion, a compact infrared gesture recognition system designed for use with a laptop, has been used in medical applications. Primarily because its touch-less operation inhibits the spreading of germs, but its accuracy and intuitiveness have also been hailed as a contributing factor (Gupta, 2014).

1.2 - Motivation for the Research Area

It seems that gestural control has been incorporated sparingly in pro-audio systems. This is in contrast to positive trends in consumer markets (Matthews, 2013). Which is particularly surprising when audio mixing and processing has moved further into the digital domain, thus simplifying the development of novel control methods. One of the hurdles has been the tendency for engineers to prefer familiar, tactile interfaces. This is made clear by the popularity of digital control surfaces in an age where the majority of processing is available “in the box” (Korff, 2014). Engineers prefer to have a ‘hands on approach’ to mixing as it can contribute towards a more efficient workflow (Sound on Sound, 2012). However, most control surfaces are confined to using pots and faders. This limitation is paralleled by DAW (Digital Audio Workstation) GUIs, where skeuomorphic (based on real-world objects) designs are often chosen to offer familiarity over enhanced usability. The ‘banks of faders’ paradigm is one of the most prevalent examples of skeuomorphic design within DAWs. It is hypothesised that gestural control could allow a mix engineer to feel more ‘connected’ to the audio and provide a more immersive mixing environment. Additionally, it is believed that gestures could help to rationalise and simplify engineer workflow, where current methods could be described being inefficient and heavily regimented (Diamante, 2007).

One of the inherent benefits of a gestural interface is the ability of users to learn the associated movements through muscle memory, much like playing a musical instrument (Bates, 1994). Will engineers experience any benefits from learning a gesture set in order to remove the ‘visual barrier’? This research project aims to answer this question while evaluating the suitability and design of a gestural interface for the control of audio processing.

In order to focus the scope of the project, testing was limited to three of the most common audio processes: equalisation, compression and gating. Reverb was considered an equally popular effect, which has been omitted due to time constraints. Furthermore, Madden et al (2011) have previously conducted a comprehensive study on the gestural control of reverb. Following audio interface design guidelines laid out by Dewey and Wakefield (2014), the intended user and purpose of the interface were clearly defined at an early stage in the project so that relevant existing products and research could be evaluated accordingly. It was concluded that the interface should be suitable for *any* potential DAW user, from beginners to frequent experts. The project was also limited to single channel processing, with navigational elements such as track changes and transport-operations being excluded from testing.

It was decided through the analysis of existing products and previous studies that two-dimensional gestures would be used. This decision was also based on assessments of the purpose and context of a DAW interface. Firstly, the ideal interface should offer a high degree of accuracy and precision. It was clear from the literature study that this was less attainable on current three dimensional gesture recognition technologies (Lech and Kostek, 2013)(Selfridge and Reiss, 2011). Secondly, touch screens are a more familiar platform to the target-user, which helps to reduce interface-familiarisation times during tests. This report continues to discuss three-dimensional systems where they have been used for relevant audio processing tasks. The author concedes that audio is intrinsically three-dimensional. In other words it can be logically represented in three domains: frequency, time and amplitude. However, this project aims to optimise the interface through *simplification* of controls, thus the mapping of parameters to two-dimensional movements is considered more suitable.

1.3 - Project Methodology

Following initial planning, the first stage of the project was a literature review. This looked extensively at the use of gestural technology in existing products and how academic studies have explored its possible implementation in audio applications. Some investigation into general HCI concepts and interface design was required to provide a knowledge base for the project, but most literature covered was audio-specific.

Preliminary testing was carried out to understand the way in which engineers interact with DAW software using traditional interfaces. By observing the workflow of engineers using a traditional WIMP interface, the more cumbersome control-processes were identified. This helped contextualise the steps taken in a typical mix scenario, which could consequently be rationalised for the proposed gestural interface.

After observing engineer workflow, the elicitation of a user-defined gesture set was achieved through surveys and testing with audio engineers. At this stage, no constraints were placed on participants, therefore encouraging their creativity and avoiding the generation of an 'over-determined' taxonomy of gestures (Hydencorn et al, 2010). The priority was to obtain a wide range of gesture-parameter associations and provide an average model for the first prototypes. The gesture elicitation tests adhered to the 'conscious, top-down' methodology (Nielson et al, 2003), which will be discussed further in Chapter 6.

Following the elicitation of user-defined gestures, the first prototype could be developed. A significant amount of objective-C programming research was required to implement the technology included in the prototype. This has been omitted from the report.

Prototype Testing was split into two sections: equalisation (EQ) and dynamics processing. This was in order to break-up test times so that participants did not become fatigued, as well as helping to distinguish between the two mixing concepts.

Initial prototype testing prompted the exploration of appropriate GUIs as a method of improving the interface. A novel visualisation method was produced, the testing of which evaluates whether a more representative GUI can improve the effectiveness of a gestural control system.

Further improvements to the interface are suggested through the combination of parameters into a higher-level rationalised control. This is referred to as a ‘gestural shortcut’ or ‘continuous preset’ that was created through the analysis of research into semantic audio descriptors. Testing was carried out to investigate whether a higher-level semantically motivated control could offer suitable accuracy to mix engineers.

Adjustments to the prototype were made between testing stages, but not between test subjects. For example, if it was discovered during EQ testing that the interface was too sensitive, informed changes would be made to the compression and gating controllers. Any significant alterations to the prototype have been detailed and justified in this report.

Statistical analysis of test results was performed using the IBM software package SPSS to assess whether results were statistically significant. Data from test results are presented graphically so that some conclusions can be drawn through inspection. Immediate observations of the test results are explored after their presentation. However, any factors that influence the project as a whole are reserved for the *discussions and conclusions* section of the report.

The discussions section will evaluate the significance of the test results. The aim of this section is to propose an ideal model for a gestural audio-processing interface. The final sections of the report will consider the possible implications on the audio industry, any ongoing work and where further work could be directed.

1.4 - Structure of Thesis

Chapter 1 introduces the project.

Chapters 2 to 4 make up the literature review.

Chapters 5 and 6 discuss the methodology and present results from the preliminary tests. These are presented earlier on in the report in order to identify the influence of their results on the design of the prototype and subsequent tests.

Chapter 7 describes the first Gestural Interface Prototype.

Chapter 8 discusses the test methodologies.

Chapter 9 presents the test results.

Chapters 10 and 11 discuss the conclusions of the project and any future work.

2 - Traditional Audio Processor Interfaces

This chapter will examine a range of processor interfaces and aims to evaluate their usability. Close attention will be paid to the mapping of control parameters and their orientation. For example, why should the clockwise rotation of a virtual potentiometer increase volume? If the control were labeled "attenuation" it would effectively serve the same purpose, with reversed operation. An engineers association with orientation and the directionality of controls has particular relevance to the mapping of gestures. It determines whether the most intuitive direction of motion (advancing or retreating) is equivalent to the corresponding parameter change (increasing or decreasing).

2.1 - Skeuomorphic Design & The Legacy of Hardware

Skeuomorphism is the act of basing a software GUI design on an existing real-world product (Judah, 2013). Its use is widespread throughout most operating systems as a design technique that helps the user feel more comfortable and familiar in a digital environment. For example, the 'Outlook' e-mail logo in Windows operating systems mimics a real-world enveloped letter and OSX still uses a 'trash can' logo for the area in which items are deleted from the hard-drive. Many designers are trying to move away from skeuomorphism; Jonathan Ive (of Apple Inc) argues that customers no longer require it for a sense of familiarity because modern users are so comfortable interfacing with digital devices (Worstall, 2013). Figure 2-1 displays a typical set of skeuomorphic designs from the Apple OSX dashboard.

Figure 2-1 Skeuomorphism in OSX

The same could be said from an audio-engineering perspective, the next generation of engineers will be more experienced mixing "in the box" than on the original hardware. Some of these engineers may never have seen the hardware that plug-in packages so famously emulate.

In digital audio interfaces, skeuomorphism relates specifically to the emulation of hardware and the mapping of audio parameters to familiar controls in the GUI. This is often for aesthetic reasons, particularly when a plug-in has been developed with an algorithm that sonically replicates a piece of hardware, it would seem logical to base the GUI on the original unit. Figure 2-2 displays one such plug-in from Waves with a design that emulates an API compressor, both sonically and graphically (Waves, 2014).

Figure 2-2 Waves API plug-in

The influence of the GUI on the outcome of a mix is a widely debated topic. Some argue that increased reliance on visual cues has the potential to distract an engineer from the sonic output of a mix, resulting in a popular mix-technique whereby an engineer turns off their screens to help improve the accuracy of listening sessions (Porter, 2011). An investigation by Mycroft et al (2013) appeared to show that the GUI had no statistically significant influence on the listening skills of the engineer. However, this study did not take into consideration the control methods, only the complexity of the visualisation. It could be the case that retrofitting analogue controls into digital systems is a decision that compromises the performance of the interface in favour of familiarity. Some designers see gestures as a fast and effective control technique that can be impeded by the skeuomorphic GUI paradigm (Pratas, 2013). It is concluded through investigation by Bragdon et al (2011) that gestural control offers numerous benefits over 'soft-button' navigation. Including, but not limited to: simplifying the GUI, reducing 'attentional load' and increasing interaction performance when multi-tasking.

2.2 - Plug-in GUI Controls

As discussed in the previous section, the GUIs of some plug-ins are deliberately modelled on existing hardware. However, this design is often adopted by other systems in order to maintain familiarity and consistency across the software package. It could be argued that hardware emulation is not the most appropriate representation for the interface. In the case of DAWs, one of the most common plug-in interfacing techniques is WIMP (Windows Icons Menus and Pointer), mouse and keyboard, operation. One downside to this environment is the inability to easily adjust controls simultaneously (hence *pointer* and not *pointers)*. For example, an engineer using a rack-mounted hardware compressor has the advantage of being able to use both hands and change the ratio and threshold settings at the same time. This is not immediately possible with WIMP interfacing methods. Ultimately this has led to software plug-ins that emulate hardware as far as aesthetics, where it is impossible to offer the same amount of control as the physical device. One solution to this problem is through the use of hardware control surfaces, where tactile controls such as knobs and faders are mapped to the plug-in parameters and controlled via MIDI. However, the varying idiosyncrasies and vast range of plug-in GUI styles can make this an impractical method of control. Therefore, some existing plug-ins address the problem by making their GUIs more suitable for 'in-the-box' operation.

Most engineers will be familiar with the X-Y orientated equalization interface, where parameters are mapped as Frequency (x) against Gain (y). Figures 2-3 and 2-4 display the EQ plug-ins included in Pro-Tools 10 and Logic X.

Figure 2-3 Logic X EQ plug-in

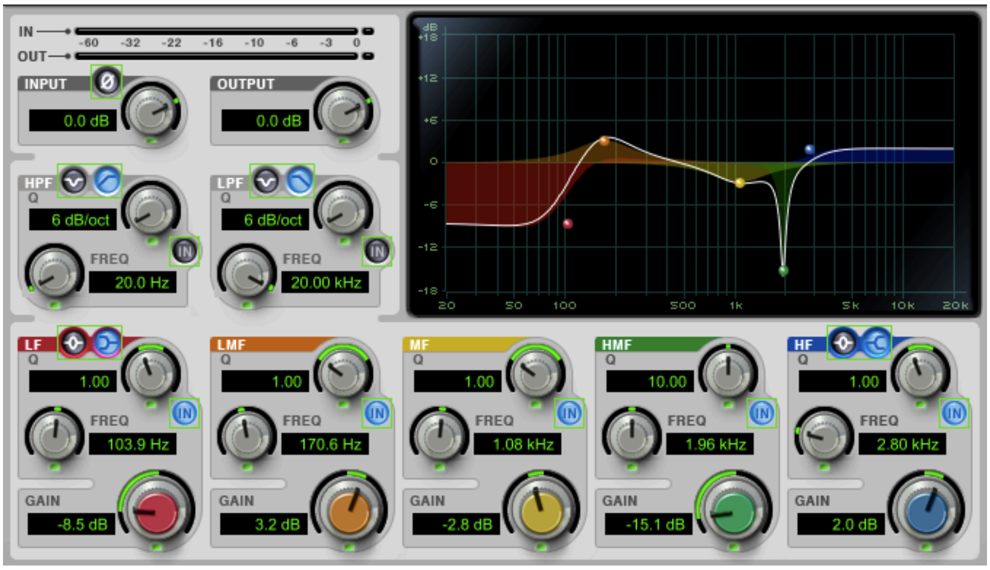


Figure 2-4 Pro Tools 10 EQ plug-in

Both of the above GUIs employ an x-y interface where the EQ response can be changed through the manipulation of ‘nodes’. Both plug-ins allow the user to adjust the gain, center frequency and Q-factor by clicking and dragging the nodes in some way. Notably, the latest iteration of Logic’s EQ has been criticised for altering the way that Q-factor is changed (Michael, 2014). To change Q-factor, the user now has to release the node and select a separate ‘vertical boundary’ rather than holding a keyboard shortcut and using the same node, as shown in figure 2-3. This has been described as inefficient because it adds another step to the workflow. Users seem to prefer increased controller complexity, or the addition of mode-indicators (such as keyboard shortcuts), in place of moving between locations on a GUI. A gestural system would require minimal changes in location or ‘active areas’. It is clear that the logic X plug-in has made more of a defined movement away from traditional controls. Whereas Pro-tools designers have chosen to include both an ‘x-y’ and a conventional ‘rotary potentiometer’ based control system. Having two sets of controls for the same parameter doesn’t seem like the best solution to optimise space on a GUI. There is a compromise that results in the inclusion of more than one control for a single parameter. This can be troublesome in a GUI, as they have limited on-screen space and can become too ‘busy’ in their design (Mycroft et al, 2013). A range of gestural controls could help to free-up space in the GUI (Bragdon et al, 2011). More detail on the preference of control types is presented during the Workflow Observations in Chapter 5.

2.3 - Plug-in GUI Visualisation

Although this project focuses on the control of plug-ins, a number of solutions offer direct interaction with visualisations as a method of manipulating the audio (as with the manipulation of the EQ curve in the Logic X plug-in). Visualisations are commonplace in plug-in GUIs as a way of assisting engineers with the mix process. For example, many modern EQ plug-ins include a spectrogram function, where a graphic representation of frequency domain information is presented to help engineers identify the frequency content of the audio. The Melda suite of EQ plug-ins advances this idea by including time-domain information in the form of a scrolling sonogram, as shown in figure 2-5 (Melda Production, 2014).

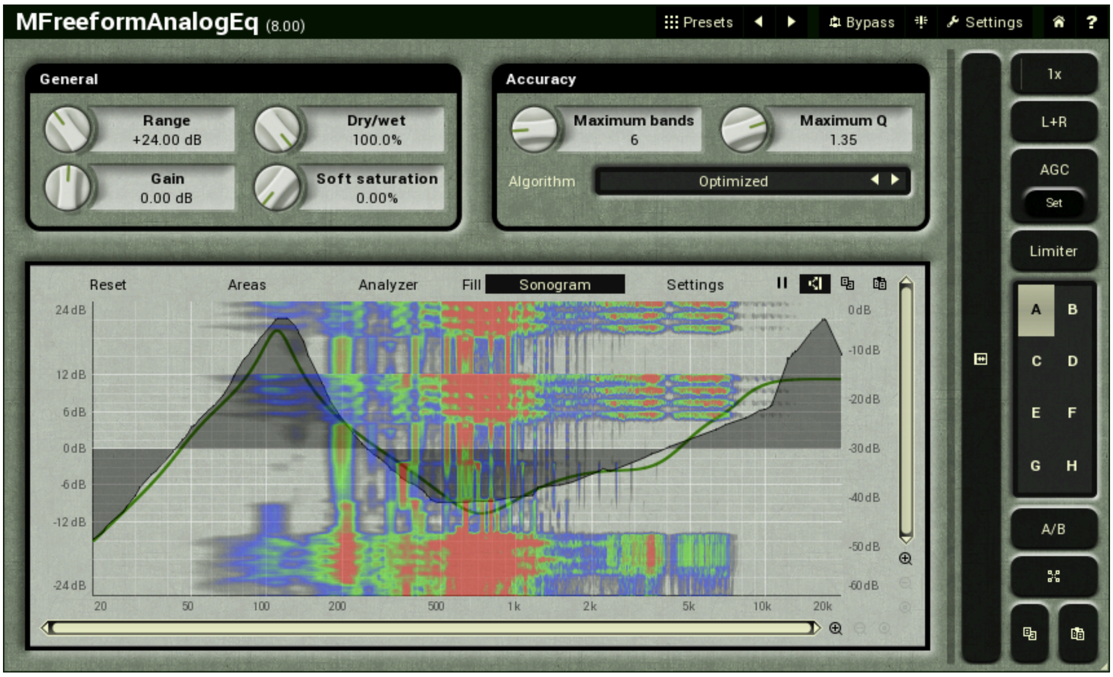


Figure 2-5 Melda EQ plug-in with Sonogram (Melda Production, 2014)

This is an example of visualisation that exceeds the skills of an engineer, and assists them accordingly. An engineer might find it easy to identify frequency content of a mix, but to remember this information at snapshots in time as a track changes is a lot more challenging (certainly this could be considered impossible at the resolution that a sonogram offers). In this case the visualisation directly *influences* the control method.

A number of pitfalls can arise when engineers begin to rely on visualisation over (or in conjunction with) auditory information. The *ventriloquism effect* (or image proximity effect) describes a perceptual phenomenon whereby a listener's localization of a sound source can be distorted by visual stimulus (Lech and Kostek, 2013). This is famously observed in the illusion of ventriloquist performances, where the audience is misdirected into localising the audio source as the ventriloquist's doll. Schutz and Lipscomb (2007) conducted a more musical investigation into this phenomenon, where test participants were asked to rate the length of a number of notes played on a marimba with and without viewing different performances of the note. They found that showing performances of sustained notes with the audio from short notes made the perceived note length longer and vice-versa. An example of the implications of this effect in the context of a Plug-in visualiser could be the graphical latency of a Loudness meter (or VU meter) making a transient audio source seem to have a longer sustain.

A further, potentially misleading, area of the GUI is the presence of parameter values. Cochrane (2013) argues that parameter values are a good guideline to the starting point of a mix, but pre-conceptions about 'suitable settings' (or any visual aspect of a DAW) shouldn't influence mix decisions. A gestural interface has the potential to bypass these problems by enabling the imageless control of audio processers through memorised gestures. Allowing mixing to continue once the screen has been turned off.

2.4 - Semantically Motivated Processor Control

The Semantic Audio Feature Extraction (SAFE) project aims to understand the linguistic associations with parameter settings (Stables et al, 2014). For example, if an engineer describes a compression setting as "Punchy", how would they map to the controls? The project works by offering free downloads of the plug-in suite so that engineers can contribute their settings for each semantic descriptor, an average of these contributions is then taken and offered as a 'ideal setting' for the corresponding descriptor. Figure 2-6 illustrates the suggested EQ response for a 'warm' sound.

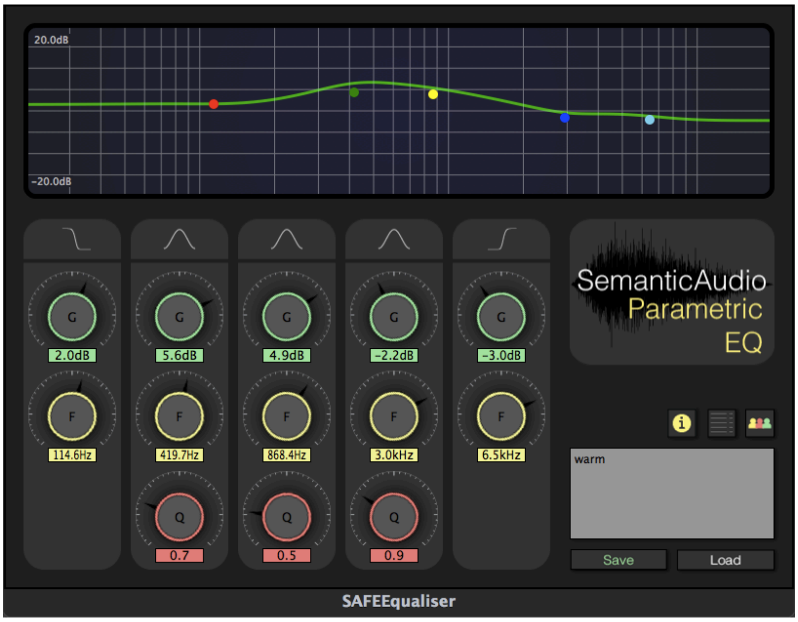


Figure 2-6 SAFE EQ plug-in

An issue to consider with this elicitation process is that the settings are source-dependent and engineers will be using a range of varying sources. However, a large enough average would represent an 'ideal' setting, furthermore the study includes more specific presets such as 'warm vocal compression' and 'rock kick drum compression' that would help make the parameters more contextually accurate. By offering 'semantically motivated presets' the SAFE project aims to improve workflow for less experienced engineers. These presets can represent a good 'starting point' for plug-in parameters and offer a semantic shortcut that has the potential to speed up the mixing process for both novices and experts.

3 - The use of 2D Gestures in Human-Computer Interaction

This chapter will explore the general application of gestures as a way of interfacing with computers. It will aim to evaluate the effectiveness of gestures in other applications so that design decisions can be made on the control of audio processes.

3.1 - 2D Gesture Principals

Gestural controllers have been successfully implemented in a wide range of applications and are subsequently commercially available in multiple incarnations. Types of gesture systems range from simple, single touch screens (Amazon, 2014) to sophisticated multi-point, three-dimensional interfaces (Leap Motion Inc, 2014). This broad availability can be attributed to the numerous contexts where gestural control has been deemed as an advantageous form of HCI. The initial design of a gesture system should evaluate the appropriate *User Input* for the *Purpose of Communication* to determine the required level of gestural complexity. Generally the purpose will either be communicative or manipulative (Westerman and Elias, 2001). Take, for example, the skeuomorphic 'virtual book' application demonstrated in figure 3-1, where the user can use the touch-sensitive screen to turn a page with a gesture.

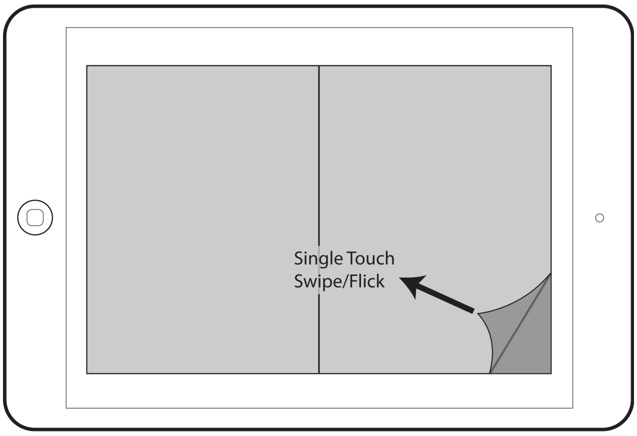


Figure 3-1 Virtual Book Application Example

An appropriate and sufficient level of input would be a single touch, where the purpose is to simulate the turning (manipulation) of a page. Further analysis can be done to identify the *type* of gesture that is being used. Gestures can be fundamentally defined by three classifications (Nielson et al, 2008):

- Static - motionless gestures such as taps. Note that these taps could be multi-point and reiterative.
- Dynamic - Moving gestures.
- Spatiotemporal - A type of dynamic gesture that requires analysis over time. The most common implementation of this is the drawing of shapes or letters.

Additionally, most gestures can be classified by whether they represent real-world actions or just arbitrary control allocations (Westerman and Elias, 2001):

- Mimetic - movements that imitate an action. These are most commonly continuous.
- Semaphoric - "Gestures from a dictionary of abstract symbols" (Balin and Loviscach, 2011). Derived from semaphores (signaling with flags).
- Deictic - acts of pointing. These are most commonly static.

The *virtual book* example is a single-touch, manipulative dynamic-gesture, which is also mimetic (imitating the real-world interaction with a book). If the application interface required a user to tap on the page to navigate through a book, this would be a single-touch, manipulative static gesture, which does not emulate a real world action and is therefore deictic.

3.2 - Pitfalls of Gestural Interfaces

One of the most widely discussed challenges when designing a gesture-system is in overcoming the 'Midas Touch' problem (Nielson et al, 2008). 'Midas' refers to 'King Midas', a character from Greek mythology whose touch turned objects into gold. Initially a blessing, this power transpired to be troublesome, turning everything he touched, including his daughter, into gold. The metaphor relates to the consideration of *whether a gesture is being performed* in a gestural recognition system. A system that was too sensitive might unintentionally interpret a user's movements as a gesture, or "turn every movement into a gesture". This can make the interface unstable and unnatural by restricting the user's 'idle movements'.

An issue with touch screens that has been highlighted by numerous sources is their prolonged use becoming uncomfortable to the user. Apple famously decided not to follow suit with other manufacturers in adding touch screen technology to their laptops. Apple designers claimed that product testing found that it was "awkward and uncomfortable" to repeatedly reach forward and touch the laptop screen (Tibken, 2014). Balin and Loviscasch (2011) experienced issues with touch screen orientation, they discovered that users find it easier to read from a screen that is mounted vertically, but it was more uncomfortable to use than a horizontal surface. This orientation compromise has been illustrated in figure 3-2.

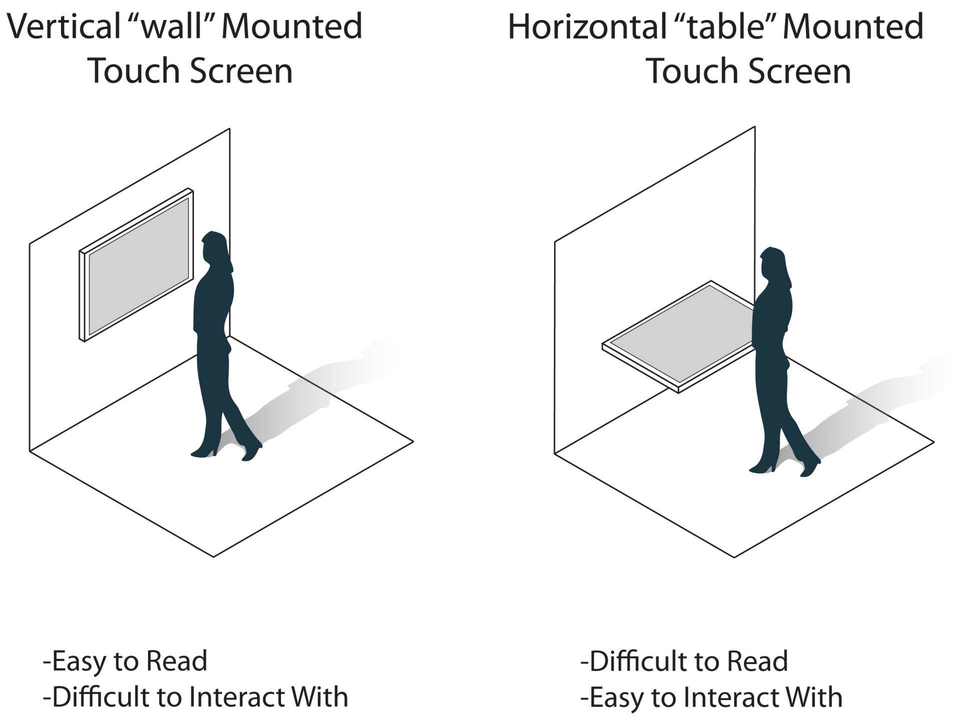


Figure 3-2 Vertical vs. Horizontal Touch Screen Mounting

Another, more fundamental, issue with touch screens is that repeated use can result in a build up of dirt and grease on the screen, which can impair smooth interaction with the device (Sullivan, 2012).

3.3 - Touch Screen Operation

A detailed explanation of touch screen technology is presented in Appendix A 'A Technological Overview of Touch Screens'. An understanding of touch screen operation is not essential for the comprehension of this report, although it is important to take note of the high *touch-resolution* of modern capacitive touch screens and the resulting gestural-recognition accuracy that this enables.

3.4 - Examples of Two-Dimensional Gestural Control in HCI

Two Dimensional Gestural Control has become ubiquitous in a wide variety of applications. The popularity of multi-touch enabled surfaces has allowed software engineers to incorporate gestures as controllers and shortcuts in their programs. One of the first incarnations of this open-platform development relationship was the addition of multi-touch 'Trackpads' to Apple Macbooks. Large software companies such as Adobe offer gestural integration with their applications. For example, while using Adobe Photoshop, the rotate gesture will rotate the canvas, while the pinch gesture can be used to zoom in and out (Osbourne, 2011).

A number of third party applications allow users to change the default gesture-set provided within the Apple operating system. Programs such as 'BetterTouchTool', 'Magic Prefs' and 'JiTouch' facilitate the mapping of customised gestures to a range of functions, keyboard shortcuts and macros (Appleyard, 2010). The popularity of these apps signifies the idiosyncratic nature of gestural control, with a high population of users disagreeing with the default gesture settings and opting instead for a unique, tailored gesture set. These tools are commonly used as a method of speeding up workflow with gestural shortcuts (Guinness, 2014).

Gestural control has begun to be implemented successfully in industries outside of audio. The automotive industry is exploring the use of gestures to control in-car-entertainment as it allows drivers to keep their eyes on the road. The car manufacturer Ford recently patented a number of 3D gestural controls that are used to adjust in-car entertainment and comfort settings, such as temperature and airflow (Ford Global Technologies, 2013). The concern with a design like this is the prevalence of Midas Touch errors. If a driver is aware that the gesture system is constantly monitoring their movements, they might begin to feel constrained. Other companies offer touch screens as a viable platform for gestural control (Krenn, 2014). A review of one such system praises the ability to control in-car features while keeping eyes on the road, but criticises the need to learn a large number of gestures (Lunderschmidt, 2014). However, other apps have proven the desirability and usability of gesture controlled app interfaces, such as 'carTunes' an app that incorporates gestures so that users can control their music library without looking at their iOS device (Virgil, 2012).

The gaming industry has been instrumental in the popularisation of gestural control. One of the earliest examples in mainstream gaming, as identified by Payne et al (2006), is in the 2001 PC game 'Black and White', by Lionhead Studios. To simplify the interface of this complex game, single-touch spatiotemporal gestures were used to replace menus and icons, thus 'increasing the user's immersion' in the gameplay. The same space-saving techniques are routinely incorporated into applications on the smartphone gaming platform. This can be particularly beneficial in large-scale games where multi-touch recognition allows navigation buttons to be replaced by gestures. For example, the smartphone game 'SimCity - BuildIt' supports pinch and rotate gestures that allow users to seamlessly "explore [their] 3D city with 360-degree controls" (Lilly, 2014). Previous single-touch incarnations of the game required a cumbersome navigation panel.

3.5 - Two-Dimensional Gestural Interface Design

A number of guidelines exist that detail the considerations that should be made when designing a gestural interface.

Hinckley (2011) highlights the importance of understanding the *input* technologies involved in an interface. He identifies four main considerations for initial interface design.

1. The physical sensor.
2. The Feedback Presented to the User.
3. The Ergonomic and Industrial Design of the Input Device.
4. The interplay between all of the interaction techniques supported by the system. (Including the interaction between multiple devices)

Hinckley suggests that evaluating the system 'from a distance' ensures that the interface is designed without too much focus on one task that might ignore the context of the application. For example, if this research project was concerned with overall control of a DAW, it could be argued that by optimising the interface for the control of processors, overall system usability is compromised. Point 4 is particularly important in a gesture system due to the limited number of unique gestures.

Heydekorn et al (2010) describe a difficulty that arises when eliciting *user-defined gestures.* If test participants are asked to pick from a pre-defined set of gestures, the results can be too 'over-determined'. They should, therefore, be given the creative freedom to choose *any* gesture for the control of a suggested parameter. However, the participants are often unaware of the importance of point 4, so any elicited gestures from a survey should be reviewed by an 'expert'. It is concluded that an ideal gesture-elicitation exercise should combine the 'creative freedom' of the average user, with contextual system awareness of an expert. Hinckley (2011) further criticises 'creative freedom' within gesture elicitation with relation to actions that are non-mimetic creating too much variation within a gesture elicitation survey. Hickley (2011) demonstrates this with the following scenario: Figure 3-3 gives three examples of gesture tasks presented to a test participant.

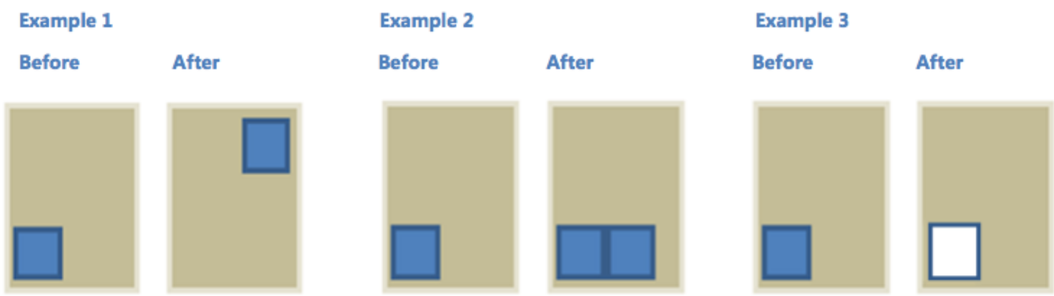


Figure 3-3 Examples of Gesture Elicitation Tasks (Hinckley, 2011)

In each example the participant is asked to perform a gesture on a touch screen that would produce the 'after' object from the 'before' object. In example 1, the consensus would most likely be a movement from bottom left to top right of the screen. In example 2, a range of swipes might be suggested. Example 3, however, is more abstract and likely to return a wide range of arbitrary, unrepresentative gestures. It is important to identify the ambiguity that can arise in the user-defined elicitation of non-mimetic gestures.

The gestural interface design guidelines proposed by Nielson et al (2003) further emphasize the desirability of gestures that are representative of their corresponding function or control. They summarize the ideal features of a gesture in an interface by specifying that they should be:

1. Easy to perform and remember.
2. Intuitive.
3. Metaphorically and iconically logical towards functionality.
4. Ergonomic (not physically stressing when used often).

Adhering to these design principals will ensure that *usability* is optimised. Nielson et al (2003) define the usability of a gestural interface with five main features.

1. Learnability - The time and effort required to become familiar with the interface.
2. Efficiency - The effective performance of the interface when used by an expert.
3. Memorability - The ease at which an intermittent user can return to using the interface.
4. Errors - The frequency of errors encountered during operation, including misinterpretation of gestures.
5. Coverage Rate - The number of successfully performed gestures to the total number of gestures (Bragdon et al, 2009).

A high level of usability will result in an interface that quickly and effectively carries out the intentions of the user. Subsequently, workflow will be streamlined.

4 - The Gestural Control of Audio Systems

A specification for a prototype gestural interface can be proposed through the evaluation of existing products and previous studies. Particular attention will be paid to the *mapping* of gestures to audio parameters and any attempts to streamline workflow.

4.1 - Touch-Screen Control

A number of commercial products already take advantage of gestures to control audio. There has been an increase in the number of mixing consoles that replace some of the traditional pots and faders with large touch screens (Avid Technology, 2014)(Calrec Audio, 2014)(Yamaha Corporation, 2014). Interestingly, the resulting touch-screen GUIs include digitally emulated pots and faders. This seems like a contradictory design decision, for the same reasons discussed in section 2.1 with regard to plug-in GUI designs. Some examples of gestural control can be found on mixing desks, such as the ability to adjust Q-factor with a pinch gesture (Avid Technology 2014)(Calrec Audio, 2014). A more comprehensive use of gestures can be found in the 'Slate Raven' MTX, as shown in figure 4-1. The 'Slate Raven' is a multi-touch enabled DAW interface. It's design claims to put the DAW "right at [the engineer's] fingertips" and likens operating plug-ins on it's high resolution 46" screen to "tweaking 19" rack gear" (Slate Media Technology, 2014).

Figure 4-1 The Slate Raven MTX Mixing Console (Slate Media Technology, 2014)

The slate raven begins to explore the use of gestures as navigational shortcuts (Robjohns, 2013). Scrub/Shuttle, waveform zoom, track zoom and track banking are all assigned to gestures, in an attempt to streamline user workflow. It could be suggested that a downside to this design is that it conforms to the traditional 'row of faders' format in a skeuomorphic fashion. However, this is a limitation within the DAW software itself and may have been a conscious decision so that experienced engineers did not have to learn a completely new interface in order to operate the desk.

One product that attempts to shift the 'bank of faders' paradigm is the Line-6 stage-scape. The stage-scape uses a touch screen GUI that emulates the layout of a typical live-performance stage. Figure 4-2 displays the hardware and an example of the on-screen controls.

Figure 4-2 Line 6 StageScape M20D (Line6, 2014)

The touchscreen control has been rationalised through the mapping of semantic audio descriptors to x-y controls. Figure 4-3 shows the 'quick tweak' screen for Bass Guitar 'Tone' settings.

Figure 4-3 StageScape 'Quick Tweak' Bass EQ Control (Line6, 2014)

The terms 'Boom', 'Snap', 'Scoop' and 'Smack' are offered as 'tone' presets for a bass guitar. The presets are mapped to each corner of an x-y pad, allowing for interpolated settings between each of the descriptors. The stagescape has sought to make the mixing GUI more suitable for touchscreen control, rather than adhering to skeuomorphism. The system has been praised for its novel interface, with numerous reports of its suitability for inexperienced engineers who are less concerned with parameter values. For example, Sound on Sound magazine described the device as "a fully featured digitally mixer… that musicians with less mixing experience than a dedicated FOH engineer could still [use to] achieve good results" (White, 2012). As such, the Stagescape has proved to be popular amongst smaller bands that are required to provide their own PA system. The addition of gesture controls and shortcuts to a system such as this could greatly improve intuitiveness and resulting workflow for the user.

A large number of tablet computer apps that operate as control surfaces for DAWs are commercially available. V-control, Mackie Control, DAW remote and DAW control are proving to be a popular alternative to small control surfaces (Sasso, 2014). As with the other products described in this section, they regularly emulate rotary controls and rarely implement gestures.

4.2 - Case Studies: Gestural Control of Processing and Mixing

The systems described previously in this chapter implemented solutions to mixing on a touch screen with the aim of improving usability and streamlining engineer workflow. However, they were often confined to using traditional, skeuomorphic GUIs. Gestural controls or shortcuts remained relatively unexplored, often leaving the interface mimicking the original that was intended for WIMP interaction. A number of research projects have sought to evaluate the suitability of gestures in a mixing environment. The majority of these projects map controllers to spatial characteristics of audio sources.

4.2.1 - Navigational and Transport Control

Balin and Loviscach (2011) propose a system that allows engineers to navigate around a DAW using gestures. As such, there is little in the way of associations to audio processing parameters. They conducted a web-based survey asking participants to choose from 30 *predefined* gestures and match them with 22 DAW commands. Their investigation only provided two continuous controls: increase value and decrease value. Predictably, a single touch, upward swipe was chosen to increase value and a single touch downward swipe to decrease value. Figure 4-4 displays the 30 gestures offered as part of the survey. Touch locations are presented as red-dots, where gesture 1 is a single tap and gesture 2 is a double-tap, and directional movements are presented as black arrows.

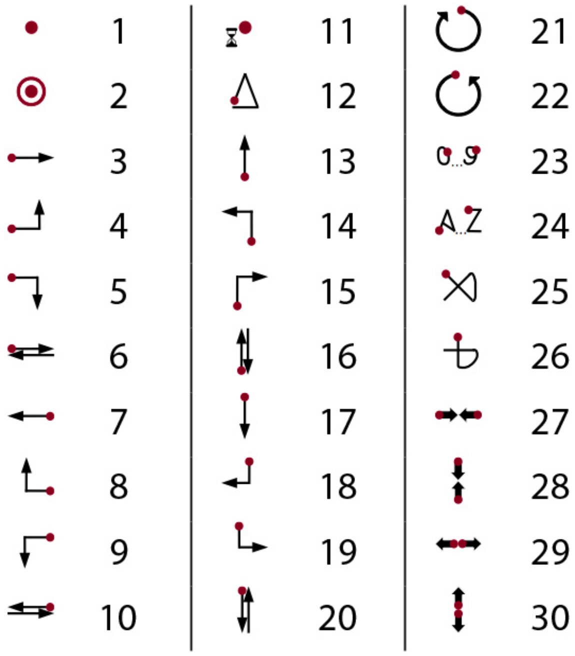


Figure 4-4 DAW Control Gesture Set (Balin and Loviscach, 2011)

A large number of the gestures are spatiotemporal; they require analysis over time to determine the type of gesture. For example, gesture number 4 (as identified in figure 4-4) requires the 90-degree 'corner' to be recognised in order to distinguish it from gesture number 3 (a straight line). Implementing similar gestures in the same system can cause disruptions to user workflow. Firstly, recognition time is increased due to similarities between spatiotemporal gestures. Secondly, the chance of a gesture being misinterpreted is increased (as both a result of system and user error). An ideal gesture system should find a good balance between the uniqueness and simplicity of the gesture set. Thus avoiding 'clashes', or similarities, while remaining intuitive. Balin and Loviscach limited their study to one and two point gestures in order to reduce the number of possible combinations. However, it is the author's belief that additional touches could be integral to the streamlining of workflow, and continuous gestures should be chosen in favour of spatiotemporal gestures wherever possible to minimise gesture-recognition latency.

4.2.2 - Spatial Control (panorama and amplitude)

Gestural control is intrinsically well suited to placing sound sources in a stereo (or multi-channel) image (Selfridge and Reiss, 2011). The implementation of deictic (pointing) gestures can rationalise the process and prove to be more intuitive for both novice and expert users. As such, a number of novel interfaces have aimed to provide functionality where the engineer ‘points’ or ‘gestures towards’ the place in the stereo field where they want the sound source to be panned.

Selfridge and Reiss (2011) produced one such system that used a wii-mote as a gestural control device. The horizontal movement of the wii-mote was translated into a MIDI control value before being used to place a sound source in the stereo field. The relationship between engineer-position and sound source panorama could be fine-tuned so that the engineer felt that they were ‘pointing at the sound source’. It was concluded that, although other parameters experienced difficulties, users could demonstrate accurate control over pan. An important addition to the design was to enable the participating engineer to tell the system when to expect a gesture by pressing a button on the wii-mote. This helped to reduce the number of misinterpreted gestures and the prevalence of the Midas Touch effect.

The ‘Motion-Mix’ system by Ratcliffe (2014) advocates the ‘stage metaphor’ design as a viable alternative to spatial mixing. The proposed system uses two pieces of gesture recognition hardware to offer a larger number of possible control mappings. The primary controller is the ‘Leap Motion’. The additional auxiliary piece of hardware is a tablet running the software ‘touch OSC’. The majority of the gestures take place using the Leap Motion, whereas the touch-screen tablet is used for navigational purposes such as changing track. The stage metaphor was interpreted on the Leap Motion by mapping pan to the x-plane and volume to the z-plane, this setup is illustrated in figure 4-5.

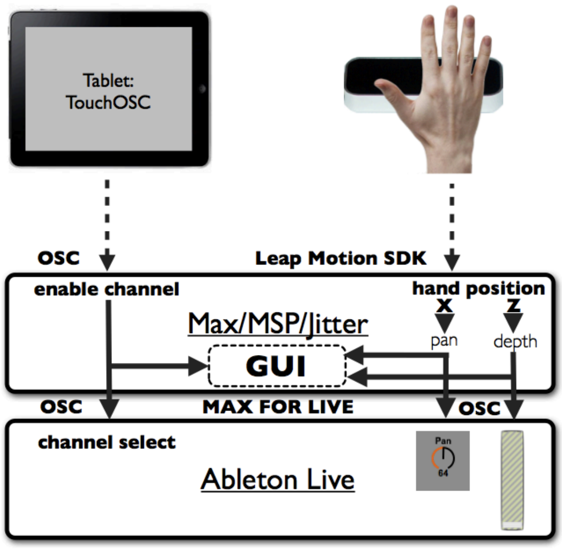


Figure 4-5 'Motion Mix' Gesture Recognition System (Ratcliffe, 2014)

Notably, a system that is capable of three-dimensional gesture recognition is only being used to measure two-dimensions of movement (x-plane and z-plane). The corresponding GUI represents each sound source with a coloured sphere. The sphere moves left and right on the 'stage' as it is panned and grows larger/smaller in diameter as it is increased/decreased in volume. The intent of the changing sizes is to make the quieter source-spheres seem as if they are further back on the 'stage' and vice-versa. Ratcliffe carried out a pilot-study on a population of 9 participants to establish the performance of the 'motion mix' in comparison to traditional interfaces. The participants were asked to mix the same sources in the same way using three different interfaces methods:

1. Motion Mix without 'stage' visualisation.
2. Motion Mix with 'stage' visualisation.
3. Ableton DAW, traditional interface.

The resulting times appeared to show that workflow and interaction were most efficient when using the MotionMix with stage visualisation. These results are presented in table 4-1.

Participant	DAW	Motion Without Visual	Motion with Visual
1	05:53	08:04	05:07
2	06:14	11:40	05:12
3	04:46	02:59	03:36
4	01:36	02:05	02:35
5	03:33	06:47	05:49
6	04:06	04:10	04:07
7	04:05	03:51	03:10
8	06:05	10:47	05:00
9	02:09	02:06	02:44
MEAN	**04:16**	**05:50**	**04:09**
SD	**01:40**	**03:40**	**01:11**

Table 4-1 Motion Mix Times (Ratcliffe, 2014)

Ratcliffe concludes that the 'Motion Mix', when used in conjunction with a stage-metaphor visualizer, does not slow down engineer workflow. However, a number of participants commented that the visualisation was prompting them to 'mix with their eyes' and 'distracted them from focusing on listening'. Testing of mix accuracy, either subjectively or objectively would be required to assess the influence of the GUI.

In similar work to Radcliffe, Drossos et all (2013) employ two dimensional gestural recognition in a three dimensional space. Figure 4-6 displays their interpretation of a 'stage metaphor' mix environment, where the left hand controls volume in the y-axis and the right controls panorama in the x-axis.

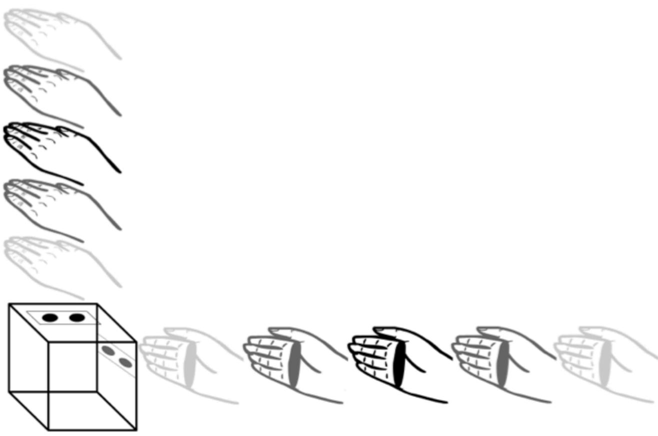


Figure 4-6 Gain vs. Panorama (Drossos et al, 2013)

Drossos et al's approach differs to Ratcliffe's in that Volume control has been mapped to the y-axis. This was a conscious decision to more closely replicate the actions of an orchestral conductor. A series of preference tests concluded that users preferred the 'artistic expression' of the interface and that it allowed them to move more freely and intuitively than a traditional interface.

4.2.3 - Mixing with the Stage Metaphor

A number of solutions that aim to make interfaces more suitable for gestural control have looked towards the 'Stage Metaphor' design, in a similar approach to the Line6 stage-scape. One such design was proposed and tested by Carrascal and Jordá (2011). They were trying to optimise the touch screen for mixing multiple sources spatially, while avoiding the 'channel strip' approach. The chosen platform was the ReacTable, a system that operates using tactile blocks called 'tangibles' as controllers (ReacTable Systems, 2014). Their design was based around the manipulation of circular 'nodes' that could be altered in size as well as their position on the 'stage'. Each node represented a channel of audio, as shown in figure 4-7.

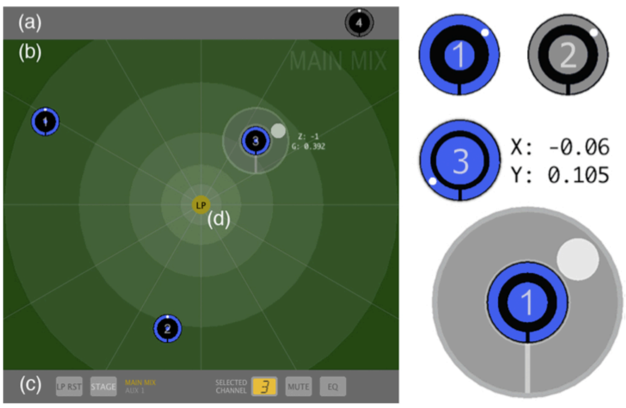


Figure 4-7 Nodal Manipulation on a Soundstage (Carrascal and Jordá, 2011)

A listening point has been included, signified by the yellow 'LP' dot, so that the interface can be used for surround-sound mixing. They concluded, through the testing of a small six-person sample, that achieving a satisfactory mix with their nodal GUI was faster than a traditional 'bank of faders' interface (Carrascal and Jorda, 2011). Perhaps a similar technique could have been included in the Line6 product to remove more of the sub-menus and auxiliary controls.

Gelineck et al (2013) produced a similar system to Carrascal and Jordá (2013). They emphasized the importance of 'passive haptic feedback' in the form of tangible objects. Their system incorporated a 'double tap' gesture for muting and soloing of audio sources.

4.2.4 - Processing and Effects Control

Madden et al (2011) explore the use of a multi-touch tablet as a controller for Reverb in a system they call 'Interactive Room Response' (IRR). Their GUI concept is based on placing a 'listener in a virtual room' to help represent the changes in reverb parameters. Two prototypes were developed, a 2-D GUI on an Apple iPad (Figure 4-8) and a simulated 3-D GUI on a Motorola Xoom (Figure 4-9).

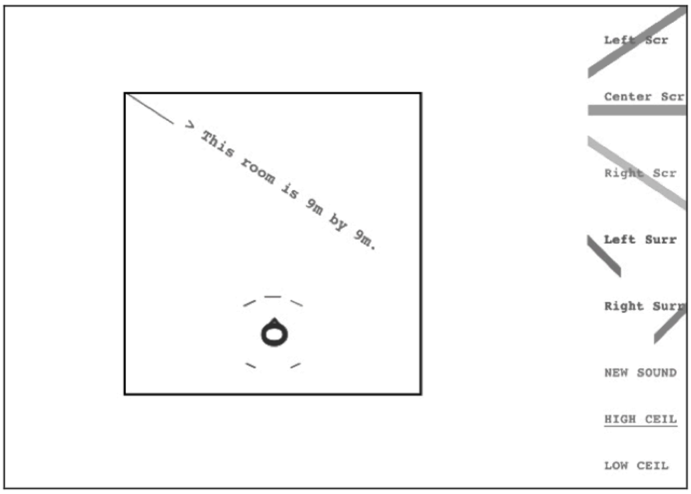


Figure 4-8 IIR iPad Visualisation (Madden et al, 2011)

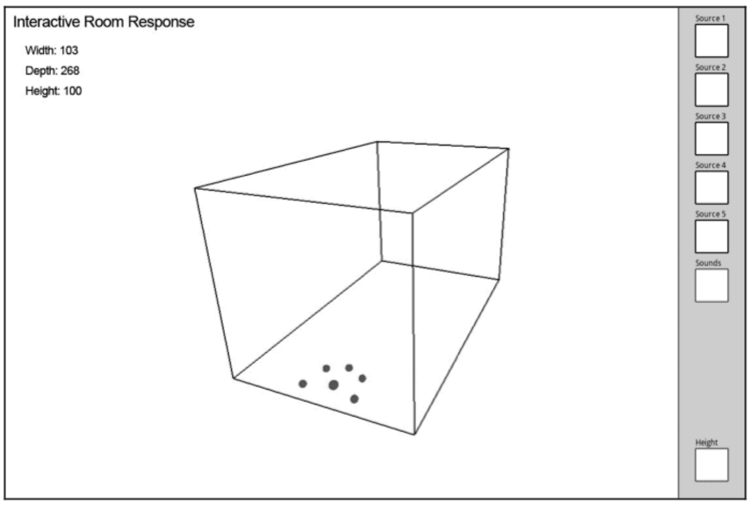


Figure 4-9 IIR Xoom Visualisation (Madden et al, 2011)

The Xoom interface was described as a more successful system because the gestures were directly mapped to the on-screen visualisation, which was more representative of the Reverb effect. This symbiotic relationship between controls and visualisation provides an intuitive interface, as mapping touches to visuals is a lot more familiar to users than mapping touches to changes in audio.

In 2013, a study by Lech and Kostec (2013) investigated the use of gestures to control a DAW (digital audio workstation). They identified that a larger number of engineers are working in small project studios where mixing desks are impractical and expensive. Therefore, they are often limited to using traditional WIMP interfaces. An inferior interface that impedes on ergonomic operation is said to be detrimental to the quality of the final mix. Lech and Kostek's proposed interface identified gestures in 3D space using a camera system. They predominantly focused on navigational elements of DAW interaction such as transport operations (play/stop/record etc) and track settings (solo/ mute), with the intention of helping engineers to rely less on visual indicators.

Lech and Kostec hypothesised that "Visualization of audio signal parameters adversely affects the aesthetic value of the mixes." and "Mixing by hand gestures leads to mixes of a higher aesthetic value than mixing with a mouse and keyboard." They were not as concerned with optimizing workflow, more so the intuitiveness and aesthetics of the interface. It could be argued that their system had a more convoluted workflow because of the number of 'parameter selection' gestures required.

In their camera based gesture system, compression requires a parameter selection layer, for example a 'T' is drawn to select threshold. 'T' is described as a "semantically associated gesture". However, it could be argued that this is an arbitrary symbol, derived from the name of the parameter rather than the audio process. In the author's opinion it should, therefore, not be considered a mimetic gesture, but rather a semaphoric gesture (A selection from dictionary of symbols). A truly mimetic gesture should be universally understandable, regardless of language. After threshold has been selected, a simple 'up to increase', 'down to decrease' control method is implemented. Their testing took 10 participants and asked them to mix the same audio sources with five varying types of interface:

1. Gestural Mixing without visualisation.
2. Gestural Mixing with visualisation.
3. Mouse & keyboard interface without visualisation.
4. Mouse & keyboard interface with visualisation.
5. Mouse, keyboard & MIDI controller interface in standard DAW environment (control)

By comparing the speed and subjective quality of the mixes, they concluded, "Better ergonomics do not necessarily mean better aesthetic results". Furthermore, the feedback from participants was that, overall, they found the interface 'inconvenient'. Lech and Kostec put this down to two main factors:

1. Engineer Fatigue - The inherent nature of the 'un-obstructed' gesture system means that engineers are unable to rest their arms. Most gestures involved a 'waving' or 'reaching' element.
2. Gesture Recognition Time - The average time it took the system to recognize a gesture was 1.5s, which most users found inconvenient. Certainly a delay of this amount would severely impede the workflow of an engineer.

The time-delay, along with prevalence of the 'midas touch', was also blamed for a reported lack of precision during parameter editing. In conclusion, they noted that dynamic (moving) gestures were better suited to their system than static gestures, with users reporting a much higher efficacy than static gestures.

Selfridge and Reiss (2011) experienced similar difficulties when experimenting with the 'Wii mote' to control EQ. Whereas gain and pan were successfully implemented, participants reported difficulty 'drawing' with a *freeform* controller. Their original prototype operated like a 31-band graphical EQ, where the user 'traced' the shape of their desired EQ response. X-axis movements were mapped to frequency, while y-axis movements adjusted the gain of the selected frequency. The problem that occurred within the system was down to time-delay, it took between 0.5 and 1 second(s) to recognize the shape and draw it's representation on the 31-band graphic. As with Lech and Kostek (2013), the time taken to recognize gestures and alter corresponding parameters has proven to be fundamentally detrimental to the success of the novel interface. Reiss and Selfridge improved their system by basing control on a parametric, rather than a graphic, EQ.

4.2.5 - Interfaces for Musical Expression

Though this project focuses on the control of processor parameters, it is still necessary to look at some of the gestural interfacing principals found in compositional systems. Whereas engineering tools (such as DAW controllers) often focus on functionality, interfaces for sonification endeavor to make the system immersive so that it feels more like a real-world instrument. In other words, the interface's priority is to alleviate the users awareness of low-level controls so that the conscious mind is free to work on higher-level goals, such as composition or mixing (Herman and Hunt, 2005). Thusly, some compositional interfaces have features that could be incorporated into a mixing interface to help the engineer 'feel closer to the audio'. Couturier (2005) emphasizes the importance of identifying features of the action-perception loop when designing an interface for musical expression, as the user must receive sufficient feedback for it to feel like 'playing an instrument'. With acoustic instruments this feedback is in the form of Auditory, Tactile and Visual stimulus as a result of playing. An equivalent level of feedback can be difficult to achieve with digital instruments. In particular, the emulation of haptic feedback is often limited by technology.

An example of a gestural physical modeling music synthesis interface that offers haptic feedback is the 'Cymatic', created by Howard and Murphy (2007). They identify that even the smallest gestural movements can influence to sonic output of an acoustic instrument, whereas this is not the case for classic synthesizers. The Cymatic system used a force-feedback mouse and joystick, intended for gaming, to provide haptic feedback to the user while composing. The resulting playing experience is described as both 'immersive and tactile'. The addition of similar feedback techniques in a DAW interface could produce a more 'immersive mixing environment'.

5 - Preliminary Test 1 - Engineer Workflow Observations

Before starting to consider gesture mappings, testing was done to determine the workflow that exists with traditional WIMP interfaces. Testing began with observations and analysis of current engineer workflow patterns when operating a

DAW. The evaluation of these observations will help to rationalise the workflow for the first gestural mixing prototype.

5.1 - Test Methodology

A test was devised to analyse workflow patterns associated with audio processing tasks within a DAW mixing environment. Participants were asked to perform common processing tasks as their actions were recorded using video screen capture software and automation recording. Twenty-two engineers took part in the testing, all with varying mixing experience raging from 1 to 20 years. All participants either past or present music technology undergraduate students and were therefore deemed to have a sufficient understanding of the three processes involved in the test, which were equalisation, compression and gating. Each participant's experience level and familiarity with the software was determined as part of an informal interview at the start of the test (for more details See Appendix C: Engineer Workflow Observation Handout). Default ProTools version 10 plug-ins were used throughout the test process.

The test comprised of four separate tasks:

- Corrective EQ - the participant is asked to match a reference snare sample by removing resonant frequencies from the source sample.
- Creative EQ - the participant is asked to match a reference guitar sample by adjust source sample tonality through the use of EQ.
- Compression - the participant is asked to reduce the dynamic range of a repeated kick sample in reference to a premixed version of the same sample..
- Gating - the participant is asked to remove guitar spill from a recording of a kick drum in a live performance context. Again, a reference sample is provided.

Participants were not given a time limit. This was to ensure that they interacted with the plug-in in the most natural way possible and reveal any idiosyncrasies in their mixing style. The reference sound source was included in order to reduce variance.

The aim of the test was to identify recurring workflow patterns that are associated with specific objectives. Two separate EQ tests were included, allowing for a distinction to be made between corrective and creative mixing styles.

5.2 - Emerging Workflow Patterns

5.2.1 - Corrective EQ

The corrective EQ workflow observations highlighted a common practice when removing resonant frequencies.

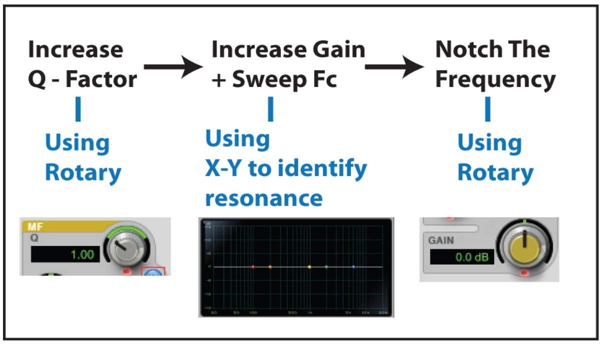


Figure 5-1 Removing Resonance, Workflow and Control

Figure 5-1 demonstrates how three separate controls were used to complete a four-step process. The graphic 'X-Y' control refers to the area of the plug-in that allows simultaneous adjustment of gain and frequency by clicking and dragging 'nodes'. This is an example of workflow improvement brought by the adaptability of digital GUIs.

The graphic 'X-Y' control of a Pro Tools EQ plug-in offers all of the functionality required to notch a resonant frequency. However, 82% of participants preferred the use of rotary controls because of their discrete mapping and therefore independent control of either frequency or gain. This resulted in a more convoluted workflow but possible improved accuracy. Could gestures offer accuracy without compromising the workflow?

5.2.2 - Creative EQ

The mixing styles of engineers in the creative EQ task were much more varied. The majority of participants spent a significantly larger amount of time using the graphic interface on this task in comparison to the 'corrective EQ' test. The tendency to choose the graphic interface over the more accurate rotary controls suggests a preference for time saving techniques and possible reliance on visual cues. One noteworthy disruption to the workflow was the need to move away from the x-y panel to adjust ‘Q - factor’. Surprisingly, 100% of participants were unaware that Q could be adjusted using the x-y interface by clicking and dragging in the coloured area between the node and the 0dB line. The relevance of this in relation to gestural control is the importance of learning techniques to improve engineer workflow.

5.2.3 - Compression

In a similar way to the creative EQ task, the workflow patterns during the compression task were very much dependent on the participant’s mixing style. The *order* of the parameter changes were varied and, in many ways, irrelevant with respect to the final outcome. However, a characteristic of compression that effected workflow was the need to constantly switch between controls. The nature of compression control means that three parameters directly influence each other; threshold, ratio and make-up gain. As a result, time and efficiency is lost with the need to cross-reference and switch controls on the plug-ins GUI (graphic user interface). Could gestural control improve efficiency? One potential solution would be to offer simultaneous control over two or more parameters.

5.2.4 - Gating

The predominant workflow pattern when using a gate was the participants’ tendency to establish a balance between threshold and ratio before adjusting the envelope (attack, hold and release) settings. This does not prove one parameter’s importance over another, but it does indicate a workflow priority when faced with a mixing task.

6 - Preliminary Test 2 - User Defined Gesture Elicitation

The user defined gesture elicitation tests aimed to find the most commonly associated gestures for standard controls of the three processors under investigation. These tests would establish gesture to audio-process mappings, and therefore help to identify ergonomic and intuitive control methods.

6.1 - Proposed System Structure

A preliminary system structure was specified so that the gesture elicitation test could be designed effectively. Defining a control hierarchy at this stage ensured that the derived gesture set would be suitable for use globally within the system.

Please refer to Appendix B for a graphic representation of the proposed gestural control system structure. Each box in the structure diagram should be allocated a gesture, with each branch of the system requiring a set of unique gestures (This has been illustrated with the use of colour, where boxes of the same colour must not contain the same gesture). This arrangement would allow the top 'selection layer' to be accessible at any point in the system.

It is important to note that the proposed structure demonstrates the *maximum* number of unique gestures required. It was a primary goal of this project to reduce the number of gestures and subsequent steps in the workflow. It was predicted that a parameter selection layer would only be required for EQ, because of the large number of controls present in a standard multi-band EQ. It would be preferable to bypass the 'parameter selection layer' for the remaining processes.

6.2 - Test Methodology

The tests will be carried out using the 'paper prototyping' technique described by Heydekorn et al (2010). The methodology aims to formulate a gesture set through conscious top-down design and investigation (Nielson et al, 2003). Participants are asked to suggest a gesture locally, they are not confined by an awareness of global system structure. In particular this is relevant to inter-process control, for example, a user shouldn't feel like they can't suggest an *inward pinch* to increase the ratio of a compressor because they want to reserve it for adjusting the Q-factor of an EQ filter.

The tests took part in an isolated room to keep the test procedure and results hidden from other participants. Each participant was asked to mimic his or her proposed gesture on a tablet computer. As well as helping the participant to understand the context of the interface, this tactile execution gave an early indication of the ergonomic considerations of touch-screen interaction. The author then recorded their gestures; this was instrumental in maintaining a standardised set of symbols for the representation of touches, taps and movements.

The gesture elicitation procedure was split into four separate tests:

1. Process Selection Tests
2. Compression Control
3. Gating Control
4. Equalisation Control

Twenty-four participants took part in each test. The majority of these were Huddersfield Music Technology Students, with the remaining participants being lecturers and post-graduate students from the same department.

In the 'process selection' tests participants were asked to suggest gestures that they would associate with EQ, compression and gating. They were each given the following scenario: "You are in the mix window of a DAW and you want to add a plug-in to the selected track. Which gesture would you use to open EQ, compression and gating?"

The 'processor control' tests did not require as much contextualization as the 'process selection' tests, as all participants were of a suitable experience level to be familiar with the typical operation of plug-ins.

In the Compression Control tests, participants were asked to suggest gestures that best represent the following processes:

1. Decreasing Threshold
2. Increasing Ratio
3. Faster Attack
4. Slower Release
5. Increasing Make-Up Gain
6. Increasing 'Overall Compression'

In the Gating Control Tests:

1. Decreasing Threshold
2. Faster Attack
3. Longer Hold
4. Slower Release
5. Increasing Gain Reduction

In the EQ Control Tests:

1. Boost Bass
2. Cut Treble
3. Increase Q - Factor
4. Set HPF (high pass filter)
5. Notch a resonant frequency

6.3 - Results

6.3.1 - Process Selection

Figure 6-1 displays the most common gesture choices.

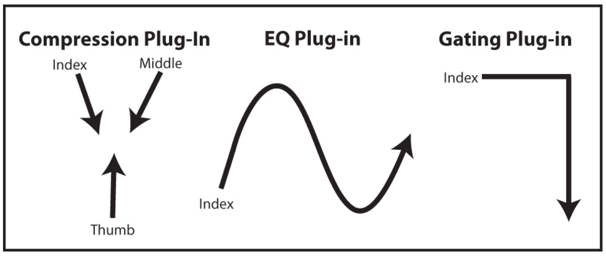


Figure 6-1 Most Popular Process Selection Gestures

The results for compression were conclusive with 83% of participants choosing the same gesture. EQ and Gating were less so, with 38% and 33% of choices respectively. The gesture choice for EQ is representative of an EQ curve from a standard graphic plug-in, a decision perhaps based on existing interfaces. Interestingly, the other two choices are symbolic of the audio process. Where the gesture for compression signifies a 'squashing' of the audio, and the gesture for gating is an interpretation of the amplitude envelope. It should be noted that there was some variance in the number of touches used to implement the gestures. For example a two-touch pinch has been included as a ‘three-touch pinch’ result. Additionally a ‘sine-wave shape’ or ‘gate shape’ gesture performed with two touches rather than one has also been categorised under the same gesture. There are two reasons for this:

1. There is very little user discretion between single and double touch spatiotemporal gestures. Users reported that they felt just as comfortable using one or two fingers to draw a shape.

2. A number of participants commented that a “single touch shape doesn’t feel like a ‘gesture’”. The author believes that they are imagining their suggested gesture as part of a larger system, which is an understandable consideration, but it shouldn’t influence the outcome of this test.

6.3.2 - Compression Control

Participant preferences were broadly varied, however, some common results still emerged:

- Horizontal (x-axis) gestures were used when making time-domain adjustments. This was observed when 73% of participants suggested controls for attack and release time. In most cases a movement left (decreasing x-value) represented faster attack or release time.

- 32% of participants suggested splitting the touch-screen into two or more active areas so that the gestures could be simplified while maintaining full functionality. For example, an upward swipe on the left side of the screen could increase threshold, while a swipe on the right side of the screen adjusts make-up gain.

- The most popular gesture selected for ‘increasing overall compression’ was the same *inward pinch* as chosen in the selection layer survey.

- A common suggestion for increasing ratio was a gesture that simulated the clockwise rotation of a potentiometer, perhaps emulating existing technology.

Figure 6-2 displays a possible gesture set that has been derived from the most common suggestions in the control layer gesture elicitation.

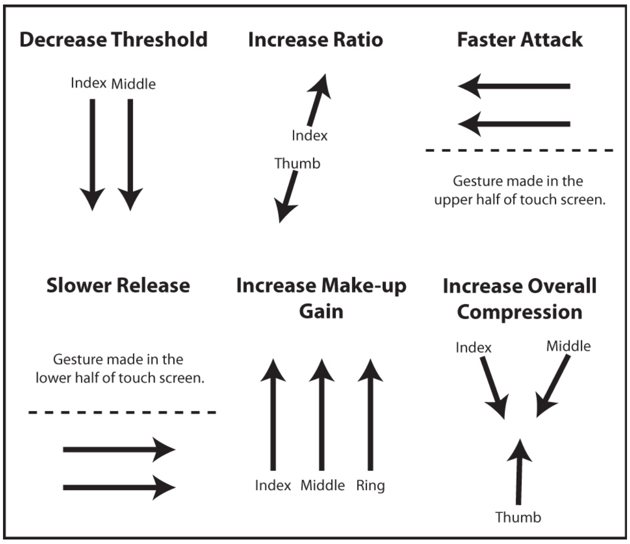


Figure 6-2 Suggested Compression Control Gestures

An observed point of contention was with the participant's interpretation of gestures as value changes rather than direct interactions with the audio. In particular, *increase ratio* was most commonly allocated an outward pinch, with 18% of participants choosing it. However, increased ratio is essentially *increasing* overall compression and therefore, one could assume increased compression should be associated with an *inward pinch,* thus 'squeezing' the audio using a mimetic gesture.

Further testing would be required to determine whether engineers are comfortable disassociating value changes from a representation of the audio process. A similar argument is made when reversing the scrolling of an Apple 'track-pad'; where the default scrolling is based on dragging the page, rather than pointing in the direction of movement (Brownlee, 2014).

6.3.3 - Gating Control

As with the compression tests, it was observed that x-axis motions were preferred for time-domain manipulations. 64% of participants attributed x-axis controls for attack and release, with 68% for hold. 23% of participants chose to split the screen into active areas (as apposed to 32% in the compression tests). In addition to these previously observed gestures, some new solutions arose that related specifically to gating:

- 36% of users imagined drawing the envelope shape to manipulate the attack, hold and release settings of the gate. The majority of these suggestions defined the attack and release times by the angle of the gesture from the x-axis of the touchscreen, whereas others preferred to use the length of the gesture. E.g. drawing a longer line from the bottom left corner would increase the attack time. A defining feature of these gestures is that they're both absolute (they reference a position on the screen.)

- In cases where a 'pivot' or rotational gesture has been suggested for a time-domain adjustment, clockwise has indicated an increase in time and vice-versa.

Figure 6-3 displays a possible set of gestures that have been based on some of the most common suggestions from the gating control layer surveys.

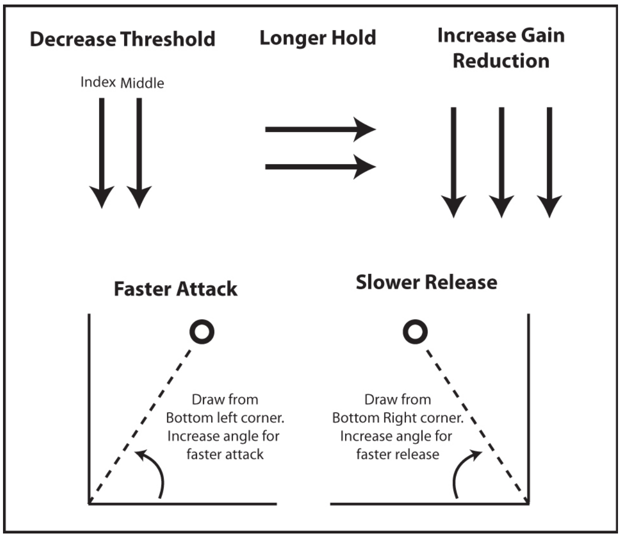


Figure 6-3 Suggested Gate Control Gestures

The attack, hold and release suggestions are examples of gestures that relate directly to the audio process rather than the manipulation of a control parameter on an existing plug-in. This is particularly relevant for the release setting, which translates a clockwise motion to a faster release time. Ordinarily, engineers are used to associating the clockwise motion of a rotary control with *increased* release time, as shown in figure 6-4.

Figure 6-4 Orientation of Release Control in Logic 9

It is reassuring that participants are starting to associate gestures with processes (in this case the amplitude envelope) rather than control values. Even if these suggestions are still x-y dependant. The gating survey differed from the compression survey in the way that envelope settings were grouped together sequentially. This, along with the addition of a hold setting, may have prompted the user to think more carefully about the envelope operation of the gating plug-in. It is important to realise the effect that subtle changes in the survey had on the user suggestions. A further example of this was in the phrasing of 'Increase Gain Reduction'. This was deliberately chosen to observe the number of participants who opted for an upward gesture to 'increase the gain reduction', without considering that this is essentially a 'lowering of the noise-floor', or range, of a gate. 23% of participants chose an upward, or increasing gesture.

6.3.4 - Equalisation Control

The results were more consistent than the broad range of suggestions for compression and gating. Most noticeably was the 'two-touch inward pinch' choice for increased Q-factor, which 91% of participants selected (the remaining 9% opted for a three-touch pinch). This gesture translates so popularly to Q-factor manipulation that it seems to be unequivocally the most intuitive. Similarly, it was 82% of participants that associated and controlled 'frequency vs. gain' with an 'x vs. y' gesture. The remaining 18% were methods that used multiple touches; for example, a one-touch gesture would control bass and a four-touch gesture would control treble. Figure 6-5 demonstrates the elicited gesture set. Again, as with Q-factor, the suggestions of the frequency and gain controls are so consistent that they point very firmly towards the most intuitive solution for a gestural interface. Although, it should be noted that 'most intuitive' does not necessarily mean best control method.

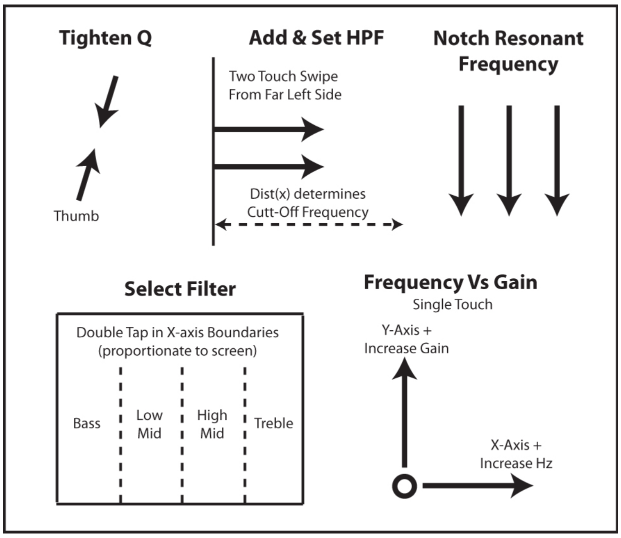


Figure 6-5 Suggested EQ Control Gestures

The issue with mapping controls in EQ units is the large number of parameters, particularly in multi-band parametric software versions. For example, an equalizer with just four filters, where each has selectable Q, cut-off frequency and gain, will have twelve low-level controls. This would simply be too many controls for unique gestures and introduce a level of complexity that would impair the memorability and learnability of the interface (Nielson et al, 2003). Therefore, a parameter selection layer must be included in the system (as discussed in section 6.1). The solution for this is shown in figure 6-5, where a double tap in a fixed x-axis position selects the corresponding filter. This conforms to the survey's results for frequency distribution and allocation across the x-axis. Regrettably, this method requires the system to determine the location of a user's actions in relation to the position on the screen, making the gestures absolute rather than relative (as preferred). Some of these 'expert' design decisions have to be made in order to maintain system stability (Heydekorn et al, 2010).

7 - Description of the Prototype Gestural Interface

This chapter will discuss the *First* Prototype, as derived from the literature review and preliminary testing. Any changes made to the prototype as a result of testing will be detailed and explained in subsequent sections.

7.1 - Justification and Analysis of the Chosen Development Platform

The proposed gestural interface will aim to offer an improvement over WIMP control in small project studios, as with the study by Lech and Kostek (2013). It has been repeatedly reported that an inexpensive solution to improving workflow when mixing 'in-the-box' is desirable, especially when engineers lack the space or funds for a large control surface (Gilder, 2014).

7.2 - Prototype Specification

The findings presented in the literature review were used as a guideline for the specification of a prototype interface. This included both the hardware and software platforms. The author made some decisions based on previous development experience and the availability of certain technologies.

The chosen platform should provide adequate *accuracy*. Ratcliffe (2014) identified that, in order to be considered a success, his 'Motion Mix' system must have an accuracy that users feel is comparable or superior to traditional interfaces. Accuracy can be described as the systems ability to correctly interpret a gesture. It is the author's belief that identifying a gesture incorrectly is one of the most workflow-disrupting occurrences in a gestural interface.

The chosen platform should be adequately *responsive*, with a suitably high level of *sensitivity*. Excessive gesture recognition time proved to be the stumbling point of Lech and Kostec's (2013) otherwise successful gesture system. They found that the speed of interaction with an interface is a pivotal design consideration. A gesture recognition time of 1.5 seconds was deemed to be too slow. Additionally, Selfridge and Reiss (2011) found that a delay of just 0.5secs when setting EQ curves caused too much disruption to the user workflow. As well as a fast reaction time, the system should be sensitive enough to detect subtle changes in gestures. Sensitivity is essentially a measurement of temporal accuracy. In other words, how quickly the system recognises movements by a suitable degree of accuracy.

An impression of the *accuracy* and *sensitivity* of a touch screen can be found by looking at two device specifications:

1. Touch Resolution - The number of 'touch active' points over the full distance of the recognition axis. For example, a touch screen with 64 x 128 touch points would be able to detect 64 points in the x-axis and 128 points in the y-axis. The size of the touch active area would then determine the resolution. If the example screen was 10cm x 20cm it would have a resolution of 6.4 points per cm. Resolution directly influences the 'pointing precision' of a gesture system (Bhalla and Bhalla, 2010).
2. Touch Latency - The time it takes for a system to check for a touch position and update the co-ordinates within the software. This value will directly influence the 'response time' and 'following speed' of the interface (ref: Bhalla and Bhalla, 2010).

High sensitivity and accuracy will allow the successful implementation of a larger gesture set, without relying too heavily on spatiotemporal gestures.

The chosen platform shouldn't be unfamiliar to the user. Familiarity is a keystone component of an intuitive interface (Gough et al, 2006). A user can easily be discouraged by an interface if they are uncomfortable using it, or if it's too difficult to learn. Balin and Loviscach (2011) were determined to keep their gesture set as small and simple as possible in order to rationalise their system. In addition to improving intuitiveness, a familiar interface will also speed up testing. 'Interface Familiarisation time' is a consistent feature of most case studies, a shorter test time will allow participants to feel more comfortable during testing.

The prototype should minimise the reliance on GUIs and visual feedback (Mycroft et al, 2013). The gestural interface should maximize the engineer's ability to mix with their eyes closed in order to facilitate the 'blind mixing technique' (Porter, 2011).

7.3 - Hardware

The literature study suggested that a number of hardware options did not provide sufficient ergonomics for sound mixing purposes (Selfridge and Reiss, 2011) (Karjalainen et al, 2006) (Lech and Kostek, 2013). This list of interfaces includes infrared sensors, accelerometers, camera systems and joystick-style devices.

In order to satisfy the specification in the previous section, the chosen Hardware platform must be capable of accurate gesture recognition with a high level of sensitivity. It should also be a platform that is familiar to users.

From the case studies discussed in the literature review, it can be concluded that two-dimensional gestures recognition systems are more stable and offer faster, more usable HCI. The most familiar platform for two-dimensional gestures is a touch-screen. Furthermore, touch screens do not incur user-fatigue, as found in free-moving gesture systems, such as Lech and Kostec's (2013). It remains to determine which touch-screen platform is most suitable. Mounted touch screens have an ergonomic trade-off, as identified by Balin and Loviscach (2011). Therefore, a handheld device will offer a more universally comfortable interface. Which would be a particularly important characteristic for professional mix-engineer who might be using the interface for extended periods of time.

A comparison of touch screen devices by Agawi TouchMark determines the *touch latency* of a range of popular consumer products (Dilger, 2013). This benchmarking process measures the time it took for the software to respond to a touch event, referred to as Minimum App Response Times (MART). This was measured in milliseconds. Figures 7-1 and 7-2 demonstrate their results for smartphones and tablet PCs.

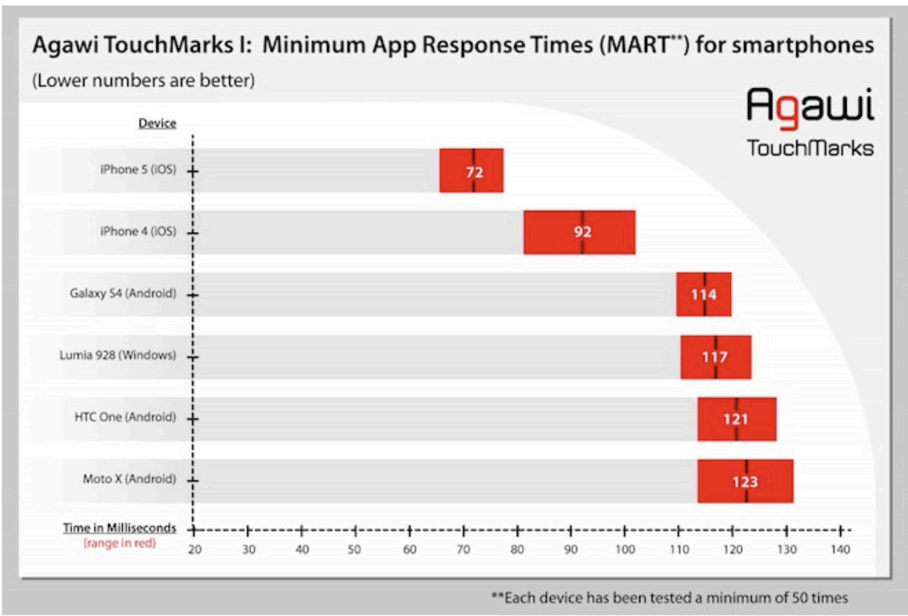


Figure 7-1 Smartphone Response Times (Dilger, 2013)

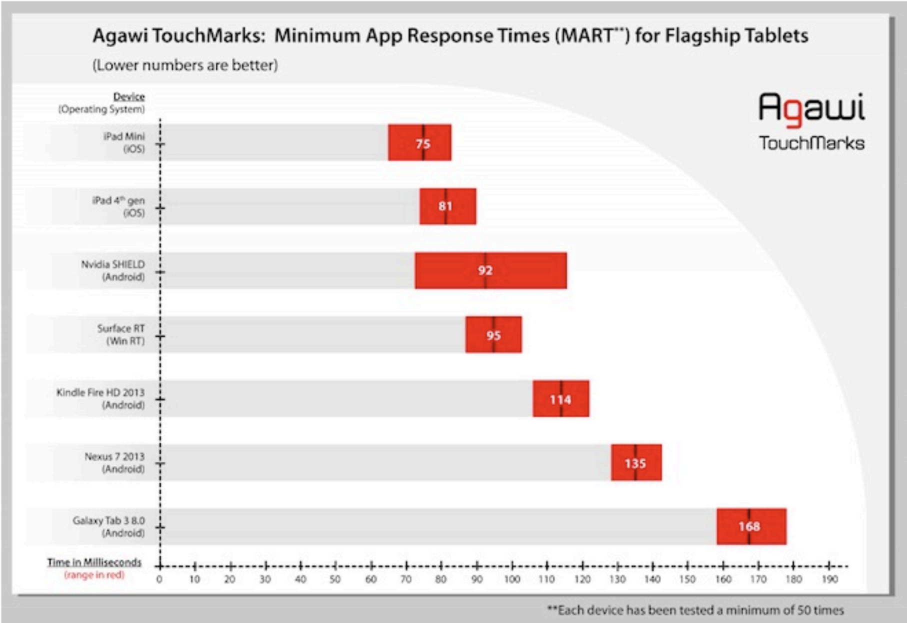


Figure 7-2 Tablet Response Times (Dilger, 2013)

Both tests conclude that Apple devices have faster average response times. However, it is by Agawi's own admission that this could be down to software factors rather than touch-screen capability (Clover, 2013). Research by Microsoft finds that a response time over 100ms can be seriously detrimental to the user-experience (Wrenn, 2012). They reported that latency doesn't go unnoticed until it is as low as 1ms. Current consumer technology cannot reach those speeds; so minimizing latency is an important design consideration.

The Apple iPad offers a competitively responsive platform. It is also an extremely ubiquitous device, one that has been the top-selling piece of tablet hardware for over 5 years (although Android is now the best-selling operating system) (Whitney, 2014).

7.4 - Development Software

A new piece of software will be developed in order to fully customise the gesture set. Default gesture recognizers such as pinch, rotate and multi-tap will not offer sufficient command variations for this system.

The development options for the chosen hardware platform are, aside from using third party applications such as TouchOSC, limited to Apple's Xcode. Xcode uses a programming language Objective-C. Objective-C is an object-oriented superset of C (Apple Inc, 2014a).

7.5 - Limitations of the Development Platform

To increase system stability, some minor changes were made from the gestures specified in Chapter 6. Firstly, the decision was made to 'distance' gestures from each other as much as possible to avoid gestural over-laps and misinterpretation. Most commonly, this manifested itself in the addition or removal of touches, an alteration that has already been established as unimportant to gestural associations. For example, test-participants chose a two-touch vertical swipe to control the threshold of a gate. But this inherently clashes with the two-touch horizontal swipes to set hold. Figure 7-3 demonstrates this issue with continuous gesture direction recognition and introduces the Response vs Accuracy trade-off.

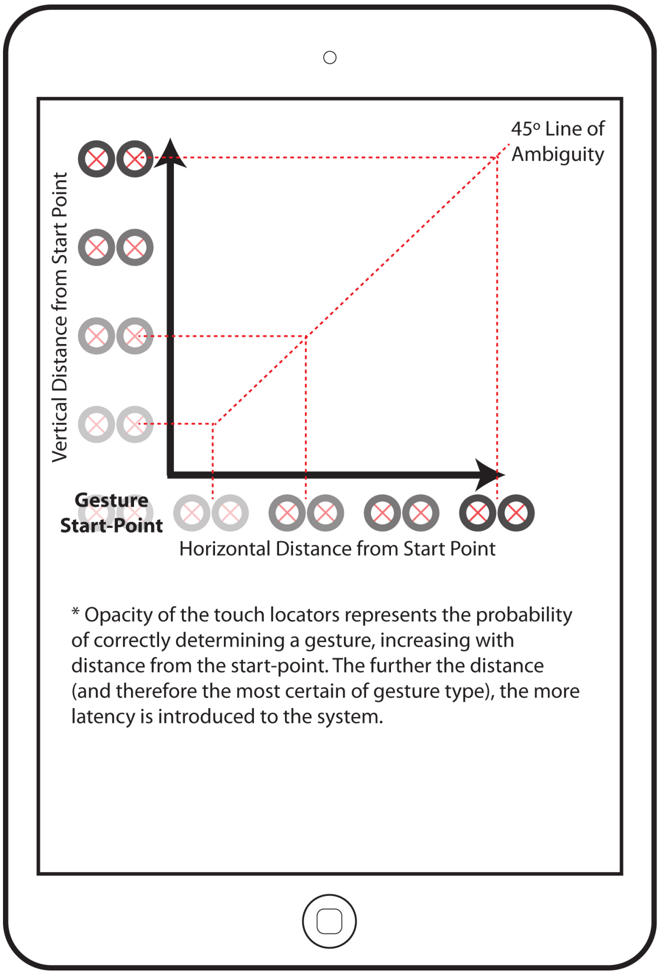


Figure 7-3 Gesture Direction Detection, Sensitivity vs. Accuracy

The Direction Dependent Gesture Recognition Process (where number of touches are equal in multiple gestures) follows the listed procedure:

1. Determine how many touches have *started* the gesture.
2. When a movement is detected, store the start location.
3. Check the location of the touches after a time delay* (typically around 100ms).
4. Calculate the probability of the gesture either being *vertical* or *horizontal* type by measuring the distance between the start point and current point.
5. If the probability is sufficiently high, report the gesture as either vertical or horizontal and begin mapping movement to corresponding parameter changes. If probability is not sufficient, repeat steps 3 and 4.

* The time delay can be changed to adjust the *sensitivity* of the gesture recogniser. Faster times = more sensitive, but more likely to misinterpret a gesture

The longer that the system waits to determine a gesture, the more likely it is to have correctly deduced it's purpose. However, this introduces latency to the system. There is a sensitivity (responsiveness) vs accuracy trade-off. This can be overcome by using unique numbers of touches for each gesture. Therefore, a single touch gesture to control the threshold of a gate would produce a more responsive, stable and optimised system. Regrettably, this requires the user to remember more gestures and touch combinations.

Inevitably, there were going to be some gestures that ended up clashing in the system. For example, setting compressor gain with a three-touch *vertical* swipe and compressor attack/release with a three-touch *horizontal* swipe. In these cases an algorithm was implemented that analysed the very first movements of a gesture to determine whether it was horizontal or vertical, as described in figure 7-3. This proved to be a stable solution, but it increased 'gesture recognition time' and reduced responsiveness.

7.6 - Hardware Specification

All details from (Apple, 2014a), unless otherwise stated.

Model: Apple iPad Mini, non-cellular (2012 - first generation)
External Dimensions: 200 x 134.7 x 7.2mm (Height x Width x Depth)

Weight: 308g

Screen Size: 200mm (diagonal)

Screen Resolution: 1024x768 at 163ppi (pixels per inch)

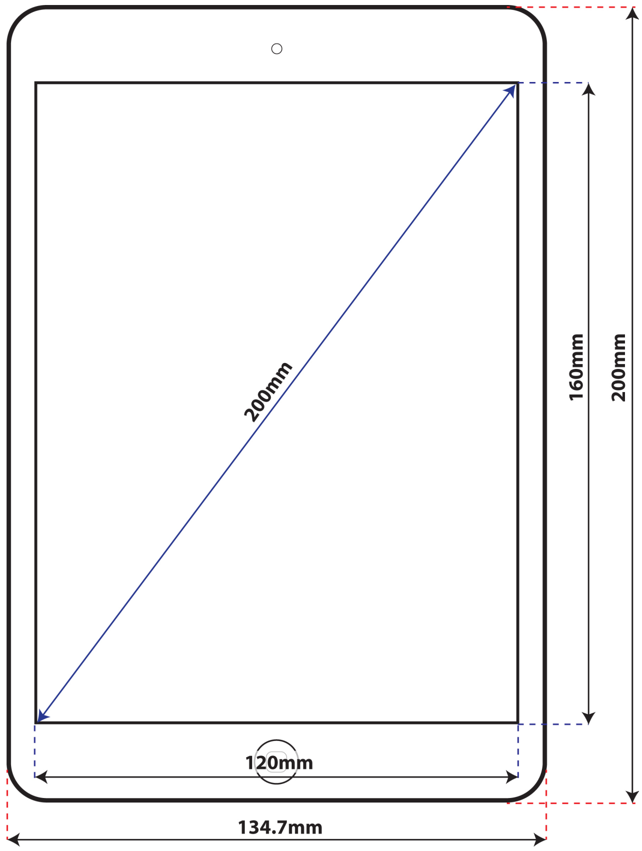


Figure 7-4 iPad Dimensions (based on: Apple, 2014a)

Touch Resolution: 1024 x 768 (estimated)

Processor: Apple A5r2 (S5L8942)

Bit Depth: 32bit

Speed: 1Ghz

Core: ARM Cortex-A9 (dual) (Shimpi et al, 2011)

Screen Treatment: Fingerprint-resistant Oleophobic (oil repellant) Coating

Operating System: iOS 8

The use of an iPad *mini* has influenced the ergonomics of the gestural interface. This has particular relevance to the orientation of the device. Test participants found it comfortable to hold in one hand and operate with the other hand. This may not be the case with a full-sized iPad.

7.7 - System Structure and Navigation

The system structure remained similar to the one specified in the User Suggested Gesture Tests in chapter 6 and presented in Appendix B. Figure 7-5 displays a navigational map of the interface.

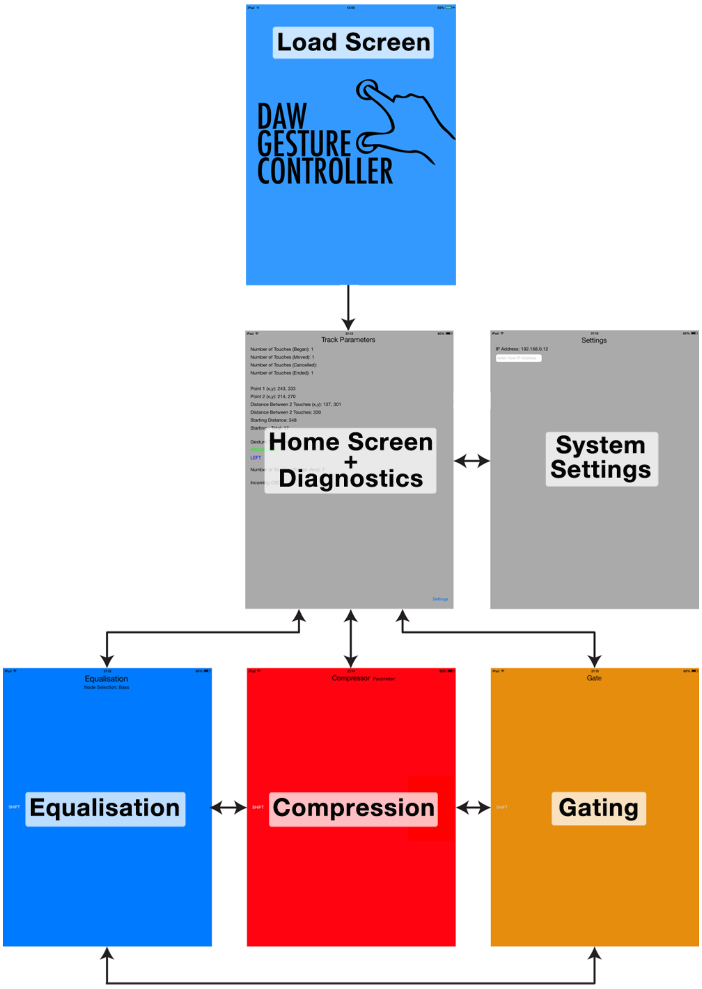


Figure 7-5 System Navigation

The simple page layout allowed movement between processing functions from any system location (excluding the 'system settings' page) through the use of 'process selection gestures'. Due to numerous overlaps between gestures for *selection* and *control* it was necessary to include a 'shift' button that allows users to move between processors or return to the home screen. The shift button was positioned in such a way that it could be classified as an additional, two-handed gesture. Figure 7-6 displays the operation of this two-handed gesture.

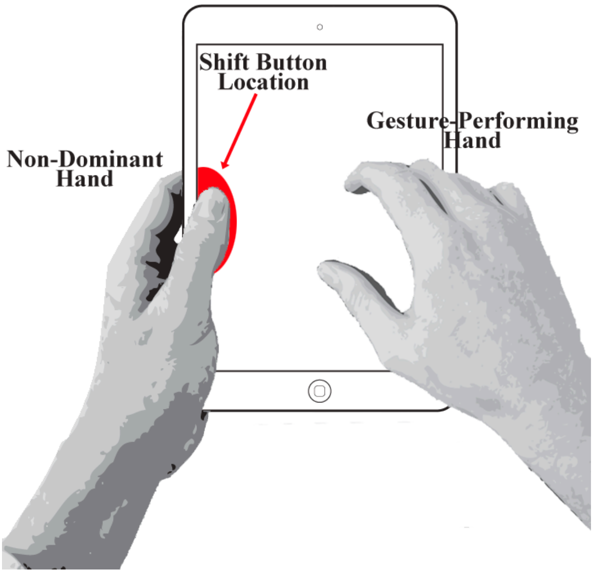


Figure 7-6 Location of the Shift Button

The ideal location for the shift button is highlighted above. In order to consider the ‘shift button’ a gestural technique, the system should allow its position to be moved. Users would then be able to customise the gestures to a grip that was comfortable; this is particularly relevant if a user naturally performs gestures with their left hand. Wigdor et al (2011) describe the use of the non-dominant hand as a mode-indicator in gestural systems, bringing more functionality to the gesture-performing hand. It is likened to using a keyboard shortcut in combination with a mouse movement, with the conclusion that they allow a system to use more gestures without any on-screen affordances or “reducing the expressiveness of the language”. The disadvantage of including a non-dominant hand mode-indicator is that additional limitations are placed upon the user. The user might want to lay the iPad on a surface and use one hand, rather than holding it on one while gesturing with another. For the purposes of testing, users are asked to use both hands when using the interface. It can still be orientated both ways and held in either hand.

7.8 - An Overview of Gesture Recognition Algorithms

The gesture recognition algorithm operates by interrupting the main code when the user touches the screen (For a more detailed explanation of a simplified code example, see Appendix D). These are described as ‘Touch Events’. There are four main touch events within the system (Apple Inc, 2014b):

1. Touches Began - called when the system first detects a new touch. Including the addition of subsequent touches, such as a second touch during the execution of a single touch gesture.
2. Touches Moved - called when a touch is moved.
3. Touches Ended - called when a touch leaves the screen. As in ‘touches began’ this is also called as subsequent touches end.
4. Touches Cancelled - called when a system event (such as a low-battery warning) interrupts a touch.

Continuous gestures could sometimes be successfully detected using only the *touches moved* method. However, using the other touch events during gestural control significantly increased performance of the system. One such scenario that required attention was when a user performed a two-touch gesture, where one touch became static for a short period of time. This resulted in the *touches moved* method only reading a single touch. A series of latches and ‘touch tracking’ algorithms were implemented to ensure the system remained stable.

Spatiotemporal gestures required the use of timers to measure the speed of a gesture. The touch co-ordinates could then be analysed as the gesture progressed to determine which gesture was being performed. A series of probability algorithms monitored the changing co-ordinates to calculate the likelihood of a particular gesture being performed. This gesture detection process was made as flexible as possible. A good example of this can be found in the EQ selection layer gesture, where the ‘amplitude’ and ‘frequency’ of the wave-shape do not stop the gesture from being detected. However, the phase of the ‘wave shape’ has to be the same.

7.9 - Gesture Sets

7.9.1 - Process Selection

The process selection gestures allow the user to switch between compression, EQ and gating. The gesture set has been based on the findings of the gesture elicitation, as discussed in chapter 6.

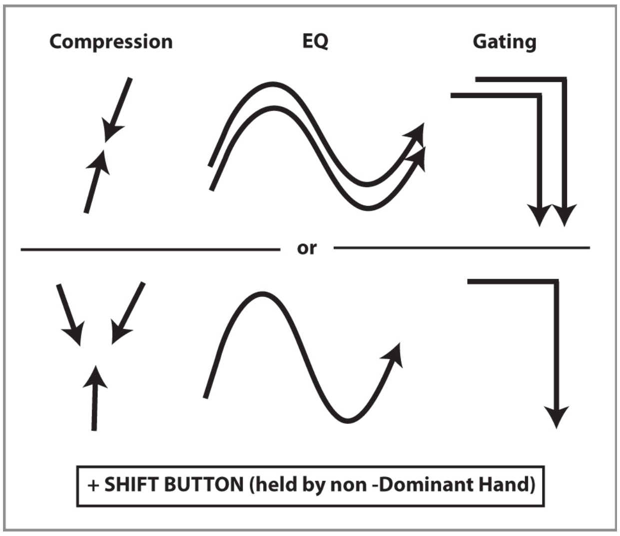


Figure 7-7 Process Selection Layer plus Alternative Operation

Figure 7-7 illustrates the features of the chosen gestures. They remain similar to the specified gestures from the survey. With the exception of the number of touches used to execute each gesture. It was found during pilot testing of the prototype at the 136th AES convention that some users struggled to perform the three-finger compression gesture. Often this was a result of ergonomics or the difficultly of simultaneously starting and finishing a gesture with three touches. For this reason a two-touch pinch was implemented in addition to the three-touch pinch. The resulting response time was faster and more reliable. Another recurring scenario that transpired at the AES convention was that first-time users would forget whether to execute gestures with one or two touches, even though they had remembered the correct shape. For this reason the prototype was adapted so that either single or double touch gestures would select the relevant process. An added benefit of making number of touches arbitrary in the selection layer was that users only have to remember the shape of the gesture, not the number of gestures. Furthermore, ergonomics and usability was improved for a wider number of people.

7.9.2 - EQ Control

The EQ control gestures remained unchanged from the survey-result specification (as detailed in chapter 6). However, functionality was added so that gestures could be simultaneously controlled. This was achieved by allowing the two-touch 'pinch' gesture to be move in an x-y direction to adjust the gain and frequency setting of the selected filter at the same time as setting the Q. A 'sensitivity latch' meant that Q could be changed on its own if no significant movement in the x or y directions was made.

7.9.3 - Compression Control

Figure 7-8 illustrates the compression control gestures implemented in the first prototype.

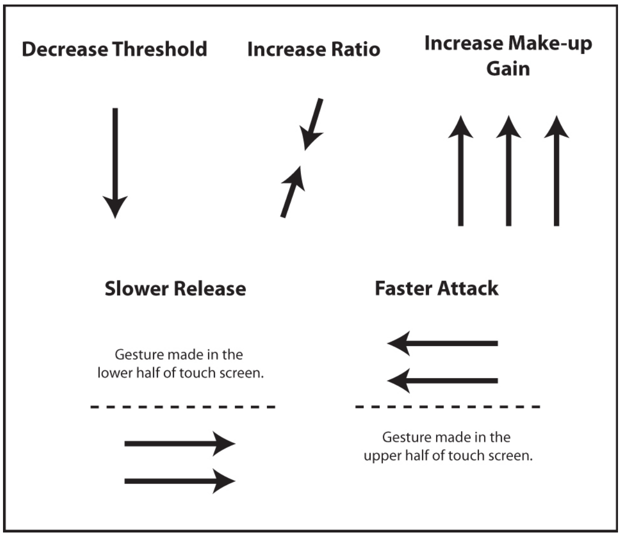


Figure 7-8 Compression Control Gestures

As a consideration of system stability, threshold control has been reduced from two touches to a single touch. A two-touch threshold gesture was more likely to be misinterpreted as an attack or release gesture, this technological consideration is detailed in section 7.5.

The most significant design consideration of the compression control layer is the orientation of the pinch gesture to control ratio. Of the six participants that allocated a pinch gesture for the control of ratio, only two of them chose an inward pinch. However, it is the author’s belief that mimetic gestures will be better suited to an audio gestural control system. Firstly, this will help the user to remember the gesture-set during operation through semantic association, for example a ‘squashing of the audio’ is associated with a ‘squashing’ mimetic gesture. Furthermore, it is hypothesised that users will be less reliant on the value of parameters in a gestural system. Evidence of this has already been observed in the results for EQ control layer survey. Where 100% of participants chose an inward pinch (either 2 or 3 touch) to *increase* Q-factor. This is therefore an example of the user favouring mimetic gestures over direct parameter manipulation. It could be argued that this a result of engineers being more used to looking at graphical representations of EQ curves.

7.9.4 - Gating Control

Gating controls remained similar to the suggested gestures, with the exception of a single touch for threshold and adaptation of the envelope settings. Refinement and rationalisation of the attack and release gestures was required for improved usability and system stability. The suggested gestures used the extremities of the touchpad, in particular the corners of the pad were proposed as a reference point for drawing the envelope. It is the author's belief that this would be constraining for the user, and the reference point should be able to be found without looking at the touch-screen. Therefore, a solution that used the angle between two touches to set attack and release times was developed.

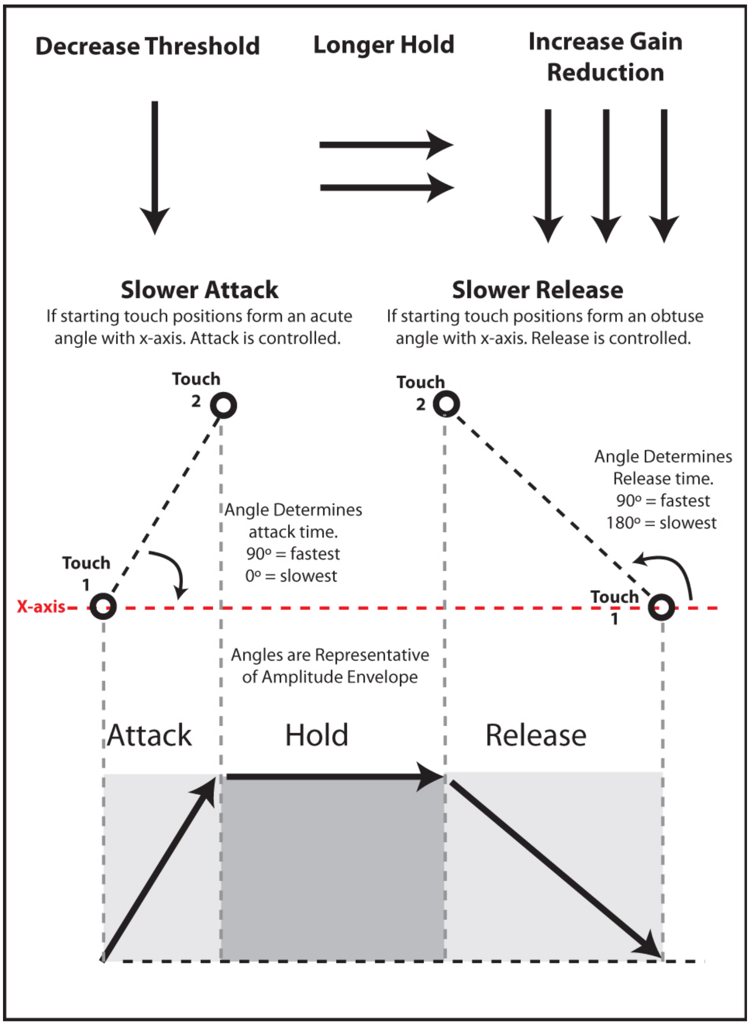


Figure 7-9 Gate Control Gestures & Envelope Explanation

The above figure demonstrates how the attack and release gestures are designed to mimic the amplitude envelope of the gate. Attack and Release are controlled separately, and are selected by the two-touch gesture start position (as illustrated in figure 7-9). The angle between two touches is then mapped to the envelope parameters in a way that is representative of the envelope shape. This design has been adopted in an attempt to move user's perception of control closer to the audio process, rather than preconceived ideas of parameter values.

One notable difference between the 'angle control' gestures and other control gestures is that they are absolute (relative to their orientation on the screen), whereas most gestures within the system are relative to the touch start position. This can cause problems when users are trying to make small adjustments to the their settings, as the setting will be 'forgot' by the system and reset when a new gesture is performed.

7.10 - Implementation

The system transmits Open Sound Control (OSC) messages over a wi-fi network using User Datagram Protocol (UDP). The code was adapted from open-source library files included in Oscpack1.1.0 by Ross Bencina (Bencina, 2013). The OSC messages are then received by an OSC to MIDI bridging application called OSCulator (Wildora, 2014). OSCulator allows OSC data packets to be mapped to MIDI messages, which subsequently allows them to be detected and processed by a DAW. In this instance Logic Pro 9 was used.

8 - Description of Testing

Each control layer will be tested separately to determine the suitability of different audio processes to gestural control and the effectiveness of the interfacing techniques. Equalisation was the first process to undergo testing.

8.1 - Equalisation Control

The intention of this test was to evaluate the *objective* performance of the interface in comparison to traditional methods. It was crucial to assess whether a gestural controller could improve workflow with a task as simple as matching target EQ settings before any subjective elements, such as audio processing, are introduced.

It was hypothesized that the interface for EQ would be easier for users to operate because of its conceptual similarity to existing plug-in interfaces. For that reason, a slightly less-experienced population of participants was chosen.

Testing took place in the Music Technology department at Blackpool Sixth Form College. A meeting with the Head of Music was arranged prior to testing to assess the experience level of the students. It was concluded that second-year students would understand equalization to an adequate level; they had been frequently using EQ plug-ins in their classes for over a year. Their course consists of numerous modules that require intensive use of equalization, including a recording module and a sound for film module. They can be classified as moderate-advanced frequent users (Dewey and Wakefield, 2014).

8.1.1 - Test Methodology

Twenty-Two music technology students were selected for the test process. Each test was intended to take no more than thirty minutes to complete, though there were no limitations introduced to the participants. The test took part in an isolated room to reduce influence on participants and keep the purpose of the test hidden from other students.

Two interfaces were being compared, the novel gestural interface and a traditional WIMP interface that comprised of a standard, wired Apple mouse. Two handouts were produced for the purpose of testing (See Appendix E: EQ Control Test Handouts); one that detailed interface operation instructions and another that illustrated the target settings for the test. Figures 8-1 and 8-2 show the two target EQ settings.

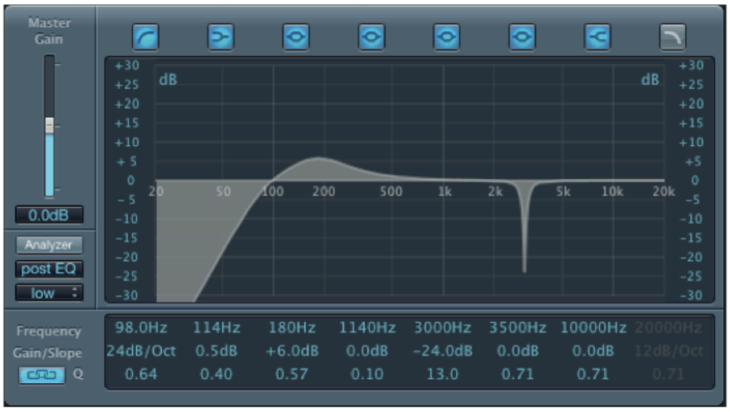


Figure 8-1 Target EQ Setting 1

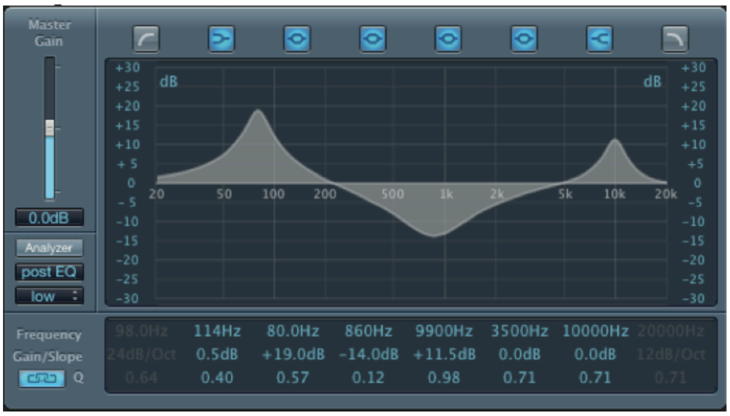


Figure 8-2 Target EQ Setting 2

It should be noted that the target settings, presented in the above figures, are of a similar complexity. They each have three filter settings to match. However, 'Target EQ Setting 1', has a High Pass Filter (HPF) in place of another parametric filter. Throughout the test, the dB/octave setting of the HPF was at a fixed value (participants were made aware of this). Therefore, It could be argued that Target EQ Setting 1 is an 'easier target' because fewer parameters need to be adjusted to reach it.

Before each timed test began, the EQ plug-in was returned to its default, flat, position. The test process was as follows:

1. Introduction - The participant was handed the iPad and asked which orientation they preferred. They were told that the interface worked in both portrait and landscape. Additionally they were then asked to try using the mouse to make sure it felt comfortable and that the sensitivity was correct.

2. Familiarisation Period - The participant was asked to refer to the 'Operation Instructions' sheet and practice mixing an audio source until they felt comfortable using the interface. The average familiarisation time was around 5 minutes.
3. Explanation of Test Process - The participants were introduced to the test process. The level of required accuracy when matching the target settings was described as within 10% and an example was given. They were also reminded that there were no time limitations.
4. First Target EQ Setting - Participant was asked to match the settings using the *Gestural Interface*. Timer begins when the target setting is revealed to the participant. Timer stopped when the participant states that they have finished.
5. First Target EQ Setting - Participant asked to match the settings using the *WIMP Interface*. Timed accordingly.
6. Second Target EQ Setting - Participant asked to match the settings using the *WIMP Interface*. (Note that interface order has been changed for the second target to reduce the influence of target EQ familiarity). Timed accordingly.
7. Second Target EQ Setting - Participant asked to match the settings using the *Gestural Interface*. Timed accordingly.
8. Questionnaire - A questionnaire was handed to the participant for them to fill out in their own time and return to their lecturer by the end of the lab session. This was an attempt to reduce any of the authors influence on the students' answers (A summary of answers are presented in Chapter 9.1.2).

The test has three variables: Interface Type, participant and Test Completion Time (TCT). TCT was measured using a stopwatch. Originally, screen capture software was used, but the running of this introduced a small amount of latency to the system, which was deemed significant enough to compromise the performance of the interface.

8.2 - Dynamics Processor Control

Compression and gating tests took place at Huddersfield University with undergraduate and postgraduate students. It was a deliberate decision to have a variation in participant-experience levels as it would help to determine whether the interface was suitable for both novice and experienced mix engineers, as specified in the introductory chapter.

8.2.1 - Test Methodology

Dynamics Control Testing followed an identical methodology to the EQ control testing (See Appendix F for dynamics control testing hand-outs and questionnaire). Both gating and compression were under test in the same session. The participants were given the opportunity to take a small break between tests.

The Compression and Gating Target settings are presented in figures 8-3, 8-4, 8-5 & 8-6.

Figure 8-3 Compression Target Setting 1

Figure 8-4 Compression Target Setting 2

Note that, even though the 'knee' value is shown as different in both settings, this parameter was not under investigation and did not influence this test.

Figure 8-5 Gating Target Setting 1

Figure 8-6 Gating Target Setting 2

8.3 - Influence of a GUI in the Dynamics Processor Interface

After discouraging results from the first dynamics processor control tests (presented in Chapter 9.2), it was concluded that further investigation of the gestural control of dynamics processing would be required to determine any potential benefits over traditional WIMP methods. Therefore, it was necessary to better understand the relationship between a mix engineer and the mix environment. A test was devised to observe the influence of a GUI when using a gestural interface to mix to an audio reference. This differed from previous tests, as an audio target reference is used, rather than a graphical target setting. The test aims to establish how much an engineer relies on visual stimulus during mixing, and whether a gestural interface might be better suited to a less-traditional GUI.

8.3.1 - Description of the Proposed Visualisation

In order to test the suitability of traditional GUIs in a gesture-controlled system, an alternative visualisation method was produced and displayed on the touch pad. An inherent benefit of this design was that the visuals were located in the same place as the controls, which meant that the user could interact directly with the GUI.

The proposed gating visualisation was based on an amplitude envelope (Time vs Amplitude) design. Figure 8-7 illustrates the visualisation of two different gate settings to demonstrate the operation of the GUI.

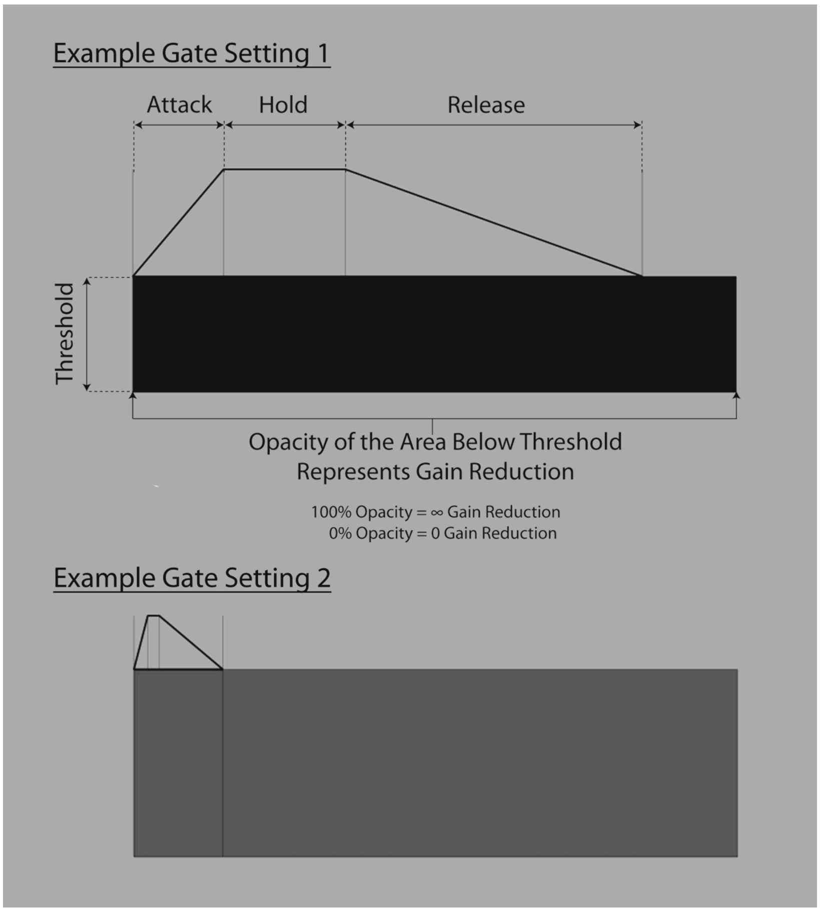


Figure 8-7 Gate Visualisation Operation

As shown in the figure above, a typical envelope representation is adapted to display gating settings. The resulting GUI illustrates the operation of the gate by combining a time domain display (the envelope) with an impression of the effects of threshold and gain reduction. Figure 8-8 shows how the visualisation looks within the gesture control application:

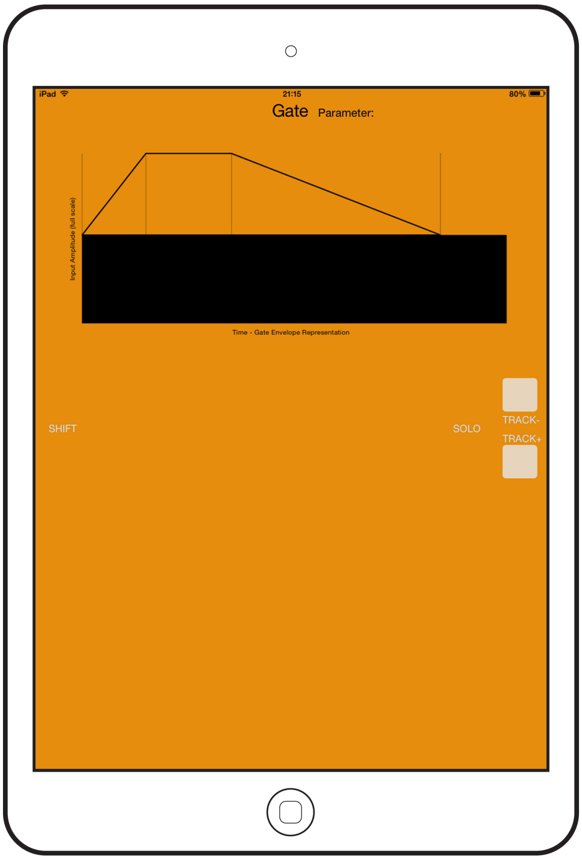


Figure 8-8 Gating Control Visualisation

The compression GUI employed a similar technique, as shown in figure 8-9. This was a design decision that helped maintain consistency, and therefore familiarity, within the application.

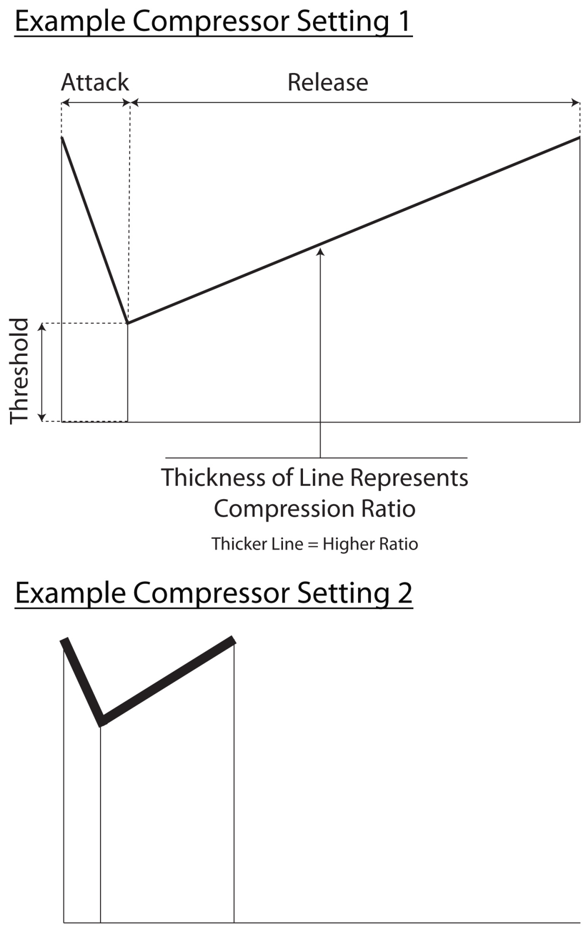


Figure 8-9 Compression Visualisation Operation

The two example settings given in the above figure display the difference between a low-threshold, low-ratio setting (setting1) and a high-ratio, high threshold setting (setting2). The 'line-thickness' method of visualisation intends to communicate the metaphor of 'brickwall compression', where an extremely high ratio prevents the audio signal passing the threshold level. It is hoped that this perspective will help to justify mapping inward pinch to increased ratio (squashing the audio with a pinch). Figure 8-10 shows how the visualisation fits into the gesture control application:

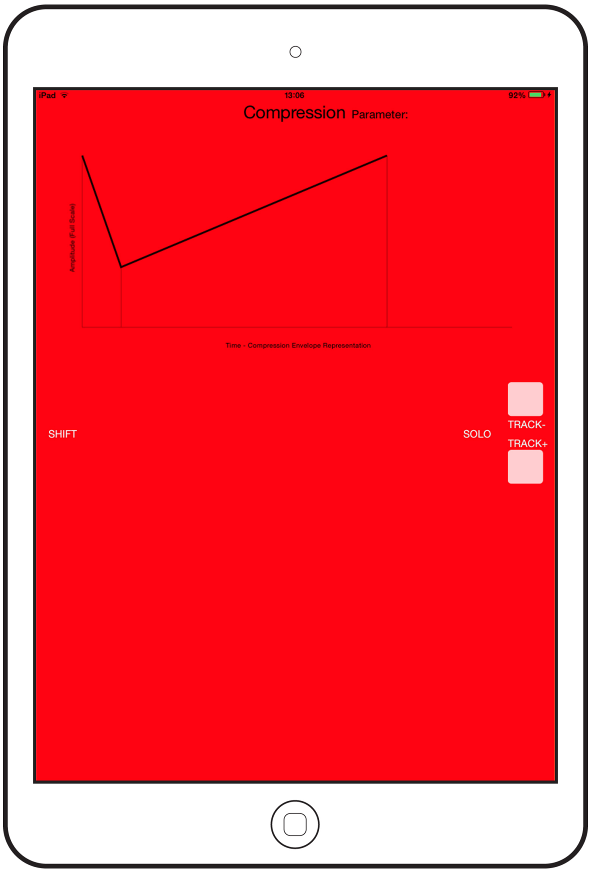


Figure 8-10 Compression Control Visualisation

8.3.2 - Test Methodology

22 expert-frequent participants were chosen, the majority of which were University of Huddersfield Music Technology students and lecturers. The tests took place in a semi-anechoic chamber with a pair of Genelec 8040A reference monitors. A standard, wired apple mouse was used for the WIMP interface tests. Logic 9 and its suite of stock plug-ins were used for all testing.

Reference tracks were created from kick drum audio samples. The sample used for gating tests was a kick from a live recording that had a significant amount of guitar spill present; the mix task was to remove the guitar spill as much as possible (with reference to a preprocessed version). The sample used for compression was a kick drum from the Logic Sample library, the sample was processed so that there were four hits, each with accumulatively less on-velocity. The mix task was to compress the sample so that each hit was more consistent, with reference to a preprocessed version. The test-participant was told that they could check the reference sample at any point during the tests (A set of navigation buttons, as shown in figure 8-10, were added to the gestural interface which allowed the reference to be soloed, this way the mouse didn't have to be used at any point in the gestural tests).

The tests were timed using a stopwatch. The timing was stopped when the participant announced that they were satisfied with their processor settings.

Each participant carried out Gating and Compression tests in the same session; both tests followed an identical methodology. They were given the opportunity to take a short break between the gating and compression stages tests.

1. Prior to testing, the iPad was cleaned to reduce the amount of grease and dirt on the touch-surface. This was for both hygienic and interface-performance reasons, as a build up of fingerprints can impair the fluidity of executing gestures.
2. Introduction - The participant was handed the iPad and asked which orientation they preferred. They were told that the interface worked in both portrait and landscape. They were then asked to try using the mouse to make sure it felt comfortable and that the sensitivity was correct.
3. Familiarisation Period - Participants were told that they could have up to ten minutes to practice using the gestural interface. They were also introduced to the custom visualiser. A practice track was provided for the participant to experiment with the gestural interface, this contained a different audio sample to the one in the test.
4. WIMP Interface Test - The participant was asked to match the reference sample using a traditional WIMP interface. *Note: plug-in was returned to default position after each stage of the test.*
5. Gestural Interface with Plug-In GUI - The participant was asked to match the reference sample using the gestural interface, with the touch-screen visualisation turned *off*.

   Note: The order of steps 4 and 5 was randomized between tests to reduce the influence of familiarization with repeated settings on the overall results.
6. Gestural Interface with Touch-Screen Visualisation - The participant was asked to match the reference sample using the gestural interface, with the touch-screen visualisation turned *on* and the plug-in GUI hidden.

7. Gestural Interface, Blind - The participant was asked to match the reference sample using the gestural interface without *any* visualisation.

8.4 - Semantically Motivated Combination of Compression Controls

Throughout this project, the gestural control of compression has proved the most troublesome to test participants. This has been the topic of discussion in a number of investigation projects, including Giannoulis et al (2013), who attribute the difficulty of working with a compressor to its non-linear, time dependent operation. They suggest that automating the parameters, through the analysis of the input signal, will reduce the required amount of user interaction and the number of control parameters, thus simplifying and improving the interface. Similarly, Cartwright et al (2014) propose the combination of parameters into a single control. Their 'Mixploration' interface is operated by moving a ball in a two-dimensional plane, where movements are mapped to changes in spatial characteristics of the mix.

It is hypothesised, through observations made in this study, that the difficulty in controlling compression with a *gestural* interface was down to two main reasons:

1. Inter-parameter influence and dependence - Within a compression processor, similar sonic outputs can be achieved with vastly different parameter settings. This was apparent during the accuracy analysis, where plug-in settings were distant from the reference settings, but the cross-correlation accuracy result remained close to the reference. For example, a setting with high threshold, high ratio can be perceived as sonically similar to a setting with low ratio, low threshold. This can cause confusion to a mix engineer, especially when mixing to a reference, as it creates an element of uncertainty when experimenting with combinations of settings. Make-up gain is also a contributor to the interdependence of parameter values.
2. Difficulty perceiving envelope changes (source dependence) - When an engineer is making adjustments to the envelope of a compressor, the changes are not immediately noticeable as they are dependent on transients within the source. For example, the release of a compressor might be adjusted before the audio source has crossed the threshold, which would have no audible effect. The author believes that this is the main reason why participants struggled

with compression in the Blind Gestural Mixing Tests. If the source was not at a transient part of the audio, they could not perceive changes in the parameters. This is why visualisation might be essential for a compression processor.

A method of further improving the gestural interface with regard to Compression might be to rationalise the parameter changes and offer 'gestural - shortcuts' to the engineer. Ultimately, this could be mapped to the 'overall compression gesture' specified in chapter 6 (gesture elicitation).

One of the disadvantages of fixed parameter presets is their inability to adapt to changing sources. For this reason, a certain level of engineer interaction is required to evaluate the subjective features of a mix. However, a 'gestural shortcut' could offer continuous control over a 'preset ratio'. For example, threshold and ratio could be controlled simultaneously through a ratio that moves between *hard* and *soft* compression presets. Effectively this would produce a 'continuous preset'.

8.4.1 - Parameter Elicitation Process

We can elicit a relationship between threshold and ratio by interpolating between two compression presets, hard and soft compression. This will provide a control that can be mapped to 'overall compression'. The parameter examples for hard and soft compression will be taken from the SAFE (semantic audio feature extraction) audio plug-in suite. As introduced in Chapter 2.4, The SAFE semantic audio project is an investigation that aims to find an average audio-processor preset for a given semantic descriptor. The project operates by offering a free plug-in suite that, in return, asks engineers to contribute by uploading settings that they deem to fit under certain descriptors. Two such descriptors are 'hard' and 'soft' compression. Figure 8-11 displays the parameter settings for hard and soft compression.

Figure 8-11 Hard and Soft Presets of the SAFE Compressor Plug-in

The parameter values from both soft and hard SAFE presets are listed below:

	Soft Compression	Hard Compression
Threshold:	-28.9dB	-33.5dB
Ratio:	3.4:1	5.1:1
Attack:	8.9ms	10.3ms
Release:	218.5ms	1291.3ms
Gain:	6.2	8.1

By interpolating between the parameter values, it is possible to provide *overall compression* control. This will rationalise the control process and could be implemented as a gestural shortcut.

Plotting *ratio* against *threshold* value produces a linear gradient between the two settings. The reader should note that, although the threshold is logarithmically scales (measured in dB), it is still deemed adequate to determine a relationship between threshold and ratio on a linear scale. Figure 8-12 displays this graphically.

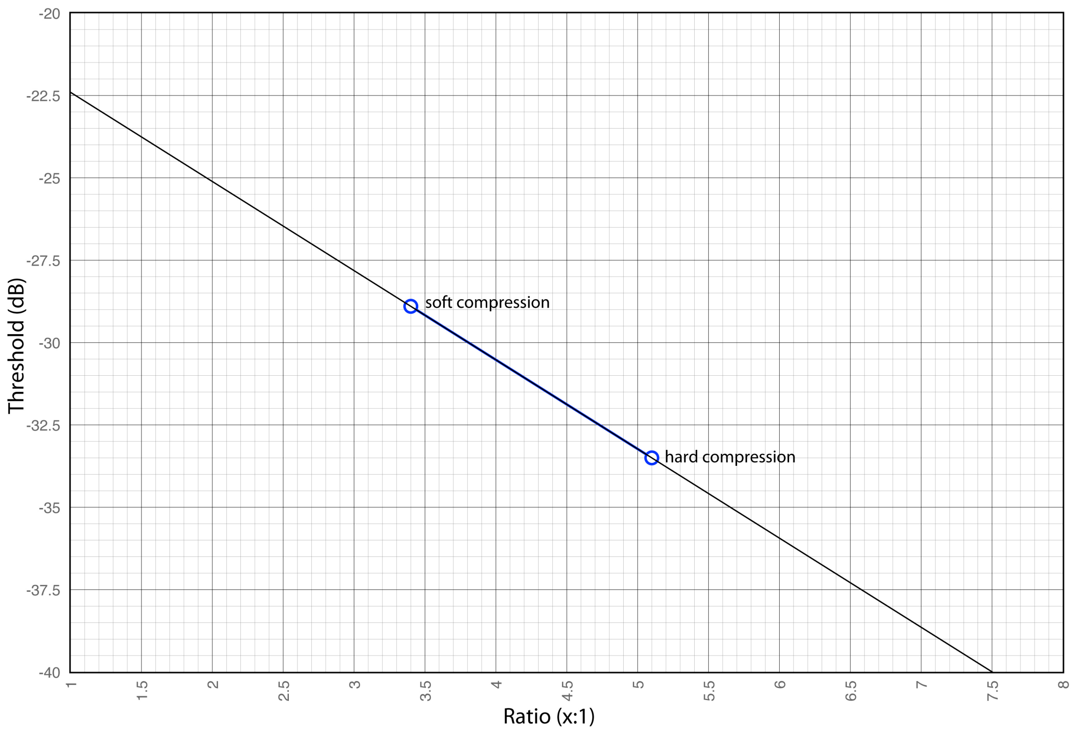


Figure 8-12 SAFE Preset Interpolation (Ratio vs. Threshold)

The relationship between ratio and threshold is given by the equation:

$y = -2.7059x - 19.7$, where x = ratio and y = threshold.

One detrimental outcome of this rationalization is that the extreme compressor settings cannot be reached. The equation can be scaled to allow for 'extreme' settings. A setting is classified as extreme when its values lie outside the boundaries of the hard and soft SAFE pre-sets.

Cartwright et al (2014) described this parameter combination process as 'hill-climbing'. They experienced the same scaling issue during the development of their Mixploration system, where the 'local maximum is less than the global maximum'. In the SAFE example, this occurs as the ratio reaches 1:1, thus limiting the threshold to a maximum of -22.5dB.

Figure 8-13 shows one scaling solution that allows the full range of thresholds to be selected.

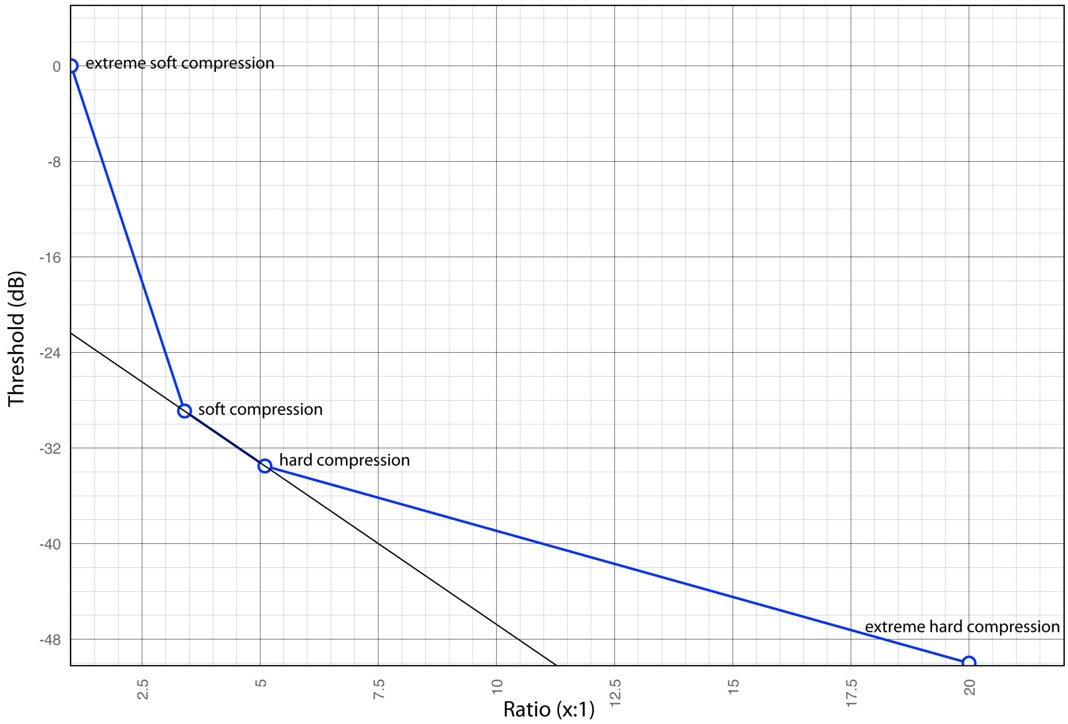


Figure 8-13 SAFE Preset Interpolation with Extremes

However, this approach would distort the relationship of parameters within the SAFE gradient. For the purposes of this test, a linear response was deemed adequate.

By repeating the elicitation process (in reference to threshold changes) with the remaining compression parameters, an 'overall compression' control value that ramps between *hard* and *soft* compression can be produced. This value can then be mapped to a gesture for rationalised control, which could be referred to as a 'continuous preset' or 'gestural shortcut'.

<u>Gain(x)</u>

y = -2.4211x - 13.8895

<u>Attack(x)</u>

y = -3.2857x + 0.3429

<u>Release(x)</u>

y = -0.0043x - 27.9628

This relationship can be given in a single expression by:

$$t = -2.7059r - 19.7 = -2.4211g - 13.8895 = -3.2857a + 0.3429 = -0.0043R - 27.9628$$

where: t=threshold, r=ratio, g=gain, a=attack, R=release.

8.4.2 - Test Methodology

A test was designed to evaluate the effectiveness and usability of the rationalised controller. The test aims to determine whether engineers can suitably match a reference track when the parameters are 'locked' into interpolated values between SAFE presets. For the purposes of this test, a simple one-touch gesture will be implemented, as we are more focused on the performance of the preset parameters than the gestures themselves. For example, at a later date, the SAFE control could be mapped to an 'overall compression' gesture. Effectively this would produce a high-level 'gestural shortcut'.

Participants, reference samples (a kick drum) and all equipment were kept consistent from the previous compression tests. This allows the comparison of interface performance during analysis of results. A total of three reference samples were produced, as shown in figures 8-14, 8-15 and 8-16. The compression settings for each reference were chosen to evaluate the accuracy capabilities of the 'SAFE continuous preset'. Reference one was the same audio sample used in the previous compression test, the compression values used to set create this sample *could not* be matched using the SAFE continuous preset. Reference two had a compression setting that could be replicated *exactly* by the continuous preset. Reference three was chosen to represent a setting that was far from the boundaries of the continuous preset.

Figure 8-14 Reference 1 Compression Settings

Figure 8-15 Reference 2 Compression Settings

Figure 8-16 Reference 3 Compression Settings

No parameter values are presented to the participant, thus encouraging them to 'trust their ears' (Cartwright et al, 2014).

1. Familiarisation Period - Participants were handed the interface and give a 'practice track' to get used to the operation and responsiveness of the control.
2. Interface used to match Reference 1.
3. Interface used to match Reference 2.
4. Interface used to match Reference 3.
5. Participant asked to complete a short preference questionnaire (see Appendix G).

   *** The order of steps three, four and five were randomised (using a random number generator) for each participant. This helped to alleviate the influence of both interface and test familiarity on the average results.

9 - RESULTS

9.1 - Equalisation Control

9.1.1 - Target Matching Times

The test completion times presented in table 9-1 have been normalised so that the results can be compared independently of participant performance, where 1.0 represents the longest time taken per subject and the remaining values are calculated proportionally. This is necessary because some test participants were much quicker at matching target settings and this should not reflect on the effectiveness of the interface, generally this was due to level of experience with EQ plug-ins.

	Normalised Target Matching Time (NTMT)			
	Target Setting 1		Target Setting 2	
Interface Type	**Gestural**	**Traditional**	**Traditional**	**Gestural**
Participant 1	0.43	0.80	1.00	0.37
Participant 2	0.84	0.92	1.00	0.46
Participant 3	0.58	0.78	1.00	0.49
Participant 4	0.43	1.00	0.53	0.50
Participant 5	1.00	0.54	0.87	0.51
Participant 6	0.60	0.40	1.00	0.76
Participant 7	0.61	0.86	1.00	0.48
Participant 8	0.60	0.59	0.76	1.00
Participant 9	0.90	0.66	1.00	0.66
Participant 10	0.64	1.00	0.89	0.44
Participant 11	1.00	0.96	0.63	0.65
Participant 12	0.66	0.56	1.00	0.91
Participant 13	0.39	0.28	1.00	0.57
Participant 14	0.57	0.81	1.00	0.46
Participant 15	0.40	0.42	1.00	0.42
Participant 16	0.70	0.71	1.00	0.92
Participant 17	0.50	0.64	1.00	0.59
Participant 18	0.30	0.71	1.00	0.26
Participant 19	0.96	0.92	1.00	0.92
Participant 20	0.52	0.67	1.00	0.64
Participant 21	0.75	0.58	1.00	0.93
Participant 22	0.56	0.88	1.00	0.66
MEAN	**0.63**	**0.71**	**0.94**	**0.62**
Standard Dev	0.20	0.20	0.13	0.21

Table 9-1 Normalised Target Matching Times (NTMTs) for EQ

By inspecting Table 9-1, it can be observed that the longest average time was taken with the traditional *WIMP* interface for the second target setting, by contrast the fastest average time was the *gestural* interface for the second target setting. The categories will hereby be referred to as Gestural Interface Time - Target 1 (GIT-1), Gestural Interface Time - Target 2 (GIT-2), WIMP Interface Time - Target 1 (WIT-1), WIMP Interface Time - Target 2 (WIT-2). Figure 9-1 provides a graphical representation of the data.

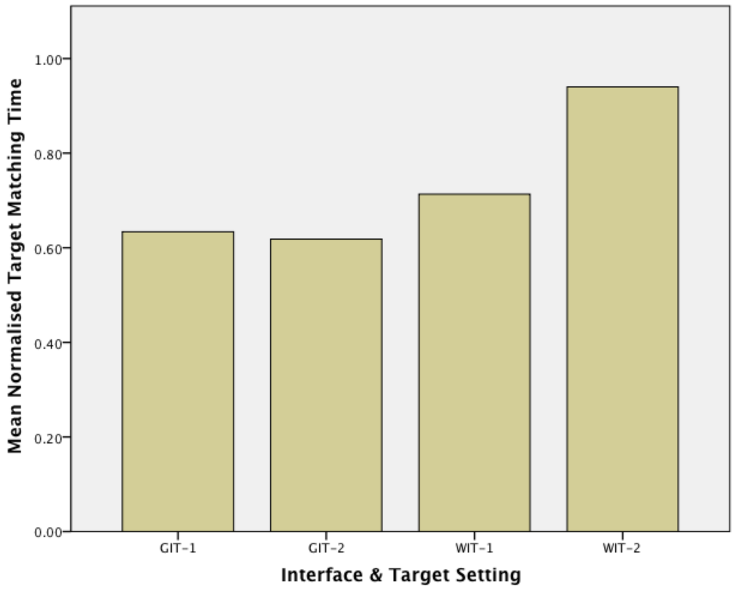


Figure 9-1 Mean NTMTs for Each Interface & Target

In order to test the data for statistical significance, it should first be determined whether the task completion times are normally distributed. The times presented in table 9-1 are normally distributed data sets, as assessed by the Shapiro-Wilk test ($p > .05$) (Lund Research, 2014), with the exception of WIT-2, which returned a value of ($p < .05$).

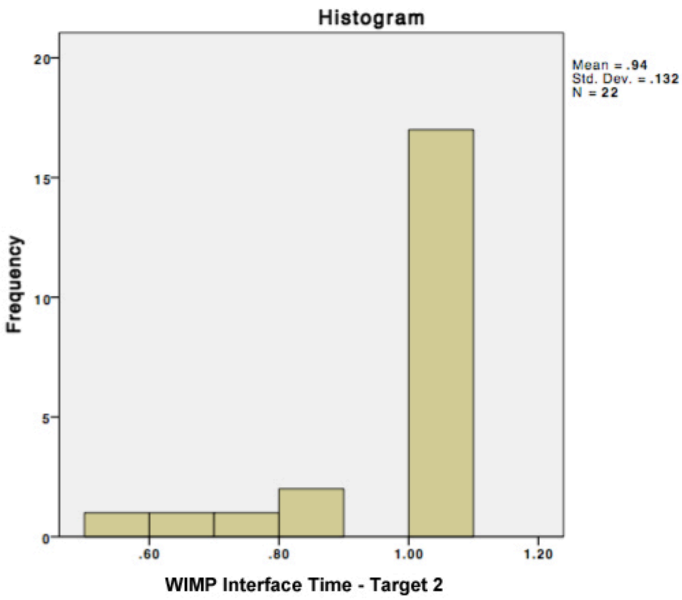
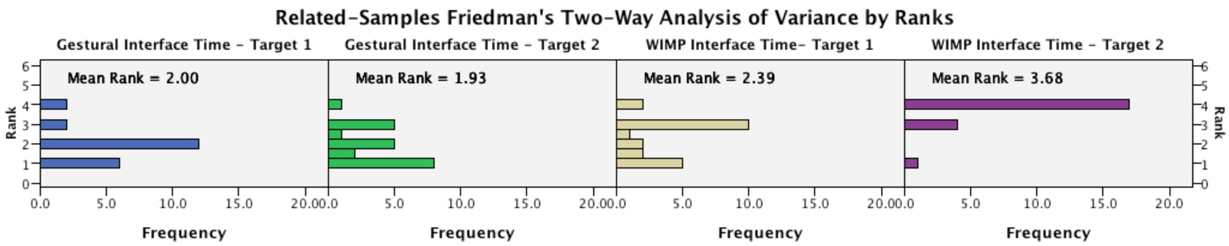


Figure 9-2 Negatively Skewed Distribution of WIT-2

The histogram in Figure 9-2 displays the results of the normality test for the WIT-2 dataset. Visual inspection confirms that the results are not normally distributed, due to significant negative skewness of the data. In other words, most people were slowest with the WIT-2 test, therefore the results are unevenly distributed. Resultantly, the non-parametric Friedman test was used to evaluate the statistical significance of the data.

The time taken with each interface is statistically significantly different, as reported by The Friedman Test, with a returned value of $X^2(3) = 26.530$, ($p<0.0005$). Where X^2 is the distribution type, (3) is the degrees of freedom, 26.530 is the Friedman Test Result and ($p<0.0005$) is the significance level. Figure 9-3 displays the graphical results of the Friedman test.



Total N	22
Test Statistic	26.530
Degrees of Freedom	3
Asymptotic Sig. (2-sided test)	.000

Figure 9-3 Results of Friedman Test for Statistical Significance of EQ NTMTs

The Friedman test concludes that Interface Target Matching Times were significantly different. Therefore we can reject the null hypothesis that "there is no difference between NTMTs for each interface". Further post-hoc analysis identified the significance of pair-wise comparisons with Bonferroni adjustments for multiple corrections (Lund Research, 2014). It was found that there was statistically significant difference in NTMTs between GIT-2 and WIT-2 ($p < .0005$), GIT-T1 and WIT-T2 ($p < .0005$), as well as WIT-T1 and WIT-T2 ($p = 0.005$).

9.1.2 - Participant Preferences & Ratings

Of the 22 participants, three did not return their questionnaire and therefore have been omitted from preference and ratings analysis. Of the remaining 19 participants, 100% reported that they preferred using the gestural interface. In addition, they were asked to report reasons for their preference. This was an open question and did not limit the number of answers that could be given. The responses were categorised according to the nearest connotation. Figure 9-4 illustrates the most frequent suggestions for preference of the gestural interface.

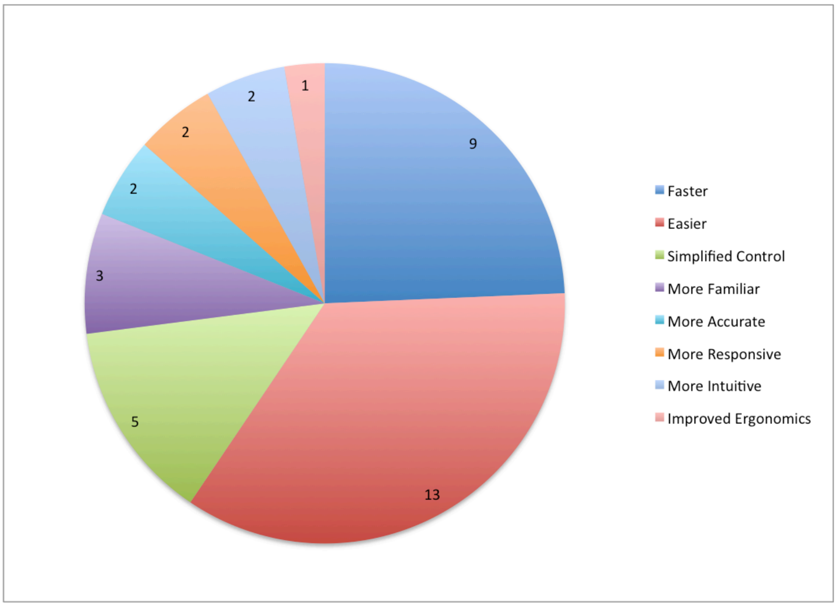


Figure 9-4 Frequency of Reasons for the Preference of Gestural Control

The modal answer was that the gestural interface was 'easier to use' than the WIMP interface.

To help assess the performance of the interface, participants were asked to give the gestural interface a score between 1 and 10 for the accuracy of control. Where 1 is not accurate and 10 is extremely accurate.

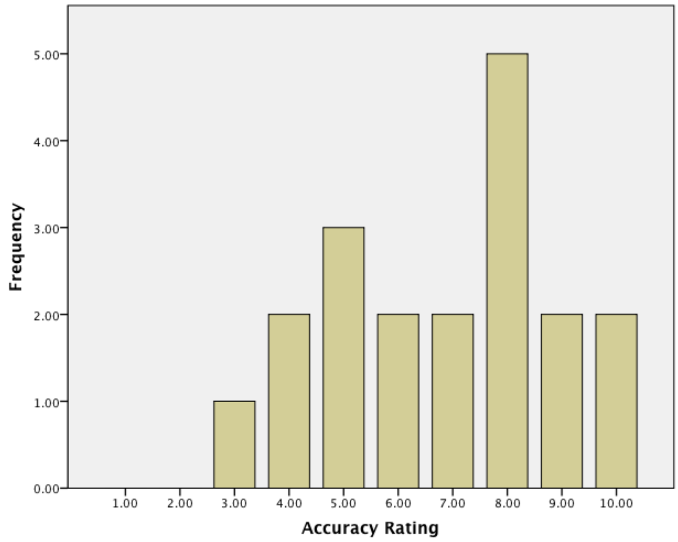


Figure 9-5 Accuracy Scores for the Gestural EQ Interface

Inspection of Figure 9-5 suggests a positive response to the accuracy rating question. With a mean score of 6.84/10.

The same was asked of Sensitivity. This time the ideal rating is 5. Where 1 is not sensitive enough, 5 is the correct amount of sensitivity and 10 is too sensitive.

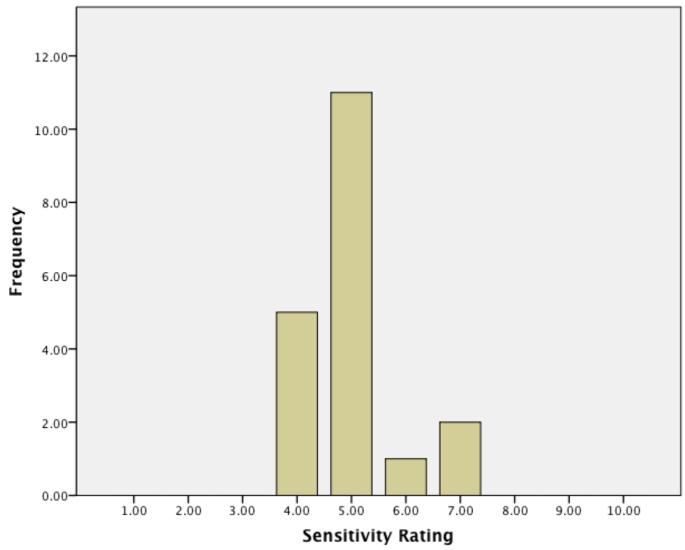


Figure 9-6 Sensitivity Scores for the Gestural EQ Interface

By inspection of figure 9-6, it can be concluded that the response from sensitivity ratings was positive, with a mean score of 4.97/10 (where 5 is ideal).

A final part of the questionnaire asked the participants if they preferred using the tablet in portrait or landscape orientation. 63% preferred portrait orientation. This figure is close to 50%, and supports the idea that a universally ergonomic tablet-based interface should be usable in both orientations.

9.2 - Dynamics Processor Control

Tables 9-2 and 9-3 present the results for gating and compression, respectively. The results have been normalised independently.

	GATING - Normalised Target Matching Time			
	Target Setting 1		Target Setting 2	
Participant #	**Gestural**	**WIMP**	**WIMP**	**Gestural**
Participant 1	1.00	0.65	0.52	0.55
Participant 2	0.95	0.64	0.94	1.00
Participant 3	0.87	0.84	0.92	1.00
Participant 4	1.00	0.59	0.66	0.93
Participant 5	0.87	0.67	1.00	0.81
Participant 6	0.99	0.76	0.79	1.00
Participant 7	0.83	0.70	0.65	1.00
Participant 8	0.88	0.51	0.68	1.00
Participant 9	0.92	0.49	0.58	1.00
MEAN	**0.92**	**0.65**	**0.75**	**0.92**
STANDARD DEV	**0.06**	**0.11**	**0.17**	**0.15**

Table 9-2 Gating NTMTs

	COMPRESSION - Target Matching Time			
	Target Setting 1		Target Setting 2	
Participant #	**Gestural**	**WIMP**	**WIMP**	**Gestural**
Participant 1	0.73	0.52	0.64	1.00
Participant 2	0.94	0.56	0.78	1.00
Participant 3	0.96	0.74	0.62	1.00
Participant 4	1.00	0.62	0.97	0.78
Participant 5	1.00	0.94	0.76	0.85
Participant 6	1.00	0.81	0.70	0.85
Participant 7	1.00	0.96	0.95	1.00
Participant 8	1.00	0.38	0.52	0.70
Participant 9	0.89	0.66	0.55	1.00
MEAN	0.95	0.69	0.72	0.91
STANDARD DEV	**0.09**	**0.20**	**0.16**	**0.12**

Table 9-3 Compression NTMTs

As can be observed in the above tables, the number of participants has reduced from the previous tests. The decision was made to abort testing after 9 participants because of the impaired performance of the gestural interface. It was clear that differences between the EQ and Dynamics controllers were causing a distinct variation in Target Matching Times. Furthermore, participants were regularly reporting difficulties during testing, including inability to perform desired gestures through issues with memorability and errors (misinterpretation of gestures by the system).

In addition to the Gestural Interface returning slower target matching times, the questionnaires revealed that only 56% of participants preferred using the tablet prototype. Although still a majority, it is markedly less that the 100% preference reported in the EQ tests.

9.2.1 - Adjustments to the Prototype

A recurring comment during gate control testing was that the gestures for attack and release were uncomfortable to perform because of the 180º rotational range. For this reason the minimum and maximum range was reduced so that:

- At 25º, Attack Time = Slowest.

- At 90º, Attack Time = Fastest.
- At 155º, Release Time = Slowest.
- At 90º, Release Time = Fastest.

The resulting range of rotation was 130º, as illustrated in Figure 9-7.

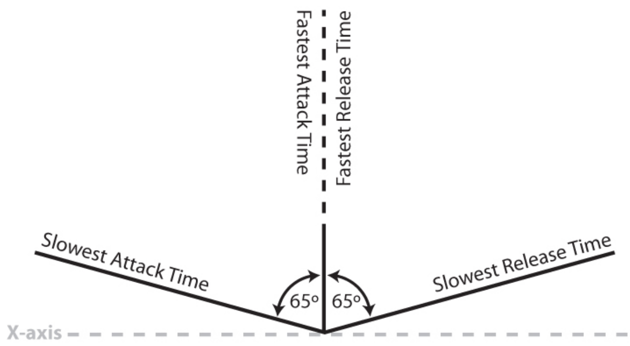


Figure 9-7 Revised Rotational Range of Attack and Release Gesture Mapping

9.3 - Influence of a GUI in the Dynamics Processor Interface

The results for gating and compression will be presented and analysed separately in order to limit variance between plug-in familiarity. For example, one user might be more comfortable using a compressor than a gate because of their mixing-style or experience level. All results have been normalised, as in previous analysis. Test results will hereby be referred to as Normalised Reference Matching Times (NRMT) for each given GUI method.

9.3.1 - Reference Matching Times - Gating

Table 9-4 presents the results for each visualization method and the WIMP control.

	GATING - Normalised Reference Matching Time			
Participant #	**WIMP**	**Gestural - Plug-In GUI**	**Gestural - Visualisation**	**Gestural - No Visualisation**
Participant 1	0.75	0.76	1.00	0.54
Participant 2	0.90	0.47	0.18	1.00
Participant 3	0.43	0.70	0.32	1.00
Participant 4	0.42	0.99	0.50	1.00
Participant 5	1.00	0.92	0.47	0.59
Participant 6	1.00	0.49	0.86	0.81
Participant 7	0.21	0.28	0.20	1.00
Participant 8	0.11	0.22	0.26	1.00
Participant 9	0.60	0.28	0.38	1.00
Participant 10	0.84	1.00	0.53	0.72
Participant 11	0.83	0.66	0.45	1.00
Participant 12	0.25	0.22	0.11	1.00
Participant 13	1.00	0.79	0.72	0.66
Participant 14	1.00	0.41	0.41	0.36
Participant 15	0.94	0.55	0.48	1.00
Participant 16	1.00	0.26	0.27	0.13
Participant 17	0.95	0.69	0.91	1.00
Participant 18	1.00	0.55	0.49	0.73
Participant 19	1.00	0.89	0.79	0.64
Participant 20	1.00	0.23	0.23	0.45
Participant 21	0.46	1.00	0.68	0.43
Participant 22	0.98	1.00	0.50	0.92
MEAN	**0.76**	**0.61**	**0.49**	**0.77**
STANDARD DEV	**0.30**	**0.29**	**0.25**	**0.26**

Table 9-4 Gating Visualisation NRMTs

Calculating z-values for Skewness and Kurtosis assessed the normality of distribution. It was concluded that all sets of data had normal distribution, with skewness and kurtosis z-values between ±2.58 (which accepts results with a statistical significance level of .01). Table 9-5 Displays the calculated z-values for Normalised Reference Matching Times (NRMT).

	GATING Z-Values			
	WIMP	Gestural - Plug-In GUI	Gestural - Visualisation	Gestural - No Visualisation
Z-Score (Skewness)	-2.02	0.037	1.071	-1.778
Z-Score (Kurtosis)	-0.528	-1.51	-0.537	-0.179

Table 9-5 Gating NRMT Z-values

Because the data is reported to have normal distribution, a one-way repeated measures ANOVA test was conducted to determine statistical significance (Lund Research, 2014). Further evidence of normality can be seen by inspection of the boxplots in figure 9-8 which confirm the absence of outliers.

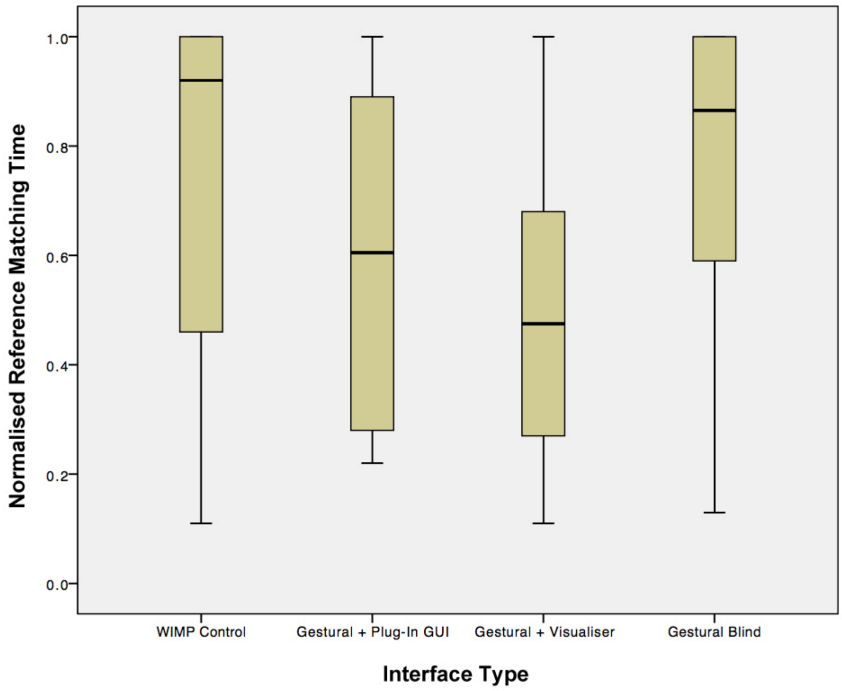


Figure 9-8 Boxplot to Confirm the Absence of Outliers for Gating NRMTs

Mauchly's test of sphericity reported that the assumption of sphericity has not been violated, with a value of $X^2(2) = 10.433$, p=0.064. Therefore, a repeated measures ANOVA will return a reliable result.

The repeated measures ANOVA test concluded that Normalised Reference Matching Times (NRMTs) have statistically significant differences, with a value of F(3, 63) = 5.752 p<0.05.

9.3.2 - Reference Matching Times - Compression

Table 9-6 presents the normalised reference matching times for each visualisation method.

	COMPRESSION - Normalised Reference Matching Time			
Participant #	**WIMP**	**Gestural - Plug-In GUI**	**Gestural - Visualisation**	**Gestural - No Visualisation**
Participant 1	0.51	1.00	0.51	0.73
Participant 2	1.00	0.66	0.67	0.36
Participant 3	1.00	0.60	0.68	0.51
Participant 4	0.72	0.98	0.71	1.00
Participant 5	0.50	1.00	0.93	0.84
Participant 6	0.86	0.57	0.51	1.00
Participant 7	0.60	0.58	0.47	1.00
Participant 8	1.00	0.44	0.48	0.59
Participant 9	0.48	0.36	0.94	1.00
Participant 10	1.00	0.77	0.61	0.33
Participant 11	0.73	0.62	1.00	0.84
Participant 12	0.70	1.00	0.47	0.62
Participant 13	0.67	1.00	0.37	0.58
Participant 14	1.00	0.77	0.66	0.78
Participant 15	0.79	1.00	0.57	0.95
Participant 16	1.00	0.67	0.56	0.35
Participant 17	1.00	0.78	0.65	0.93
Participant 18	1.00	0.62	0.38	0.53
Participant 19	0.63	0.59	0.82	1.00
Participant 20	0.40	0.47	1.00	0.53
Participant 21	0.38	0.35	0.33	1.00
Participant 22	0.72	0.24	0.68	1.00
MEAN	**0.76**	**0.69**	**0.64**	**0.75**
STANDARD DEV	**0.22**	**0.24**	**0.20**	**0.24**

Table 9-6 Compression Visualisation NRMTs

Normality of distribution was calculated through analysis of skewness and kurtosis in the same way as the gating NRMTs. Table 9-7 displays the resulting z-values.

	COMPRESSION Z-Values			
	WIMP	Gestural - Plug-In GUI	Gestural - Visualisation	Gestural - No Visualisation
Z-Score (Skewness)	-0.487	-0.0428	0.0009	-0.845
Z-Score (Kurtosis)	-1.373	-0.988	-0.626	-1.386

Table 9-7 Compression Visualisation NRMT Z-values

The boxplot in Figure 9-9 illustrates the distribution of data and confirms the absence of outliers.

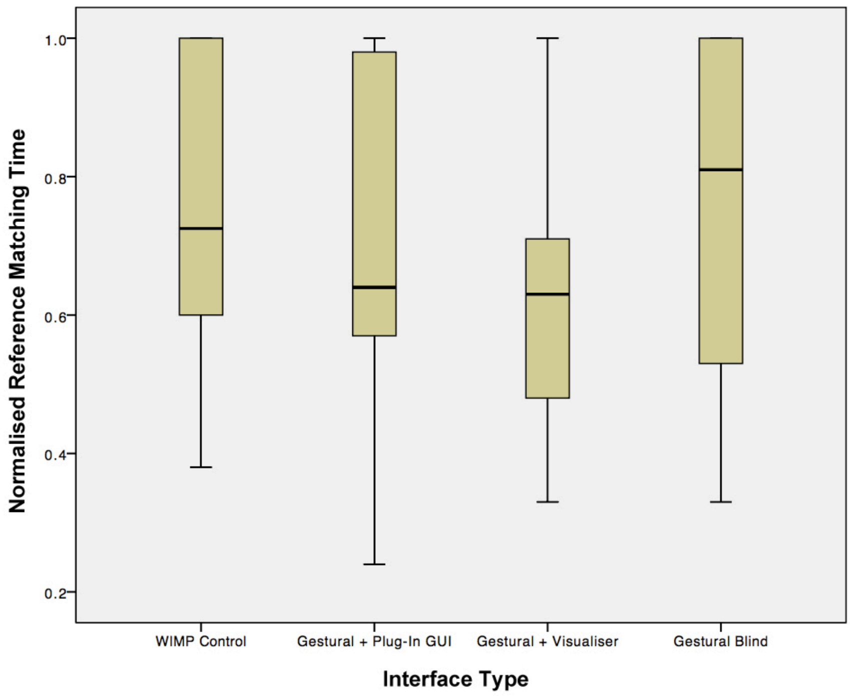


Figure 9-9 Boxplots to Confirm the absence of Outliers for Compression NRMTs

As with gating results, the compression data has returned z-values that imply relatively normal distribution. Therefore a repeated ANOVA is a suitable test to determine the statistical significance of the data.

Mauchly's test of sphericity reported that the assumption of sphericity has not been violated, with a value of $X^2(2) = 3.744$, p=0.587. Therefore, a repeated measures ANOVA will return a reliable result.

P values were returned > 0.05, therefore statistical significance CANNOT be deduced. This is a result of the mean values being closer. However, it can be suggested (through inspection of the box-plots) that further improvements to the interface would reach the same conclusions for compression as were discovered for gating.

9.3.3 - Accuracy Measurements

As compression NRMTs lacked statistical significance, the accuracies of user-settings were calculated.

The accuracy of participant's gate and comp setting were assessed using the xCorr cross-correlation function in MATLAB to compare the reference and the user's setting (Mathworks, 2014). This returned a value between 0 and 1 that represented the similarity between two waveforms (where 1 is identical and 0 is no similarities). Tables 9-8 and 9-9 display the results for compression and gating, respectively.

	Compression - Reference Matching ACCURACY			
Participant #	**WIMP**	**Gestural - TRAD GUI**	**Gestural - Visualisation**	**Gestural - Blind**
Participant 1	0.9992	0.9948	0.9638	0.9844
Participant 2	0.9906	0.9975	0.9770	0.9977
Participant 3	0.9895	0.9961	0.9956	0.9854
Participant 4	0.9879	0.9765	0.9986	0.9950
Participant 5	0.9700	0.9760	0.9529	0.9645
Participant 6	0.9433	0.9905	0.9926	0.9734
Participant 7	0.9971	0.9962	0.9956	0.9839
Participant 8	0.9819	0.9649	0.9473	0.9792
Participant 9	0.9711	0.9711	0.9882	0.9588
Participant 10	0.9912	0.9934	0.9944	0.9967
Participant 11	0.9738	0.9608	0.9737	0.9415
Participant 12	0.9892	0.9877	0.9531	0.9779
Participant 13	0.9839	0.9814	0.9930	0.9634
Participant 14	0.9928	0.9966	0.9965	0.9953
Participant 15	0.9863	0.9776	0.9935	0.9673
Participant 16	0.9685	0.9668	0.9695	0.9541
Participant 17	0.9953	0.9993	0.9969	0.9408
Participant 18	0.9722	0.9762	0.9698	0.9399
Participant 19	0.9944	0.9985	0.9974	0.9961
Participant 20	0.9859	0.9752	0.9633	0.9970
Participant 21	0.9892	0.9810	0.9672	0.9435
Participant 22	0.9898	0.9936	0.9959	0.9944
MEAN	**0.9838**	**0.9842**	**0.9807**	**0.9741**
STANDARD DEV	**0.0128**	**0.0122**	**0.0172**	**0.0206**

Table 9-8 Compression Accuracy Results

	GATING - Reference Matching ACCURACY			
Participant #	**WIMP**	**Gestural - TRAD GUI**	**Gestural - Visualisation**	**Gestural - Blind**
Participant 1	0.9984	0.9983	0.9987	0.9988
Participant 2	0.9632	0.9746	0.9993	0.9979
Participant 3	0.9975	0.9973	0.9980	0.9965
Participant 4	0.9613	0.9957	0.9729	0.9996
Participant 5	0.9874	0.9558	0.9867	0.9974
Participant 6	0.9988	0.9907	0.9058	0.9989
Participant 7	0.9994	0.9997	0.9989	0.9984
~~Participant 8~~	~~0.9974~~	~~0.9796~~	~~0.9961~~	0.2028
Participant 9	0.9987	0.9840	0.9822	0.9237
Participant 10	0.9961	0.9996	0.9998	0.9992
Participant 11	0.9729	0.9848	0.8955	0.9944
Participant 12	0.9925	0.9489	0.9461	0.9979
Participant 13	0.9918	0.9602	0.8168	0.8841
Participant 14	0.9987	0.9902	0.9650	0.9964
Participant 15	0.9922	0.9872	0.9992	0.9945
Participant 16	0.9771	0.9275	0.9248	0.9103
Participant 17	0.9756	0.9870	0.8935	0.9983
Participant 18	0.9956	0.9996	0.9960	0.9938
Participant 19	0.9845	0.9995	0.9975	0.9973
Participant 20	0.9819	0.9991	0.9973	0.9966
Participant 21	0.9576	0.8761	0.9821	0.9863
Participant 22	0.9970	0.9990	0.9961	0.9967
MEAN	**0.9866**	**0.9788**	**0.9644**	**0.9837**
STANDARD DEV	**0.0136**	**0.0308**	**0.0490**	**0.1696**

Table 9-9 Gating Accuracy Results

Note that participant 8 has been removed from the gating accuracy results because of a significant outlier for the blind test. After checking the original audio samples, it was concluded that the outlier was produced due to a gate setting with a high threshold and extremely fast attack and release times. The resulting waveform was very dissimilar to the reference and caused the outlier when the cross-correlation algorithm was run. The participant in question was a professional live-sound engineer and may have been gating with a 'trigger source' in mind, rather than matching the reference. The averages and standard deviations in table 9-9 have been calculated with participant 8 omitted. Participant 8 will be removed from any further gating analysis.

The closeness of the normalised results makes it difficult to draw any conclusions by inspection of tables. The boxplots in figure 9-10 and 9-11 provide a better representation of the data. The averages in the boxplots are Median values.

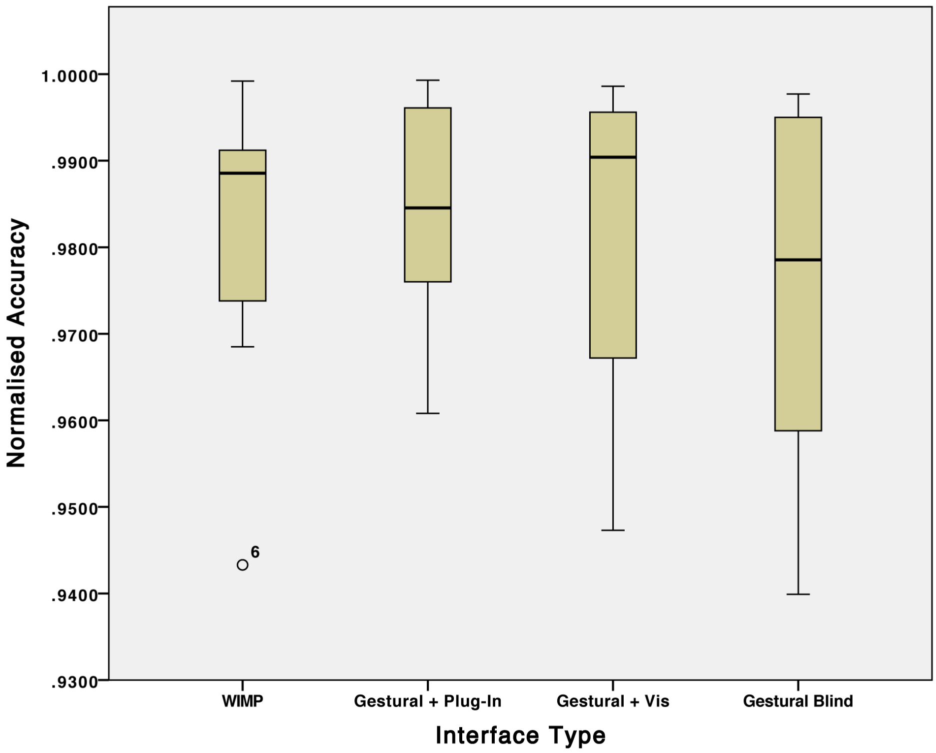


Figure 9-10 Compression Accuracy Boxplots

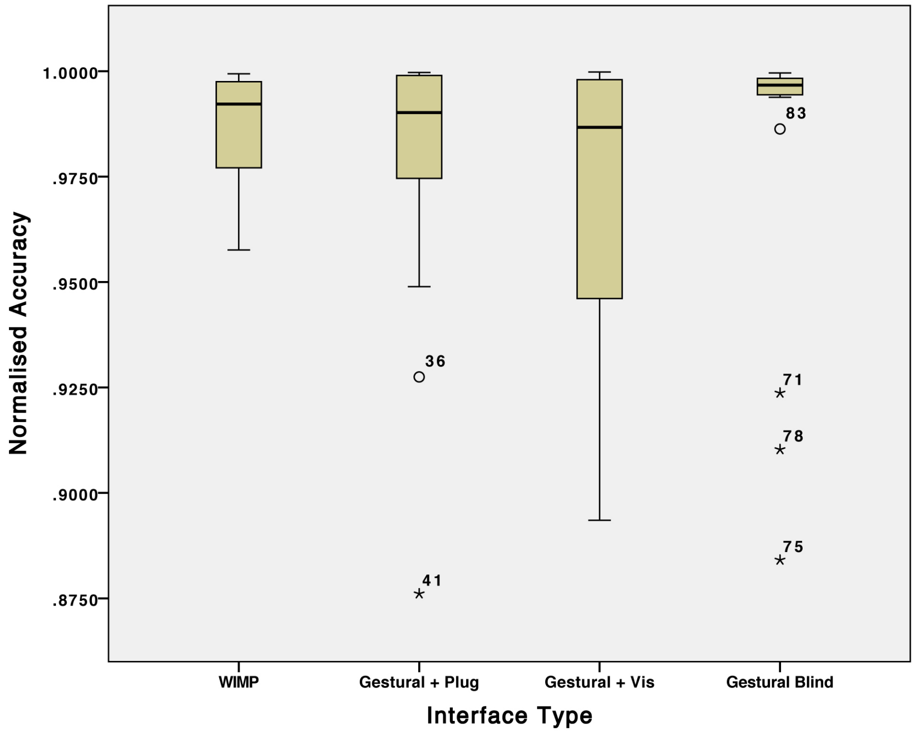
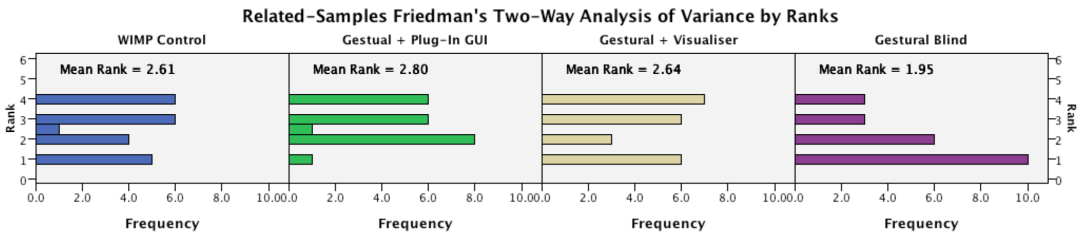


Figure 9-11 Gating Accuracy Boxplots

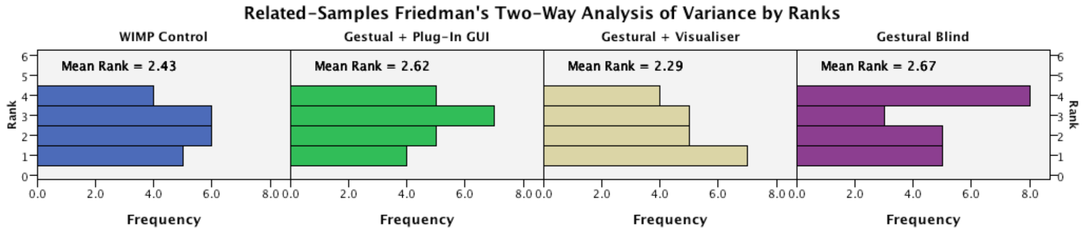
Neither of the data sets exhibit normal distribution (as assessed by the Shapiro-Wilk test), therefore the results were tested for statistical significance using the Friedman test.



Total N	22
Test Statistic	5.521
Degrees of Freedom	3
Asymptotic Sig. (2-sided test)	.137

1. Multiple comparisons are not performed because the overall test retained the null hypothesis of no differences.

Figure 9-12 Friedman Test Results for Compression Setting Accuracy



Total N	21
Test Statistic	1.171
Degrees of Freedom	3
Asymptotic Sig. (2-sided test)	.760

1. Multiple comparisons are not performed because the overall test retained the null hypothesis of no differences.

Figure 9-13 Friedman Test Results for Gating Setting Accuracy

Friedman tests results, as shown in figures 9-12 and 9-13, concluded that the accuracy measurements did not deviate far enough from the mean to be considered statistically significant ($p > .05$). Therefore we can conclude that the type of visualisation did not impair (or improve) the accuracy of a user's setting.

9.3.4 - Preferences and Questionnaire Answers

Participants were asked to choose whether they preferred using the gestural interface with or without visualisation when making fast or accurate parameter adjustments. Figure 9-14 displays the frequency of responses.

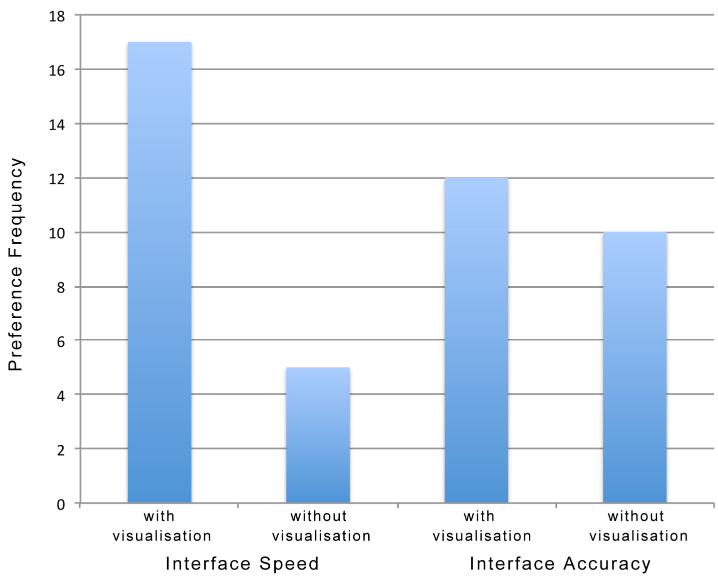


Figure 9-14 Visualisation Preference Frequencies for Interface Speed and Accuracy

A recurring comment from the preference questionnaire was that the 'Blind' mixing test proved difficult when making small adjustments to a setting. This reiterates the importance of visualisation as a guide whilst mixing. The most commonly suggested reason for interface preference *with visualisation* concerned small adjustments to the dynamics processors. Of the 29 instances where visualisation was preferred (for both speed and accuracy of the interface) 45% reported some kind of difficulty making small adjustments or problems executing small gestural movements without visual indication that the correct gesture had been (or was being) performed. The latter may be a result of user familiarity and lack of confidence using the new interface.

Participants were asked whether they preferred making attack and release adjustments with the gate or the compressor. Where the gate used a representative 'angle' setting and the compressor used arbitrary multi-touch swipes. 55% reported a preference for the compressor method; therefore neither method was preferred significantly by this participant population.

The final section of the questionnaire asked participants to describe the 'envelope shape' pictured in figure 9-15.

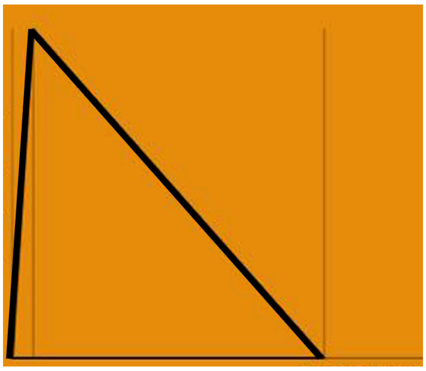


Figure 9-15 Questionnaire Envelope Shape

Participants were asked whether they considered the attack and release to be fast or slow, and to suggest a value in milliseconds that it might represent. The purpose of this question was to evaluate whether the visualisation is representative of a user's perception of envelope time.

The actual settings of the envelope represented in Figure 9-15 are 13ms (fast) attack and 1400ms (slow) release.

100% of participants correctly described the attack as ‘fast’ and 95% described the release as ‘slow’. With such a strong majority, we can conclude that the visualisation is correctly representative of user’s perception of ‘fast’ and ‘slow’ envelope times.

9.4 - Semantically Motivated Combination of Compression Controls

Firstly, the NRMT (normalised reference matching times) for each compression setting will be presented, giving an impression of which reference was easiest to match. Following this the accuracy will be measured, proving whether the ‘continuous preset’ could sufficiently match the accuracy of settings made with traditional multi-parameter control.

Following this, the times and accuracies for ‘reference 1’ will be compared to those from the previous compression tests. This will demonstrate the speed and efficiency of the ‘continuous preset’ interface against the other four interface types:

1. WIMP
2. Gestural With Plug-in GUI
3. Gestural With Representative GUI
4. Gestural Blind

Only 20 of the 22 original test participants could make this round of testing. Therefore, Participant 11 and Participant 16 have been omitted from the comparisons in section 9.4.2. Averages and normalisations were recalculated accordingly.

9.4.1 - Comparison of Reference Matching Times

Table 9-10 displays the NRMTs for each of the three references, all matched using the semantically rationalised high-level gestural controller.

	(N) Reference Matching Time		
Participant #	**Ref 1**	**Ref 2**	**Ref 3**
Participant 1	1.00	0.99	0.70
Participant 2	0.35	0.53	1.00
Participant 3	0.18	0.73	1.00
Participant 4	0.34	1.00	0.79
Participant 5	1.00	0.54	0.92
Participant 6	0.86	0.85	1.00
Participant 7	0.41	0.39	1.00
Participant 8	0.77	1.00	0.72
Participant 9	0.88	0.49	1.00
Participant 10	0.63	1.00	0.60
Participant 12	1.00	0.77	0.48
Participant 13	1.00	0.75	0.86
Participant 14	0.68	0.64	1.00
Participant 15	0.33	1.00	0.39
Participant 17	0.73	1.00	0.55
Participant 18	0.30	0.34	1.00
Participant 19	0.63	1.00	0.60
Participant 20	0.47	0.43	1.00
Participant 21	0.39	0.88	1.00
Participant 22	0.69	0.95	1.00
MEAN	**0.63**	**0.76**	**0.83**
STANDARD DEV	**0.27**	**0.24**	**0.21**

Table 9-10 - NRMTs for 'Continuous Preset'

The datasets were tested for normal distribution. Both reference 1 and reference 2 displayed uneven distribution (as assessed with the Shapiro-Wilk Test). Therefore, the non-parametric Friedman test was used to test for statistical significance. The Friedman test reported that there was no statistically significant difference between the NRMTs with a value of p = .522. Therefore, on average, each reference was matched within a statistically similar time.

9.4.2 - Comparison of Interfacing Methods

Table 9-11 presents the normalised reference matching times (NRMTs) for all interfacing methods, where participants, process and reference sample used are consistent.

	COMPRESSION - Reference Matching Time				
Participant #	**WIMP**	**Gestural - TRAD GUI**	**Gestural - Visualisation**	**Gestural - No Visualisation**	**SAFE 'Continuous Preset'**
Participant 1	0.51	1.00	0.51	0.73	0.67
Participant 2	1.00	0.66	0.67	0.36	0.05
Participant 3	1.00	0.60	0.68	0.51	0.08
Participant 4	0.72	0.98	0.71	1.00	0.14
Participant 5	0.50	1.00	0.93	0.84	0.32
Participant 6	0.86	0.57	0.51	1.00	0.22
Participant 7	0.60	0.58	0.47	1.00	0.08
Participant 8	1.00	0.44	0.48	0.59	0.47
Participant 9	0.48	0.36	0.94	1.00	0.29
Participant 10	1.00	0.77	0.61	0.33	0.16
Participant 12	0.70	1.00	0.47	0.62	0.24
Participant 13	0.67	1.00	0.37	0.58	0.77
Participant 14	1.00	0.77	0.66	0.78	0.66
Participant 15	0.79	1.00	0.57	0.95	0.19
Participant 17	1.00	0.78	0.65	0.93	0.25
Participant 18	1.00	0.62	0.38	0.53	0.42
Participant 19	0.63	0.59	0.82	1.00	0.32
Participant 20	0.40	0.47	1.00	0.53	0.19
Participant 21	0.38	0.35	0.33	1.00	0.10
Participant 22	0.72	0.24	0.68	1.00	0.22
MEAN	**0.75**	**0.69**	**0.62**	**0.76**	**0.29**
STANDARD DEV	**0.22**	**0.25**	**0.19**	**0.24**	**0.21**

Table 9-11 - NRMTs of all interfacing methods

By inspection of the average NMRTs in Table 9-11 it can be observed that the SAFE 'continuous preset' allowed participants to reach a satisfactory compression setting in less than half the time of any other interfacing method. Figure 9-16 illustrates this data in a bar graph.

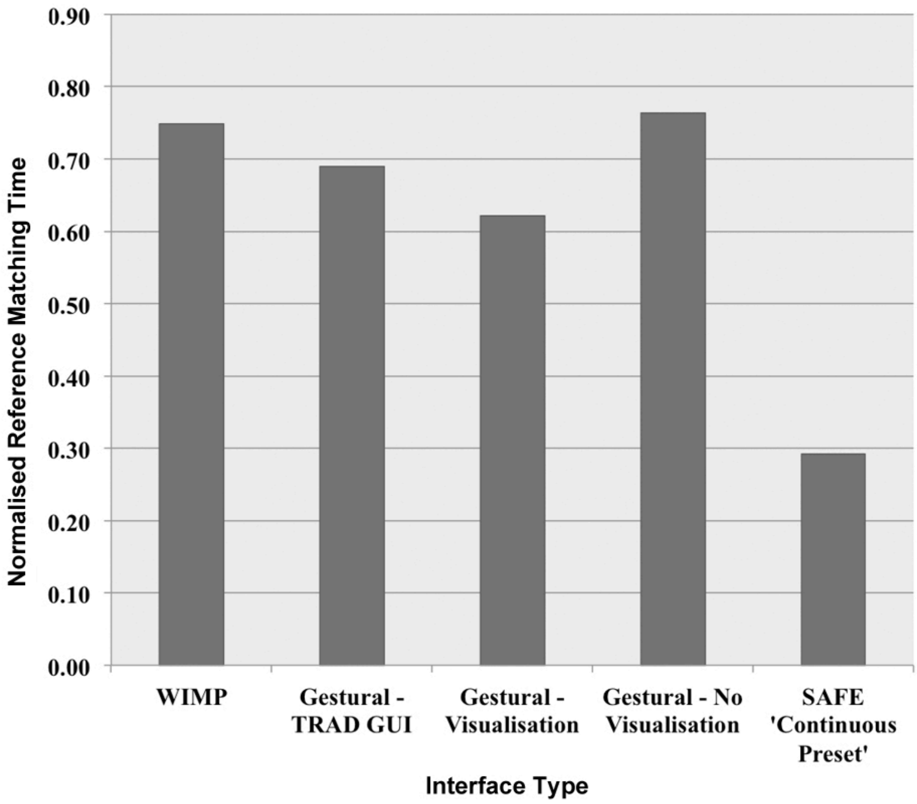


Figure 9-16 Bar Graph Illustration of NRMTs for Every Compressor Interface Method

What remains to be determined is whether the participant's settings were suitably accurate with the novel interface, as speed alone does not indicate a successful interfacing method. Table 9-12 Displays the accuracy measurements (taken with the same cross-correlation methodology as the results in Section 9.3.3) of all interfaces.

	Compression - Reference Matching ACCURACY				
Participant #	**WIMP**	**Gestural - TRAD GUI**	**Gestural - Visualisation**	**Gestural - No Visualisation**	**SAFE 'continuous preset'**
Participant 1	0.9992	0.9948	0.9638	0.9844	0.9994
Participant 2	0.9906	0.9975	0.9770	0.9977	0.9995
Participant 3	0.9895	0.9961	0.9956	0.9854	0.9422
Participant 4	0.9879	0.9765	0.9986	0.9950	0.9953
Participant 5	0.9700	0.9760	0.9529	0.9645	0.9858
Participant 6	0.9433	0.9905	0.9926	0.9734	0.9485
Participant 7	0.9971	0.9962	0.9956	0.9839	0.9986
Participant 8	0.9819	0.9649	0.9473	0.9792	0.9886
Participant 9	0.9711	0.9711	0.9882	0.9588	0.9955
Participant 10	0.9912	0.9934	0.9944	0.9967	0.9951
Participant 12	0.9892	0.9877	0.9531	0.9779	0.9964
Participant 13	0.9839	0.9814	0.9930	0.9634	0.9855
Participant 14	0.9928	0.9966	0.9965	0.9953	0.9911
Participant 15	0.9863	0.9776	0.9935	0.9673	0.9991
Participant 17	0.9953	0.9993	0.9969	0.9408	0.9946
Participant 18	0.9722	0.9762	0.9698	0.9399	0.9908
Participant 19	0.9944	0.9985	0.9974	0.9961	0.9991
Participant 20	0.9859	0.9752	0.9633	0.9970	0.9982
Participant 21	0.9892	0.9810	0.9672	0.9435	0.9422
Participant 22	0.9898	0.9936	0.9959	0.9944	0.9964
MEAN	**0.9850**	**0.9862**	**0.9816**	**0.9767**	**0.9871**
STANDARD DEV	**0.0128**	**0.0107**	**0.0178**	**0.0196**	**0.0190**

Table 9-12 Accuracy of All Interfacing Methods

Again, the datasets are unevenly distributed. The Friedman Test concluded that there is statistically significant difference between the Interface accuracy values, $X^2 (4) = 11.980$, $p < .05$.

The boxplots in Figure 9-17 illustrate the improved accuracy results of the SAFE interface.

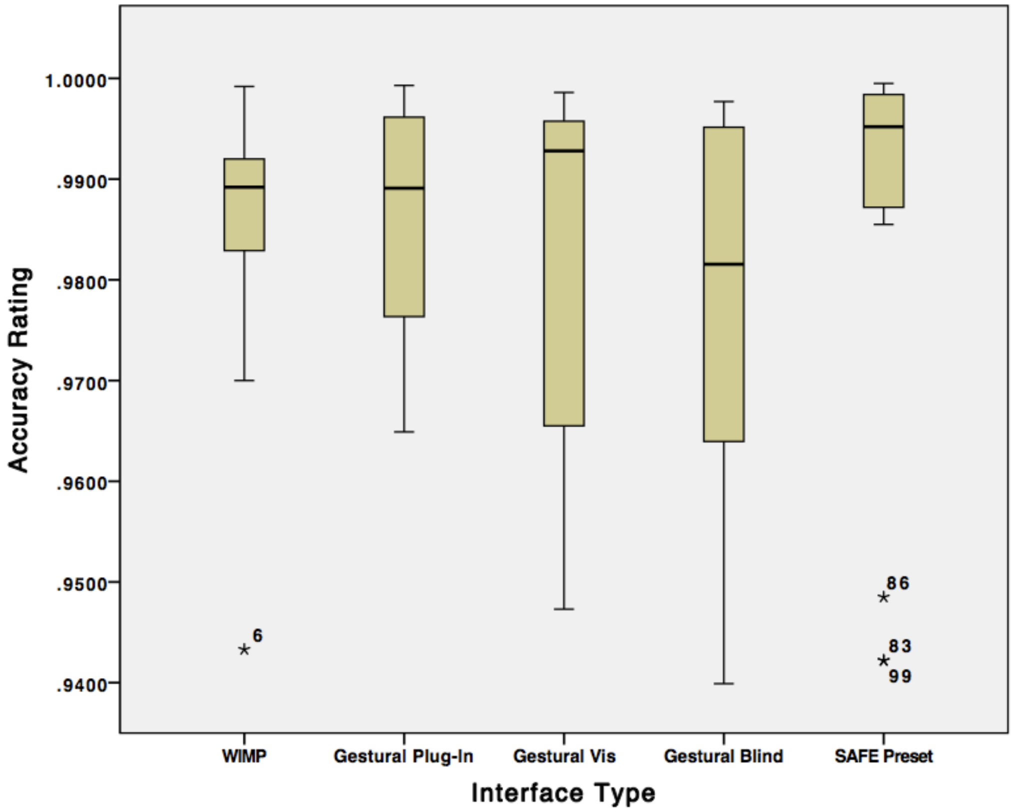


Figure 9-17 Accuracy Ratings for Each Interface Type - Boxplot

Therefore, one can conclude that the SAFE continuous preset allowed participants to make more accurate settings in a shorter amount of time.

10 - Discussions and Conclusions

10.1 - Evaluation of the Proposed Gesture Set

The intrinsic lack of mimetic gestures, representative of low-level control parameters, limits the overall usability of the system as a ‘blind mixing tool’. Elicitation of a gesture set through *user defined gestures* proved the selection layer to be the most easily defined:

- Select Compression - Inward Pinch, thus ‘squashing’ the audio.
- Select Gating - Shape of Amplitude Envelope for a ‘closing gate’.
- Select EQ - Shape Representative of an EQ response (freq vs time).

Conversely, continuous controllers are more complex and resulted in arbitrary semaphoric gesture allocations. For example, only the following system gestures can be described as mimetic, three of which were altered by the author to produce a more representative action:

- Increase Q Factor - Inward Pinch, thus 'tightening' the bandwidth of the filter.
- Increase Ratio - Inward Pinch, thus 'squashing' the audio.
- Increase Attack Time (gate) - Draw an angle representative of Amplitude Envelope.
- Increase Release Time (gate) - Draw an angle representative of Amplitude Envelope.

With the remaining allocated thusly:

- Threshold - Single Touch movement in Y-axis.
- Make-Up Gain - Three-Touch movement in Y-axis.
- Attack Time (comp) - Three-Touch movement in the X-axis (made in top half of touch screen)
- Release Time (comp) - Three-Touch movement in the X-axis (made in bottom half of touch screen)
- Increase Gain Reduction - Three-Touch movement in the Y-axis.

EQ Control can be described as Deictic (pointing) where the user points towards a desired place on an x-y axis:

- Increase Centre (or cut-off frequency) - Point in the direction of increased x-value.
- Increase Gain - Point in the direction of increased y-value.

Preference testing reported that 55% of participants favoured the arbitrary gestural allocation for control of attack and release (as found in the compressor). This could be down to ergonomic factors, as identified in Chapter 9.2.1 where the rotational range of the envelope control settings had to be changed.

It was concluded that the lack of representative gestures for low-level controls was detrimental to the learnability and memorability of the gestural audio interface. Fundamentally this could be attributed to the abstract concept of directly associating a hand movement with small changes in specific sonic characteristics (such as compressor release time). A representative GUI is required to assist the engineer with these, more complex, controls.

10.2 - Gestural Interfaces for the Control of EQ

Tests revealed that the gestural interface was, on average, faster at matching both target settings. When comparing all four tests (both target settings with both interfaces), it can be observed that the slowest average target matching time was with the *WIMP* interface, while was the quickest was with the *Gestural Interface.* This could be caused by three factors present in the gestural interface:

1. Combination of controls - Gain, frequency and Q-factor could be controlled simultaneously by the gestural system by moving a two-touch pinch in x and y directions. This allowed users to make multiple adjustments without stopping to perform a different gesture. It was still necessary to include gestures that were mapped exclusively to single parameters for the sake of accuracy. In particular when resonant frequencies had to be notched (as identified in the engineer workflow observations).
2. No requirement for locational selection (movement between parameters) - It was reported from the preliminary tests that users spent a large amount of time navigating between parameters. The gestural interface does not encounter this problem, an intrinsic benefit is that the user does not need to look at the screen to make control changes.
3. Intuitive and Familiar relationship with visualisation - The x-y, frequency verses gain, representation of EQ is something that is familiar to the majority of engineers because of its widespread use in plug-ins, visualisers and spectrograms. Therefore, it is intuitive for an engineer to associate changes to the frequency content of a signal with movements on a 2D surface.

The significantly improved performance of the gestural interface for the control of EQ set the benchmark for other audio processing tasks. It should be the case that adhering to points 1, 2 and 3 when designing a gestural audio interface will produce a system that improves usability and workflow over a WIMP interface. Where familiarity is not already in place (point 3), a suitably representative GUI should be presented. This was found to be the case with the gestural control of dynamics processing.

10.3 - Gestural Interfaces for the Control of Dynamics Processing

Initial tests of Gestural Dynamics control, where the standard plug-in GUI was used, showed no benefits over WIMP interaction. Therefore, changes to the interface were made and tested to try and match the performance of the EQ gestural control.

10.3.1 - Suitability of Visualisation

As identified in EQ gesture testing, the familiarity and intuitiveness of the visualisation helped to optimise the interface. Standard plug-in GUIs have been identified as unsuitable for gestural control because of their unrepresentative, skeuomorphic designs.

Testing of novel visualisation, based on amplitude envelope designs, showed that a gating reference could be matched 35% quicker with a gestural interface and a representative GUI than a WIMP interface with traditional visualisation methods. The average times for compression appeared to show the same trends, however, the variances were too small to be considered statistically significant.

Blind mixing tests proved that users could maintain the same level of accuracy when using the gestural interface *without* any visualisation. It can therefore be concluded that the purpose of visualisation is to improve the speed of settings through familiarity with a representative GUI. The fact that users were unable to view parameter values also had no effect on accuracy.

It is important to note that the gestural interface still facilitates blind mixing (as specified at the start of development), however visualisation is required as a reference point and helps provide more feedback to the user. The benefit of a Gestural Interface over a WIMP equivalent is that the visualisation is not a necessity.

10.3.2 - Combination of Controls into 'Continuous Presets' for Compression

Low-level Gestural control alone, proved inadequate to optimise the control of dynamics processing. In order to try and match the performance of the gestural control of EQ, the compression interface should try to offer simultaneous control over multiple plug-ins. This could have been achieved in two ways; firstly, simultaneous parameter mappings to suitable gestures. For example, the distance between two touches is mapped to ratio, while the rotation of those touches is mapped to threshold.

Secondly, the parameters could be combined intelligently by the interface and offer the user a higher-level control. For the purposes of this research project, it was more insightful to evaluate the effectiveness of combined parameters into higher-level controls.

For the potential of a gestural system to be realised (as benchmarked by the EQ tests), semantically motivated gestural shortcuts were suggested as a means of implementing 'continuous presets'. Testing revealed that users found it easier to match a reference when the number of controls had been reduced. Results showed that participants could achieve a higher level of accuracy in a faster time by using the 'Continuous Preset' prototype to compress a kick drum sample.

Some reservations should be held with regard to the 'continuous preset' test results as the interface is directly source dependent. The kick drum sample used throughout the compression testing proved to be a suitable, transient source for the 'continuous preset'. It may be the case that the gestural interface would be less effective at processing different audio sources. However, by making improvements to the preset interpolation response (The preset elicitation process was carried out with a linear interpolation between parameters) or offering more specific semantic descriptors (such as categories for instrumentation) the gestural control of a 'continuous preset' could prove to optimise processor control in any mix scenario.

10.4 - Possible Applications

10.4.1 - Studio Controller

In its current form, the Gestural Mix Interface would be best suited as an 'auxiliary controller' in a studio. It would provide a fast and intuitive 'quick mix' interface for starting a mix. But it would require the addition of numerous navigational elements to make its operation more practical in a global DAW environment.

10.4.2 - Broadcast Engineering

The prototype was taken for a short demonstration at Calrec Audio in Holmfirth, West Yorkshire. It was hoped that the opinions and feedback from industry experts would help identify the potential applications of the interface. Henry Bourne, head design engineer, introduced a design problem that Calrec encounter when incorporating touch-screens into mixing desks for broadcast production (such as the

Summa Console). He stated that they always have to include a tactile control (rotary, fader or switch) for each on-screen parameter because broadcast engineers need to watch a live-stream while simultaneously mixing the audio. Bourne suggested that the gestural interface could provide a suitable solution, commenting that "The interface would allow the operator to achieve everything they need without ever looking away from the video".

10.4.3 - Live Sound Engineering

The fast and intuitive implementation of the 'continuous presets' would be particularly suitable for applications where speed and efficiency of processor settings are paramount, such as 'sound-checking' bands during Live Music Production.

11 - Further Work

11.1 Gestural Control in More Typical Mix Scenarios

The gestural control system remains to be tested in a more practical, subjective mix scenario. A navigational gesture-set could be added to the system which would allow its usability and efficiency to be compared to a large control surface. A 'stage metaphor' design might prove to be a suitable approach.

11.2 Enhanced Visualisation

Testing proved the importance of a representative GUI in a gestural control system. Further improvements could produce a GUI that can be directly manipulated by gestures, as with the IIR gestural reverb interface developed by Madden et al (2011). Such a system would help to close the 'action perception loop' thus providing more intuitive feedback to the engineer.

As discussed by Giannoulis et al (2013), the ideal compression processor would be intelligent and react to changes in the audio signal. The same was identified for the controller and visualisation in a gestural system. For example, monitoring of the input signal would facilitate the automation of the threshold and a time-domain representation of the amplitude. The gesture prototype was adjusted so that audio data could be streamed back to the iPad via Wi-Fi. However, latency was introduced to the gesture recognition time of such a magnitude that was deemed to compromise the performance of the overall interface. The proposed visualiser for the reactive compressor controller was based around a circle that represented the average level of an audio source. Essentially this was achieved by quickly plotting a VU meter in a clockwise motion over time. Figure 11-1 displays two of the audio source representations of this GUI that were produced autonomously by the system. It shows the resulting plots of a looped, transient audio source. One with and one without compression applied.

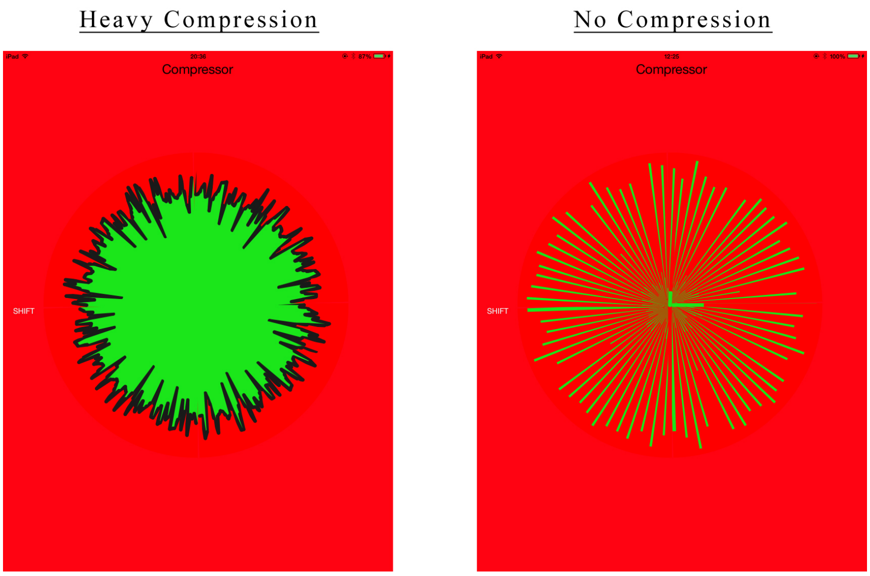


Figure 11-1 Proposed Circular Compression Visualisation

The ultimate goal was to combine control and visualisation, so that the user could 'shape' the circle with a gesture and the program would make the corresponding parameter changes (which were calculated intelligently through analysis of the audio stream). This GUI would demonstrate a marriage between visualisation and control that could greatly improve intuitiveness and effectiveness of the interface, in much the same was as was observed with EQ interaction. Further development of the audio streaming and gesture recognition algorithms would allow this technique to be tested. The disadvantages of such a visually active system might be that users are distracted from the audio (Mycroft et al, 2013). It would also prevent the gestural interface from being used to 'mix blind', an advantage that was received favourably by test participants.

11.3 Exploring the Global Benefits of Non-Locational Controls

An advantage of the Gestural Interface that was repeatedly observed during this research project was the ability to move between plug-ins and controls without the need for finding the relevant menus or windows within a DAW. Nash and Blackwell (2011) reinforce this claim with results from their extensive DAW operation observations, finding that engineers spend 24.8% of their time moving between windows and menus. If the testing of the gestural interface was opened up to more global mixing tasks, the observed engineer workflow could be significantly improved. Much like playing an instrument, the engineer would have access to all the processing controls of a DAW through memorised gestures, with the added assistance of representative GUIs.

References

Apple Inc. (2009) Touch Screen Device, Method and graphical User interface for determining commands by applying Heuristics. US Patent: us7479949

Apple Inc. (2014) Mac Developer Library: About Objective-C [online] available at: https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html [Accessed 22nd December 2014]

Apple Inc. (2014a) iPad mini [online] available at: http://www.apple.com/ipad-mini/specs/ [Accessed 22nd December 2014]

Apple Inc. (2014b) Event Handling Guide for iOS: Gesture Recognisers [online] available at: https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/GestureRecognizer_basics/GestureRecognizer_basics.html [Accessed 22nd December 2014]

Appleyard, D. (2010) 5 Apps to Enhance Your Trackpad & Magic Mouse [online] available at: http://mac.appstorm.net/roundups/utilities-roundups/5-apps-to-enhance-your-trackpad-magic-mouse/ [Accessed 14th December 2014]

Amazon, Inc. (2014) Kindle [online] available at: http://www.amazon.com/gp/product/B00I15SB16 [Accessed 12th December 2014]

Avid Technology. (2014) Pro Tools S6 [online] available at: http://www.avid.com/US/products/avid-S6 [Accessed 13th December 2014]

Balin, W. Loviscach, J. (2011) *Gestures to Operate DAW Software* In: Audio Engineering Society 130th Convention, London, UK, 13 - 16 May

Barrett, G. Omote, R. (2010) Projected-Capacitive Touch Technology [online] available at: http://large.stanford.edu/courses/2012/ph250/lee2/docs/art6.pdf [Accessed 19th November 2014]

Bates, E. (1994) *Social Interactions, Musical Arrangement and the Production of Digital Audio in Istanbul Recording Studios,* PhD Thesis, University of California, Berkeley.

BBC. (2011) Microsoft Kinect 'Fastest-Selling Device on Record' [online] available at: http://www.bbc.com/news/business-12697975 [Accessed 14th October 2014]

Bencina, R. (2013) oscpack [online] available at: http://www.rossbencina.com/code/oscpack [Accessed 29th December 2014]

Bhalla, M, D. Bhalla, A, V. (2010) Comparative Study of Various Touchscreen Technologies. In: *International Journal of Computer Applications (0975 - 8887) Volume 6 - No.8, September 2010.*

Blindmann, G. (2011) Multitouch Overview [online] available at: http://www.multi-touch-solution.com/en/knowledge-base-en/multitouch-technologies-en/ [Accessed 16th November 2014]

Bragdon, A. Zeleznik, R. Williamson, B. Miller, T. LaViola, J. J. (2009) *GestureBar: Improving the approachability of Gesture-Based Interfaces.* In: Proceedings of the Conference for Human-Computer Interaction (CHI 2009), Boston, USA, 4 - 9 April.

Bragdon, A. Nelson, E. Li, Y. Hinckley, K. (2011) *Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments* In: Proceedings of the Conference for Human-Computer Interaction (CHI 2011), Vancouver, Canada, 7 - 12 May.

Brownlee, J. (2014) Why OS X Lion's "Reverse Scrolling" is Awesome and How to Use it [online] available at: http://www.cultofmac.com/104335/os-x-lions-reverse-scrolling-is-awesome-here-why-how-to-use-it-in-snow-leopard/ [Accessed 14th December 2014]

Buxton, B. (2014a) Multi-Touch Systems that I Have Known and Loved [online] available at: http://www.billbuxton.com/multitouchOverview.html [Accessed 11th October, 2014]

Buxton, B. (2014b) Buxton Collection: CF-10 Digital Fader [online] available at: http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/detail.aspx?id=216 [Accessed 8th December 2014]

Calrec Audio, Ltd. (2014) Summa [online] available at: http://calrec.com/summa/ [Accessed 13th December 2014]

Cartwright, M. Pardo, B. Reiss, J. D. (2014) *MIXPLORATION: Rethinking the Audio Mixer Interface.* In: Proceedings of The International Conference on Intelligent User Interfaces (IUI 2014). Haifa, Israel, 24 - 27 February.

Clover, J. (2013) iPhone 5 Touch Screen as Fast as Android Touch Screen [online] available at: http://www.macrumors.com/2013/09/21/iphone-5-touch-screen-twice-as-fast-as-android-touch-screen/ [Accessed 23rd December 2014]

Carrascal, J, P. Jordá, S. (2011) *Multitouch Interface for Audio Mixing.* In: Proceedings of New Interfaces for Musical Expression (NIME 2011). Oslo, Norway.

Cochrane, G. (2013) The Case For Mixing With Your Eyes Closed [online] available at: http://therecordingrevolution.com/2013/06/28/the-case-for-mixing-with-your-eyes-closed/ [Accessed 5th November 2014]

Couturier, J, M. (2005) Gestural Control of Audio Systems and Digital Musical Instruments [online] available at: http://www.enterface.net/enterface05/docs/seminars/couturier.pdf [Accessed 19th December 2014]

Dewey, C. Wakefield, J, P. (2014) *A Guide to the Design and Evaluation of New User Interfaces for the Audio Industry* In: Audio Engineering Society 136th Convention, Berlin, Germany, 26-29th April.

Diamante, V. (2007) AWOL: Control Surfaces and Visualization for Surround Creation. University of Southern California: USA

Dilger, D, E. (2013) Agawi TouchMark contrasts iPad's fast screen response to laggy Android tablets [online] available at: http://appleinsider.com/articles/13/10/08/agawi-touchmark-contrasts-ipads-fast-screen-response-to-laggy-android-tablets [Accessed 21st December 2014]

Drossos, K. Floros, A. Koukoudis, K. (2013). *Gestural User Interface for Audio Multitrack Real-Time Stereo Mixing.* In: Proceedings of the 8th Audio Mostly Conference (AM 2013). Piteå, Sweden.

Elgan, M. (2014) Why In-Air Gestures Failed, And Why They'll Soon Win [online] available at: http://www.computerworld.com/article/2599359/emerging-technology/why-in-air-gestures-failed-and-why-theyll-soon-win.html [Accessed 4th November 2014]

Finn, G. (2010). Construction of a Resistive Touch Screen, image [online] available at: http://www.analog.com/library/analogdialogue/archives/44-02/touch_screen.html [Accessed 14th October 2014]

Ford Global Technologies. (2013) Interacting With Vehicle Controls Through Gesture Recognition. US Patent: us20130204457.

Freescale Semiconductor. (2014) CRTOUCH: Freescale Touch-Sensing Platform [online] available at: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CRTOUCH [Accessed 5th November 2014]

Fujitsu Microelectronics. (2014) Capacitive Touch Sensors: Application Fields, technology overview and implementation example [online] available at: http://www.fujitsu.com/downloads/MICRO/fme/articles/fujitsu-whitepaper-capacitive-touch-sensors.pdf [Accessed 9th November 2014]

Gelineck, S. Overholt, D. Büchert, M. Andersen, J. (2013) *Towards an Interface for Music Mixing based on Smart Tangibles and Multitouch.* In: Proceedings of New Interfaces for Musical Expression (NIME 2013), Daejeon, Korea, 27 - 30 May

Giannoulis, D. Massberg, M. Reiss, J, D. (2013) Parameter Automation in a Dynamic Range Compressor. In: *Journal of the Audio Engineering Society, vol.61, no.10, October 2013*

Gilder, J. (2014) Why Control Surfaces Aren't Worth It [online] available at: http://www.homestudiocorner.com/why-control-surfaces-arent-worth-it/ [Accessed 19th December 2014]

Gough, C. Green, M. Billinghurst, M. (2006) *Accounting for User Familiarity in User Interfaces* In: Image and Vision Computing New Zealand.

Guinness, H. (2014) Speeding Up Your Workflow With Custom Gestures. [online] available at: http://computers.tutsplus.com/tutorials/speeding-up-your-workflow-with-custom-trackpad-gestures--mac-60046 [Accessed 12th November 2014]

Gupta, A. (2014) How DriftCoast is Bringing Leap Motion Control to Heart Surgery. [online] available at: http://blog.leapmotion.com/how-driftcoast-is-bringing-leap-motion-control-to-heart-surgery/

Hammond, E, V. (2000) *Multi-Modal Mixing: Gestural Control for Musical Mixing Systems.* MSc Thesis. Massachusetts Institute of Technology: USA.

Heller, N. (2011) Five Key Ted Talks. *The New Yorker,* July 2012 [online] available at: http://www.newyorker.com/culture/culture-desk/five-key-ted-talks [Accessed 3rd January 2015]

Herman, T. Hunt, A. (2005) An Introduction to Interactive Sonification. IEEE Multimedia. pp. 20-24.

Heydekorn, J. Frisch, M. Dachselt, R. (2010) *Prospects of User Elicited Gestural Interaction Techniques.* In: Conference on Human Factors in Computing Systems (CHI 2010), Altlanta, Georgia, USA, 15 - 20 April.

Hinckley, K. (2011) Chapter 9: Input Technologies and Techniques. In: Sears, A. Jacko, J, A. *The Human-Computer Interaction Handbook.* New Jersey: Lawrence Erlbaum Associates, pp. 161-176.

Howard, D. M. Murphy, D. T. (2007) *Cymatic: a real-time tactile-controlled physical modelling musical instrument.* In: Proceedings of 19th International Congress on Acoustics (ICA 2007), Madrid, Spain, 2 - 7 September.

Lilly, P. (2014) SimCity BuildIt on iOS and Android Devices [online] available at: http://hothardware.com/news/SimCity-Buildit-Lets-You-Build-Booming-Skylines-On-iOS-And-Android-Devices [Accessed 27th October 2014]

Judah, S. (2013) What is Skeuomorphism? *BBC News Magazine,* June 2013 [online] available at: http://www.bbc.co.uk/news/magazine-22840833 [Accessed 1st November 2014]

Karjalainen, M. Maki-Patola, T. Kanerva, A. Huovilainen, A. (2006) Virtual Air Guitar, *Journal of the Audio Engineering Society*, vol. 54, no. 10, pp. 964–980

Korff, C. (2014) Controlling In The Deep. *Sound on Sound,* November 2014 [online] available at: http://www.soundonsound.com/sos/nov14/articles/spotlight-1114.htm [Accessed 3rd December 2014]

Krenn, M. (2014) A New Car UI [online] available at: http://matthaeuskrenn.com/new-car-ui/ [Accessed 6th December]

Lech, M. Kostek, B. (2013) Testing A Novel Gesture-Based Mixing Interface. In: *Journal of the Audio Engineering Society,* vol.61, no.5, 2013 May.

Line6, Inc. (2014) StageScape M20d [online] available at: http://uk.line6.com/stagescape-m20d [Accessed 12th December 2014]

Long, A, C. Landay, J, A. Rowe, L, A. Michiels, J. (2000) *Visual Similarity of Pen Gestures.* In: Proceedings of Human Factors in Computing Systems (SIGCHI 2000). The Hague, Netherlands.

Lund Research, Ltd. (2014) Testing for Normality in SPSS [online] available at: statistics.laerd.com/premium/tfn/testing-for-normality-in-spss.php [Accessed 30th November]

Lunderschmidt, J. (2014) A Gesture-Controlled Car User Interface [online] available at: http://johannesluderschmidt.de/a-gesture-controlled-car-user-interface/2913/ [Accessed 14th December 2014]

Madden, A. Blumenthal, P. Andreopoulou, A. Boren, B. Hu, S. Shi, Z. Roginska, A. (2011) *Multi-Touch Room Expansion Controller for Real-Time Acoustic Gestures*. In: Audio Engineering Society 131st Convention*,* New York, 20-23rd October.

Matthews, S. (2013) IndustryARC: Gesture Recognition in Consumer Electronics Market (2013-2018) [online] available at: http://industryarc.com/Report/81/gesture-recognition-in-consumer-electronics-market.html [Accessed 15th October 2014]

Melda Production. (2014) Equalizers Comparison [online] available at: http://www.meldaproduction.com/plugins/comparison_equalizers.php [Accessed 19th December 2014]

Michael, A. (2014) New Logic Pro X Channel EQ [online] available at: https://www.gearslutz.com/board/logic-pro-x/897556-new-logic-pro-x-channel-eq.html [Accessed 9th December 2014]

Mycroft, J. Reiss, J, D. Stockman, T. (2013) *The Influence of Graphical User Interface Design on Critical Listening Skills*. In: Proceedings of the Sound and Music Computing Conference 2013, Stockholm

Nash, C. Blackwell, A.F. (2011) *Tracking Virtuosity and Flow in Computer Music*. In: Proceedings of the International Computer Music Conference (ICMC 2011), Huddersfield, UK, 31 July - 5 August.

Narasimhan, S. (2014) Combining Self- & Mutual-Capacitive Sensing for Distinct User Advantages [online] available at: http://www.designnews.com/author.asp?doc_id=271356&dfpPParams=ind_184,aid_271356&dfpLayout=blog&dfpPParams=ind_184,aid_271356&dfpLayout=blog [Accessed 19th November 2014]

Nielson, M. Störring, M. Moeslund, T. B. Granum, E. (2003) *A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for Man-Machine Interaction,* Technical Report, Aalborg University, Denmark.

Nielson, M. Störring, M. Moeslund, T. B. Granum, E. (2008) Gesture Interfaces. In: Kortum, P. *HCI Beyond the GUI.* London: Kaufmann, pp. 75-106.

Osborne, I. (2011) How to use Multi-Touch Gestures on your Mac [online] available at: http://www.techradar.com/news/computing/apple/how-to-use-multi-touch-gestures-on-your-mac-973617 [Accessed 19th December 2014]

Park, H, P. Nieto, O. (2013) *Fortissimo: Force-Feedback for Mobile Devices.* In: Proceedings of New Interfaces for Musical Expression (NIME 2013), Daejeon, Korea.

Payne, J. Keir, P. Elgoyhen, J. McLundie, M. Naef, M. Horner, M. Anderson, P. (2006) *Gameplay Issues in the Design of 3D gestures for Video Games.* In: Proceedings of the Conference for Human-Computer Interaction (CHI 2006), Montréal, Canada, 22 - 27 April.

Porter, C. (2011) Quick Tip: Mixing With Your Eyes Closed [online] available at: http://music.tutsplus.com/tutorials/quick-tip-mixing-with-your-eyes-closed--audio-11318 [Accessed 29th December 2014]

Pratas, A. (2013) Skeuomorphic vs Digital Interfaces and What Will 2013 Bring Us [online] available from: http://designmodo.com/skeuomorphic-vs-digital-interfaces/ [Accessed 4th November 2014]

Ratcliffe, J. (2014). *MotionMix: A Gestural Audio Mixing Controller*. In: Audio Engineering Society 137th Convention, Los Angeles, USA, 9 - 12th October.

Reactable Systems. (2014) Reactable: Tech Specs [online] available at: http://www.reactable.com/products/experience/tech-specs/ [Accessed 3rd December 2014]

Robjohns, H. (2013). Slate Raven MTX. *Sound On Sound,* May 2013 [online] available at: http://www.soundonsound.com/sos/may13/articles/raven.htm [Accessed 15th November 2014]

Saffer, D. (2009) *Designing Gestural Interfaces*. Sebastopol, CA: O'Reilly Media, Inc.

Sasso, L. (2014) Remote Control. *Sound On Sound,* April 2014 [online] available at: http://www.soundonsound.com/sos/apr14/articles/live-tech-0414.htm [Accessed 15th December 2014]

Schuts, M. Lipscomb, S. (2007) Hearing Gestures, Seeing Music: Vision influences perceived tone duration. In: *Perception,* volume 36(6), pp. 888-897.

Selfridge, R. Reiss, J. (2011) *Interactive Mixing Using Wii Controller*. In: Audio Engineering Society 130th Convention, London, UK, May 13 - 16.

Shimpi, A. L. Klug, B. Gowri, V. (2011) Apple iPad 2 Preview [online] available at: http://www.anandtech.com/show/4215/apple-ipad-2-benchmarked-dualcore-cortex-a9-powervr-sgx-543mp2/2 [Accessed 14th November 2014]

Slate Media Technology. (2014) Raven MTX, Multi-Touch Production Console [online] available at: http://www.slatemt.com/products/raven-mtx/ [Accessed 15th November 2014]

Sound on Sound. (2012) 'DAW Control Surfaces - Roundup'. *Sound on Sound,* November 2014 [online] available at: http://www.soundonsound.com/sos/nov12/articles/spotlight-1112.htm [Accessed 2nd November 2014]

Stables, R. Enderby, S. Man, B. D. Fazekas, G. Reiss, J. (2014) SAFE: *A System for the Extraction and Retrieval of Semantic Audio Descriptors* In: 15th International Society for Music Information Retrieval Conference (ISMIR 2014) Taipei, Taiwan, 27 - 31 October.

Steinmetz, K. (2014) 'This Technology Could Change the Way Deaf People Live'. *Time Magazine,* October 2014 [online] available at: http://time.com/3529629/motionsavvy-deaf/ [Accessed 2nd November 2014]

Virgil, J. (2012) CarTunes: A Gesture Based Music Player [online] available at: http://iphone.appstorm.net/reviews/music/cartunes-a-gesture-based-music-player/ [Accessed 15th December 2014]

Sullivan, M. (2012) Fingerprints everywhere! Are we ready for 400 million dirty Windows 8 touchscreens? [online] available at: http://www.pcworld.com/article/2011665/fingerprints-everywhere-are-we-ready-for-4-million-dirty-windows-8-touchscreens.html [Accessed 11th December 2014]

Tibken, S. (2014) Touch-Screen Mac unlikely, says Apple's Federighi [online] available at: http://www.cnet.com/uk/news/touch-screen-mac-unlikely-says-apples-federighi/ [Accessed 5th November 2014]

Waves Audio Ltd. (2014) API 2500 [online] available at: http://www.waves.com/plugins/api-2500 [Accessed 25th December 2014]

Westerman, W. Elias, J, G. (2001) Multi-Touch: *A New Tactile 2-D Gesture Interface For Human - Computer Interaction* In: Proceedings of the Human Factors And Ergonomics Society 45th Annual Meeting, Minneapolis, Minnesota, 8 - 12th October.

White, P. (2012) 'Line 6 StageScape M20D'. *Sound on Sound,* November 2012 [online] available at: http://www.soundonsound.com/sos/nov12/articles/line-6-m20d.htm [Accessed 1st July 2015]

Whitney, L. (2014) Android beat Apple in Tablet Sales Last Year [online] available at: http://www.cnet.com/news/android-beat-apple-in-tablet-sales-last-year-gartner/ [Accessed 24th December 2014]

Wigdor, D. Benko, H. Pella, J. Lombardo, J. Williams, S. (2011) Rock and Rails: Extending Multi-touch Interactions with Shape Gestures to Enable Precise Spatial Manipulations, Microsoft Research Application Note, Washington, USA

Wildora (2014) Osculator [online] available at: http://www.osculator.net/ [Accessed 22nd December 2014]

Wrenn, E. (2012) No Time For Delay: Microsoft Demos Touchscreen that Responds to your Finger in Less than One Millisecond [online] available at: http://www.dailymail.co.uk/sciencetech/article-2113783/No-time-delay-Microsoft-demos-touchscreen-responds-finger-millisecond.html [Accessed 23rd December 2014]

Worstall, T. (2013) Apples iOS7, Well, It Was Time for Skeuomorphism To Die [online] available at: http://www.forbes.com/sites/timworstall/2013/09/19/apples-ios7-well-it-was-time-for-skeuomorphism-to-die/ [Accessed 5th November 2014]

Yap, J. (2010) More Touch Enabled Devices Expected in Future [online] available at: http://www.zdnet.com/article/more-touch-enabled-devices-expected-in-future/ [Accessed 1st December 2014]

Yamaha Corporation. (2014) QL Series [online] available at: http://www.yamahaproaudio.com/global/en/products/mixers/ql/index.jsp [Accessed 13th December 2014]

Appendix A - A Technological Overview of Touch Screen Operation

Although there are many ways to gather x-y positional data from a flat surface, historically, there have been two main types or touch screen: resistive and capacitive.

Traditional resistive screens are built from two transparent layers with a small gap in-between them. Each layer in covered with a conductive coating. The coating is generally indium tin oxide (IDO), which has a uniform (linear) resistance. Figure A-1 demonstrates the orientation of these layers and how they can be 'sandwiched' with an LCD screen to produce an interactive touch surface.

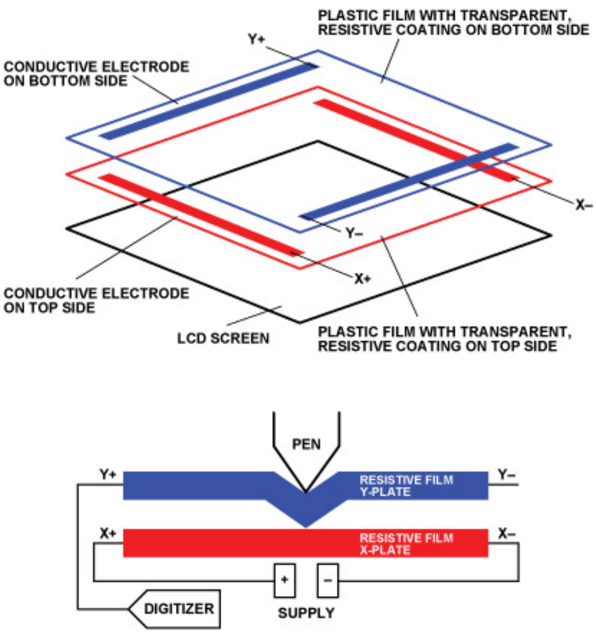


Figure A-1 Resistive Screen Operation (Finn, 2010)

When a user presses on the screen, the two layers join at the point of touch. This produces a connection between the layers that can be localised. In a resistive screen, x and y positions have to be gathered separately. Figure A-2 shows how the x and y-axis positions can be located. A voltage, often 5V, is passed between the Electrode strip (x+) and the common strip (x-). The voltage at the point of connection can then be read from the y-layer because the resistive coatings form a potential divider. This voltage can be used to find the location of the touch along the x-axis. The process has to be repeated using the opposite layers to find the y-axis position.

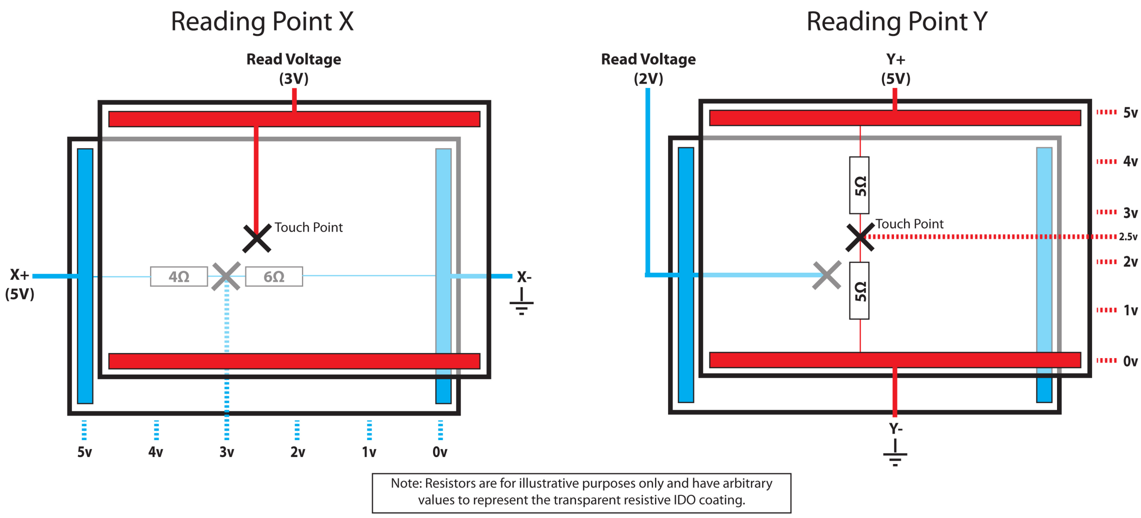


Figure A-2 Finding X and Y on a Resistive Screen

The example in Figure A-2 displays how an x-axis touch that occurs closer to the supply rail will have less resistance (4Ω). The y-axis position is central, therefore the resistance is equal either side of the touch. Essentially the two layers operate in the same way as a potentiometer (variable resistor), where layer X is the resistor track and layer Y is the wiper position.

The problem with this resistive design is that multi-touch cannot be achieved, as only one voltage can be read at a time. Some of the more modern resistive screens are capable of reading multiple touches, but generally capacitive screens offer more accurate readings of multiple touches (Freescale, 2014).

Capacitive touch screens take advantage of the human body's intrinsic capacitance. Capacitance can be given by the equation (Fujitsu, 2014):

$$C = \varepsilon \left(\frac{A}{d} \right)$$

$$\varepsilon = \varepsilon_0 * \varepsilon_r$$

Where:

C is the capacitance

ε_r is the relative permittivity (dielectric constant) of the insulating material between the capacitor plates.

ε_0 is the permittivity of free space (8.854x10-12F/m)

A is the area of the plates.

d is the distance between the plates.

Figure A-3 shows the practical implications of these values when a person touches a capacitive surface.

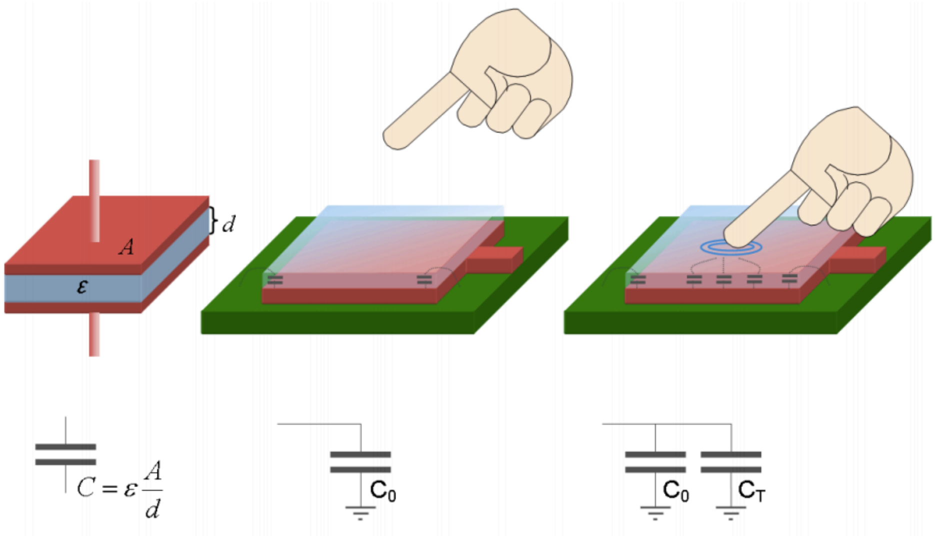


Figure A-3 The Body's Influence on a Capacitive Screen (Fujitsu, 2014)

The addition of a touch on the surface of the screen adds capacitance (C_T) to the system. The resulting change in capacitance can be measured by a microcontroller. This simple method of capacitive touch sensing is known as 'Surface Capacitance'.

A variation of surface capacitance, referred to as projective capacitance, detects disturbances in a capacitive field, rather changes in capacitance value (Blindmann, 2011). To detect this, capacitive screens implement a series of pads, or alternatively columns and rows to detect the location of a touch. Figures A-4 and A-5 display the difference between the two approaches.

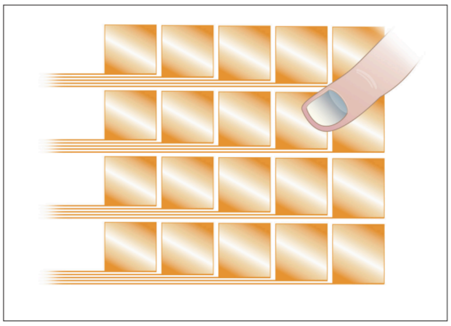


Figure A-4 Capacitive Sensing with Multi-Pads (3M, 2013)

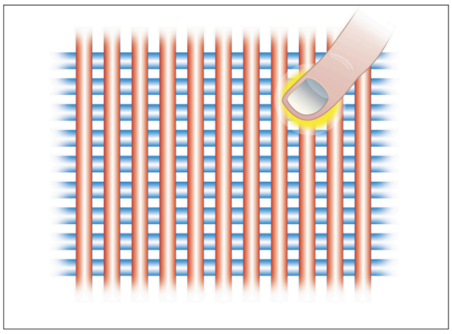


Figure A-5 Capacitive Sensing with Rows and Columns (3M, 2013)

Employing the multi-pad design allows multiple touches to be detected with ease, however, each pad has to be addressed individually by the controller circuit, which compromises speed and, ultimately, increases cost of the system. The disadvantage of the rows and columns design is that it cannot accurately detect multiple points (without software optimization). Figure A-6 displays the 'ghost points' that are produced in a column and row design. It demonstrates how the system is only able to locate simultaneous touches with a low degree of accuracy. For example, a microcontroller can detect a touch has occurred on Rows X1 and X2. It can also detect that a touch has occurred on Columns Y0 and Y3. What it cannot do is determine which X-position relates to which Y-position.

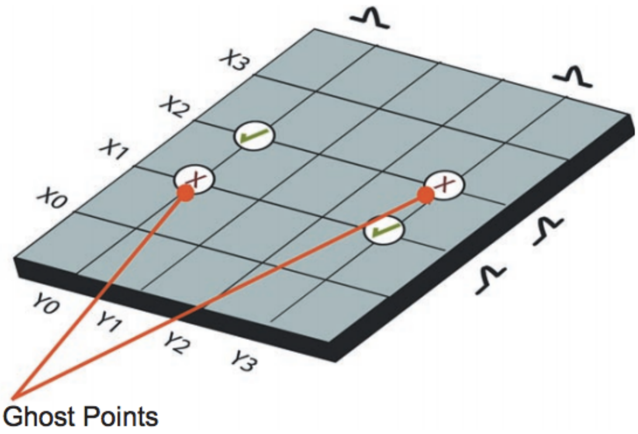


Figure A-6 Ghost Points (Barrett and Omote, 2010)

One way to combat this is to use mutually capacitive touch screens. All of the previous examples are described as 'self-capacitive' where a touch introduces capacitance to a system. In a mutually capacitive design, the 'charge field' (capacitance) between two objects is altered by the presence of another capacitive object, such as a finger (Barrett and Omote, 2010). Essentially the capacitance of the human body 'steals' some of the charge at the intersections between the rows and columns, as shown in figure A-7.

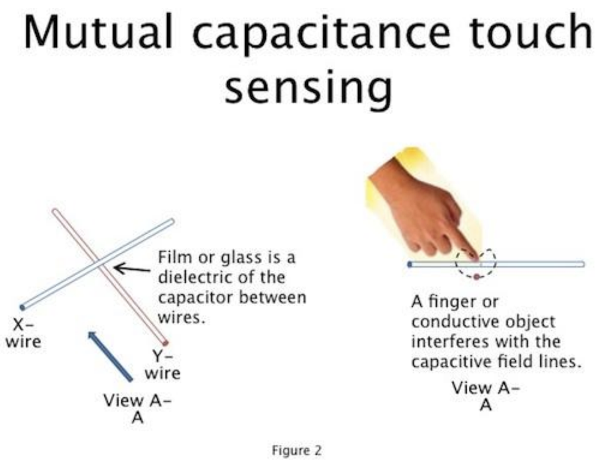
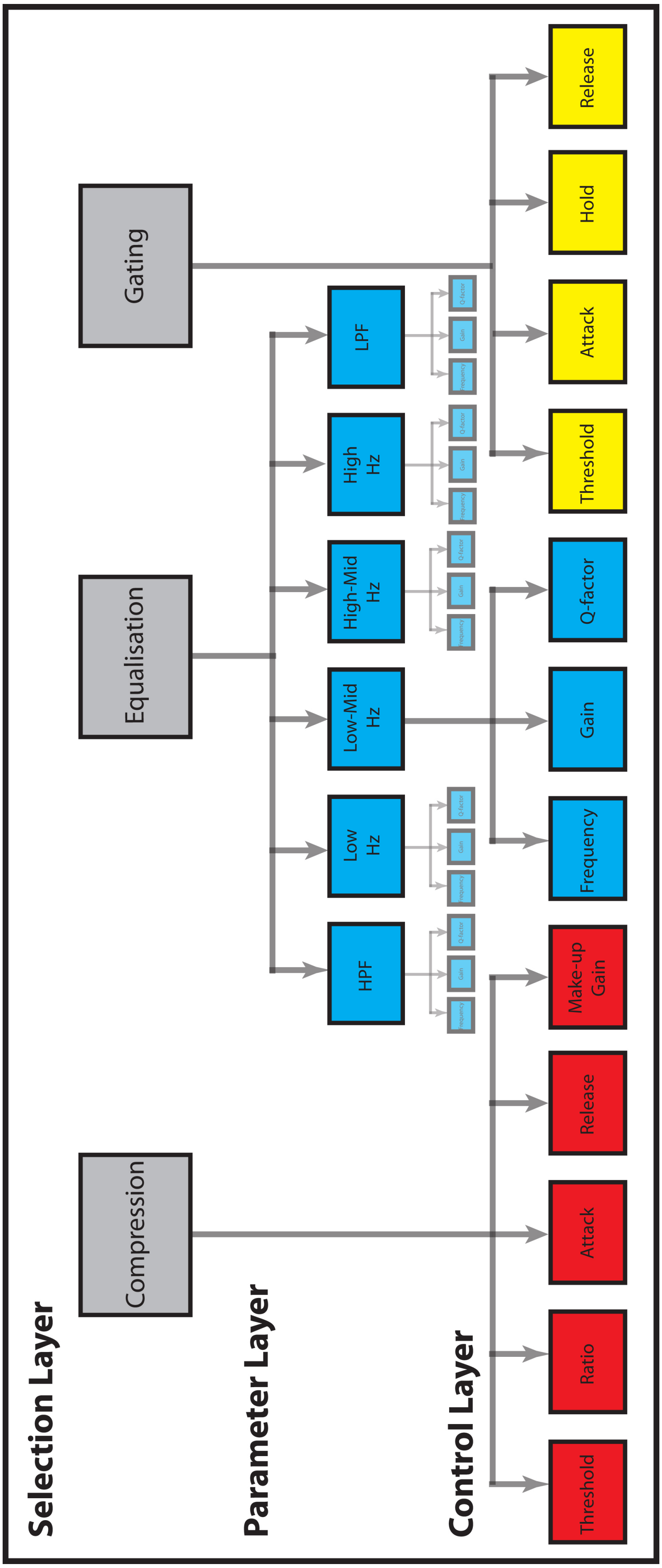


Figure A-7 Interference between X and Y Crossing Points (Narasimhan, 2014)

This results in a co-ordinate (x-y) position being produced for each touch. Therefore, ghost points are not an issue with mutually capacitive screens. Mutually Capacitive screens require more processor intensive and complex 'scanning' methods to determine locations of each touch, but the accuracy of their multi-touch measurements mean that these types of screens remain the most popular.

Other methods of touch screen gesture recognition include frustrated total internal reflection (FTIR) and acoustic pulse recognition. These, along with other optical designs, are often used where large touch surfaces need to be created at low cost.

Appendix B - Proposed System Structure



Appendix C - Engineer Workflow Observation Hand-out

Hello, Thank you for taking part in my Investigation. I will remain in the room while you follow the steps, simply to make observations and save the project when you finish each stage. The whole test should take no more than 20 minutes, though there are no time limits or expectations.

There will be 4 parts to the test. In each part you will be asked to match the sample to the reference using the following processes:

- Corrective EQ - Remove The Resonant Frequency
- Creative EQ - Match the tonality of the reference.
- Gating - Isolate the kick drum, as demonstrated in the reference.
- Compression - This test mimics an *inconsistent drummer.* Use compression to make the hits more dynamically consistent.

The plug-ins have already been loaded, please DO NOT change the plug-in.

1. Are the Keyboard and Mouse Set up in a comfortable position? Make sure the mouse sensitivity is suitable.
2. Have you used pro-tools before? Are you familiar with the standard plug-ins? Let me know if not.
3. Please familiarise yourself with the mix window. You will only need to adjust the plug-in and switch between the source and reference tracks. X-OR solo mode has been enabled to make this easier.
4. Each part of the test will be recorded using Quicktime Screen Capture software.
5. The test will begin when the Quicktime screen recording is initialised.
6. When you feel you have finished the first part of the test (Corrective EQ), you can stop the recording by clicking the stop button in the top right corner of the menu bar.

Appendix D - Gesture Recognition Algorithm Example

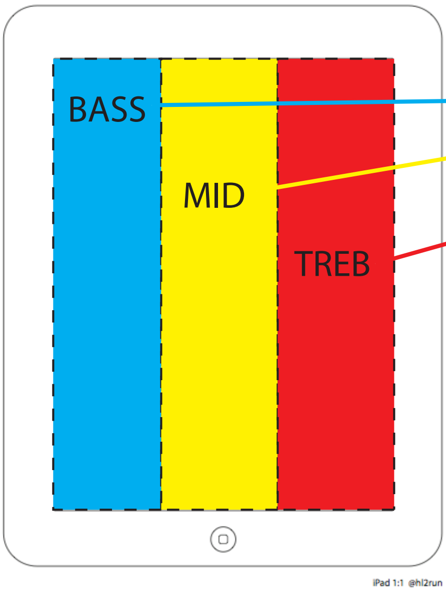
This Appendix presents a generalised and simplified Objective-C code example for one of the custom gesture recognition algorithms used in the system. Other parts of the system follow a similar recognition process, with different nuances, particularly with regard to shape detection and temporal probability calculations. Full source code is provided on the CD accompanying this thesis.

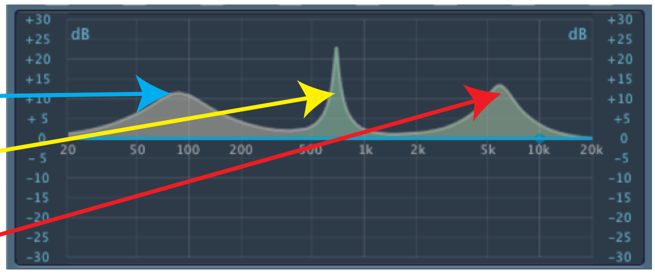
```objc
touchesArray = [touches allObjects];          //Store the touch point data in a local Array
NSUInteger numTouches = [touchesArray count];   //Count the number of Touches that have been detected

if (shiftHeld == 1)                             //the SHIFT button is used to perform 'selection layer' gestures.
{
    if (numTouches == 2)                            //if two touches have been detected
    {

        touch1 = [touchesArray objectAtIndex: 0];       //store the first touch as UITouch object 'touch1'
        touch2 = [touchesArray objectAtIndex: 1];       //store the second touch as UITouch object 'touch2'
        point1 = [touch1 locationInView:self.view];     //store the co-ordinate data from touch1 as 'point1'
        point2 = [touch2 locationInView:self.view];     //store the co-ordinate data from touch2 as 'point2'

        //Testing revealed that more stable gesture recognition was achieved by taking an average of the two points,
        //rather than tracking both points individually. This also made the code a little more efficient.

        midPoint.x = (point1.x + point2.x)/2;           //Average x-position
        midPoint.y = (point1.y + point2.y)/2;           //Average y-position

        //<<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>> Timer <<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>>

        if (stateMachineTimer == 0)                     //start timer on first 2touch movement. Use statemachine to prevent reiteration.
        {
            clockInt = 0;
            timer = [NSTimer scheduledTimerWithTimeInterval:0.01 target:self selector:@selector(countUp) userInfo:nil repeats:YES];

            //Interval 0.01 = miliseconds, selector = The function that is excecuted on each count. The function 'countUp'
            //is just a simple incremental timer that  uses a globally accessible variable for full-system syncronisation.

            stateMachineTimer = 1;                      //State Machine is reset when 'TouchesEnded' is called.
        }

        //<<>><<>><<>><<>><<>><<>><<>><<>><<> PINCH RECOGNITION - COMPRESSION <<>><<>><<>><<>><<>><<>><<>><<>><<>><<>>

        float distX = abs(point1.x - point2.x);         //Calculate the distance between two touches (x direction)
        float distY = abs(point1.y - point2.y);         //Calculate the distance between two touches (y direction)

        float totalDist = sqrt((distX * distX) + (distY * distY));  //Calculate total distance using pythagoras

        //store the Gesture start points so that they can be compared as the users moves through the phases of a gesture.

        if (shiftStateMachine == 0)
        {
            startingDist = totalDist;                   //Store first iteration of current distance as 'starting distance'
            shiftStateMachine = 1;                      //State Machine reset when TouchesEnded called.
        }

        //find difference between initial distance and current distance between points.
        float startingMinusTot;
        startingMinusTot = startingDist - totalDist;

        if (startingMinusTot >= 200) {                  //If the gesture has moved by a sufficient ammount, send command

            //COMPRESSION SELECTION GESTURE has been recognised

            NSLog(@"Time Passed: %d", clockInt);        //Get the time taken for gesture to be detected. (Used for probability testing)

            [self performSegueWithIdentifier: @"compressorSegue" sender: self]; //Perform transistion
        }
```

Appendix E - EQ Control Test Hand-outs

Selecting & Controlling The Filter

You Can Control 3 Seperate 'Filters'. Bass, Mid and Treble. You can Select each of these by DOUBLE TAPPING in the left, middle or right of the screen.





Don't worry about 'mixing up' the position of these filters, the iPad Controller will always make sure that BASS selects Lowest, Treble selects Highest and Mid selects middle.

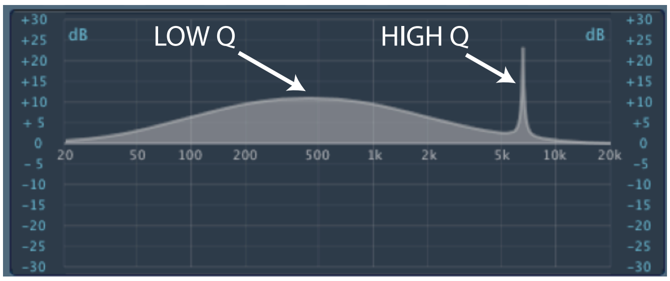
Adjusting Filters(X-Y):

X-axis = Frequency(Hz)

Y-axis = Gain(dB)

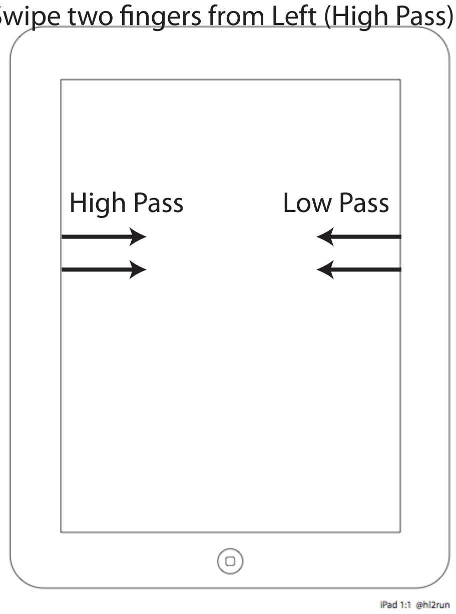
Adjusting Filters(Q - Factor):

Inward Pinch = Increased Q



Adding High Pass and Low Pass Filters

Swipe two fingers from Left (High Pass) or Right (Low Pass)

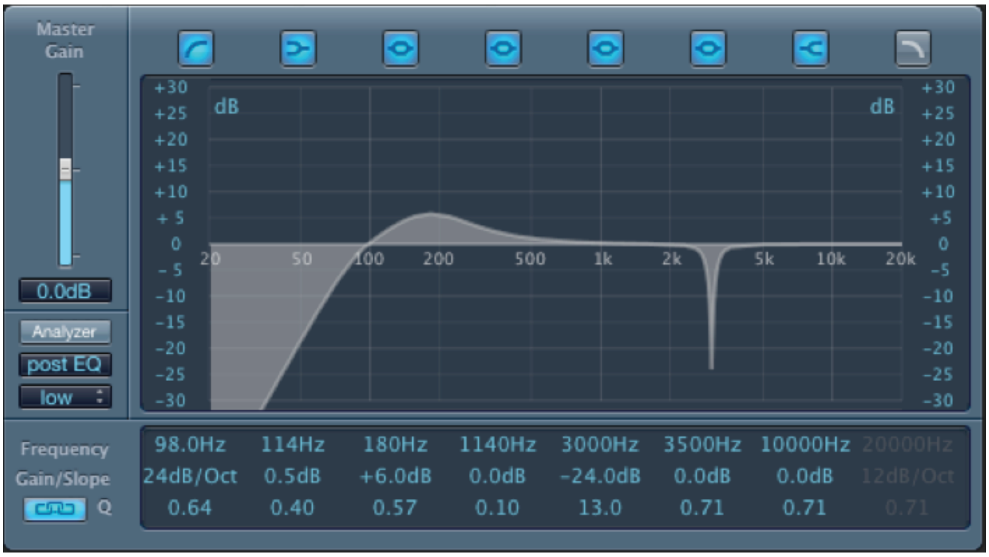


The distance that you swipe from the edge of the screen determines the Cut-off frequency of the Filter.

The High Pass or Low Pass filters are turned off by returning to the edge of the screen.

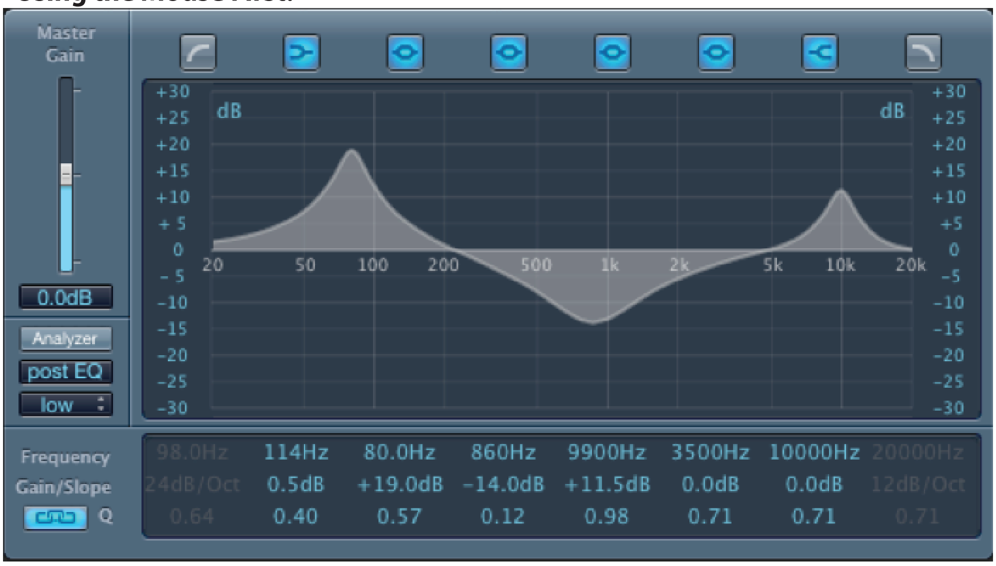
Match The following EQ Settings As quickly as possible with both the iPad and mouse/keyboard, while maintaining a 10% degree of accuracy:

Using the iPad First:



- High Pass Filter:
  Fc between 90 and 110 Hz
- bass boost:
  Fc between 162 and 198 Hz
  Gain between 5.4 and 6.6dB
  Q between 0.54 and 0.66
- Treble Cut:
  Fc between 2700 and 3300 Hz
  Gain between -20 and -24dB
  Q between 11.3 and 14.3

Using the Mouse First:



- bass boost:
  Fc between 72 and 88 Hz
  Gain between 17.1 and 20.9dB
  Q between 0.36 and 0.44
- Mid Cut:
  Fc between 774 and 946 Hz
  Gain between -12.6 and -15.4
  Q between 0.9 and 1.1
- Treble Boost:
  Fc between 8910 and 10890 Hz
  Gain Between 9.9 and 12.1dB
  Q between 0.63 and 0.77

Gestural Equalisation Control - Questionnaire

Name ______________________________

1. How often do you use touch screen devices?
a) Very Often (hourly) b) Often (daily)
c) Rarely (weekly) d) Very Rarely (Monthly) e) Not at all

2. How often do you use equalisation plug-ins / processors for mixing/composing.
a)Very Often (More than Once a day) b)Often (Once a day)
c)Rarely (weekly) d)Very Rarely (Monthly) e)Not at all

3. Did you Prefer using the Mouse or the iPad Controller?
a)Mouse
b)iPad
What are your reasons for this preference?

____

____

____

4. How would you rate the sensitivity of the iPad Controller?
Please give a number between 1 and 10.
1- Not Sensitive enough. 5- The correct Sensitivity. 10 - far too sensitive.

4. How would you rate the accuracy of the iPad Controller (How easy was it to match the target value)? Please give a number between 1 and 10.
1- Not Accurate Enough 5- Moderate accuracy 10 - very accurate

5. Did you prefer using the iPad Controller in portrait or Landscape orientation?

____

Thank you for helping me with my research. Please include any notes or suggestions on the back of this sheet. I'm particularly interested in ways you might think the controller could be improved!

Appendix F - Dynamics Control Test Hand-outs

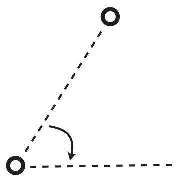
Controlling The Gate

Threshold

- Single Touch
- Y-Axis
- Decrease Thresh

- Single Touch
- Y-Axis
- Increase Thresh

Attack



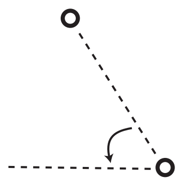
- Two Touch
- Acute Angle between touches
- Angle determines Attack time
- Smaller Angle == Slower Attack

Hold

- Two Touch
- Swipe Right
- increase hold time

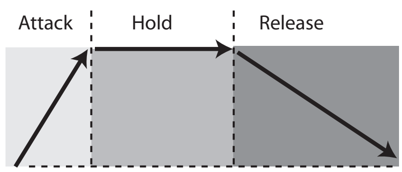
- Two Touch
- Swipe Left
- Descrease hold time

Release



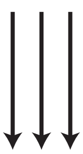
- Two Touch
- Acute Angle between touches
- Angle determines release time
- Smaller Angle == Slower Attack

*Note: Controlling The Envelope



- When controlling the Attack, Hold and Release of a gate, imagine you are drawing the Envelope.
- Angle of attack. Steeper angle produces a faster attack. i.e. gate opens more quickly.
- Hold == horizontal swipe.
- Angle of Release. Shallow angle produces a slower release. i.e. gate closes more slowly.

Reduction



- Three Touch
- Swipe Down
- increase reduction

- Three Touch
- Swipe Up
- descrease reduction

Controlling The Compressor

Threshold

- Single Touch
- Y-Axis
- Decrease Thresh

- Single Touch
- Y-Axis
- Increase Thresh

Ratio

- Two Touch
- Inward Pinch
- Increase Ratio

- Two Touch
- Outward Pinch
- Decrease Ratio

Attack

- Three Touch
- Swipe Left
- Faster Attack

Gesture made in the upper half of touch screen.

- Three Touch
- Swipe Right
- Slower Attack

Gesture made in the upper half of touch screen.

Release

Gesture made in the lower half of touch screen.

- Three Touch
- Swipe Left
- Faster Release

Gesture made in the lower half of touch screen.

- Three Touch
- Swipe Right
- Slower Release

Gain

- Three Touch
- Swipe Up
- Increase Gain

- Three Touch
- Swipe Down
- Decrease Gain

Match The following Gate Settings As quickly as possible with both the iPad and mouse/keyboard, while maintaining a 10% degree of accuracy:

Using the iPad First:

- Attack:
19ms > attack < 23ms
- Hold:
432ms > hold < 528ms
- Reduction:
-100 > reduction < -90
- Release:
54ms > release < 66ms
- Threshold:
-44 > threshold < -36

Using the Mouse First:

- Attack:
47ms > attack < 57ms
- Hold:
207ms > hold < 253ms
- Gain:
10.8 > gain < 13.2
- Release:
95ms > release < 116ms
- Threshold:
-11 > threshold < -9

Match The following Compression Settings As quickly as possible with both the iPad and mouse/keyboard, while maintaining a 10% degree of accuracy:

Using the iPad First:

• Attack:
9ms > attack < 11ms

• Release:
525ms > release < 635ms

• Ratio:
5.5 > ratio < 6.7

• Threshold:
-17.6 > threshold < -14.4

• Gain:
3.6 > gain < 4.4

Using the Mouse First:

• Attack:
104ms > attack < 128ms

• Release:
36ms > release < 44ms

• Ratio:
2.25 > ratio < 2.75

• Threshold:
-31.9 > threshold < -26.1

• Gain:
10.8 > gain < 13.2

Gestural Gate/Comp Control - Questionnaire

Name ______________________________

1. How often do you use touch screen devices?
a) Very Often (hourly) b) Often (daily)
c) Rarely (weekly) d) Very Rarely (Monthly) e) Not at all

2. How often do you use comp/gate plug-ins for mixing/composing.
a)Very Often (More than Once a day) b)Often (Once a day)
c)Rarely (weekly) d)Very Rarely (Monthly) e)Not at all

3. Did you Prefer using the Mouse or the iPad Controller?
a)Mouse
b)iPad
What are your reasons for this preference?

____

____

____

4. How would you rate the sensitivity of the iPad Controller?

Please give a number between 1 and 10.

1- Not Sensitive enough. 5- The correct Sensitivity. 10 - far too sensitive.

[ ]

4. How would you rate the accuracy of the iPad Controller (How easy was it to match the target value)? Please give a number between 1 and 10.

1- Not Accurate Enough 5- Moderate accuracy 10 - very accurate

[ ]

5. Did you find that an INWARD pinch to INCREASE compression ratio felt intuitive? Or should an INWARD pinch DECREASE the compression ratio?

- ☐ Inward Pinch = Increased Ratio & Outward Pinch = Decreased Ratio.
- ☐ Inward Pinch = Decreased Ratio & Outward Pinch = Increased Ratio.

6. Did you prefer the method of setting Attack and Release on the gate or compressor?

Compressor - 3 touch swipe in top or bottom half of screen.

Gate - Attack and release set by the angle (ramp) between two touches.

What are you reasons for this choice?

____

____

Thank you for taking part in my test, if you have any other comments or suggestions, please note them on the back of this page.

Appendix G - SAFE Rationalised Control Questionnaire

SAFE Gesture Shortcut - Questionnaire

Name ______________________________

1. Please give a rating for how EASILY and how ACCURATELY you feel reference SAMPLE 1 was matched using the interface.

EASE OF USE ☐ **(between 1 & 5, where 5 is very easily and 1 is not easily set)**

ACCURACY ☐ **(between 1 & 5, where 5 is very Accurately and 1 is not Accurately)**

2. Please give a rating for how EASILY and how ACCURATELY you feel reference SAMPLE 2 was matched using the interface.

EASE OF USE ☐ **(between 1 & 5, where 5 is very easily and 1 is not easily set)**

ACCURACY ☐ **(between 1 & 5, where 5 is very Accurately and 1 is not Accurately)**

3. Please give a rating for how EASILY and how ACCURATELY you feel reference SAMPLE 3 was matched using the interface.

EASE OF USE ☐ **(between 1 & 5, where 5 is very easily and 1 is not easily)**

ACCURACY ☐ **(between 1 & 5, where 5 is very Accurately and 1 is not Accurately)**

4. Do you think that the level of accuracy offered by the interface was sufficient to match each reference sample?

☐ **yes**

☐ **no**

5. What aspect of the audio did you have most difficulty matching?

☐ **Loudness** ☐ **Dynamics (Envelope Characteristics)**

☐ **Tonality** ☐ **Other** ______________________________

6. In order to improve the mixing accuracy. Which parameter would you find most useful to have independent control over?

☐ **Threshold** ☐ **Ratio**

☐ **Make-Up Gain** ☐ **Attack** ☐ **Release**

Thank you for taking part in my test, if you have any other comments or suggestions, please note them on the back of this page.