Touchpoint: Dynamically Re-Routable Effects Processing as a Multi-Touch Tablet Instrument

Nicholas K. Suda
California Institute of the Arts
24700 McBean Pkwy.
Valencia, California 91355
United States
nicholassuda@alum.calarts.edu

Owen S. Vallis, Ph.D
California Institute of the Arts
24700 McBean Pkwy.
Valencia, California 91355
United States
ovallis@calarts.edu

ABSTRACT

Touchpoint is a multi-touch tablet instrument which presents the chaining-together of non-linear effects processors as its core music synthesis technique. In doing so, it utilizes the on-the-fly re-combination of effects processors as the central mechanic of performance.

Effects Processing as Synthesis is justified by the fact that that the order in which non-linear systems are arranged results in a diverse range of different output signals. Because the *Effects Processor Instrument* is a collection of software, the signal processing ecosystem is virtual. This means that processors can be re-defined, re-configured, created, and destroyed instantaneously, as a "note-level" musical decision within a performance.

The software of **Touchpoint** consists of three components. The signal processing component, which is addressed via Open Sound Control (OSC), runs in Reaktor Core. The touchscreen component runs in the iOS version of Lemur, and the networking component uses ChucK.

The resulting instrument unifies many perceptual domains of modulation into a consistent interface, encouraging an expressive exploration of the areas between their borders. **Touchpoint** attempts to embody and make vital an aspect of contemporary music which is typically treated as augmentative and secondary.

1. INTRODUCTION

In the 21st century, digitally automated, computerized recording is the mainstream method of music engineering. More often than not, any style of music – be it rock music, jazz, folk, bluegrass, dance music, film scores, and especially hip-hop and experimental music – is tracked to hard disk, and variously employs the techniques of computer-based creation. [1]

Thanks to plug-ins and the Digital Audio Workstation (DAW), granular synthesis, phase vocoding, sampling, and a serialist's sense of parametric automation have found a home in the toolset of the average recording studio.

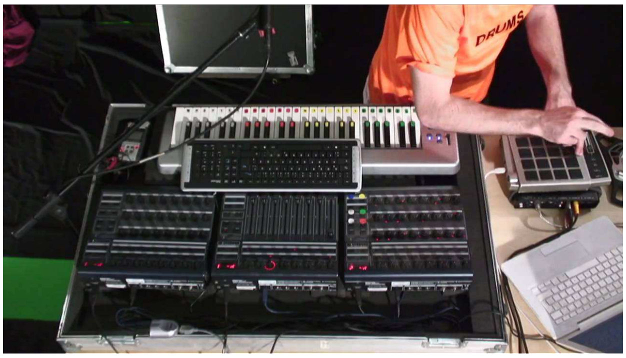

Figure 1. The sprawl of peripherals used to control IDM artist Tim Exile's live performance.

In time stretching, pitch correcting, sample replacing, noise reducing, sound file convolving, and transient flagging their recordings, studio engineers of every style are regularly applying the tools of art music composition from decades past in their work. [2] The lines between recording, editing, composition, and sound design have blurred, and the definition of what is considered a "computer music" technique is obscuring.

As the compounding of elaborate digital techniques becomes easier to apply, popular new sounds and genres are more frequently emerging from non-performative, non-instrumental, or virtual sources. [3]

1.1 Live Presentation

It is generally accepted that the concert presentation of most electroacoustic / academic computer music is inherently a playback of a series of heavily edited, through-composed moments. However, the ordinary assumption made in popular music is that every musical element is being presented live, as an instrumental performance. Anything other than a totally mimetic presentation of the music is simply excused as "magic." [4]

Too often, the disembodiment and unperformability of recreating these emergent forms of popular music live is replaced with pure spectacle. However, those who attempt to re-enact the studio process in a live setting find themselves buried under a pile of generic controllers, like in Figure 1. The commitment to the re-embodiment of these processes interferes with an emotional connection to the recitation of the music itself.

Certainly, the idea of the performative effects processor is nothing new in computer music studies. *The Hands* of Michel Waisvisz [5] explored the idea of gesturally manipulating audio buffers to great effect nearly thirty years ago.

The pursuit of embodied computer music systems with performative interfaces has seen such an investment from institutions like Stanford's CCRMA [6], UC Berkeley's CNMAT [7], and the Princeton Sound Lab [8] that an entire conference dedicated to *New Interfaces for Musical Expression* (or NIME) was created in 2001, annually collating the developments of computer music performance.

Thanks to the iPhone, the mobile device has been responsible for a recent explosion of dynamic touchscreen-based interfaces. [9, 10] In pursuit of embodying the use of effects processing in contemporary pop music, even previous attempts to sandbox and consolidate plug-in based compositional ecosytems have been realized before. [11, 12]

1.2 The Signal Processing Instrument

This paper presents **Touchpoint**, which is a software instrument that networks an iPad-based touchscreen interface with a signal processing engine that lives on a remote host computer.

It is primarily inspired by a wave of recent DAW plugins which are multi-effects processors. Products such as Native Instruments' *The Finger* (2009) [1] , iZotope's *Stutter Edit* (2011) [2] , and Sugar Bytes' *Turnado* (2011) [3] represent the nascent efforts of the commercial plug-in industry responding to the growing reliance upon elaborate signal processing techniques in contemporary popular music.

Touchpoint presents effects processing as a momentary, performative mechanic by presenting a performer with a series of two-dimensional touch fields on an iPad interface, similar to a KORG *KAOSS Pad*. Touching a field crossfades in an effects process onto the end of a dynamically constructed serial stack. This makes the order that effects are activated in important. Whether starting with a sine wave or an entire ensemble of live instruments, performing with **Touchpoint** centers around gesturally introducing, removing, and manipulating effects processors in such a transformative way that the original sound merely becomes an excitation signal for an entirely different texture, synthesized by the combination of non-linear processes.

Virtualization is a key aspect of **Touchpoint** in several regards. Signal processing sources and controller interfaces share an asymmetrical relationship, meaning that one or several instances can be performed by one or several performers in any arrangement. Furthermore, the objects within a signal processing chain are virtual, meaning that the mechanics of object creation, patching, and destruction are lucid. This forms the core of a performance practice.

[1] http://www.native-instruments.com/en/products/komplete/effects/the-finger

[2] http://www.izotope.com/en/products/effects-instruments/stutter-edit

[3] http://www.sugar-bytes.com/content/products/Turnado/index.php?lang=en

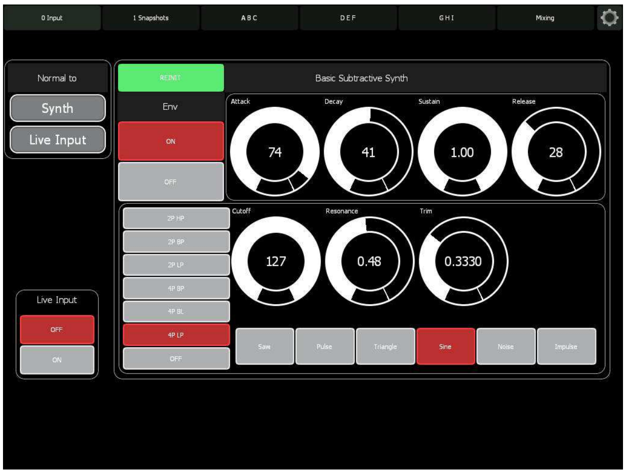

Figure 2. The *Input* page of **Touchpoint**.

1.3 Overview

Section 2 will describe in greater detail the layout of the iPad interface of the instrument. The effects processors available to the system are then overviewed in Section 3. A scenario of the average sound design process when using the instrument follows in Section 4. The mechanics of non-linear processor arrangement are discussed in Section 5, and then the networked structure of interface support is explained in Section 6. Evaluation of the system by other performers is documented in Section 7, followed by a reflective conclusion in Section 8.

2. INTERFACE

The iPad-based interface of **Touchpoint** is divided into six pages of functionality. The *Input* page determines a sound source, offering either a subtractive synthesizer or a stereo audio input. Next, a *Snapshots* page allows the performer to define "presets" of available effects processors. The main area is divided into three *Performance Pages*, where the effects processors are performed. Finally, a *Mixer* page allows for the crossfading of any currently active effects processes to be gesturally controlled. Each section will now be examined in further detail.

2.1 *Input*

Shown in Figure 2, the *Input* page defines what the sound source that excites the signal processing chain will be. If *Live Input* is selected, a stereo audio input is made constantly audible, unmuted by any actions upon any signal processors.

If *Synth* is selected, a sub-window is displayed that contains all the basic functionality of a simple subtractive synthesizer. A multi-waveform oscillator (which includes white noise and an impulse) is run through an ADSR amplitude envelope that can be filtered by a state-variable filter of low-pass, high-pass, band-pass varieties in both 2-pole and 4-pole versions. Both the filter and the envelope can be turned off. If the envelope is turned off, then the oscillator will be constantly heard regardless of processor states, as with *Live Input*.

If the performer hooks up any kind of standard keyboard MIDI controller into the computer which is running the Reaktor Ensemble, they can perform the synthesizer, with note events abiding by the amplitude envelope. Alternatively, a button on every processor window allows the performer to trigger the amplitude envelope of the *Input Synth*, which is discussed in Section 2.3.1.

By default, the oscillator will play a root A440Hz pitch, but this can be changed in the presence of any new MIDI pitch. Tapping the *Input Synth*'s *REINIT* button will normal the synth to a basic amplitude envelope, low-pass filter, sine wave, and gain state.

2.2 *Snapshots*

Performing with **Touchpoint** tends to be a very exploratory affair, with each recital taking very different turns. However, if the performer recalls a certain set of processors which work particularly well for achieving a certain kind of sound, the instrument takes advantage of Reaktor's capacity for *Snapshot* recall. In **Touchpoint**, *Snapshots* refer to the state of the instrument's current selection of processors, their settings, their internal phase, and their active/inactive status within a potential signal flow.

Although *Snapshot* creation is a manual process, switching to a *Snapshot* is instantaneous. Tapping a button on this page will completely erase the state of all nine processors and replace them with another previously defined state.

Figure 3. One of the *Performance Pages* of **Touchpoint**.

2.3 *Performance Pages*

The three *Performance Pages*, partially shown in Figure 3, consist of nine two-dimensional "XY" fields which are comparable to a KORG *KAOSS Pad*. Each XY field is reponsible for the activation of an effects process when it is touched. The effect is faded out when the performer's finger is released. Each dimension corresponds to a certain parameter of that effects process.

Each processor "Slot" is labeled with a letter, from Slots A, B, and C on the first page to Slot I on the third page.

In general, the Y-axis of each XY field is mapped to the depth at which the effect is present upon the input signal. The X-axis is mapped to a modulating frequency, the range of which is scalable. If the performer taps a second finger anywhere within the same XY field, this will reset the phase of the modulating oscillator back to its starting point.

2.3.1 *Slot configuration*

Above each XY field, there are two features. Up top is a drop-down menu, manipulated with a single tap-and-drag motion. This menu picks the type of effect that the XY field below it pertains to. Below this, there is a two-page window of parameters for that effect.

The first page is the *Global* page, which describes parameters that are shared across any effect placed in that Slot, regardless of the effect. This includes the "Onset" and "Release" times of the presence of the effect when a finger is placed on its XY field, the range of the X-axis' modulation frequencies, and a toggle for selecting whether or not the input signal is auditioned when the performer touches an XY field. (This feature allows for the changing of X- and Y-positions to be passive, so that they can be modified without re-triggering an envelope of the audibility of the input signal.)

The second parameter page per-processor describes features that are relevant to the type of processor selected in that Slot. This can include things like a basic waveform shape, pulse width, polarity, wet/dry amount, and modulation depth. Both pages have an effect re-initialization button, and a button for "latching" the effect, i.e. holding a finger down on that effect at the last-registered X and Y positions.

As the performer touches an XY field, that processor is crossfaded onto the end of a serial stack, over a time specified by the "Onset" value on that effect slot's *Global* page. Removing a finger from an XY field crossfades the presence of this effect out of the signal as per that effect slot's "Release" value. The implementation of this is described in Section 5.

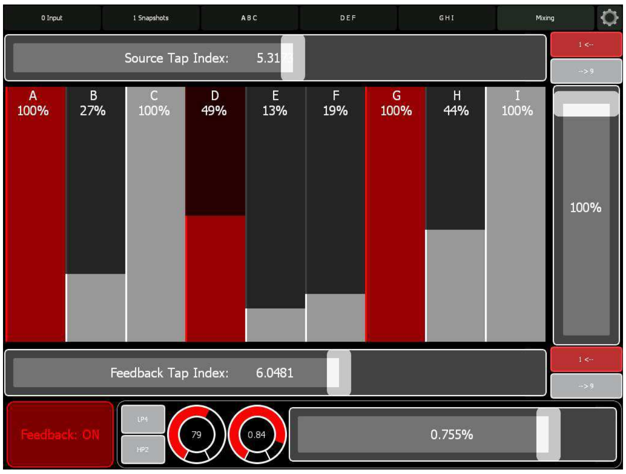

Figure 4. The *Mixer* page of **Touchpoint**.

2.4 *Mixer*

At the *Mixer* page (shown in Figure 4), the performer has access to an additional wet/dry control per effect slot, which

Table 1. Percieved modulation effects, Sinusoid

Modulation Rate	Volume	Pitch
Slow (1Hz-5Hz)	Fade-In/Out	Wow/Flutter
Medium (5Hz-20Hz)	Tremolo	Vibrato
Fast (20Hz-20KHz)	Ring Modulation	FM Synthesis

are contextually created, describing the order of the effects that are present. Effects are indexed by their Slot letter, which goes from A to I. Above the maximum nine faders is an additional horizontal fader that allows for the performer to only feed the input signal to a specific subset of audible processors out of the total chain.

This feature works best when there are close to the maximum number of processors active, over a sustained period. It allows the performer to effectively "tap" the point within the chain at which they want to feed the input signal in. The direction of the processor "tap" can be reversed with a toggle menu next to this fader.

At the right of the *Mixer* page is a global *Master Volume* fader, expressed in percent. This is an easy option for ending a performance, for instance.

2.4.1 Whole-system feedback

By turning on the *Feedback* button at the bottom of this page, the end of the signal chain is recursively fed into the beginning at an attenuated percentage specified by a newly-displayed crossfader which goes up to 90%.

Like the "source tap", the *Feedback* component of the *Mixer* is also "tappable", with a reversible direction toggle menu.

Since each processor has a hyperbolic saturator built into it for protection, this can lead to some really interesting, chirpy, overdriven artifacts from the whole system. This feedback signal is also filterable by a state-variable filter with variable cutoff frequency and resonance.

3. OBJECT CHOICES

Rather than inviting a diverse array of existing signal processor types (of varying degrees of complexity) into the instrument's design, just for the sake of variety, there are only three objects in the **Touchpoint** ecosystem.

These objects are focused from the *bottom up*, where each processor only modifies one attribute of the signal that is fed into it. However, when viewed from the *top down*, these parameters actually describe several different "effects", when driven by a periodic modulator at different frequency ranges. This *modulation spectrum* is outlined per-effect by Tables 1, 2, 3.

By presenting these phenomena on a traversable continuum, the performer can create moments where the stutter edit begins to blur into the table lookup oscillator, or where tremolo begins to create sum-and-difference tones, or where vibrato becomes deep and fast enough to become an operator pair with the incoming audio to become FM synthesis.

Thus, the **AM** object simply changes a signal's gain, the **FM** object just shifts a signal's pitch by an amount specified in semitones, and the **Comb** object is a sample buffer with two modes; with or without feedback. The "feedback" mode acts as a rudimentary Karplus-Strong plucked string [13], and the "no-feedback" mode simply repeats the input material at the specified length until release.

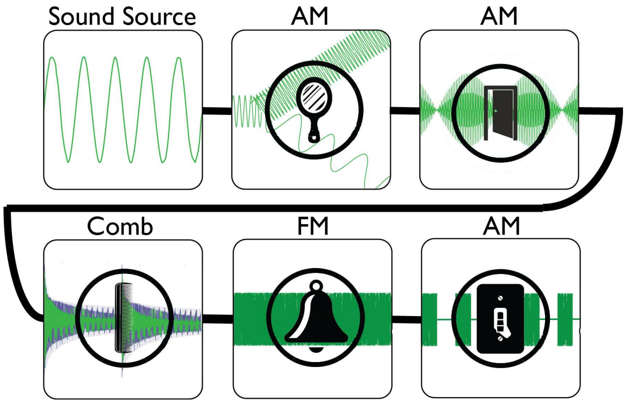

Figure 5. A visual metaphor of each process in a typical signal chain.

4. WORKFLOW EXAMPLE

Figure 5 provides a subjective visual analogy for a typical thought process behind creating a "patch" for this environment. What follows is another example, described in text.

The performer begins a session with a single A440Hz sine wave for an input signal. The *Input Synth*'s amplitude envelope is turned off, so that they can hear it streaming through the environment at all times.

The performer adds an **AM** object to the empty chain, intending to use it as a ring modulator. The object's modulation range is scaled at the minimum faster than perceivable rhythmic units, but slower than frequencies that will alias. They make the modulator bipolar, and place their finger on the object's XY field, splitting 440Hz into two midrange sine tones. They press the *Latch* button on that slot's *Global* page, holding the activated ring modulator at its current settings.

Next, the performer adds an **FM** object onto the chain, intending to use it for FM synthesis. The modulation range is scaled within an audible frequency range, and its depth is set to 12 semitones or more. The effect is triggered and latched.

Next, the performer adds another **AM** object, scaling its modulation range within rhythmic units. They make the modulator a unipolar, inverted sawtooth wave, and they activate the effect. This creates rapid 16th note pulses out of their inharmonic, dense cloud of an input sound.

The performer then uses those pulses as an excitation for a Karplus-Strong model by adding a **Comb** object to the chain, turning feedback on, scaling its modulation oscillator to a midrange frequency and activating the effect.

The performer calls up another **FM** object in order to gradually drift the pitch of these string excitations within +0.5 semitones of the original pitch, latches it, and then calls up another **Comb** object with no feedback and the full

Table 2. Percieved modulation effects, Pulse wave

Modulation Rate	Volume	Pitch
Slow (1Hz-5Hz)	Mute	Siren
Medium (5Hz-20Hz)	Gate	Yodel/Arpeggio
Fast (20Hz-20KHz)	Ring Modulation	FM Synthesis

Table 3. Percieved effects, Buffer manipulation

Modulation Rate	Feedback	No feedback
Slow (1Hz-5Hz)	Echo	Loop
Medium (5Hz-20Hz)	Delay	Stutter Edit
Fast (20Hz-20KHz)	Comb Filter	Table Lookup

range of repetition frequencies, with a quick attack-release activation crossfade envelope. This is used to "performatively stutter" the sound, occasionally re-sampling the input by tapping the XY field rhythmically with a second finger, while also moving around the pitch of the previous **Comb** instance.

In this example, each of the objects was used in several different applications by adjusting their settings. The **AM** object was used as a frequency-refracting ring modulator at one time, and as a makeshift amplitude envelope / clocked step sequencer in another. They differed only in waveform shape, polarity, and frequency. The **FM** object was used as an FM Synthesis engine, and an emulation of tape wow and flutter at different points. The **Comb** module was used as a pitched, resonating string and as a performative stutter buffer, differing only in their ranges and whether or not feedback was present in the signal path.

5. NON-LINEAR PROCESSOR ARRANGEMENT

The key conceit of the concept of *Effects Processing as Synthesis* is that the order of non-linear processing systems matters. In DSP (Digital Signal Processing) theory, a *Linear System* refers to any process which, amongst other characteristics, exhibits *additivity* in combination. [14] This means that a sum of several processes can still be inferred as being the combination of a series of individual operations. Each transformation is observable in the final result; the order of operations is unimportant.

In combining a series of non-linear processors, the output result is inherently tied to the order in which transformations are applied. Individual processors will respond with different output results depending on the characteristics of the incoming signal, such as its spectrum and loudness. In **Touchpoint**, this principle is applied by treating each processor as an entry within a dynamically compressed stack, represented in Figure 6.

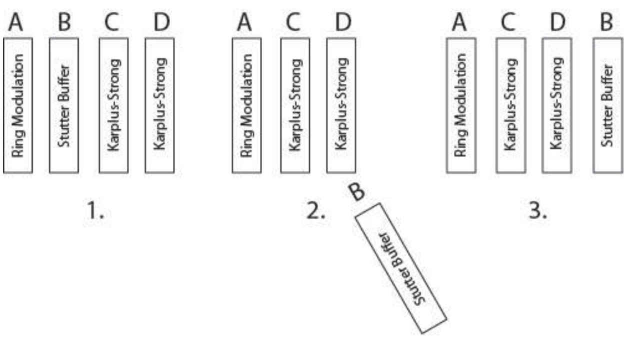

Figure 6. Processors are serially appended in the order they are activated.

5.1 Example

In Figure 7, a spectrograph was rendered, comparing an input signal of an A440Hz sine wave with two different combinations of the same two objects placed after it. Each configuration is played for four seconds, lasting twelve seconds total. Each of the effects processors has the same frequency and wet/dry strength in its modulation oscillator.

In the center configuration, an **AM** object is placed before an **FM** object. In this scenario, the fundamental pitch appears to have been tuned a chromatic half-step up, with the appearance of at least two perfect octaves somewhere below it, and a lot of typical clangorous Bessel Function FM activity in the midrange.

In the configuration at right, the same **AM** object is placed after the **FM** object. The resulting sound has a dominant fundamental a whole step above the original pitch, with many overlaid pure tones above it that make it sound like a two-tone DTMF dial tone-type sound. Many more sustained, pure frequencies are generated in this configuration than with the previous configuration.

If these processors were *Linear Systems*, their order would not matter; both cases should produce the same spectrum. However, each of these processors subtract and insert their own spectral content and amplitude envelope modifications which are dependent upon the signal being fed to them, thus creating a dynamic output result that is dependent upon order.

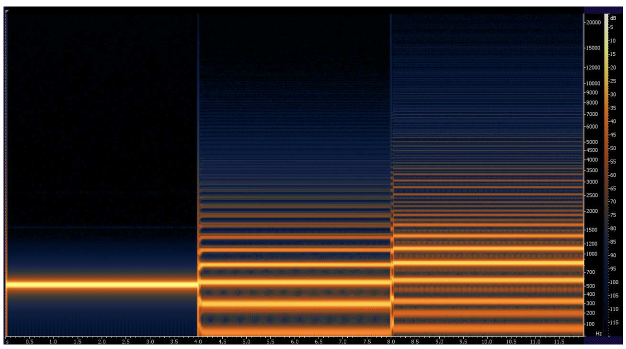

Figure 7. Sine, Sine/AM/FM, and Sine/FM/AM.

6. "N-PLAYER" SUPPORT

The networked implementation of the exchange of control messages between the tablet interface and the software engine allows for multiple tablets to control one instance of **Touchpoint**. This is a natural manifestation of the fact that this instrument is the embodiment of a virtual ecosystem with many simultaneously operating components.

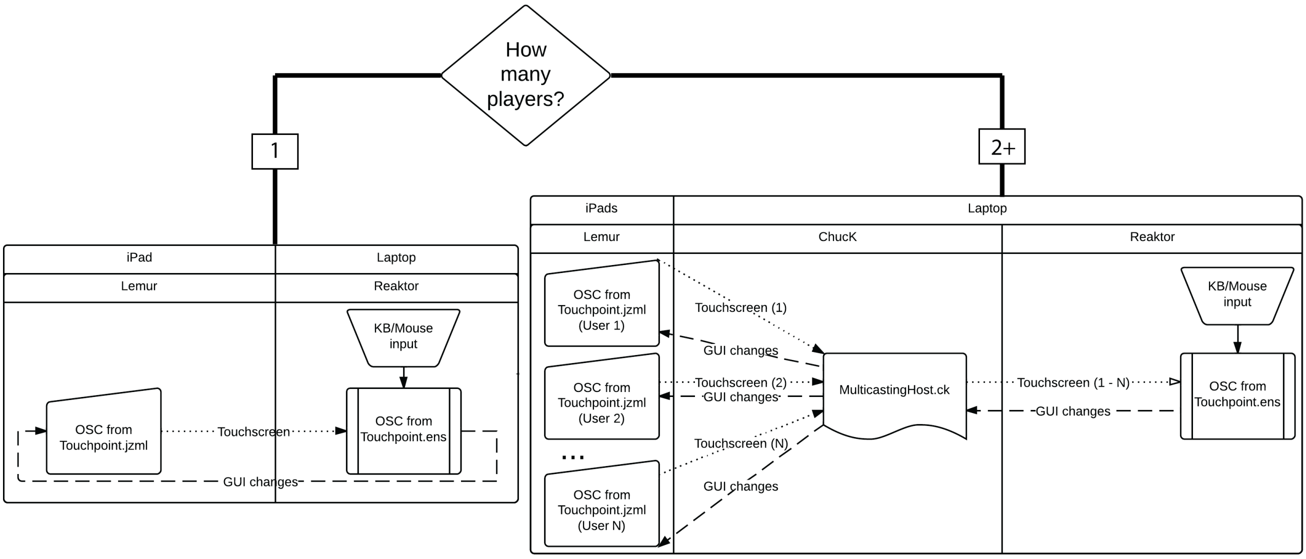

Figure 8. Software requirements of "*N-player*" vs. single-performer configurations of **Touchpoint**

As depicted in Figure 8, if only one performer is operating **Touchpoint**, then the OSC messages generated by the tablet can go directly between Lemur and Reaktor, without the use of a ChucK server. ChucK becomes involved if, for example, the performance was centered around three performers manipulating one instance of **Touchpoint**.

This feature is made possible by the addition of general port listening and UDP[4] multicasting in newer versions of ChucK. Because ChucK no longer needs to search for a specific address path from a specific Open Sound Control source, it acts as an open-door middleman between Lemur and Reaktor.

At startup time, the IP address and output port of each tablet is specified. Each tablet is given its own thread on the CPU, with an additional thread broadcasting any GUI (Graphical User Interface) changes received Reaktor-side to be multicasted back to all of the tablets.

This can also lead to a totally asymmetrical model of **Touchpoint** instances and controller interfaces. Two sessions could be manipulating two different audio input sources, with three players manipulating the first session, and the third player also using his control data to manipulate the same area of the second session. This kind of flexibility leads to all kinds of interesting networked performance settings.

7. EVALUATION

Touchpoint has met the performing hands of several other people throughout the previous year of its development. Given the goal to eventually perform in a collaborative improvisation, two people were deliberately introduced to the instrument with very disparate amounts of time to learn and practice with it.

This strategy was conducted in the interest of seeing if significantly divergent performance styles could arise, particularly on such a personalized style of DMI.

[4] User Datagram Protocol, a form of network packet which is utilized in the Open Sound Control format.

Christopher Knollmeyer only had a mere couple of days with **Touchpoint**. He corroborated its capacity for a reflexive and impulsive sound design process, while criticizing on its relative lack of performance interface feedback.

"***Touchpoint*** *is a valuable tool for performance because it offers the user immediate control of complex processes. Upon seeing the visual interface, the first elements to gain attention are the three x-y fields. These are obviously very hands-on and can be used impulsively. This impulsive capacity allows for a greater sensation of spontaneity, an element missing from much live electronic music.*

The option to configure three spaces with three processors gives significant freedom to the range of playability, but is nonetheless limited to these three qualities of effect. If I were to ask for something more within the ***Touchpoint*** *interface, it would be the ability to enter a specified range of each x-y field for more specified results. This could lend use to more arranged moments between a group of players with one or more musician performing on* ***Touchpoint****. For example, I would like to have a delay time or modulator frequency happening close to a specific song tempo. A choice of scaling options could behoove the x-y field as well.*"

Colin Honigman, in contrast, was given ample time and opportunity to develop a relationship with **Touchpoint** unmitigated by instruction or aesthetic tutoring from the developer. Over a period of nearly six months, Colin regularly performed with **Touchpoint** in a traditional electroacoustic orchestral ensemble.

"***Touchpoint*** *is a novel combination of touch interface and synthesis technique that results in a multi-faceted and expressive instrument. To begin with, it is a fairly simple interface that can be more or less understood without much explanation. On the surface, it is easy to begin playing, and it is obvious that many sounds can be achieved. With practice and exploration, the instrument is found to be capable of a multitude of sounds, with the added ability to manipulate those sounds on micro and macro levels.*

The choice of three sound generators for each unit allows *for a surprising amount of variation, and provides continuously surprising and pleasing effects. Looping, while not an independent feature, is achievable through knowledgeable manipulation of the comb filter. In a performing context,* ***Touchpoint*** *allows for great control and expressivity, and is especially useful in an improvisatory context. Repeatable performances are achievable, but require practice, and the careful use of presets.*

Figure 9. L-R: Suda, Knollmeyer, Honigman in concert.

However, the exploratory approach is my personal favorite. This approach also makes ***Touchpoint*** *an excellent tool for sound design and sample creation. While extended performance requires more practice, for production purposes, this instrument can be used to easily create a large amount of content with very little effort. Moving back and forth from simple to complex timbres, many exciting sonic possibilities reveal themselves, through the interaction of the modules.*

Sonically, ***Touchpoint*** *seems very similar to a modular synthesizer. However, the ability to traverse the timbral spectrum is unprecedented when compared to the analog method of patching cables. Although, because it is digital, it suffers and benefits from its digital behaviours. There were a few "glitches", in the developer's mind, that I found to be unique features that allowed for the creation and performance of different styles. For instance, tapping a second finger on the xy pad to restart the phase could cause a percussive clicking sound that allowed for precise rhythmic performance otherwise not really achievable without this "glitch."*

While ***Touchpoint*** *is similar in theory to a Kaoss pad, this is only a consequence of the xy touch interface itself. In my past experiences with it, the Kaoss pad feels like one is merely manipulating the parameters of different effects. With* ***Touchpoint****, the interaction is more expressive, and the output can be more surprising, especially as modules are locked, stacked, and manipulated simultaneously.*

I learned and played this instrument while rehearsing with an Electronic Music Ensemble, an improvising electroacoustic ensemble. I found that I had the ability to improvise a large range of dynamics and sounds that worked well with electronic and acoustic instruments alike.

The instrument intrigued the other ensemble members, who would say things like, "I don't understand what you're doing, but I like it." Like the acoustic instrumentalists, I found myself "warming up," creating a basic starting pallete at the beginning of rehearsal that I would change and explore variations of throughout each piece. I was noting the combinations of parameter values, positions, gestures, and modules that seemed to work well.

With each practice (both ensemble and personal), I found that I had more control and increased ability to repeat sounds and transitions from one sound to another. There was always the element of surprise, as there are so many combinations available that even very small changes can have drastic results, especially when creating complex signal chains. Personally, I look forward to continue playing this instrument, for both performance and production purposes."

Knollmeyer and Honigman's reactions help to illustrate that the balance between the opaque and the discrete in the instrument's current interface has mutually valuable benefits and drawbacks. While it can lead to the sensation of wonder in Honigman's "exploratory" method, it can also appear to be a bit too continuous – like an unlabeled violin neck – to those who wish to compose for it.

8. CONCLUSION

In this paper, it has been suggested that many of the sounds of DSP-based computer music have entered into popular music, as music recording has become computerized and accessible. While the need to present the techniques of computer music in a recitative fashion is not a requirement in academic settings, this disembodiment has become problematic for the traditionally "live" style of pop music.

Meanwhile, the academic scene has substantiated the embodiment of computer music for decades. Platforms such as the iPhone and iPad are providing an avenue for years of gesture mapping [15] and sensor-based interface research to be embraced by the public.

By associating a complex signal processing engine with a touchscreen-based hardware interface, the potential bloat of needing many physical controllers to control an ecosystem of objects is reduced to a single device. While there are notable drawbacks to the use of a touchscreen controller as an intuitive peripheral – namely, the lack of haptic feedback – the modularity of expanding the interface of such a peripheral with just a little bit of extra code exceeds the urgency of this concern.

While devices such as guitar effects pedal boards, modular synthesizers, rack-mounted studio gear, the software plug-in, and other forms have embodied the signal processor onstage before, **Touchpoint** cherry-picks many of their mechanics, seeking to represent a comprehensive embodiment of the signal processor as an instrument, in the name of accessibility. Its interface is neither too opaque and nonrepresentative, like a new form of gestural DMI (Digital Music Instrument), nor is it too much a mess of windows and keyboard and mouse operations, like a computer application.

Virtualization – in both the hardware interface, and the software engine – was identified as the most important aspect behind **Touchpoint**'s construction. A touchscreen-based virtual hardware interface allows for the dynamic representation of an ecosystem which contextually spawns and removes subsets of controller widgets.

Virtualized objects within this environment allows for the existence of objects and their relationship to each other to become a stream-of-consciousness musical decision. Finally, a virtualized link between hardware interface and software engine allows for asymmetrical pairings of instruments and performers in an "N-player" relationship.

Touchpoint has a great depth of settings configuration options, while also being subconscious in its actuation mechanics. Not aspiring to be the interface of a "score-level" conducting system, or an "effect-level" global parameter adjuster, the instrument is such that playing a single processor is a "note-level" action [16], akin to pressing a single key of a piano. Early appraisal of the system by external performers provides a valuable insight into the efficacy and novelty of this design strategy, as well as where it is headed in the future.

Touchpoint seeks to combine the aesthetics of audio plug-in heavy, effects processing-based music composition with the recent explosion of app-based multitouch tablet instruments, in an attempt to embody what has traditionally been neglected in the concert performance of popular music.

Principally grounded by the dynamic outcome of gesturally instantiatied non-linear processor re-combination, the instrument quickly does something that a DAW channel strip loaded with a long series of effects processors cannot do without excessive preparation.

By liberating the effects processor from the burden of premeditated preparation, the performer is potentially obliged to performatively embody the previously offline aspects of their composition which are so important. Also, the re-contextualization of this practice as an instrument can lead to new forms of musical expression and genre by itself. The "effects processor recital" - solo, or even in an ensemble - suddenly becomes a coherent notion.

9. REFERENCES

[1] A. Hearst, "Technology puts the recording studio on a hard drive," *New York Times*, Nov. 1999. [Online]. Available: http://events.nytimes.com/1999/11/11/technology/11down.html

[2] M. McGowan, "Auto-tune's effect on musicians, genres, and culture," Ph.D. dissertation, Carleton University, 2012. [Online]. Available: https://curve.carleton.ca/system/files/theses/28492.pdf

[3] K. Cascone, "The aesthetics of failure:post-digital tendencies in contemporary computer music," *Computer Music Journal*, vol. 24, no. 4, pp. 12–18, 2000.

[4] W. A. Schloss, "Using contemporary technology in live performance: The dilemma of the performer," vol. 31, no. 1, 2002.

[5] M. Waisvisz, "The hands: A set of remote MIDI-Controllers," in *Proceedings of the International Computer Music Conference*, 1985.

[6] B. Verplank, M. Gurevich, and M. Mathews, "The plank: designing a simple haptic controller," in *Proceedings of the 2002 conference on New interfaces for musical expression*. National University of Singapore, 2002, pp. 1–4.

[7] M. Zbyszynski, M. Wright, A. Momeni, and D. Cullen, "Ten years of tablet musical interfaces at cnmat," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2007, pp. 100–105. [Online]. Available: http://www.nime.org/proceedings/2007/nime2007_100.pdf

[8] D. Trueman and P. Cook, "Bossa: The deconstructed violin reconstructed," *Journal of New Music Research*, vol. 29, no. 2, pp. 121–130, 2000.

[9] F. Calegario, J. Barbosa, G. Ramalho, G. Cabral, and G. Finch, "Sketchument: Empowering users to build dmis through prototyping," *Organised Sound*, vol. 18, no. 03, pp. 314–327, 2013.

[10] S. Tarakajian, D. Zicarelli, and J. K. Clayton, "Mira: Liveness in iPad Controllers for Max/MSP," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013. [Online]. Available: http://nime2013.kaist.ac.kr/program/papers/day3/poster3/241/241_Paper.pdf

[11] D. Gibson and R. Polfreman, "An architecture for creating hosting plug-ins for use in digital audio workstations," in *Proceedings of the International Computer Music Conference*. International Computer Music Association (ICMA), 2011.

[12] W. Kleinsasser, "Dsp.rack: laptop-based modular, programmable digital signal processing and mixing for live performance," in *Proceedings of the 2003 conference on New interfaces for musical expression*. National University of Singapore, 2003, pp. 213–215.

[13] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.

[14] S. W. Smith *et al.*, "The scientist and engineer's guide to digital signal processing," 1997.

[15] K. Schlei, "Tc-11: A programmable multi-touch synthesizer for the ipad," *Proc. of New Interfaces for Musical Expression (Ann Arbor, USA, 2012)*.

[16] M. M. Wanderley and N. Orio, "Evaluation of input devices for musical expression: Borrowing tools from hci," *Computer Music Journal*, vol. 26, no. 3, pp. 62–76, 2002.