3-28-2006

# Expanding guitar production techniques : building the guitar application toolkit (GATK)

Roberto Horacio Aramburu
*Florida International University*

**DOI:** 10.25148/etd.FI14032330

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

# EXPANDING GUITAR PRODUCTION TECHNIQUES: BUILDING THE GUITAR APPLICATION TOOLKIT (GATK)

A thesis submitted in partial fulfillment of the

requirements for the degree of

MASTER OF MUSIC

by

Roberto Horacio Aramburu

2006

To: Interim Dean Mark Szuchman
College of Arts and Sciences

This thesis, written by Roberto Horacio Aramburu, and entitled Expanding Guitar Production Techniques: Building the Guitar Application Toolkit (GATK), having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

David Dolata

William Dan Hardin

Kristine Burns, Major Professor

Date of Defense: March 28, 2006

The thesis of Roberto Horacio Aramburu is approved.

Interim Dean Mark Szuchman
College of Arts and Sciences

Interim Dean Stephan L. Mintz
University Graduate School

Florida International University, 2006

# DEDICATION

I dedicate this thesis project to my wonderful wife. Without her tolerance, understanding, support, and most of all love, the completion of this work would not have been possible.

# ACKNOWLEDGMENTS

I wish to thank the members of my committee for their unconditional support, patience, and especially for their dedication and professionalism demonstrated through the development process of this thesis. Dr. Hardin was a role model as teacher and particularly helpful in the research area. He helped me to develop all the necessary skills to be able to write a thesis at an academic level. Dr. Dolata helped me to clarify my ideas, to polish the content, and to improve the format of the project. Professor Tom Owen was my main artistic inspiration. His level of expertise as a Max/MSP programmer as well as his generosity as a teacher really helped me to implement my aesthetics and technical concepts. Finally, I would like to thank my major professor, Dr. Burns, who accepted me into her program and gave me the opportunity not only to complete a Master’s degree with excellence, but also to accomplish a personal dream.

# ABSTRACT OF THE THESIS

# EXPANDING GUITAR PRODUCTION TECHNIQUES:

# BUILDING THE GUITAR APPLICATION TOOLKIT (GATK)

by

Roberto Horacio Aramburu

Florida International University, 2006

Miami, Florida

Professor Kristine Burns, Major Professor

The purpose of this thesis was to build the Guitar Application ToolKit (GATK), a series of applications used to expand the sonic capabilities of the acoustic/electric stereo guitar. Furthermore, the goal of the GATK was to extend improvisational capabilities and the compositional techniques generated by this innovative instrument.

During the GATK creation process, the current production guitar techniques and overall sonic result were enhanced by planning and implementing a personalized electro-acoustic performance set up, designing custom-made performance interfaces, creating interactive compositional strategies, crafting non-standardized sounds, and controlling various music parameters in real-time using the Max/MSP programming environment.

This was the first thesis project of its kind. It is expected that this thesis will be useful as a reference paper for electronic musicians and music technology students; as a product demonstration for companies that manufacture the relevant software; and as a personal portfolio for future technology related-jobs.

# TABLE OF CONTENTS

CHAPTER PAGE

# LIST OF TABLES

TABLE PAGE

## LIST OF FIGURES

FIGURE PAGE

# CHAPTER I

## INTRODUCTION

For the electro-acoustic composer and performer, the guitar may seem limited by a static acoustic envelope, short sustain, narrow dynamics and frequency range. These limitations were first addressed by electric guitars, through the design of different pick-up models (from single coil to Twinblade and vintage Humbucker), implementing active/passive guitar modules, redesigning tube and transistors amplifiers, and increasing the external effects boxes options. In order to have an electrified version of the acoustic and nylon classical guitar, a built-in preamplifier with four-band equalizer, notch filter, and phase switch controls was added.

Currently, one of the finest hybrid instruments is the Carvin AE 185 Acoustic/Electric stereo guitar; and the most sophisticated technological advance applied to the guitar is probably the new modeling guitar implemented in the Variax Series from the Line 6 company. There is almost no research, however, concerning the stereo guitar music production techniques field. Most software is designed for the standard electric guitar in the rock and pop context. Much of the commercially released software is effects oriented (e.g. chorus, delays, reverbs, compressors, etc) and hardware dependent —meaning that the only way to run the software is through the company's own proprietary hardware devices.

The objective of this thesis project is to develop several applications that not only enhance the qualities of the guitar and address the previously mentioned issues, but also offer the possibility to augment improvisational and compositional strategies through the computer software interaction, for example, via the Polyphonic Dreams application.

During the GATK development, current production guitar techniques are enhanced by implementing a personalized electro-acoustic performance set up, designing customized performance interfaces, creating interactive compositional strategies, crafting distinctive sounds, and controlling a variety of musical parameters in real time. Max/MSP was used as an interactive music system to create real-time sound environments for improvising in performance or compositional settings. To sum up, the GATK is a series of applications designed to expand the guitar's sonic capabilities as well as to extend the improvisational ideas and the compositional techniques inspired and created by this ground-breaking instrument.

The content of this thesis project is organized into four chapters: "Introduction", "Music Production: Core Concepts", "Max/MSP-Music Production Software Interaction", and the "Guitar Application Toolkit (GATK)". In addition, a data DVD-ROM accompanies the paper portion of the thesis to enhance the reading and comprehension experience through the support of multimedia data and extensive audio and visual examples. Concise video tutorials describe my studio setting and equipment, illustrating the main concepts and explaining selected production techniques. Audio fragments and somewhat longer examples provide an aural reference to many of the advanced technologies presented. In Appendix A: "Hardware Setups and Software

Settings", there are several examples that contain detailed instructions. Appendix B: "DVD Content", works as a cross reference to the examples included in the DVD-ROM, offering a brief description of each of them, and Appendix C contains a Free/Shareware VST Plug-Ins list.[1]

The paper-DVD combination approach was also selected to provide the reader with current production examples, who is better able to understand specific technical topics included in the paper version by watching and analyzing the video tutorial examples. For example, the reader can immediately examine Appendix A to build a Digital Audio Workstation (DAW).

Chapter Two, "Music Production: Core Concepts", describes and analyzes each of the main DAW components used for this thesis as well as the role of MIDI in music production, emphasizing the use of the MIDI guitar and several auxiliary controllers. Moreover, various traditional and experimental stereo guitar production techniques are proposed, plus Digital Signal Process (DSP), audio plug-ins, and ReWire technologies are explained and exemplified.

On Chapter Three, the main subject is music production software interaction—specifically, how Max/MSP manages MIDI, audio, VST[2] plug-ins, and the ReWire protocol. The VST plug-in controller interface is introduced and several techniques and examples are presented.

[1] Essentially, plug-ins are additional software programs that would instantly 'plug in' to the host environment, thereby adding functionality.

[2] Virtual Studio Technology (VST) developed by Steinberg.

Chapter Four is the core of this thesis project because it represents my personal approach to guitar production techniques, from the MIDI guitar to the enhanced hybrid stereo guitar. In this section, I describe the creation of my own electro-acoustic performance set up, and the design of several custom-made performance interfaces. In particular, the stereo guitar performance interface (SGPI) is presented as a macro framework for the GATK and then, each GATK application is described and exemplified.

# CHAPTER II

## MUSIC PRODUCTION: CORE CONCEPTS

### Digital Audio Workstation (DAW)

A DAW signifies a dedicated or computer-based hard-disk recording system that offers advance multitask editing, signal processing and integrated peripheral features in one package.[3] The DAW can successfully replace and enclose almost all the functionality present in a traditional console and outboard gear-based studio. Aside from the addition of a digital audio interface, hardware controllers, microphones and loudspeakers, the modern DAW does not require any audio external equipment, etc.[4]

I based my electro-acoustic music production on the DAW, relying on its strengths such as integration, communication, speed and flexibility, expandability, user-friendly operation, value and quality. A brief explanation of the main DAW components is found in Table 1.

Computers are the core component of any DAW-based studio. In the case of this research project, a Mac G4 is connected to two computer monitors (17" and 15"). In addition, a Sony Vaio laptop works in conjunction with the Mac via home networking or as an independent DAW. The Glyph GT050 is the dedicated external audio disk and is partitioned in two segments: music production software and sound sample libraries, respectively. For back ups and storage purposes, the Maxtor external disk is used.

[3] Davis Miles Huber and Robert E.Runstein, *Modern Recording Techniques*, 5th ed. (Boston: Focal Press, 2001), 227.

[4] Colby N. Leider, *Digital Audio Workstation* (New York: McGraw-Hill, 2004), 46.

TABLE 1

DAW COMPONENTS

| DAW Hardware | | | | | | |
|---|---|---|---|---|---|---|
| **Computers** | **Digital Audio Interface** | **External Hard-drives** | **Loudspeakers and Headphones** | **Other External Devices** | **Networking and FTP Devices** | **MIDI System Components** |
| Dual 1.25 GHz PowerPC G4<br>Sony Vaio PCG-K33 | Digidesign Digi 002 Rack digital audio interface. | Glyph Gt 050<br>Maxtor One touch | Event Tuned Reference 5<br>Sony MDR-7506<br>AKG K-240 studio | Marshall JMP 1 Valve MIDI Pre-Amp<br>Alesis Quadraverb Digital Effects Processor<br>Roland XV 5050 Sound Module<br>Plextor PX 708 UF DVD+/-R/RW | Buffalo AirStation WZR-RS-G54 125* High-Speed Mode™ Wireless Secure Remote Gateway<br>Buffalo LinkStation TM Network Storage Center HD-H250LAN | MIDIman Radium 61 Keyboard Controller<br>Roland GI-20 Guitar Interface<br>Roland GK-3 divided pickup<br>Boss 5L/5U Foot Switch<br>Boss FV 300L Volumen Expression Pedal<br>Behringer FCB 1010 MIDIFoot Controller |

| **Software** | | | |
|---|---|---|---|
| **Operating System** | | **Music Production (partial list)** | |
| **Mac OS X 10.3.7** | **Windows XP Home Edition** | **Mac** | **PC** |
| | | Digidesign Pro Tools LE 6.9 | Digidesign Pro Tools LE 6.9 |
| | | Waves Platinum Bundle | Acid 5.0 |
| | | Native Instrument Komplete 3 Bundle | Sound Forge 7.0 |
| | | East West Quantum Library Gold Edition (Symphonic Library) | Max/MSP 4.5 and Jitter Bundle |
| | | StormDrum Sample Library | |
| | | Reason 3 | |
| | | Max/MSP 4.5 and Jitter Bundle | |

The second major component of this particular DAW station is the Digidesign Digi 002 Rack.[5] This digital audio interface works with Windows XP or Mac OS X based computers and offers superior audio and MIDI production results. The Digi 002 Rack is comprised of a single unit that includes analog audio, digital audio, and MIDI inputs and outputs, along with four high-quality microphone preamps and communicates with the computer and Pro Tools LE software via a single FireWire cable.

Units listed in the other external devices and the MIDI system component columns work collectively during the electro-acoustic music production process. Furthermore, the devices mentioned in the Networking column allow me to send, receive and store files worldwide, without any size limit or file format restriction. Not only is the content of all the project studio machines transferable into the local area network, but through the FTP server feature of the external Buffalo Linkstation, I am able to offer online collaborations and a fast, reliable solution regarding file transfer.

Each artistic project requires a particular production approach. Most of my commercial work is recorded, edited, processed and mastered using the Pro Tools platform, along with third party plug-ins and the ReWire protocol. Conversely, in order to expand current guitar production techniques and develop a unique guitar production style, I maximized and extended the actual tendencies using my GATK as a main sound design, improvisational and compositional tool.

[5] The Digi 002 Rack MIDI features are listed in Appendix 1.

## MIDI in Music Production

In electronic music production, MIDI systems perform an expanding range of production tasks, such as music production, stage productions, live performances, audio-for-video and film postproduction and multimedia. Nevertheless, in order to understand the relationships between my audio and MIDI networks, the following explanation focuses on the MIDI setup and its use in the creative process.

MIDI hardware systems are designed to interface, distribute, process, and diagnose MIDI data. In this particular MIDI system, I use a "Daisy Chain" configuration. This method relays MIDI from one device to the next by retransmitting data that is received at a device's MIDI in port out to another device via its MIDI thru port.[6] Besides the aforementioned MIDI interface (in this case the DIGI 002 Rack), one of the main components of the MIDI system is the MIDI keyboard controller.[7] The MIDIman Radium 61 has a 5-octave keyboard combined with 16 controllers that can be mapped to any MIDI-assignable parameters included in the performer/composer's hardware and software rig.[8] For mixing through virtual faders, the eight MIDI-assignable sliders are ideal, while the eight MIDI-assignable knobs are useful for tasks like panning and effects sends. Faders and knobs are very valuable for real-time performances, giving the

[6] Or MIDI echo port.

[7] This is a MIDI interface with a keyboard whose only use is to control other MIDI devices connected to its MIDI out port. There is no internal sound module.

[8] The MIDIman Radium MIDI features are listed in the Appendix 1.

performer/composer the possibility of controlling several synthesizers' parameters, virtual instruments, and samplers in real-time.

Another alternative to working with MIDI is through a MIDI guitar. My particular system contains the GI-20 guitar interface and the GK-3 divided pickup, both from Roland. The GI-20 guitar interface handles the following MIDI messages: note on/off, bend, bank select and program messages, control change and SysEx. [9] In Mono mode, each string uses a separate channel. With a multi-timbral sound module, the performer can select six different sounds (one per string).[10] In Poly mode, the information for all six strings is transmitted over a single channel with the same sound. Moreover, the performer/composer is capable of creating and saving his own patch[11] settings using the front panel buttons and knobs. *See DVD, Chapter 2, ex. 1.*

In relation to the divided pick up, the GK-3 detects the vibrations of the electric guitar strings. *See DVD Chapter 2, ex. 2.* It is attached between the guitar bridge and the bridge pick up, and also includes volume, down/S1 switch, up S2 switch and select switch controls. In addition, a power indicator, a GK connector and a normal pickup input jack are also available.

Several auxiliary controllers can be appended to the system such as the FS-5L/5U Foot Switch and the FV-300L Volume/ Expression Pedal. [12] The FS-5U is a momentary

[9] The GI-20 features are listed in the Appendix 1.

[10] Such as the Roland XV5050.

[11] Basically, the terms "patch" or "program" usually refer to the entire collection of smaller objects and subpatches.

[12] The FV-300L features are listed in the Appendix 1.

"unlatch"-type footswitch and can be connected using ordinary guitar cables. The FV-300L is a sturdy stereo volume pedal that has low-impedance and is perfect for keyboards or for connecting after effects units. Table 2 describes the functions that can be assigned to the controllers.[13]

TABLE 2

GI-20 MIDI INTERFACE CONTROLLER'S FUNCTIONS

| Controller | Function | Author Setup |
|---|---|---|
| Expression Pedal | CC#1-31, 64-95,Pitch Bend 1-6,Vibrato 1-4 | CC#7 (Main Volume) |
| Foot Switch | CC#1-31, 64-95, Patch Up, Patch Down, Octave up, Octave Down, GR Hold 1-3,Glide Up 1-8, glide down 1-8,Vibrato 1-4 | H1 (Sustain) |
| GK Volume | CC#1-31, 64-95 | CC#95 (Phaser Depth) |
| S1/S2 Switch | Patch Up/Down, Octave Up/Down | oC (Octave Up/Down) |

One of the most important controllers for my live performance setup is the FCB 1010 MIDI Foot Controller from Behringer. Its main features are: [14]

- 10 banks with 10 user-editable presets each.
- Two expression pedals with freely assignable MIDI channel, controller number and range.
- Five MIDI program change and two MIDI controller commands per preset (simultaneously transmittable).
- MIDI note-on commands transmittable for trigger and tap-tempo applications.
- Two programmable relay-controlled switch jacks, e.g. for amp channel selection.
- MIDI merge and SysEx backup functions.

[13] Roland Corporation, *GI-20 GK-MIDI Interface Owner's Manual* (Los Angeles, Calif., Roland Corp, 2003), 20.

[14] Summarized and adapted from the following webpage: http://www.behringer.com/FCB1010/index.cfm?lang=ENG.

I usually use the MIDI guitar and the MIDI keyboard controller in the studio for all music productions activities and the FCB 1010 MIDI Foot Controller for electro-acoustic live performances, in conjunction with his SGPI. *See DVD Chapter 2, ex. 3.* Table 3 shows the MIDI system signal flow used in this thesis project.

TABLE 3

MIDI SYSTEM SIGNAL FLOW

| From | To |
|---|---|
| Radium MIDI Out<br>Radium USB | Digi 002 MIDI In<br>USB Hub-Mac |
| MIDI Guitar GK Out | GI-20 GK Input |
| GI-20 MIDI Out | Roland XV 5050 Sound Module MIDI in |
| Roland XV 5050 Sound Module MIDI Thru | Digi 002 MIDI In |
| FS-5L/5U Foot Switch Out | GI-20 Footswitch In |
| FV-300L Volume/ Expression Pedal Out | GI-20 Expression Pedal In |
| FCB 1010 MIDI Foot Controller MIDI Out | Digi 002 MIDI In |

Next, several ideas and suggestions about the use of the MIDI guitar are briefly described. The performer is able to control a low frequency oscillator (LFO) with an expression pedal or map Pitch Bend messages to change vibrato depth (e.g. as the performer bends a string farther, the vibrato gets deeper and faster). Regarding Pitch Bend settings, the performer must be sure that the guitar-to-MIDI converter's Pitch Bend range matches the synthesizer or sampler's Pitch Bend range. In relation to the transposition feature, most MIDI sequencing software allows the performer to transpose incoming MIDI data in real-time, either by MIDI channel, by track, or (as in this particular case) within the GI-20 MIDI Interface via the transpose function. In this way, the performer is able to create interesting harmonization effects by blending the sound of his guitar with the synthesizer transposed to a unique interval. Additionally, assigning multiple destinations to a MIDI Track is an excellent technique for orchestration and sound design using virtual instruments and samplers. *See Appendix A, ex. 1, and DVD Chapter 2, ex. 4.*

In brief, the final experience related to input MIDI data via MIDI guitar involves several factors such as how the pick up is installed, the proper right hand position in relation with the pick up, what kind of articulation has been used and which are the sensitivity and guitar feel settings, etc.

The keyboard controller functions well for producing complex polyphonic performances based on notes and dynamics, but not so well when expression is required. On the other hand, the guitar controller works well for simple polyphonic performances using notes, dynamics and expression, but is limited to only six notes simultaneously.[15] The performer's musical criteria should decide in which situations are more appropriate to use a particular controller based on the final aural result.

## Extending the Current Guitar Production Techniques

"Reamping" consists of recording a direct guitar with no amplifier. Then the engineer sends the direct track out to an amplifier and records it to a new track. This technique is extremely useful because the engineer is not only able to change amplifier

[15] Martin Russ, *Sound Synthesis and Sampling*, 2d ed. (Amsterdam: Focal Press, 2004), 349-350.

and microphone configurations until the perfect tone is achieved, but is also capable of capturing the room's ambience. During the production process the guitarist will probably need a direct box for recording the tracks,[16] and then the engineer uses an interface box to convert the line-level output of the recorder back to instrument level to feed an amplifier. Another choice is to send the direct guitar track through a modeling processor (instead of sending the direct guitar track to an amplifier), and record the processor back to a new track.[17] The performer is able to exploit the reamping technique using one or multiple amplifiers modeling plug-ins.[18]

Sometimes recording a guitar through several different amplifiers and cabinets simultaneously is the answer to get that particular sound. Through devices like the Radial JD7 Injector (Direct Input (DI), Signal splitter, and re-amp kit) or the LittleLabs guitar splitter box, the performer has the option to drive the guitar signal through as many as seven amplifiers simultaneously without loss or coloration. Consequently, the guitar player is able to combine amplifiers, effect pedals and modeling devices in order to create fresh new sounds. This analog technique can be emulated in the DAW environment if the performer sends the direct guitar track through several

[16] Huber and Runstein, 121. The direct box serves to interface an electric instrument to the audio console reducing an instrument's line level output to mic level for direct insertion into the console; changing an instrument's unbalanced line to a balanced signal and electrically isolating audio paths. DI examples: Radial, Manley, Groove Tubes, Summit Audio, Avalon, etc.

[17] In a simplified way, modeling processors are hardware devices that emulate several guitar amplifiers, cabinets and stomp boxes. For example, Line 6 POD, Zoom, Alesis, Korg, Digitech, Vox etc.

[18] Such as Line 6 Amp Farm, Native Instruments Guitar Rig, TC Electronic TC Thirty, Bomb Factory SansAmp, IK Multimedia Amplitube or the brand new Waves GTR.

amplifiers modeling plug-ins at the same time. An overview of guitar production techniques used by internationally recognized producers are summarized in Table 4.[19] Table 5 describes several microphones and production techniques. In addition, a few examples about DSP applied to guitar mixing ideas are added in the continuation of Table 5.

TABLE 4

GUITAR PRODUCTION TECHNIQUES: ACOUSTIC GUITARS

| **Producers** | **Topic: Acoustic Guitars** |
|---|---|
| **Bill Bottrell**[20] | Selected microphones: Telefunken 251s, KM 54s and the tube version of the KM 84s. |
| **Chuck Ainlay**[21] | Selected microphones: Neumann KM84 or the tube version of the KM56 or 54 via Focusrite modules. He determines the microphone distance and placement based on the guitar brand and playing technique. |
| **Julian King**[22] | Selected microphone: Gefell (UM 70S). He records via direct out, mic and then process the direct out through an Amp Farm. |
| **Elliot Scheiner**[23] | Selected microphones: Shure SM81 for strumming rhythm parts and a tube U47 for picking parts. Compressors: Summit TLA-100 and dbx 160-SL |

[19] Maureen Droney, *Mix Masters: Platinum Engineers Reveal Their Secrets for Success* (Boston: Berklee Press, 2003). All the producers credits included in this table are extracted and summarized from the aforementioned source.

[20] Credits: Sheryl Crow, Thomas Dolby, Elton John, Madonna, Michael Jackson, etc.

[21] Credits: Dire Straits, Mark Knopfler, Steve Earie, Vince Gill, Waylon Jennings, etc.

[22] Credits: Faith Hill, Toby Keith, Tim McGraw, John Anderson, Randy Travis, etc.

[23] Credits: Queen, The Eagles, Fleetwood Mac, Steely Dan, Toto, Natalie Cole, etc.

## TABLE 5

## ELECTRIC GUITAR: MICROPHONES AND PRODUCTION TECHNIQUES

| **Producers** | **Topic: Electric Guitars: microphones and production techniques.** |
|---|---|
| **Dave Jerden**[24] | He uses a SM57 angled in at 45 degrees coming between the center and the edge of the cone. Triple Tracking concept: one dedicated amplifier for each frequency range; low, mid and high through the Lucas Deceiver (active splitter). |
| **Steve Marcantonio**[25] | Royers, Neumann and the SM57 are his microphone's choices. Compressor: Fairchild by Universal Audio (UA) |
| **David Thoener**[26] | Interesting concepts: No Eq, No compression. He proposes-listening to the amplifier, thinking about the song and trying to create a guitar sound that captures the spirit of the part. |
| **Neil Dorfsman**[27] | He records a DI and then reamplified those tracks. Then, he compress the signal via LA2A, 3A or 1176 (Leveling amplifiers designed by UA. |
| **Alan Moulder**[28] | He uses filters more than Eq. Besides that, he does multi-reamping, routing one side of the Digitech 2112 stereo preamp to a tube preamp and the other side to a solid-state unit. Then he sends them to a Messa Boogie pre amp and out to a pair of Boogie 4-by-12s. |
| **Brian Malouf**[29] | He does a lot of subtractive EQ, panning, and compression via UA units. In addition, he records guitars using the POD.[30] |
| **Dave Way**[31] | Essentially, he uses POD to get good distortion tones. |
| **Dave Pensado**[32] | He uses the Korg A1 for guitar effects and several Waves plug-ins. |

[24] Credits: Rolling Stones, Red Hot Chili Peppers, The Offspring, Herbie Hancock, Brian Eno/David Byrne, etc.

[25] Credits: Alabama, Deana Carter, Willie Nelson, Mark Chesnutt, Roseanne Cash, etc.

[26] Credits: AC/DC, Aerosmith, Santana, John Mellencamp, k.d. lang, etc.

[27] Credits: Laurie Anderson, Bob Dylan, Kiss, Sting, Billy Idol, Paul McCartney, Bobby McFerrin, etc.

[28] Credits: U2, Nine Inch Nails, Moby, BT, Marilyn Manson, The Smashing Pumpinks, etc.

[29] Credits: Queen, Pearl Jam, Madonna, Macy Gray, Lisa Loeb, Sarah McLachlan, etc.

[30] The POD is the ultimate guitar recording/performance tool that integrates the Line 6's modeling and built-in stereo effects.

[31] Credits: Christina Aguilera, India Arie, Michelle Branch, Foo Figthers, Mick Jagger, etc.

[32] Credits: Take 6, Pink, Ice Cube, Brian McKnight, Black Eyed Peas, etc.

"TABLE 5—*Continued.*"

| **Producers/Engineers** | **Topics: DSP applied to guitars and mixing ideas.** |
|---|---|
| **Bruce Swedien**[33] | He uses delays time that goes from 12 to 30 ms with hard panning, or 25ms on one side and 50ms on the other |
| **Don Smith**[34] | He applies 8:1 or 10:1 compression ratios with the attack and release timed to the song's pulse. |
| **Benny Faccone**[35] | 4:1 to 8:1 are his compression ratios and the Lexicon PCM 42 is one of his favorites reverb units. |
| **Jerry Finn**[36] | He just compresses clean guitars, probably using the dbx 160SL stereo compressor limiter. Hard left and right panning. |
| **Andy Johns**[37] | Smart concept: Instead of using Eq, he suggests to change the guitar and/or the amp until the guitar sound works properly with the track. Besides that, he uses the EMT 140 plate reverb from TC Electronics for several purposes. |
| **George Massenburg**[38] | He compresses guitars, then adds reverb and gate the result of that process. Related to the use of effects he is always trying to invent everything from scratch. |
| **Ed Seay**[39] | He applies the curving eq technique and pan as if he was sitting in the audience, giving each musician a place on the stage. |
| **Allen Sides**[40] | He likes tube stereo plates. Specially, non linear reverbs for guitars, setting a 4 sec reverb with a 10 ms pre-delay and equalizing the left and right side differently. |

Table 6 contains several signal routing and setup options specifically designed by myself for the extended stereo guitar. *See DVD Chapter 2, ex. 5-8.*

[33] Credits: Count Basie, Tommy Dorsey, Duke Ellington, George Benson, Nat "King" Cole, etc.

[34] Credits: Tom Petty, Talking Heads, The Eurythmics, Rolling Stones, U2, etc.

[35] Credits: Sting, Ricky Martin, Luis Miguel, Tony Braxton, etc.

[36] Credits: Green Day, Rancid, Goo Goo Dolls, Beck, etc.

[37] Credits: Led Zeppelin, Rolling Stones, Van Halen, Traffic, Blind Faith, etc.

[38] Credits: Billy Joel, Kenny Loggins, James Taylor, Randy Newman, Toto, etc.

## TABLE 6

## EXTENDED STEREO GUITAR TECHNIQUES: SIGNAL ROUTING AND SETUPS OPTIONS

| From Stereo Guitar | | To/From | To/From | To/From | To |
|---|---|---|---|---|---|
| **A** | Acoustic Output (AO) | | Digi 002 Rack (D002R) Ch 3 Input (Ch3 I) | | To Studio Monitors or to console and PA system |
| | Electric Output (EO) | | D002R<br>Ch 4 I | | |
| **B** | AO | Direct Box A | D002R Ch3 I | | Idem |
| | EO | Direct Box B | D002R Ch 4 I | | |
| **C** | AO | Volume Pedal A | D002R Ch3 I | | Idem |
| | EO | Volume Pedal B and/or after the DI | D002R Ch 4 I | | |
| **D** | AO | One or multiple stompboxes and/or Volume Pedal A | D002R Ch3 I | | Idem |
| | EO | Volume Pedal B | D002R Ch 4 I | | |
| **E** | AO | | Carvin AG 100D Amp (CAG100D) | D002R Ch 1 Mic Input (MI) | Idem |
| | EO | | D002R Ch 4 I | | |
| **F** | AO | | CAG100D In/Stereo Line Out (CAG100DSLO) | D002R Ch 7-8 I | Idem |
| | EO | Direct Box B | D002R Ch4 I | | |
| **G** | AO | Direct Box A | D002R Ch3 I | | Idem |
| | EO | | CAG100D In/Miked Amp (MA) | D002R Ch 1 MI | |
| **H** | AO | Direct Box A | D002R Ch3 I | | Idem |
| | EO | | CAG100D In/SLO | D002R Ch 7-8 I | |
| **I** | AO | | Peavy Amp(PAmp) In/MA | D002R Ch 2 MI | Idem |
| | EO | | CAG100D In/MA | D002R Ch 1 MI | |
| **J** | AO | | PAmp In /Line Out (PAmpLO) | D002R Ch 2 Line In (LI) | Idem |
| | EO | | CAG100D In/MA | D002R Ch 1 MI | |
| **K** | AO | | Pamp In/MA | D002R Ch 2 MI | Idem |
| | EO | | CAG100D In/SLO | D002R Ch 7-8 I | |
| **L** | AO | | Pamp In/LO | D002R Ch 2 LI | Idem |
| | EO | | CAG100D In/SLO | D002R Ch 7-8 I | |

[39] Credits: Paul Davis, Peabo Byson, Melissa Manchester, Pam Tillis, etc.

[40] Credits: Little Richard, Aretha Franklin, Natalie Cole, Brian Setzer Big Band, Phil Collins, etc.

“TABLE 6—*Continued*.”

| | | | | | |
|---|---|---|---|---|---|
| **M** | AO | | CAG100D In/MA | D002R Ch 1 MI | Idem |
| | EO | | PAmp In/MA | D002R Ch 2 MI | |
| **N** | AO | | CAG100D In/SLO | D002R Ch 7-8 I | Idem |
| | EO | | PAmp In/MA | D002R Ch 2 MI | |
| **O** | AO | | CAG100D In/MA | D002R Ch 1 MI | Idem |
| | EO | | JMP1 In/Speaker Emulator Out (SEO) | D002R Ch 5-6 I | |
| **P** | AO | | CAG100D In/SLO | D002R Ch 7-8 I | Idem |
| | EO | | JMP1 In/SEO | D002R Ch 5-6 I | |
| **Q** | AO | | JMP1 In/SEO | D002R Ch 5-6 I | Idem |
| | EO | | CAG100D In/MA | D002R Ch 1 MI | |
| **R** | AO | | JMP1 In/SEO | D002R Ch 5-6 I | Idem |
| | EO | | CAG100D In/SLO | D002R Ch 7-8 I | |
| **S** | AO | | CAG100DInput3/MA, Peavy In/MA or DI | D002R Ch 3 | Idem |
| | EO | JMP1Input/Main Outputs Left and Right Ch (MOLR) | CAG100D Input 1-2/ Miked in stereo | D002R Ch 1-2 MI | |
| **T** | AO | JMP1Input/ MOLR | CAG100D Input 1-2/ Miked in stereo | D002R Ch 1-2 MI | Idem |
| | EO | | CAG100DInput3/MA, Peavy/MA or DI | D002R Ch 3 | |
| **U** | AO | | DI | D002R Ch 3 | Idem |
| | EO | JMP1Input/Main Output Left | CAG100D Input 1/MA | D002R Ch 1 MI | |
| | | JMP1Input/Main Outputs Right | Peavy Input/MA | D002R Ch 2 MI | |
| **V** | AO | | CAG100D Input 1/MA | D002R Ch 1MI | Idem |
| | Mic 1 (In the Hole) | | | D002R Ch 3 MI | |
| | EO | | JMP1 SEO | D002R Ch 5-6 I | |
| **W** | AO | | CAG100D Input1/SLO | D002R Ch 7-8 I | Idem |
| | Mic 1 (Hole) | | | D002R Ch 3 MI | |
| | EO | | JMP1 In/SEO | D002R Ch 5-6 I | |
| **X** | AO | | CAG100D Input 1 | D002R Ch 1MI | Idem |
| | Mic 1 (Hole) | | | D002R Ch 3 MI | |
| | Mic 2 (fret 12) | | | D002R Ch 4 MI | |
| | EO | | JMP1 SEO | D002R Ch 5-6 I | |
| **Y** | AO | | CAG100D Input1/SLO | | Idem |
| | Mic 1 (Hole) | | | D002R Ch 3 MI | |
| | Mic 2(fret12) | | | D002R Ch 4 MI | |
| | EO | | JMP1 SEO | D002R Ch 5-6 | |
| **Z** | Options A to V plus two ambience microphones | Options A to V | Options A to V | Options A to V | Idem |
| | | | | Mic Inputs X | |

Once the signals are recorded into Pro Tools, the engineer is able to insert several DSP plug-ins or amplifier modeling plug-ins in each of the two audio tracks or send each signal via a bus channel to one or multiple stereo auxiliary tracks. In this manner, the engineer is capable of having several signal routing options, from direct to Pro Tools to all the A-Z variations aforementioned. Additionally, the engineer is able to combine several plug-ins options serially or in parallel, merging outboard gear or devices with specific guitar programs such as SansAmp, Amplitube and/or Guitar Rig 2, inserting and using DSP bundles, adapting analog techniques to digital and converging acoustic/electric methods, in other words, expanding the current guitar production techniques. *See Appendix A, ex. 2-3 and DVD Chapter 2, ex. 9-10.*

When composer and guitarist Steve Mackey performs live or records with a traditional orchestra, he conceives of the electric guitar as a member of the orchestra. [41] For that purpose, he uses a specially designed spherical speaker system and an augmented guitar. Essentially, the concept is to blend the electric guitar sounds and the acoustic sounds in a natural and effortless way. In order to adapt this notion to the electro-acoustic music production process, the engineer can pan the stereo guitar between 32–48 (Left) and 96–112 (Right)—on a 127 range or 50-50 on a 100 range—and experiment with the amount of reverb as well as the effect placement. The objective is to find the proper spot in the stereo field and to give the adequate front-rear dimension, constantly evaluating how the stereo guitar blends and fuses with the virtual symphonic orchestra.

[41] Curtis Bahn, Dan Trueman and Perry Cook, “Alternative Voices for Electronic Sound,” *The Journal of the Acoustical Society of America* 108, no. 5 (2000): 25-38.

An interesting practice is to emulate the sound of the main guitar players via amplifier modeling technique and DSP (e.g. imitating specific playing techniques, instruments, cabinets, speakers, combo amplifiers, effects units, etc). The ideal learning approach would be via a Line 6 Variax Guitar with the Workbench software that allows the composer/performer to create his own custom guitar. In that manner, the performer/composer/sound designer is able to virtually recreate the largest part of the sound chain. For instance, the guitar model could be imitated via Line 6 software, cabinets and speakers through the amplifier modeling aforementioned software, and the microphone types and placement probably via the Antares Microphone Modeler program. Of course, the most important aspect is to be able to play like Hendrix, Clapton, Joe Pass, or John Scofield but that is something for which there is currently no plug-in. The musician must have skills, talent and a unique vision because it is not enough to merely have the same setup (hardware or software versions). Nevertheless, in order to develop the performer/composer's artistic personality and following the Renaissance apprentice learning method, I encourage the imitation of real world models, learning by doing via critical listening and intense research. *See DVD, Chapter 2, ex.11-14.*

## Digital Signal Process and Audio Plug-ins

The most important signal-processing operations are dynamic range alteration, filters, convolution, time delay effects, and time/pitch changing. Another class of signal processing operations is related to sound analysis techniques such as pitch detection, rhythm recognition, and spectrum analysis.[42]

In the digital domain, all the signal-processing operations are carried out by digital signal processors—a computer processing chip which is optimized for dealing with computations on numbers— with the help of several plug-ins.[43]

*Pro Tools plug-ins formats and basic characteristics*

There are four types of plug-ins that can be used with Pro Tools software systems: Time Division Multiplexing (TDM), Host TDM (HTDM), Real-time Audio Suite (RTAS) and AudioSuite. In Pro Tools LE systems only RTAS and AudioSuite plug-ins are available. Basically, RTAS plug-ins works in real-time. The musician inserts these plug-ins into Pro Tools mixer channels to provide instant effects. RTAS plug-ins function as pre-fader inserts meaning that their input levels are not affected by a track's volume fader. [44]

Real-time plug-ins are available as in-line inserts on audio tracks, auxiliary inputs, and master faders. If multiple inserts are used on a track, they are processed in series, from top to bottom in the Mix window. Inserts can be used in two ways: on single tracks or auxiliary inputs. With in-line inserts, the musician controls the level of effect by adjusting the controls of the plug-in. As well, an insert can be used as a shared resource in a send-and-return arrangement by bussing signals from several tracks to an auxiliary

[42] Curtis Roads, *The Computer Music Tutorial,* 2d ed. (Cambridge, Mass.: MIT Press, 1996), 390.

[43] Russ, 409.

[44] Except on Master Fader tracks, where inserts are post-fader.

input, and applying the insert to the auxiliary input track. In this manner, the musician is able to control the send level for each track and the overall level of the effect can be controlled from the auxiliary input track. Additionally, RTAS can be used in mono, multi-mono, or multichannel formats, depending on the type of plug-in and whether the destination is a mono or multichannel track. When using an RTAS instrument plug-in (such as Native Instruments FM7) on an auxiliary input, the instrument must be the first plug-in in the signal flow, then other RTAS plug-ins may be inserted. *See DVD, Chapter 2, ex.15-17*.

With AudioSuite plug-ins, the musician needs to first-select an audio region in the Pro Tools edit window then choose the desired plug-in from the Audiosuite menu. Pro Tools processes the audio via the computer CPU creating a new file on disk so the musician has the option to replace the original file or use it alongside the original file. AudioSuite plug-ins differ from TDM and RTAS plug-ins in that they are used to process and modify audio files on disk rather than non-destructively in real-time.

DigiRack is a set of basic plug-ins provided by Digidesign for each Pro Tools system that contains several options such as 4-Band EQ, Chorus, Flange, Delay Expander-Gate, Dither and DigiReWire. In order to understand how Pro Tools categorizes its Audiosuite and RTAS plug-ins, see Table 7.

After the musician installs a new plug-in, the host software will automatically recognize and categorize it. In addition, there are additional plug-ins included with Digi 002 and Digi 002 Rack Systems, such as Digidesign Maxim, Digidesign D-Fi, Lo-Fi, Sci-Fi, Recti-Fi, Vari-Fi. Furthermore, there are two more plug-ins packages included in the new bundled software: Free Bomb Factory Plug-ins and the Pro Tools Ignition Pack™. The Free Bomb Factory package contains a Compressor, a Clip Remover, a Correlation Meter Bridge, a Noise Meter, a Tuner, and the Funk Logic Mastererizer. Pro Tools' Ignition Pack™ includes Ableton Live Lite 4, Reason Adapted, IK Multimedia Bundle, and the Celemony Melodyne software.

TABLE 7

PRO TOOLS AUDIOSUITE AND RTAS PLUG-INS CATEGORIZATON

| Audiosuite | RTAS Multi-channel | RTAS Multi-mono |
|---|---|---|
| EQ | EQ | EQ |
| Dynamics | Dynamics | Dynamics |
| Pitch Shift | | *Pitch Shift* |
| Reverb | Reverb | Reverb |
| Delay | Delay | Delay |
| Modulation | Modulation | Modulation |
| Harmonic | Harmonic | Harmonic |
| Dither | Dither | Dither |
| Sound Field | Sound Field | Sound Field |
| Other | *Instrument* | *Instrument* |
| | Other | Other |

Clear examples of third party plug-ins are the Free Bomb Factory Plug-ins and Pro Tools Ignition Pack™. In short, the musician has the option to expand his plug-ins collection via optional Digidesign plug-ins, Digidesign distributed plug-ins, Digidesign virtual instruments, third-party plug-ins, software samplers, or creating his own plug-ins via Cycling 74 software.[45]

Considering that the plug-in market is expanding on an almost daily basis and each performer/composer has a different set up, it is beyond the scope of this thesis to try to cover all the available options. A procedure that is particularly effective, however, is plug-in automation. This allows the performer/composer to build up complex automation in stages, controlling one or two plug-in parameters at a time and creating a more dynamic sonic treatment. *See Appendix A, ex. 4 and DVD Chapter 2, ex. 18.*

Below are descriptions of Practical Audio Plug-in Techniques and Examples.

### <u>Side-Chain Processing</u>

Side-chain processing allows the performer to trigger a plug-in from a separate reference track or external audio source. The source used for triggering is referred to as the Key input. A classic use is to control the dynamics of one audio signal using the dynamics of the key input. For example, a rhythm guitar track could be used to gate a keyboard pad. *See Appendix A, ex. 5, and DVD Chapter 2, ex. 19.*

[45] Essentially, virtual instruments are software emulations of 'real' instruments such as Hammond organs, Rhodes pianos, Moog synthesizers as well as samplers, drum machines, and almost any kind of audio signal processor. These software simulations are called virtual instruments because they are constructed using computer code rather than real hardware.

### Key Input Filters

Some plug-ins feature key high-pass and low-pass filters. These controls let the musician define a specific frequency range in the key input signal with which to trigger the plug-in effect. A common production technique is to employ these controls to filter a drum track so that only specific high frequencies (e.g. a hi-hat or low frequencies (e.g. a tom or a kick) trigger the effect.

Another smart plug-in technique is learning how to use plug-ins on the way in. This means inserting a plug-in in a stereo auxiliary track and setting the guitar signal as an input. After that, the engineer needs to send the signal via bus and receive it as an input in a stereo audio track. Therefore, the musician has the opportunity to modify the direct signal before to be recorded. This option could be very helpful in recording acoustic guitars with strong and pumping strumming patterns. In this manner, the engineer is treating plug-ins like external hardware devices such as a dbx compressor unit. *See Appendix A, ex. 6, and DVD Chapter 2, ex. 20.*

### Multiple outputs for instrument plug-ins

Performers may route the output of an instrument plug-in to the main track outputs and to individual track inputs on other tracks. Thus, the musician can use a single instrument insert to route multiple samples or synthesizer sounds through separate outputs for individual processing. When a plug-in that has both main and auxiliary outputs is inserted on a track, the plug-in's main outputs are routed through the track on which the plug-in is inserted, and the plug-in's auxiliary mono or stereo outputs are available in the track input pop-up menu on other tracks. *See DVD, Chapter 2, ex.21.*

Plug-ins benefits:

The use of plug-ins instead of hardware devices has several advantages. Commonly, plug-ins works in 24 or 32 bits; whereas, first, second and third generation hardware samplers and synthesizers work in 8 to 16 bits. In addition, the user is able to save unlimited presets in software. Most hardware devices, however, have a limited amount of presets available. Regarding automation, the user is able to automate all the performance movements, record, edit, and save the best takes in real-time. Instead, using hardware devices implies that the producer has to work conditioned by the unit's memory and/or the engineer's skills (e.g., if a mix is inadvertently not saved, the whole recording process must start again). As well, the assistant engineer needs to write all the setups and settings on paper or save them as a text file for future recall. Another advantage is that the engineer may have multiple instances of a particular virtual instrument in the DAW environment (e.g. a different reverb unit preset per track running at the same time). On the hardware side, the studio owner would need to buy one external reverb unit per track or instrument.

Additionally, there at least four more benefits: portability, maintenance, space and price. Through an Ilok device, the user is able to move the plug-ins' authorizations from desktop to laptop computers.[46] Physical devices require transportation, additional space, and are expensive. Finally, one of the most important options concerning plug-ins creativity is that the musician is capable of designing custom applications for specific purposes.

## ReWire: Definition and Examples

ReWire is a software protocol developed by Propellerheads in 1998[47] that transfers audio data between two computer applications in real-time. Moreover, ReWire 2's[48] technology offers the option of streaming in real-time up to 256 individual audio channels and 4080 individual MIDI channels. In addition, there are several features such as high-precision synchronization, common transport functions and additional querying. Basically, ReWire allows two compatible programs to work together synergistically in a host-client relationship.[49] In order to make both programs work in conjunction, the musician has to first open the host and insert a ReWire compatible program. Essentially, ReWire provides virtual patch chords that connect the client's outs into the host's virtual mixer (e.g. if Pro Tools is the host and Reason is the client, Reason's instrument outs would appear in the Pro Tools mixer channels). In essence, the host takes over all MIDI

[46] The iLok is a USB smart key that can store over 100 separate licenses for participating protected software.

[47] It first appeared in Propellerheads' ReBirth RB-338 and Steinberg Soft- & Hardware's Cubase VST.

[48] Summarized from the http://www.propellerheads.se/technologies/rewire/index.cfm?fuseaction=mainframe.

[49] Some programs can be hosts, some clients and some either one.

and audio routing for the client. Besides that, the musician is able to play the client instruments from a MIDI keyboard, sending the data into the host first, which then passes it onto the client. In addition, if both applications support ReWire 2 the musician is able to route MIDI from the host application to the client. This means that any sequencer track in the host application can play a sound source in the client application. In effect, the musician is able to control everything from the host program. *See Appendix A, ex. 7–12, and DVD Chapter 2, ex. 22-23.*

Moreover, the musician can use up to 64 different ReWire channels and use Reason's mixer to mix most of the devices to the "Reason Mix L/R" ReWire channel (and route a few devices directly to the hardware interface) or could route all devices to separate ReWire channels (mixing in the host application). *See DVD Chapter 2, ex. 24.*

I also discovered that in using the common Rewire setup between Pro Tools and Reason, the performer is able to record automation in a “live performance approach.” Basically, the performer needs to record into the MIDI track and simultaneously operate the MIDI Controller Keyboard knobs and faders. After that, almost all the movements are recorded as a sound Controller 1, etc. Then, the musician is able to edit the previous controller data or re-record as midi and audio at the same time in order to capture those unique performances ideas.

### *How to use ReWire with MSP*

Musicians are able to use MSP as a ReWire device via the ad_rewire driver. When this driver is selected, audio from MSP can be routed to any of 16 inter-application ReWire channels which will appear as inputs in ReWire host applications. MSP can also be used as a host application for ReWire devices such as Reason using the *rewire~* object. "Objects" are computer algorithms that precipitate actions. Essentially, Max/MSP works with three types of objects; the *standard* objects contained in the default library, the *custom* objects (created by grouping several standard objects) and the *external* objects (new objects programmed in C).

The `hostsync~`, `hostphasor~`, and `hostcontrol~` MSP objects can work in conjunction with the ad_rewire driver in order to provide synchronization, transport, and tempo information to and from ReWire devices and control the host's transport. For example, the musician can choose one of the Max ReWire MIDI ports as a MIDI output device in Pro Tools and then use standard Max MIDI objects (such as `notein`) to control Max/MSP created synthesizer. Similarly, the musician can send MIDI into Pro Tools using the max MIDI objects and the ReWire MIDI ports, and record the results to a track for further manipulation or playback.

# CHAPTER III

## Max/MSP-MUSIC PRODUCTION SOFTWARE INTERACTION

Max/MSP is a graphical programming environment specially designed for music, audio, and multimedia. Max includes MIDI, control, user interface, and timing objects.[50] In addition, MSP is a set of audio processing objects built on top of Max that offers a wide range of functions, from interactive filter design to hard-disk recording. As well, there is a set of matrix data processing objects optimized for video and 3-D graphics called Jitter that works in conjunction with Max/MSP.

### How Max/MSP handles MIDI

In order to transmit and receive MIDI data to and from the existing MIDI system ports, Max uses various built-in objects. All the complexity of the MIDI messages' structure can be simplified in Max using particular MIDI objects. Essentially, those objects convert all the critical data into integers, managing the input and output from Max, and separating the specific data for the MIDI stream. Then Max repacks the final output in the form of complete MIDI messages before sending the data out as a MIDI message.

Max handles continuous controller input and output through the `ctlin` and `ctlout` objects. The `ctlin` object receives controller data—MIDI channel, controller

[50] In a simplified way, objects are Max's central building blocks.

number and controller value —which can then be remapped in Max using the incoming values to control several functions within the program. For example, various synthesis parameters or the computer's tempo could be altered or modified using the modulation wheel.

For each MIDI message there are several specifically designed objects. For example, pitchbend messages are managed by the `bendin` and `bendout` objects, after touch messages are handled by the `touchin` and `touchout` objects, and program change messages are controlled by the `pgmin` and `pgmout` objects. In addition, the `midiin` and `midiout` objects, receives and sends the entire MIDI stream respectively. Other specialized MIDI input objects are `polyin` for polyphonic key pressure data, `rtin` for real-time messages, and `SysEx in` for system exclusive messages.[51] *See DVD Chapter 3, ex.1.*

## How Max/MSP manages Audio

If we consider that a digital representation of sound is just a list of numbers, any list of numbers can theoretically be considered a digital representation of a sound. In order to be audible as sound, the numerical values must oscillate at an audio rate perceptible to the human's ear. Therefore, the performer/composer can have an aural feedback of any such list by sending the numbers to a DAC where they are converted to

[51] For additional information please check the Max 4.5 Tutorials and Topics pdf file, located in the documentation folder of your Max/MSP folder.

voltages.[52] This is the basic starting point of computer sound synthesis. Once the sound is digitalized—whether it was quantized from a "real world" sound or it was synthesized by the computer—it becomes a sequence of numbers. For that reason, any arithmetic operation performed with those numbers becomes a form of basic digital signal processing (DSP).

Although the Max and MSP objects are found in the same program, the inter-communication between MSP objects is conceptually different than the Max objects. Rather than establishing a path for messages to be sent, MSP connections establish a relationship between the connected objects, and that relationship is used to calculate the audio information necessary at any particular instant. This configuration of MSP objects is known as the "signal network."

Max/MSP uses both control and audio rates. Some MSP objects are specifically created to provide a link among Max and MSP, and to translate between the control rate and the audio rate. These objects are very important because they give Max and the performer/composer, direct control over the signal network activities. In brief, Max/MSP enables the performer/composer to:

- Design numerous programs for controlling MIDI synthesizers, samplers, and effects processors.
- Create several digital audio device designs and computer musical instruments.
- Build an instrument that accomplishes a specific task not readily available in his setup.
- Establish the relationship between gestural control and audio result.
- Integrate audio processing into his composition or performance programs.[53] *See DVD Chapter 3, ex.2.*

[52] In this context, DAC means digital audio converter object.

## Sending Audio from Max/MSP to Pro Tools via ReWire

Max/MSP is able to route its signal to other software. This is an extremely important feature because it allows the composer to merge different programs and take advantage of the best characteristics of each of them, not only enriching the final artistic product but also defining and expanding the music production techniques. The sonic personality of my aesthetic perspective is based on his stereo guitar performance interface and the guitar application toolkit. Both of these factors were developed in the Max/MSP environment. All music production work, however, is done in Pro Tools (with the exception of the guitar tracks). The logical and natural solution is to send the signal from the Max/MSP audio outputs to Pro Tools via ReWire. Essentially, all the work done in the GATK, such as synthesis of new sounds by mathematical procedures, application of digital signal processing techniques to audio signals and development of real-time compositional strategies, can be routed to Pro Tools or any other program that supports the ReWire technology. *See Appendix A, Chapter 3, ex.1, and DVD Chapter 3, ex.3.*

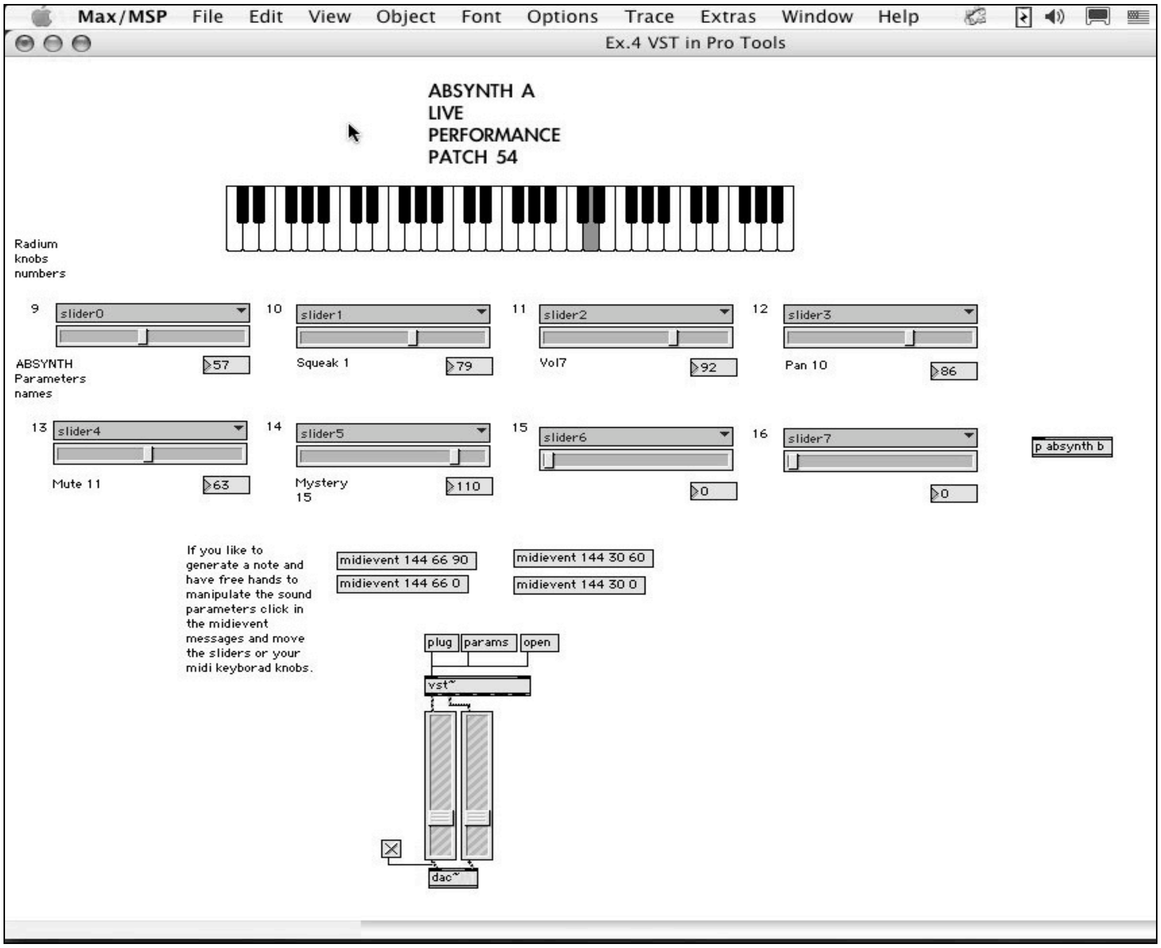
## How to use VST Plug-ins in Pro Tools via Max/MSP

There are many innovative solutions to a series of old and serious music production problems. One would think the computer would make the whole process of audio production much simpler. This is not however, always the case given proprietary

[53] For additional information please check the MSP 4.5 Tutorials and Topics pdf file, located in the documentation folder of your Max/MSP folder.

software, competing operating systems, and a wide range of other problems. For example, Pro Tools users cannot take advantage of the various VST plug-ins alternatives because this format is not supported by the software. In addition, Pro Tools does not allow the user to choose his own controller devices because the program only works with its own control surfaces. Furthermore, the number of parameters that can be controlled and automated with Digidesign devices is limited and the mapping process is complex.

In order to be able to use, control, and automate VST plug-ins in Pro Tools I designed my own VST plug-in controller interfaces. With the intention of working with just one active window, the interface is organized in the same way that appears in the MIDI keyboard controller and each `bpatcher` object is labeled with the corresponding VST plug-in parameter. In this case, I am using the Absynth plug-in from Native Instruments. This application has two patches: (1) Absynth A and (2) Absynth B. Absynth A allow the performer/composer to control the VST plug-in parameters in real-time via each of the eight MIDI keyboard controller knobs. Also, this patch extends the control over the hold notes via two `midievent` messages instead of playing and controlling parameters simultaneously. (See Figure 1).



If you like to generate a note and have free hands to manipulate the sound parameters click in the midievent messages and move the sliders or your midi keyborad knobs.

Fig. 1. VST Plug-in Controller interface v1, Absynth A patch.

The second patch, Absynth B, is embedded into the `patcher` object and has one line object per parameter with a stop message on top of each of them. (See Figure 2).The goal is to perform and control the Absynth parameters in real time with the Absynth A patch, and use the patcher Absynth B as a second generative option. In this way, the performer can choose two different presets form the same virtual instrument and use them in patch A or B, or directly use two different VST plug-ins. The audio signal is then routed from Max to Pro Tools via ReWire and the performance is recorded into it. *See Appendix A, Chapter 3, ex.2, and DVD Chapter 3, ex.4.*

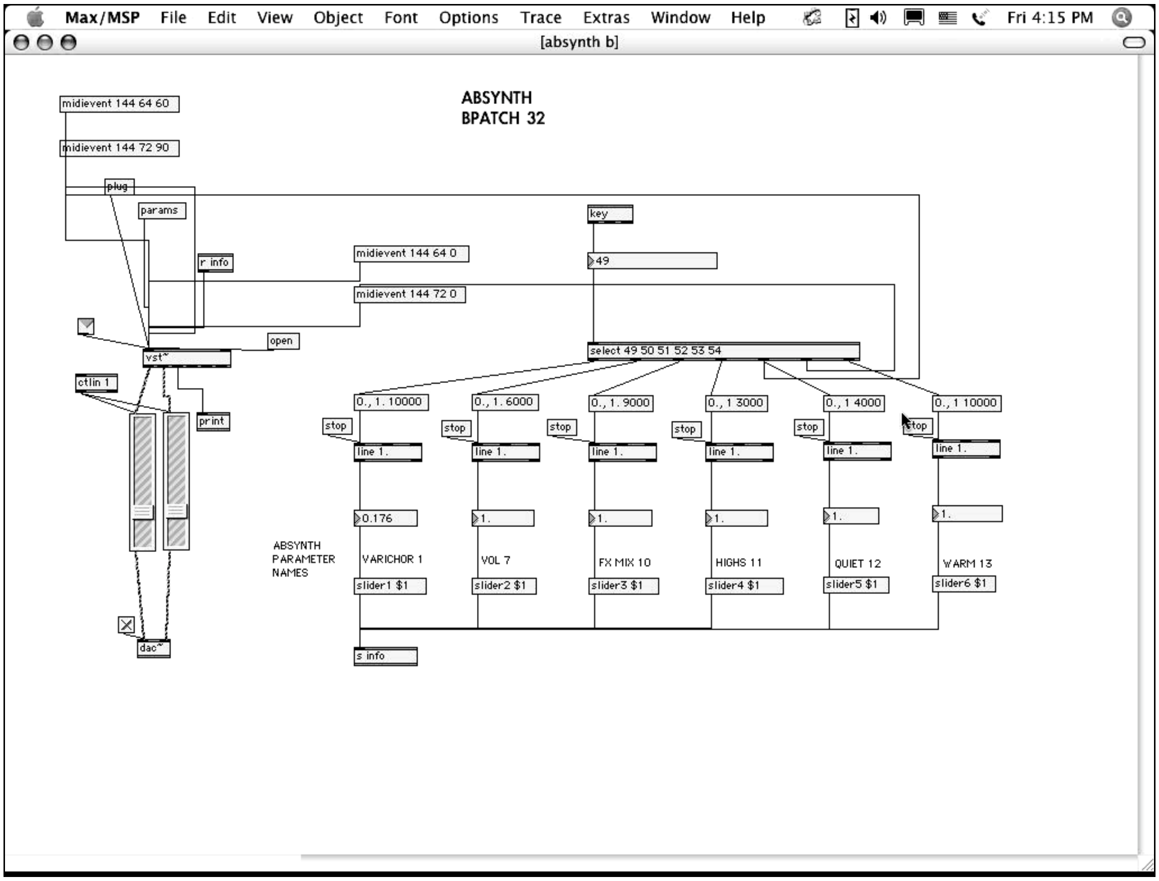


Fig. 2. VST Plug-in Controller Interface v1, Absynth B patch.

If the software instrument is extremely complex, it is very useful to design a reference chart per ensemble that includes each parameter number and the actual function. In that way, with just one template per Reaktor “ensemble,” the composer is able to control the internal parameters (50 to 100 per ensemble ) of each of the instruments contained in that particular ensemble. For these particular cases, I designed the VST Plug-In Controller Interface v2. (See Figure 3).

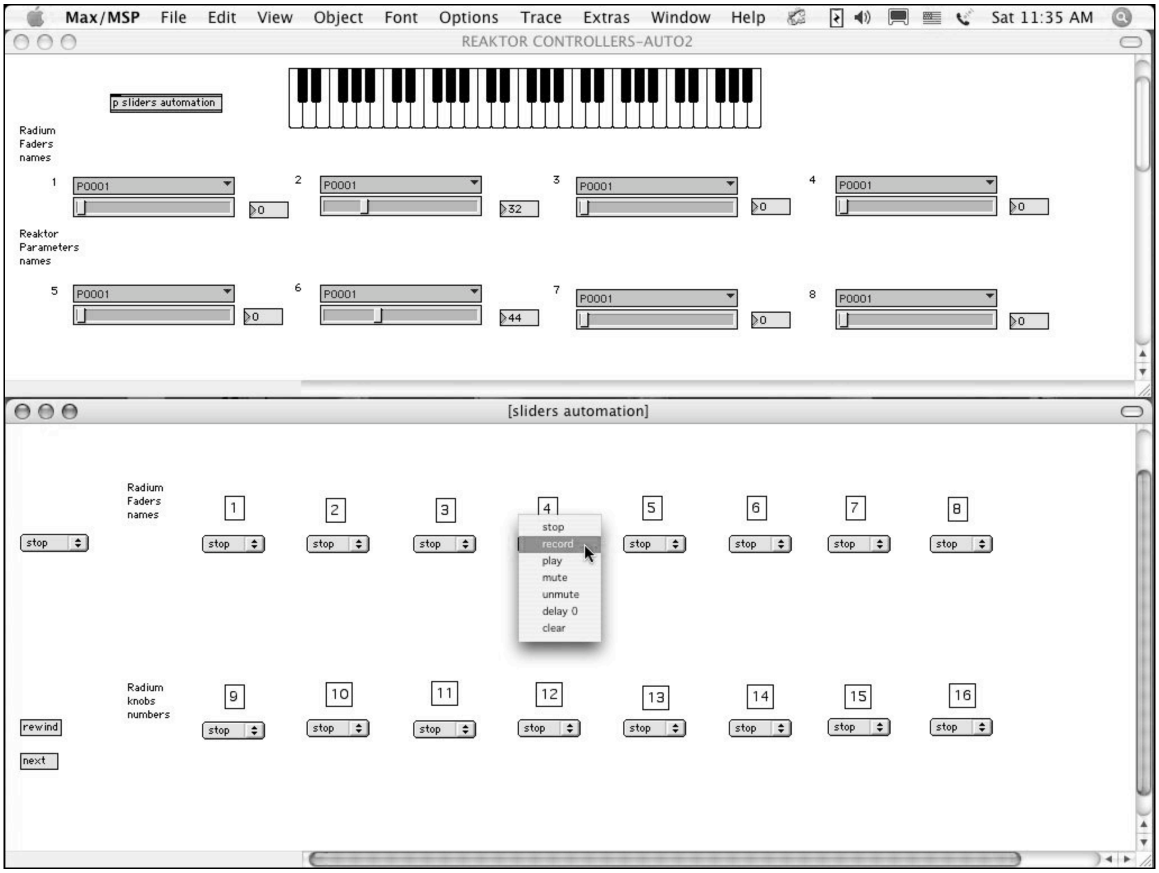


Fig. 3. VST Plug-in Controller Interface v2. Reaktor patch. Above, First 8 bpatchers; below, Sliders automation patch.

In order to have more manual control of the parameters offered by the Reaktor 4 Nanowave ensemble, I added eight more `bpatcher` objects controlled by the eight MIDI keyboard controller faders having a total of 16 freely assignable parameters to control and manipulate in real time. The performer is also able to record automation in each of the parameters through the sliders automation patcher. *See Appendix A, Chapter 3, ex.3, and DVD Chapter 3, ex.5.* Moreover, if the musical idea demands more control, the designer is able to assign the same controller number to multiple parameters (i.e., one fader can control three or more Reaktor parameters). This method provides an option to reassign controls and expand the functionality of the physical MIDI controller limitations.

Another mode to maximize the previous techniques is through the use of freeware or shareware VST plug-ins. These types of applications each have unique and different approaches to a specific topic. They offer friendly interfaces or interesting modes to perform everyday tasks. As and added benefit, the composer can contact the programmer directly, establishing a new relationship, sharing ideas and helping the freeware or shareware user's community to grow together. *See Appendix C, ex.1.*

Learning to create effective patches, front-end interfaces, MIDI controller interfaces, custom sounds, and real-time improvisational and compositional strategies are extremely important in order to be able to use the technological advances favorable to the performer/composer's musical vision.

# CHAPTER IV

## THE GUITAR APPLICATION TOOLKIT (GATK)

### Background: Interactive Computer Music

Interactive computer music is a music improvisation or composition where software interprets a live performance to affect music generated or created by computers.[54]

During the process of making music, performers respond to each other as they play, constantly varying their musical interpretation according to the sonic stimulus that they perceive from the other musicians. For that reason, music has always been an interactive art. At a certain level, many conventional musical relationships can be simulated with the computer. Moreover, computers can even partially emulate the complexity of human interactions; first, analyzing the musical information provided by the performer and second, generating a response that probably will affect the performer's future actions.

The interaction concept is based on two aspects: either the performer's actions affect the computer's output, or the computer's actions affect the performer's output.[55]

[54] Todd Winkler, *Composing Interactive Music: Techniques and Ideas Using Ma*x (Cambridge, Mass.: MIT Press, 2001), 4.

[55] Guy E. Garnett, "The Aesthetics of Interactive Computer Music," *Computer Music Journal* 25, no. 1(2001): 21-33.

For instance, performers participate in the creation of an interactive work, in some way by the amount of freedom they have to produce significant results in the computer's response. It is implied that each kind of collaboration has several levels of interaction and various approaches.

Interactions between humans and machines attempt to emulate the type of communicational options that people use regularly. The communication may range from monologue to dialogue, changing the active role of each component gradually or rapidly, modifying the participation frequency, and provoking an intelligent conversation in which each part generates interesting comments. The original idea or topic is in constant motion, keeping the discussion fresh and morphing between every opinion, agreement or contradictions. In a simplified way, we could say that in an effective conversation the vocabulary is broad, the interchange of ideas is fast and fluent, and the mood changes are unpredictable.

A real-time soundscape "conversation" created for those two people is rich and wide because they are continually changing musical dimensions. Moreover, those two people may alter pitch, dynamic range, timbre, duration, tempo, and rhythm of each word or phrase that they produce in the conversational process, simultaneously assigning significance to the generated information based on each person's cultural background.

Similarly, interactive compositions or improvisations should attempt to imitate those types of multifaceted human behaviors. Emphasizing the previous idea, Roger (Luke) DuBois writes that: "A truly interactive experience can be mediated only through creating computer response scenarios that are more fluid and less deterministic that a simple series of cues."[56]

When we discuss virtual instruments, virtual performers, virtual composers, or virtual listeners, we are describing computer simulations of models that behave like real-world objects, situations, and phenomena. The programmer's creativity and technical skills predetermine the behaviors of these virtual entities, reinforcing the concept that the key element of an interactive music system is the human input.

## Components of an Interactive System

In order to have a framework for discussing interactive systems, I included Robert Rowe's explanation: "Interactive music systems are those whose behavior changes in response to musical input."[57] Based on the previous definition and the approach taken by Todd Winkler, the five essential steps necessary to create an interactive composition plan are:

1. *Human input/instruments*. The musician's actions are translated into digital information and sent to the computer. Input could be via MIDI instruments, (keyboard, guitar, etc), through alternate controllers (such as game controllers, joysticks or MIDI foot pedals), via custom-designed sensor instruments, or through the computer keyboard, mouse, trackball, etc.

[56] Roger L. DuBois, "Applications of Generative String Substitution Systems in Computer Music" (Ph. D. diss., Columbia University, N.Y. 2003), ix.

[57] Robert Rowe, *Interactive Music Systems*. (Cambridge, Mass.: MIT Press, 1993), 1.

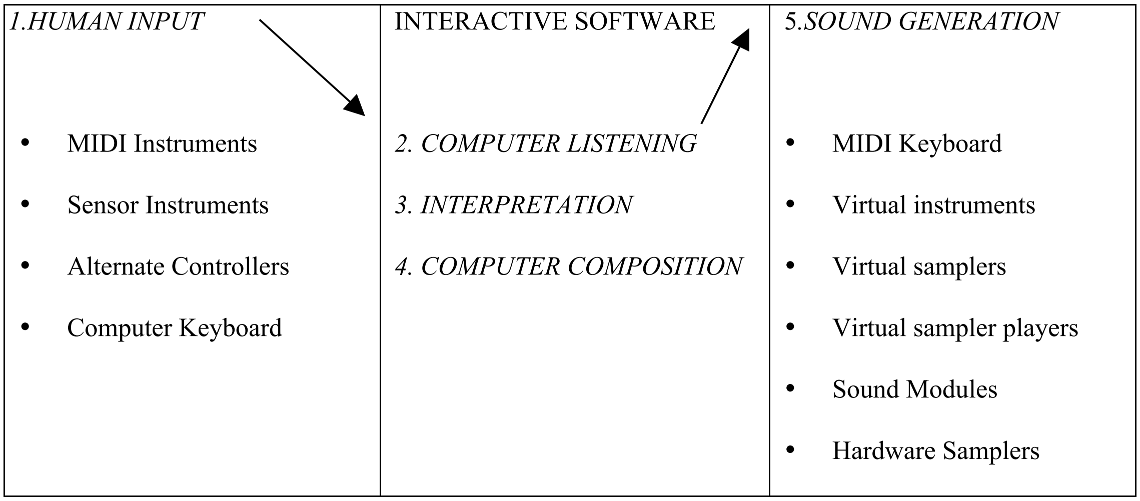
2. *Computer listening/performance analysis*. The computer analyzes several musical dimensions such as timing, pitch, dynamics, brightness, etc. For example, the Max/MSP programming environment uses the external object called analyzer.[58]

3. *Interpretation*. Software interprets the computer-listener information, generating data that will influence the composition according to the attributes and features of each object and the characteristics of the composer's patch.

4. *Computer composition*. Computer processes, responsible for all aspects of computer-generated music, are based on the results of the computer's interpretation of the performer actions.

5. *Sound Generation and output, performance*. The computer plays the music, using sounds created internally, routing the signal to virtual instruments (such as VST plug-ins), virtual samplers, virtual music libraries running via software sampler players or by sending musical information to external devices such as MIDI keyboards, sound modules, hardware samplers, etc.

Figure 4 summarizes and clarifies the structure and creation of an interactive composition.



| *1.HUMAN INPUT* | INTERACTIVE SOFTWARE | 5.*SOUND GENERATION* |
|---|---|---|
| • MIDI Instruments | *2. COMPUTER LISTENING* | • MIDI Keyboard |
| • Sensor Instruments | *3. INTERPRETATION* | • Virtual instruments |
| • Alternate Controllers | *4. COMPUTER COMPOSITION* | • Virtual samplers |
| • Computer Keyboard | | • Virtual sampler players |
| | | • Sound Modules |
| | | • Hardware Samplers |

Fig. 4. Redesign of the basic components of an interactive composition.[59]

[58] Tristan Jehan and Bern Schoner, "An Audio-Driven Perceptually Meaningful Timbre Synthesizer." *Proceedings of the International Computer Music Conference, Havana, Cuba, 2001.*

[59] Winkler, 7.

The same communicational analogy previously presented could be applied as a collective approach in which more than two performers interact with one or more computers. Furthermore, the relationship collective/computers could be regarded as a debate, forum, contest, or all the mathematical combinations between the members and the computers involved in a two-way street approach, exploring several styles and macro/micro levels of interactivity.

## Computer and Performer Interaction: Model Analysis

To create an interactive work the composer/programmer must define roles for the computer as well as the performer, creating a flexible relationship between humans and computers, one that contemplates a balance of power, fluctuating from carefully designed structures to open forms with free and spontaneous artistic expressions. Simulation of real-world models is essential for future unique designs for the digital medium. Three traditional instrumental templates are used as a model: the symphony orchestra, a string quartet, and a jazz combo.[60]

### 1. *The Conductor Model—Symphony Orchestra*

The Conductor Model implies that the composer/performer/improviser adopts the behavior of a real orchestral "conductor." A conductor has the responsibility not only of directing the timing, controlling the dynamics, and adjusting the acoustical balance, but

[60] Winkler, 21-27.

also must take care of all the macro decisions regarding interpretation. The orchestra provides its feedback in the form of musical production and facial gestures, offering useful information to the conductor who will adapt his future actions based on instantaneous analysis.

In the interactive computer model, musical content is predetermined with the performer/composer acting as conductor. Essentially, the performer provides the tempo to the computer via any MIDI device or using beat-tracking algorithms, which derive the beat from a real-time analysis of a performance. The "score-following technique" contains the performer's and computer's scores in its computer memory, matching and synchronizing the performer input to the previously stored data. Thus, the matched notes orient the computer in the composition providing exact locations and then trigger the notes or information stored in the computer score. With the intention of improving the musical result, complex systems include control of additional parameters such as dynamic, vibrato, and timbre. There are several examples of the current model such as Tod Machover's *Bug Mudra,* Max Mathews's *Radio Baton*, Gary Garnett's *Flute Fantasy,* and Pierre Boulez's *Explosante Fixe*.

### 2. *The Chamber Music Model—String Quartet*

In a traditional string quartet, four musicians influence each other's performance owing to the more democratic control distribution. This model has a complex, dynamic interaction level which may be thought of as a four-part fugue in which each voice has its own independent characteristics if analyzed horizontally. Conversely, if analyzed vertically we see a *gestalt* type of perceptional model, one in which the sum could not be possible without the complexity of the four individual parts. In a sense, the interaction level of a musical composition is determined by the control fluctuation and balance of power between the components of an interactive music system. Additionally, it is implied that the chamber music model could be applied to any kind of instrumentation and size including duets, trios, quintets, and larger.

Winkler's *Snake Charmer* is an example of this model, and the Curtis Bahn and Dan Trueman's "Interface" project is an extension and improvement of the actual concept.[61] "Interface" is an electronic chamber duo specialized in interactive computer music. Curtis Bahn plays the "SBass"—a five string vertical bass fitted with electrical pickups, motion, touch and pressure sensors which allow him to "drive" his computer during performance. Dan Trueman plays a six string electric violin with an electric bow of his own design called "RBow." The "RBow" is a normal violin bow covered with motion and pressure sensors that send information to his computer performance system. They have also begun to integrate spherical speaker arrays (which radiate sound in all directions) into their performance set-up. In summary, "Interface" creates real-time sonic environments in performance, which combine pre-composed electronic sounds with real-time digital signal processing, synthesis, algorithmic composition, and sampling.

[61] Curtis Bahn and Dan Trueman, "Interface: Electronic Chamber Ensemble," ACM CHI *Workshop in New Interfaces for Musical Expression, Seattle, April 1-2, 2001.*

### 3a. *The Improvisational Mode—Jazz Combo*

Jazz repertoire is extensive and varied, and each composition presents a formal structure and a conceptual framework in which musicians interact. In a typical performance of a jazz standard, the combo will create an introduction, then the melody will be played once or twice, followed by the beginning of the solo section. Usually, the melody is repeated after the solos until the end of the piece. What makes jazz an exciting experience for the performer and the audience, however, is not simply the recognition of the structure, but also the performer's actions in relation to the notated music; in other words—how the performer interprets the written melody, [62] reharmonizes the given progression, [63] develops a solo line, and reacts to the real-time generation of musical ideas.

On a basic level, this kind of musical intelligence can be simulated via interactive software. Effectively, computers can recognize patterns, riffs, scales, melodic motives, chord progressions, rhythmic sequences, and tempi, and using this information, a series of rules can be coded into algorithms to generate new content related to the performer material. In comparison with the two previous models, this option could demand a more serious programming work[64] but also offers numerous examples of interactivity at

[62] For example, applying several criteria of melodic transformation, such as augmentation, diminution, addition, subtraction, permutation, omission, dispersion, etc.

[63] For instance, adding chord extensions and tensions, secondary dominants, tri-tonal substitutions, modal voicings, etc.

[64] For example creating complex patches that integrates the composer's own compositional algorithms and custom abstractions, using several external objects and subpatches.

different levels, limited only by the performer's improvisational skills and the algorithm capabilities.

### 3b. *Free Improvisation*

As George Lewis[65] conceptualizes this model, neither the performer nor the computer may be "in control," but rather, each one will have some influence on how the other reacts, contributing to the final outcome of an improvisation. Basically, Lewis regards the computer as another musician—a virtual performer that is separated from and independent of the human performer. The computer does not reintroduce the collected data[66] directly, but uses it to condition and guide a separate process of real-time algorithmic composition. Consequently, in order to create an appealing interactive work, the performer/composer must be not only a proficient programmer but also a sensitive musician in order to deal with this type of open form.

In view of the preceding discussion, I assert that in an interactive music environment, a composer could assign a variety of roles to a computer such as instrument, performer, conductor, or composer. These roles may exist simultaneously or

[65] George Lewis, "Too Many Notes: Computers, Complexity and Culture in Voyager," *Leonardo Music Journal* 10, (2000): 33-39.

[66] Collected or captured data. Captured data is useful musical material and analysis information that has been stored during a performance via one of the Max's data storage objects. In addition, there are several data sources such as real time data, predetermined data, generated data, processed data and the aforementioned captured data.

change continually. The live musician may also choose one or all the aforementioned roles. (See Figure 5).

Nevertheless, in the most pure form of an artistic interaction composition the role's boundaries should not be perceptible in order to maximize the essence of the musical act and enrich the audience experience.

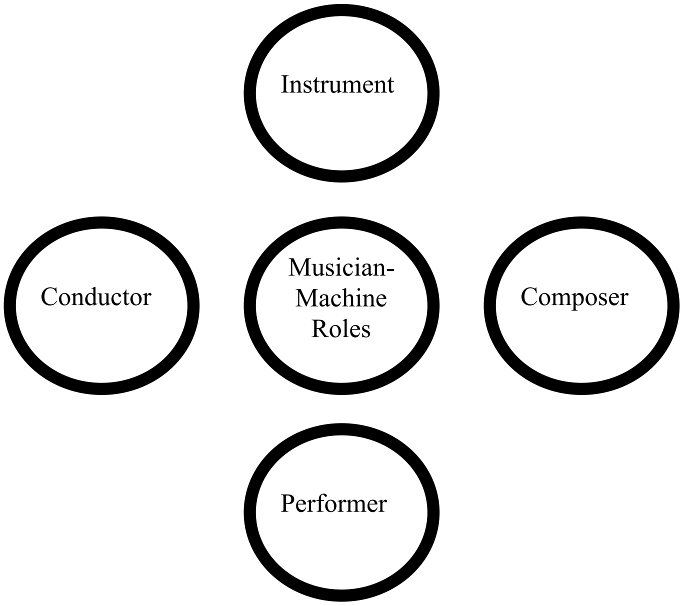


Fig. 5. Musician-machine roles

## The Guitar Application Toolkit (GATK): Concept

As stated in Chapter 1, the guitar has several limitations (e.g. sustain, envelope, dynamic range, and frequency range) that have been addressed by a variety of composers, performers, and *luthiers* during the last 50 years. These restrictions were first addressed by electric guitars through the creation of diverse pick-ups models beginning in the 1950s and 1960s, and via the implementation of active/passive guitar modules in the 1970s. Second, tube and transistors amplifiers were completely redesigned, and third, the external effects boxes options grew exponentially the 1980s. At the same time, with the purpose of having an electrified version of the acoustic guitar ca. 1975 and nylon classical guitar beginning in the 1990s, a built-in preamplifier was added.

Currently, there are two main innovations—the Carvin AE 185 Acoustic/Electric stereo guitar and the new modeling guitar implemented in the Variax Series from the Line 6. Nearly nothing, however, has been written on production techniques for stereo guitar. Furthermore, all of the present programs are designed for the traditional electric guitar with the intention of being used in the most popular scenarios such as pop, rock, alternative, heavy metal, etc. Most commercially released software is used to replace hardware "stomp" boxes and other effects units by hard dependent virtual effects pedals. These programs offer specially designed foot controller devices to control various software parameters without interfering with the performance.[67]

[67] For example, Amplitube 2 works in conjunction with the StompIO by IK Multimedia, Guitar Rig 2 is controlled by Rig Kontrol (Native Instruments) and the Waves company specially designed the GTR interface.

The goal of the GATK project is to develop several applications that expand the present “effects approach,” enhancing the guitar’s qualities and addressing the aforementioned issues. It offers the performer/composer the ability to augment his improvisational and compositional strategies through computer software interaction.

Recent production guitar techniques were improved throughout the GATK process, (e.g. by the implementation of a personalized electro-acoustic performance set up, the design of custom performance interfaces, and the creation of interactive compositional plans via the craft of unique sounds and the control of several musical parameters in real time). In order to create real-time sound environments for improvising in a performance or compositional setting, Max/MSP was used as an interactive music system. The GATK is a set of applications used to expand the guitar’s sonic spectrum, to increase the improvisational options, and to augment the compositional methods inspired and created by the Carvin AE 185 Acoustic/Electric stereo guitar. An overview of the GATK development process follows.

## Development Process

### *Planning and Implementing a Personalized Electro-Acoustic Performance Set up*

In order to emphasize the importance of the human input in the first stage of the sound generation process, we should consider the performer’s physical body and musical gestures,[68] as well as his comprehensive playing techniques,[69] and extended instruments as a complete sound generative entity. *See Figure 4, page 42, and Figure 6, page 53.*

One of the most important components of the electro-acoustic performance set up is the extended instrument AE185Acoustic/Electric Stereo Guitar. This custom-made hybrid instrument treats the stereo signal components individually, allowing the performer to have access directly from the guitar body. For instance, the performer is able to control the tone of the acoustic (piezo) or the electric (magnetic) pickups independently. In addition, the performer can select each pickup mode (dual or single) as well as phase. There is also a blend pot that directs the signal from the magnetic to the piezo pickups, a master volume, and a 3-way selector (neck, both, or bridge) that works only with the magnetic pickup. *See DVD Chapter 4, ex.1*

There are numerous routing options for stereo operations included in the Chapter 2, Table 6: Extended Stereo Guitar Techniques: Signal Routing and Setups Options. The signal flow starts connecting the Carvin guitar acoustic output (AO) to a stereo volume pedal before going to the digital audio interface (DAI) and then from the magnetic output (MO) directly to the DAI. In this manner, the performer has the option of producing a dynamic ramp in the AO and simultaneously playing in a different dynamic level with the MO. In order to invert the roles (connecting the MO to the volume pedal), the performer will need to re-patch or get an additional volume pedal. Connecting both outputs to the

[68] Curtis Bahn, Tomie Hahn and Dan Trueman, Physicality and Feedback: A Focus in the Body in the Performance of Electronic Music, quoting F. Delalande, “Le Geste, outil d’analyse: quelques enseinements d’une recherche sur la gestique de Glenn Gloud.” *Analyse Musicale*, 1er trimester, 1988.

[69] For example, transposing electric bass player’s techniques to the guitar such as popping or slapping. In addition, guitarists may utilize natural and artificial harmonics (single or double stops) as well as hybrid techniques (right hand and several kinds of picks or attack elements).

stereo pedal and then selecting each output via the guitar blend pot, the performer is then able to generate ramps in both, but in a serial way, not in parallel.

After the guitar's audio signal arrives at the DAW we use the Max/MSP environment to create the interactive music experience. I designed the SGPI, flexible and versatile, so that the performer may run up to ten multiple effects or ten real-time compositional applications per output. In this way, the performer could have a sound palette of 20 different sounds plus the option to keep the direct signals on or off independently. Moreover, the SGPI offers the possibility to assign different volume and panning for each application and then save the settings in the preset object for later use. Subsequently, each macro preset can be recalled via the FCB 1010 MIDI Foot Controller. *See DVD Chapter 4, ex.2*

The SGPI can morph between different artistic contexts and styles. For instance, the composer can add more real-time compositional applications or change the actual sub-patches with a set of new ones according with the aesthetic characteristics of each project. In addition, the SGPI is a powerful sound design tool because the composer can juxtapose several sound layers and craft a wide spectrum of timbres and textural densities.[70]

Subsequently, the performer can route the DAI outputs to a pair of studio monitors or go directly to a PA system in the case of a live performance. The performer is able to create a sub-mix first, adding a small console on the stage and directing the signal

[70] Especially if the composer is using Percolate or whichever of the several excellent external objects available on the Cycling74 website for example.

properly. After that, the performer may send the sub-mix to the main sound diffusion system for a stereo or multi-channel configuration. Figure 6 shows the performance set up.

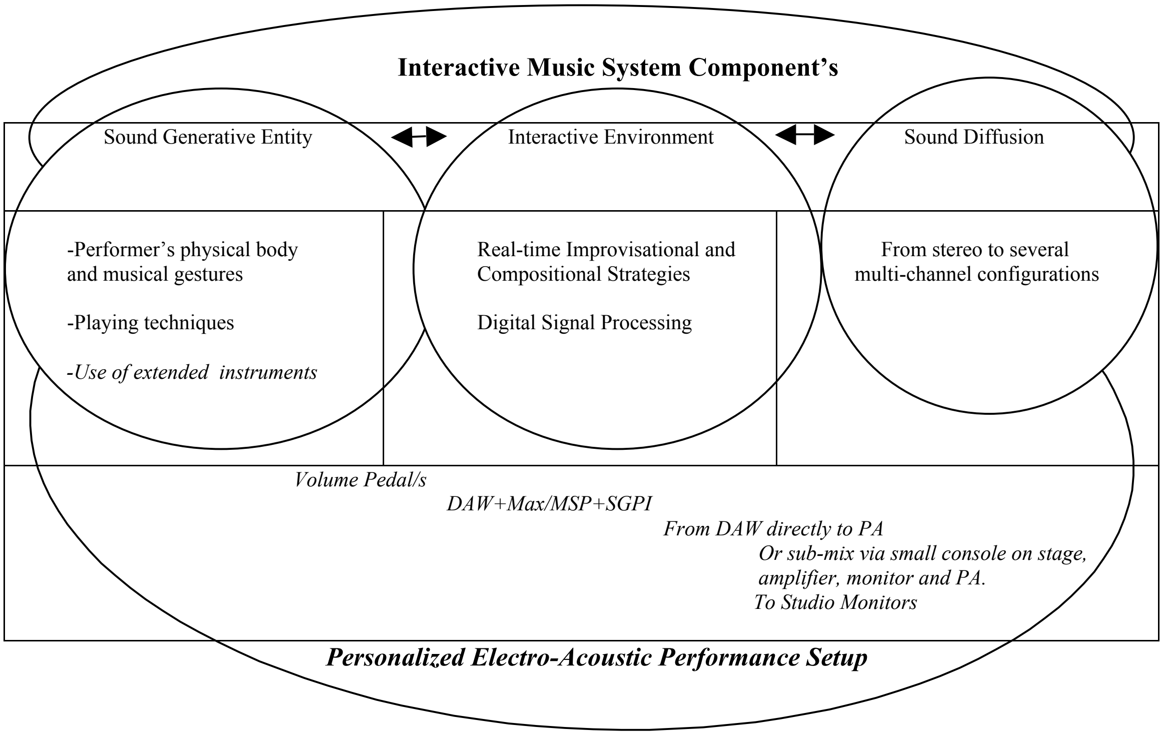

Interactive Music System Component's
Sound Generative Entity
Interactive Environment
Sound Diffusion
-Performer's physical body and musical gestures
-Playing techniques
-Use of extended instruments
Real-time Improvisational and Compositional Strategies
Digital Signal Processing
From stereo to several multi-channel configurations
Volume Pedal/s
DAW+Max/MSP+SGPI
From DAW directly to PA
Or sub-mix via small console on stage, amplifier, monitor and PA.
To Studio Monitors
Personalized Electro-Acoustic Performance Setup

Fig. 6. Electro-acoustic performance set up concept and implementation

## Building the Application

### *Problem detection, alternative solution: Defining the application purpose*

It is implied that the programmer is building an application to try to improve some aspects of an existing program or to attempt to fix previous problems. In an extreme case, a redesign or update of an older software version may be necessary; while at other times the programmer may need to rewrite the program in another computer language to best achieve the original goals. Sometimes, the goal of developing a new program is to create a software version of a hardware device.

During the GATK development process most of the problematic situations are related to the physical and sonic limitations of the electric guitar. Therefore, I designed various applications to present several possible solutions. As a result, the improvisational and compositional strategies were expanded through the computer software interaction. Table 8 shows the limitations and software solutions presented by the GATK.

TABLE 8

GATK DEVELOPMENT PROCESS: PROBLEM DETECTION AND ALTERNATIVE SOLUTIONS

| **Problem detection** | **Alternative solution** |
|---|---|
| Narrow dynamic scope | Create a Dynamic Ramp Generator application |
| Reduced frequency range | Design a Frequency Range Expander application |
| Static acoustic envelope | Develop a Sound Design application set |
| Short sustain | Built a Sustainer application |

*Creating Interactive Compositional Strategies*

Winkler states

> Compositional strategies are plans of action for creating a work. These plans are often formulated in the precompositional stages before detailed work begins, and take the form of musical sketches, diagrams or prose. Just as computer algorithms are step-by-steps plans of action to solve problems, compositional strategies are plans to produce and integrate the disparate elements of a work. These elements include the creation of a score or other instructions for the performer, primary musical material and processes, algorithms for listener and composer objects, a model or framework for structuring the work, and the design of a user interface.[71]

"Score objects" are the key components of a program and encompass the structures, data, and mechanisms for computer timing events. Furthermore, score objects work as interpreters among listener objects and composer objects by setting up the conditions to analyze incoming performance information, assigning compositions modules to receive that information, and determining how music and signal processing is created from that information. Score objects contain data structures (storage objects) that systematize the messages and other data required for a performance, as well as the mechanisms for triggering and delivering these messages to the intended objects. Techniques and methods that delineate the actions for advancing a computer score or transforming its state are called score-objects mechanisms.

The following section describes each GATK application including a brief explanation about compositional approaches using this particular system.

[71] Winkler, 260.

### *Sound Design: crafting non-standardized sounds*

"Sound design" is the way in which musicians conceive, create, produce, organize, and process sonic elements. During the sound design process the programmer can use different tools and techniques divided in three major areas:

- Arranging: including stacking, layering, and hocketing techniques.
- Timbres: this area covers topics such as multi-timbrality, polyphony, general MIDI, and the use of effects.
- Control: this aspect is focused on the use of performance controllers and editing either live or in the studio.

Stacking is a single composite sound produced for two or more timbres. Stacks can be divided into two types—composites and doubles. Composites are related to multiple sounds or timbres, while doubles take one sound and use multiple pitches derived from it. There are several methods of producing composite sounds such as adding, hybridizing, and splitting. By choosing simple sounds it is possible to use a technique similar to additive synthesis to produce composite sounds which are made up of much simpler sounds. Furthermore, hybrid sounds are produced using contrasting or complementary synthesis techniques. Some hybrid possibilities include:

- Analog with digital, where analog sound attributes can be used to complement synthesized sounds.
- Imitative sounds created synthetically, where the resulting sound has some characteristics of the real instrument, but with a certain degree of artificiality to ensure that is not mistaken for a purely imitative sound.
- Familiar with alien, where the final sound has some elements familiar to the listener, but which also includes additional elements, which are unfamiliar. The difference between imitative and familiar with alien techniques is that the second one is not linked with a traditional musical instrument as a source sound.
- Contrasting synthesis techniques, where the hybrid sound is created via the juxtaposition of additive synthesis with frequency modulation, etc.

- Sample with imitative, where the basic sample is enhanced with additional imitative sounds to make it sound more like a true acoustic instrument.

The process of splitting a sound into component sounds, then producing those sounds using separate sound sources, and finally combining them as one to produce the final sound, is related to the analysis-synthesis methods of digital synthesis. Although in terms of stacking, the splitting method chooses sounds which approximate required components and then combine them, with any analysis performed intuitively by the synthesist, for example, using a residual synthesis technique, where a sound with similar spectrum is removed from a source sample, and the residual source sample is then used as the basis for further iterative extraction of sound. In addition, “doubling” implies either a transposition or tuning change, with both the original and the doubled parts then playing together. There are several doubling techniques such as detuning, octaving[72] interval, chording, and alternatives, (for instance when the sound programmer stacks two General MIDI (GM) string sounds from different sound modules).

Layering makes use of amplitude envelope changes over time. In other words, it involves a time separation of sounds (i.e. one sound is used to provide an initial attack, while another is used for the sustain portion of the sound). Other possibilities include the combination of a fast attack/rapid decay sound with a slow attack/slow release pad, or contrasting two sounds with opposite envelopes. Another option would be to layer two sounds which have different acoustic spaces or contrasting pan positions. Moreover, the

[72] Russ’s term that means transposing a part up or down one or more octaves.

sound programmer has the option of using a sound that decays slowly and another sound that rises slowly at the same time that the first is decaying, generating a dynamic sustain.

The technique that sends successive notes from the same musical part to different instruments, assigning different timbres, pan positions, and effects is called Hocketing. This technique can be realized using several criteria:

- by note sequence order
- by note number
- by velocity
- by beat
- by time

Multi-timbrality is the ability of a single synthesizer to produce several different timbres simultaneously, essentially, an extension of the concept of stacking. Polyphony is the total number of different pitches that an instrument can make at any one time. Both multi-timbrality and polyphony tend to encourage the use of the layering, splitting and hocketing techniques. Nevertheless, the number of timbres and pitches that the sound designer is able to produce at the same time, is dependent on the applied synthesis technique and the memory and disc space of the computer or synthesizer.

Sound designers use several sample and pitch changes, as well as envelope crossfades and comb filters as editing methods for live performance or compositional strategies. Then, the sounds are catalogued, stored, and made available for subsequent retrieval. (See Table 9).

TABLE 9

SOUND SYNTHESIS TECHNIQUES

| Arranging | Stacking | Composites | Additive | |
|---|---|---|---|---|
| | | | Hybrid | Analogue with digital |
| | | | | Imitative with synthetic |
| | | | | Familiar with alien |
| | | | | Additive with Frequency Modulation (FM) |
| | | | | Sample with imitative |
| | | | Splitting | |
| | | Doubles | Detuning | |
| | | | Octaving | |
| | | | Intervals | |
| | | | Chording | |
| | | | Alternatives | |
| | Layering | Percussive and pad | | |
| | | Opposites | | |
| | | Echo/Reverb and dry | | |
| | | Pan position | | |
| | | Slow decay and slow rise | | |
| | Hocketing | by note sequence order | | |
| | | by note number | | |
| | | by velocity | | |
| | | by beat | | |
| | | by time | | |

As previously stated, the sound design process begins with the musician and performance gestures and ends with the final sound production and diffusion. The sound design process may include playing techniques, extended instruments, a variety of synthesis methods, a range of sampling techniques, and different signal processing algorithms involved in the interactive scenario. During the development of the GATK, I analyzed and tested each of the examples patches contained in the Max/MSP package. Furthermore, I researched several external Max/MSP objects in order to evaluate how they work in each particular context. After that, I adapted, redesigned, and built new applications that reflect his musical ideas and sonic personality.

### *Designing custom-made performance interfaces*

The applications contained in the GATK, as well as the other applications included in this thesis were designed following the seven basic principles of interface design proposed by Winkler:[73]

1. Design for the activity of the user (not the capability of the machine).
2. Isolate and display relevant information.
3. Have easy and logical access to controls needed at the moment.
4. Hide undue complexity; show only what counts.
5. Make it intuitive, using obvious gestures, images, or words.
6. The computer should respond naturally to user input.
7. Know the user. Think of the user's experience while running a program, considering background, experience, training, and psychology.

In the next section, each GATK application will be described following this format: application name, objective, features, and brief descriptive paragraph.

### *GATK: Description of each application*

Name: Dynamic Ramp Generator

Objective: To increase the narrow dynamic range.

[73] Winkler, 111.

Brief description:

Using the Dynamic Ramp Generator (See Figures 7-9) as an improvisational and compositional strategy, the performer/composer is able to playback a precomposed soundfile and to utilize it as a background or as a second guitar. Additionally, opening two instances of this application, the performer/composer is able to generate two individual dynamic ramps, one per output. Moreover, using multiple instances of this application the performer/composer can create complex layers of musical ideas, each with individual volume, panning, curve type, and dynamic ramp. *See DVD Chapter 4, ex.3.*

The Dynamic Ramp Generator offers the following features:

- Generation of dynamic ramps, (crescendos-diminuendos) using linear or exponential curves and freehand.
- Five visual representations of the dynamic ramps.
- The ramps may be applied to live input or to precomposed audio files.
- Mono or Stereo operation. (Via matrix selection).
- Manual or Virtual mode. (Manual via MIDI foot expression pedals 1 & 2, Virtual via software).
- Freely assignable dynamic ramp length in beats/ms. The performer inputs his work's bpm and then decides how many beats the ramp needs to reach the maximum level. This feature is available via the bpm-ms convertor patch.
- Parallel and contrary ramp direction: Acoustic and Electric output crescendo/ diminuendo, Acoustic output crescendo, Electric output diminuendo and vice versa.
- Free Hand Dynamic Generator: The user is able to draw its own dynamic curve.
- Dynamic crossfades.

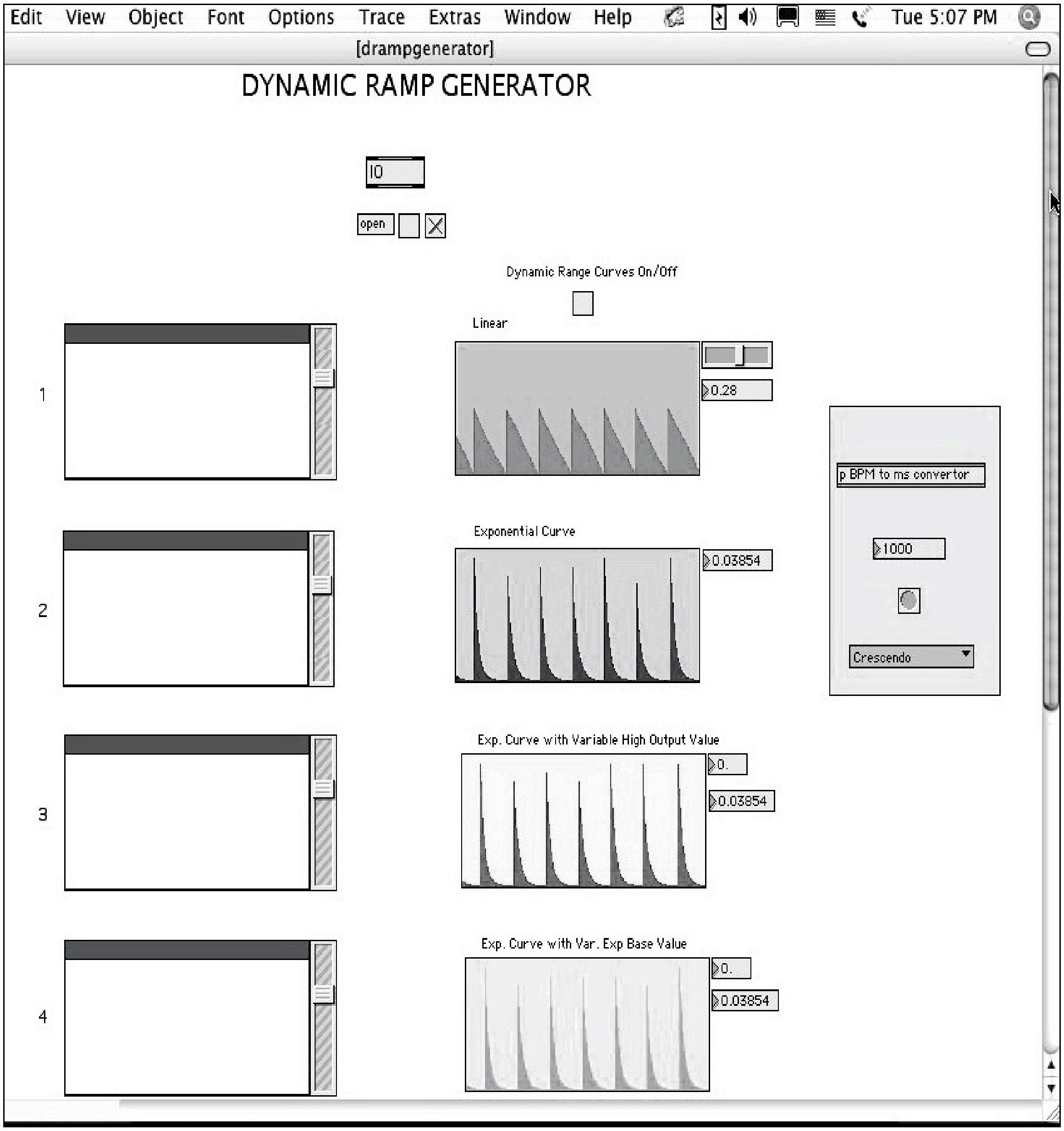


Fig. 7. Dynamic Ramp Generator. Top view.

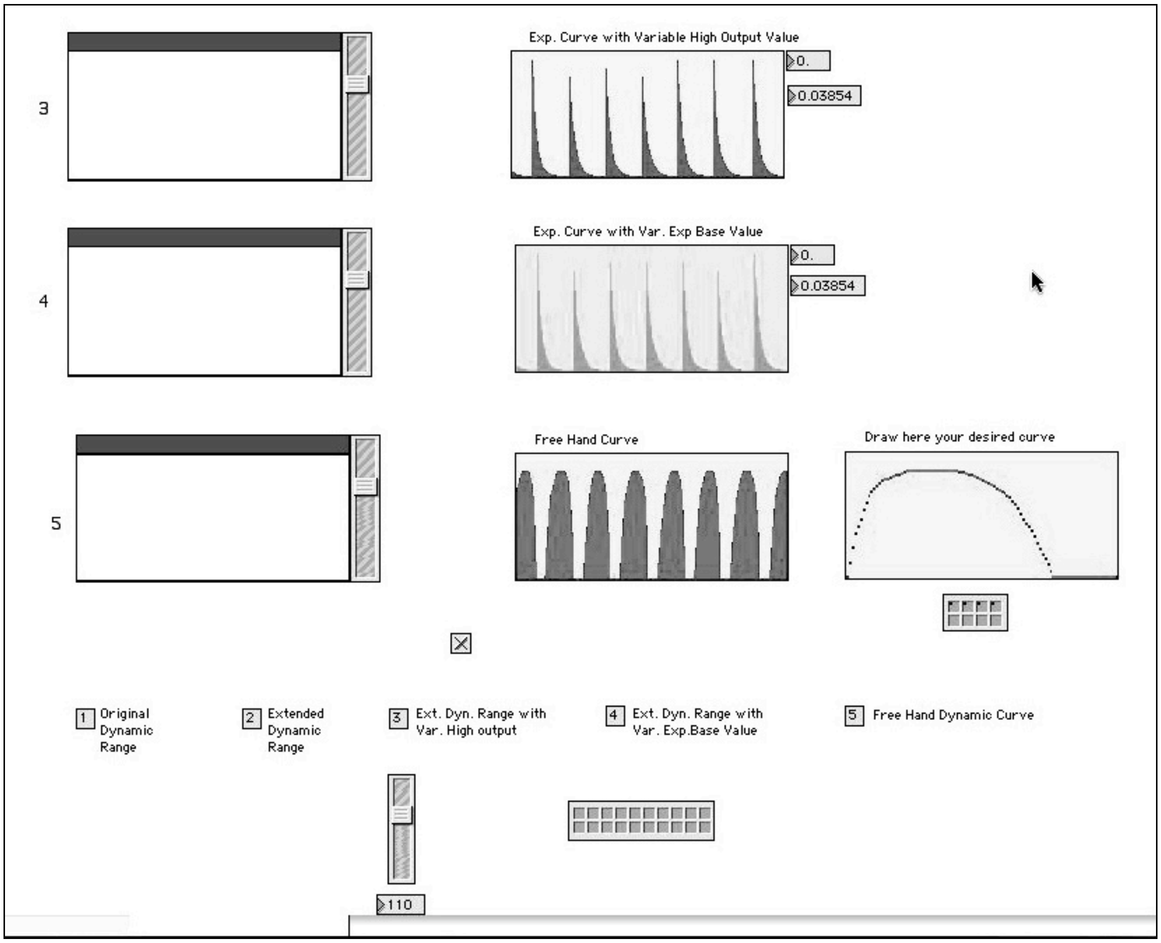
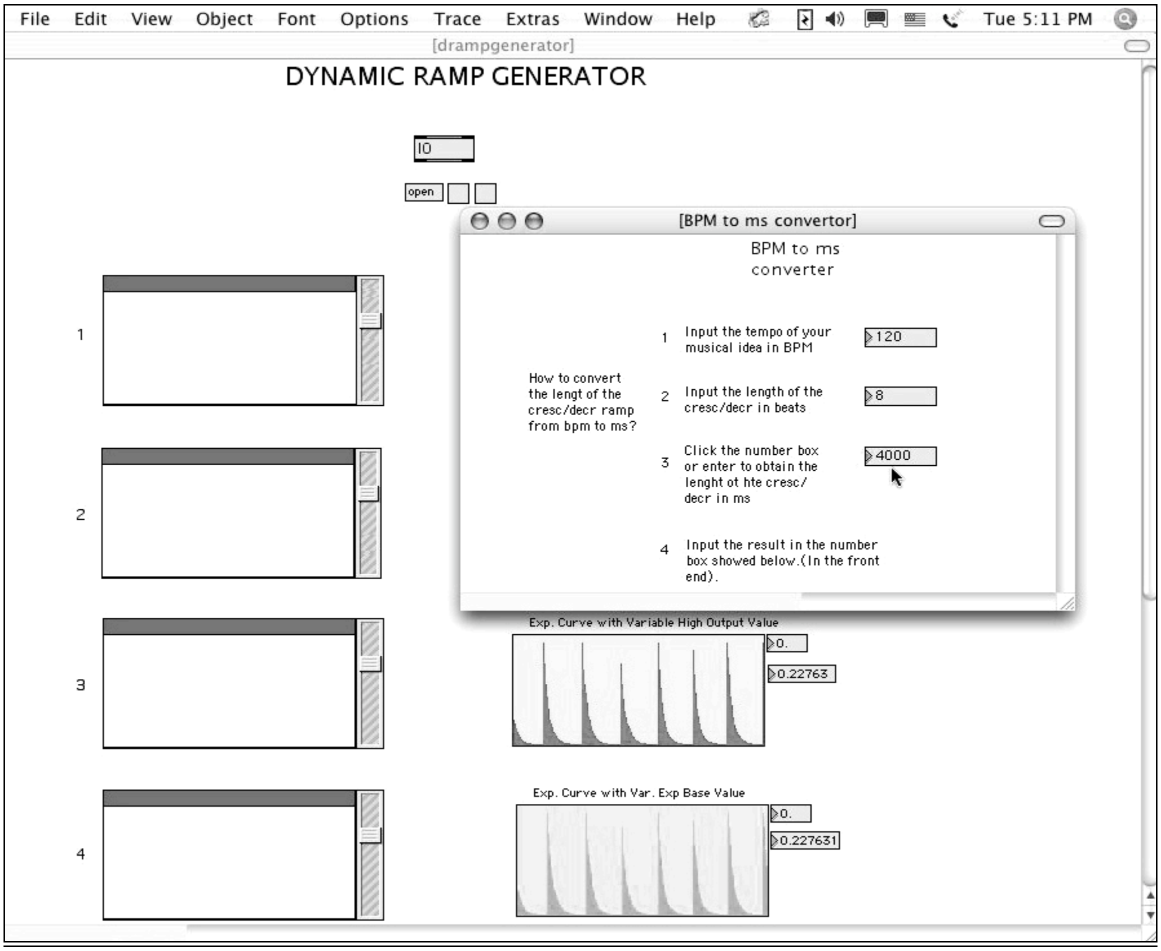

3
4
5
Exp. Curve with Variable High Output Value
0.
0.03854
Exp. Curve with Var. Exp Base Value
0.
0.03854
Free Hand Curve
Draw here your desired curve
1 Original Dynamic Range
2 Extended Dynamic Range
3 Ext. Dyn. Range with Var. High output
4 Ext. Dyn. Range with Var. Exp.Base Value
5 Free Hand Dynamic Curve
110

Fig. 8. Dynamic Ramp Generator. Bottom view.


File Edit View Object Font Options Trace Extras Window Help
Tue 5:11 PM
[drampgenerator]
DYNAMIC RAMP GENERATOR
10
open
[BPM to ms convertor]
1
2
3
4
Exp. Curve with Variable High Output Value
0.
0.22763
Exp. Curve with Var. Exp Base Value
0.
0.227631

BPM to ms converter

How to convert the lengt of the cresc/decr ramp from bpm to ms?

1 Input the tempo of your musical idea in BPM 120

2 Input the length of the cresc/decr in beats 8

3 Click the number box or enter to obtain the lenght ot hte cresc/ decr in ms 4000

4 Input the result in the number box showed below.(In the front end).

Fig. 9. Dynamic Ramp Generator. BPM to ms converter.

Name: Sound Design Set

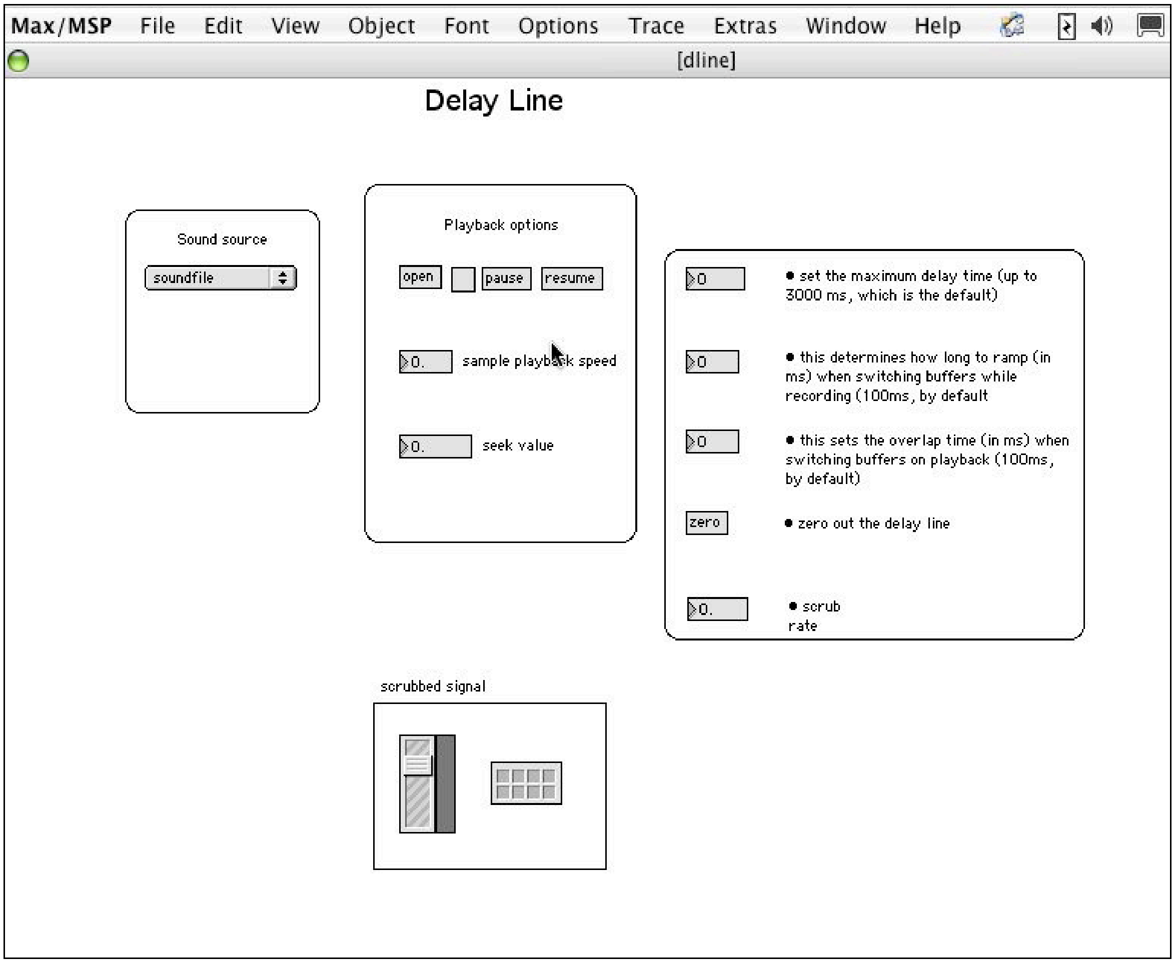
Objective: Transform a static acoustic envelope into an active one via application's crossfade. This application set is inspired in the Risset's mutation concept

Brief description of the Sound Design Set (SDS):

The SDS (See Figures 10 and 11) is a combination of three applications: Delay Line, Triggered Delay, and Stereo Granulator.[74] This set is created with the intention of

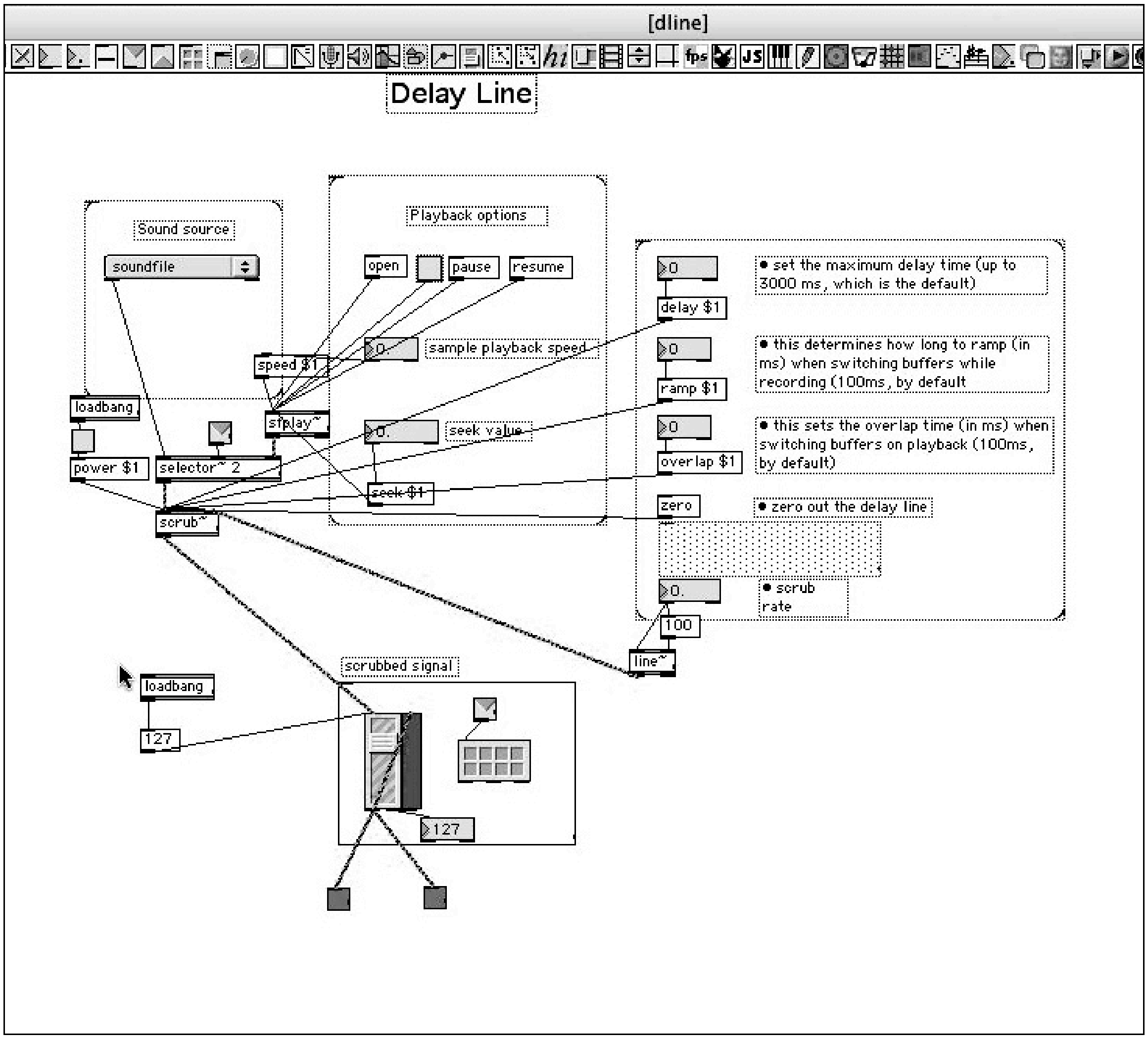
[74] Based on the Scrub, Flapper, and Munger patches designed by Dan Trueman.

offering the composer the possibility to morph from the usual guitar sounds to the unique crafted sound created through each of the aforementioned applications. *See DVD Chapter 4, ex.4-7.*



- set the maximum delay time (up to 3000 ms, which is the default)
- this determines how long to ramp (in ms) when switching buffers while recording (100ms, by default
- this sets the overlap time (in ms) when switching buffers on playback (100ms, by default)
- zero out the delay line
- scrub rate

Fig. 10. Delay line. Locked patch.



• set the maximum delay time (up to 3000 ms, which is the default)

• this determines how long to ramp (in ms) when switching buffers while recording (100ms, by default

• this sets the overlap time (in ms) when switching buffers on playback (100ms, by default)

• zero out the delay line

• scrub rate

Fig. 11. Delay line. Unlocked patch.

The Delay Line application offers the following features:

- Selectable sound source: sound input, sound file.
- Open, start/stop, pause and resume playback options.
- Changeable sample speed playback and seek value.
- Assignable delay time, ramp length and overlap time.
- Scrub rate.
- Individual volume and panning. *See DVD Chapter 4, ex.4.*

The Triggered Delay Line application (See Figures 12 and 13) offers the following features:

- Assignable flap speed.[75]
- Selectable interpolation type: linear or poly.
- Individual volume and panning.
- Variable ADSR (attack, decay, sustain, release) values.
- Trigger on/off option.
- Selectable sound source: sound input, sound file.
- Open, start/stop, pause and resume playback options.
- Changeable sample speed playback. *See DVD Chapter 4, ex.5.*

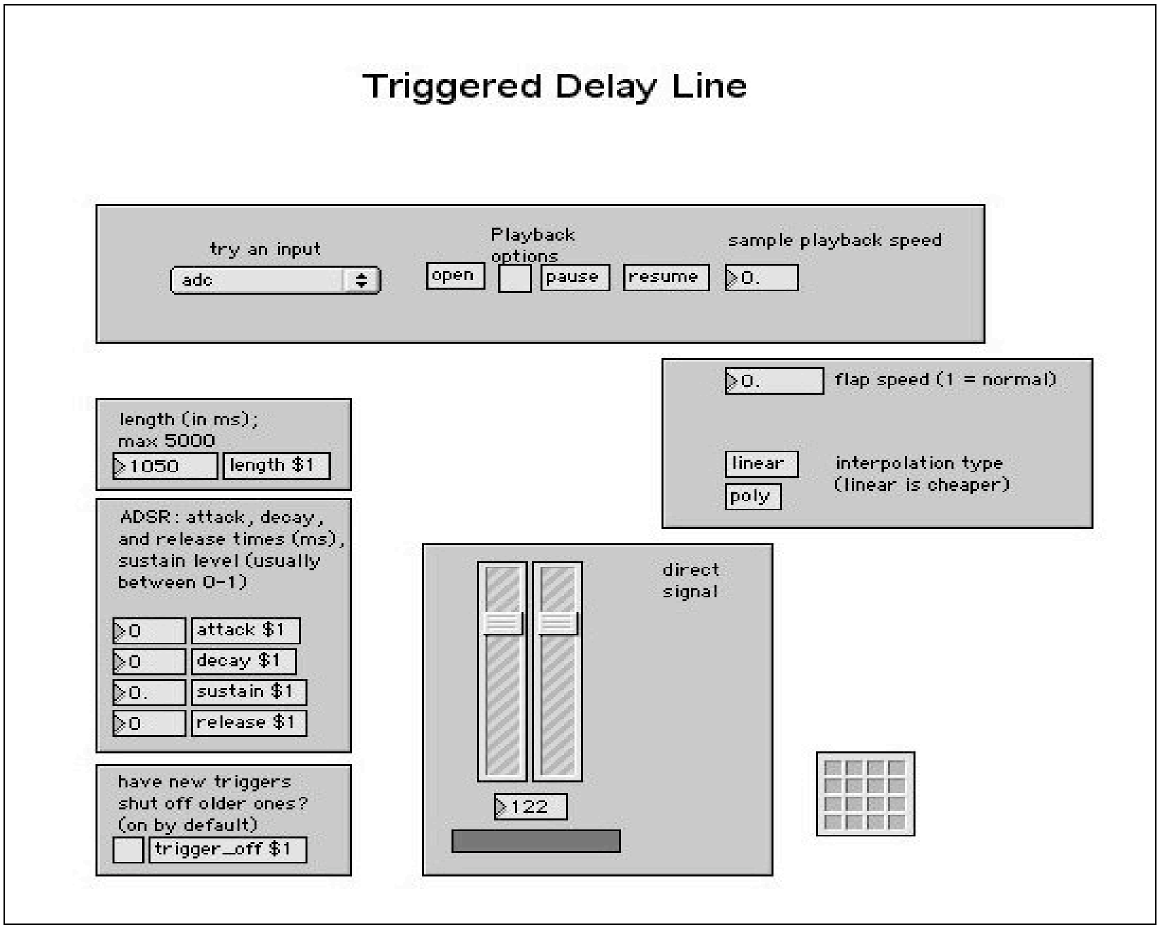


Fig. 12. Triggered Delay line. Locked patch.

[75] This feature means that the user can modify the playback speed of the flapper object, going back and forth between the trigger point and the “present” current sample.

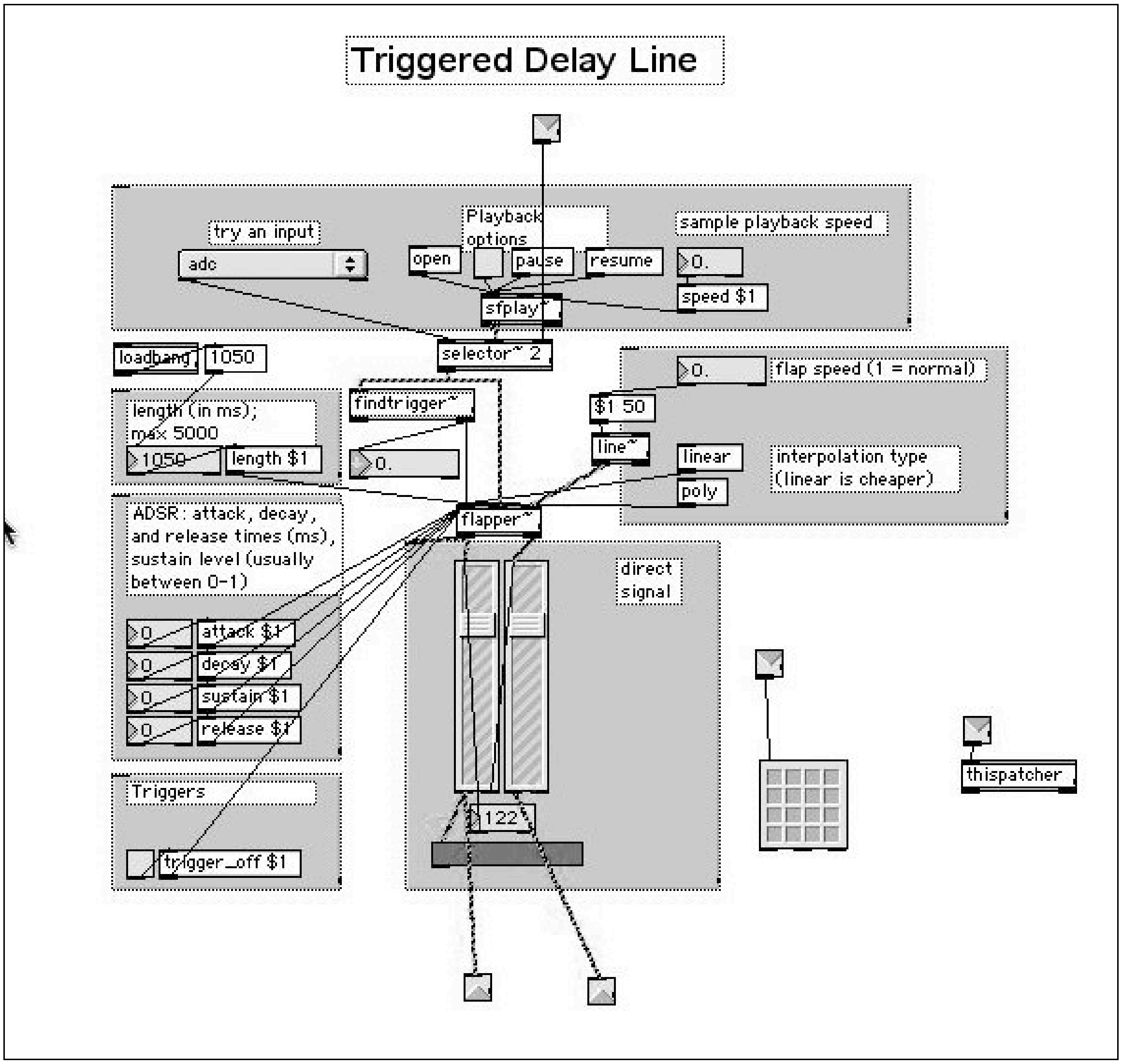


Fig. 13. Triggered Delay line. Unlocked patch.

The Stereo Granulator application (See Figures 14 and 15) offers the following features (partial list):

- Selectable sound source: sequence, sound input, sound file.
- Turn on/off the stereo granulator.
- Transpose, gain, pan, attack, decay, sustain level, and release each grain.
- Turn on/off continuous grains option.
- Selectable voices and ramptime.
- Control grain separation, rate, size, pitch and stereo spread.

- Variable grain directions: backwards and forwards (0), just forwards (1), or just backwards (-1).
- Selectable playback position within the buffer (between 0 and 1). -1 = randomized position (which is the default). *See DVD Chapter 4, ex.6.*

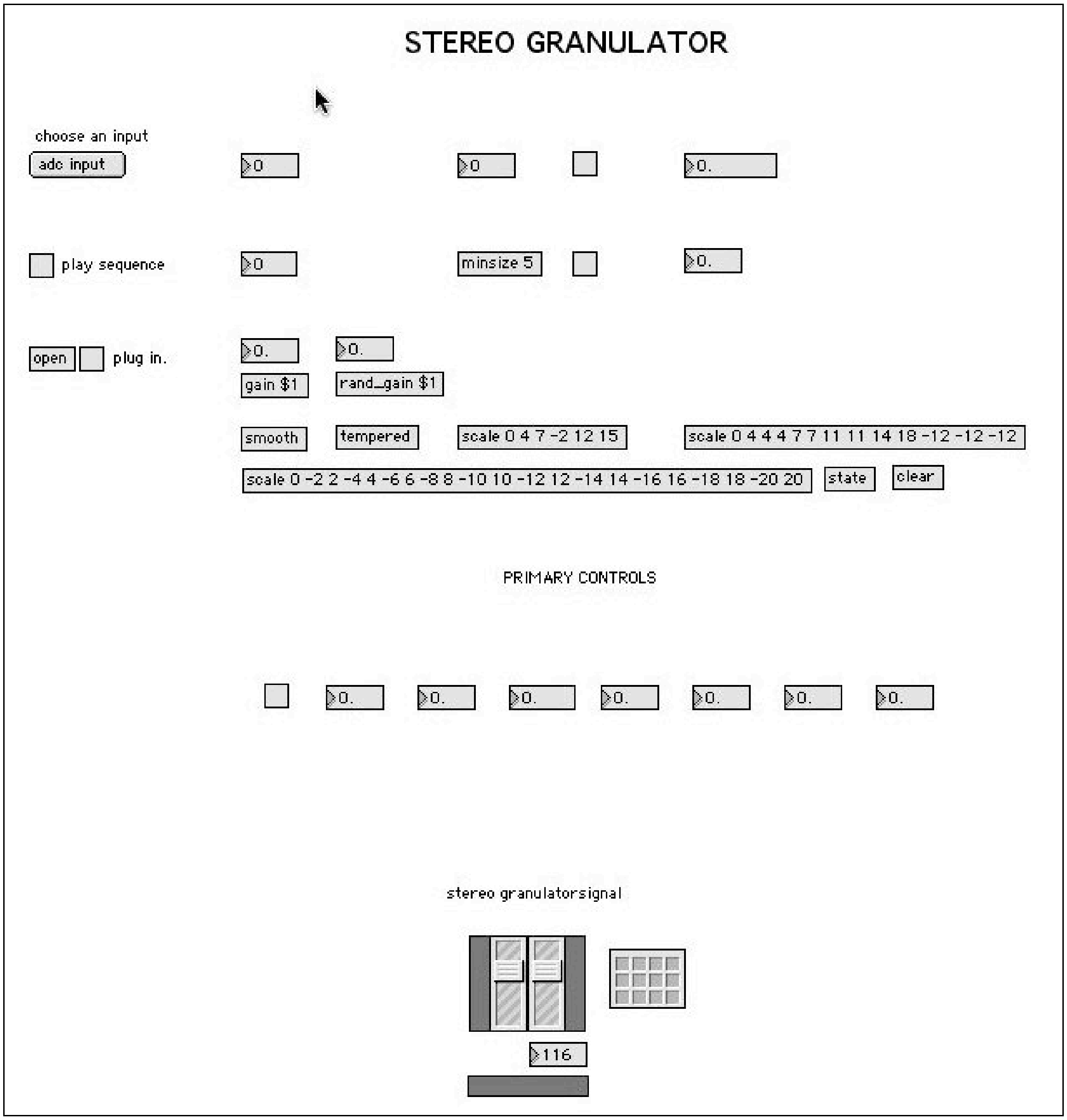
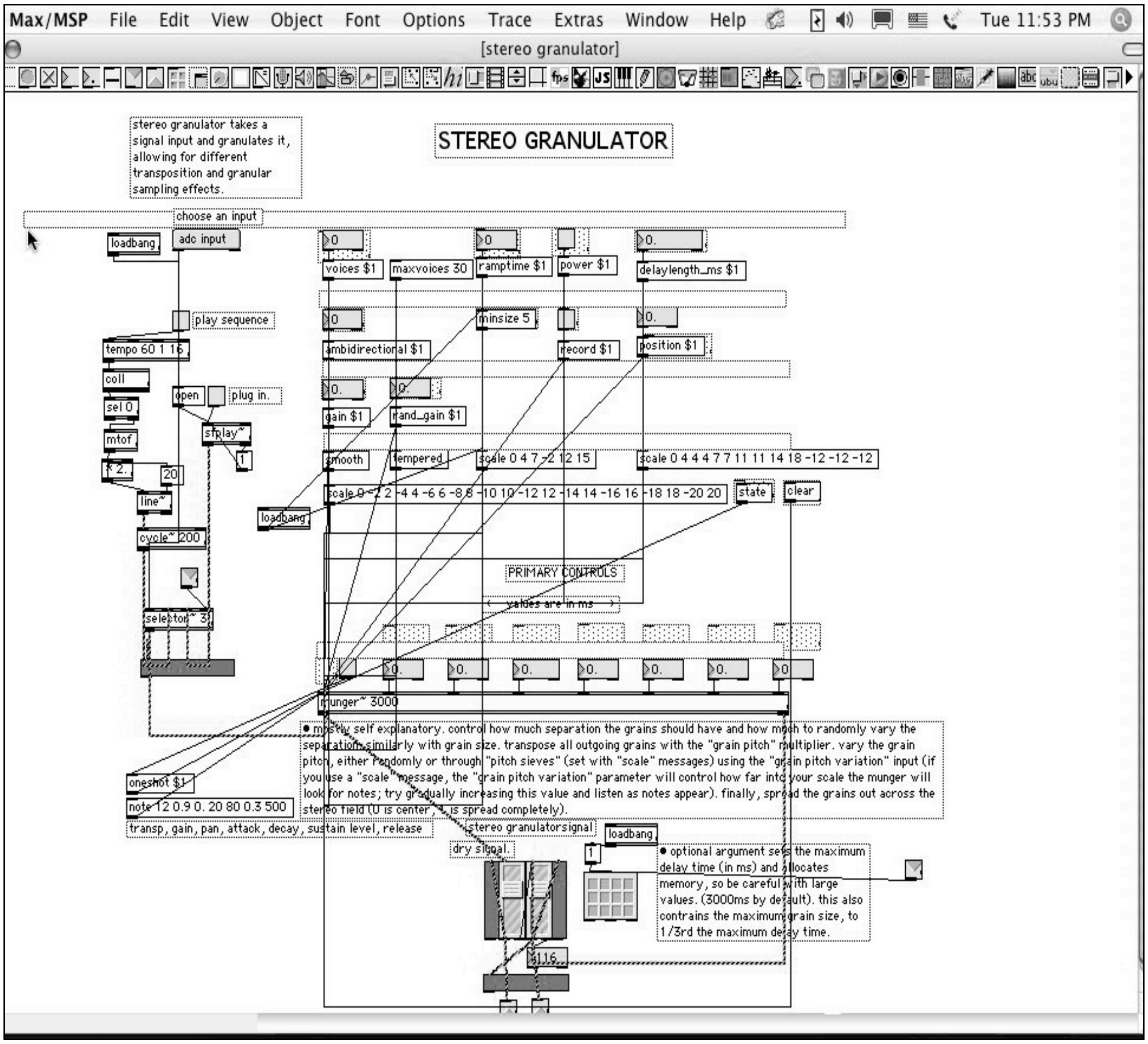


Fig. 14. Stereo Granulator. Locked patch.



stereo granulator takes a signal input and granulates it, allowing for different transposition and granular sampling effects.

● mostly self explanatory. control how much separation the grains should have and how much to randomly vary the separation. similarly with grain size. transpose all outgoing grains with the "grain pitch" multiplier. vary the grain pitch, either randomly or through "pitch sieves" (set with "scale" messages) using the "grain pitch variation" input (if you use a "scale" message, the "grain pitch variation" parameter will control how far into your scale the munger will look for notes; try gradually increasing this value and listen as notes appear). finally, spread the grains out across the stereo field (0 is center, 1 is spread completely).

● optional argument sets the maximum delay time (in ms) and allocates memory, so be careful with large values. (3000ms by default). this also contrains the maximum grain size, to 1/3rd the maximum delay time.

Fig. 15. Stereo Granulator. Unlocked patch.

*See DVD Chapter 4, ex.7.*

Name: Frequency Expander.

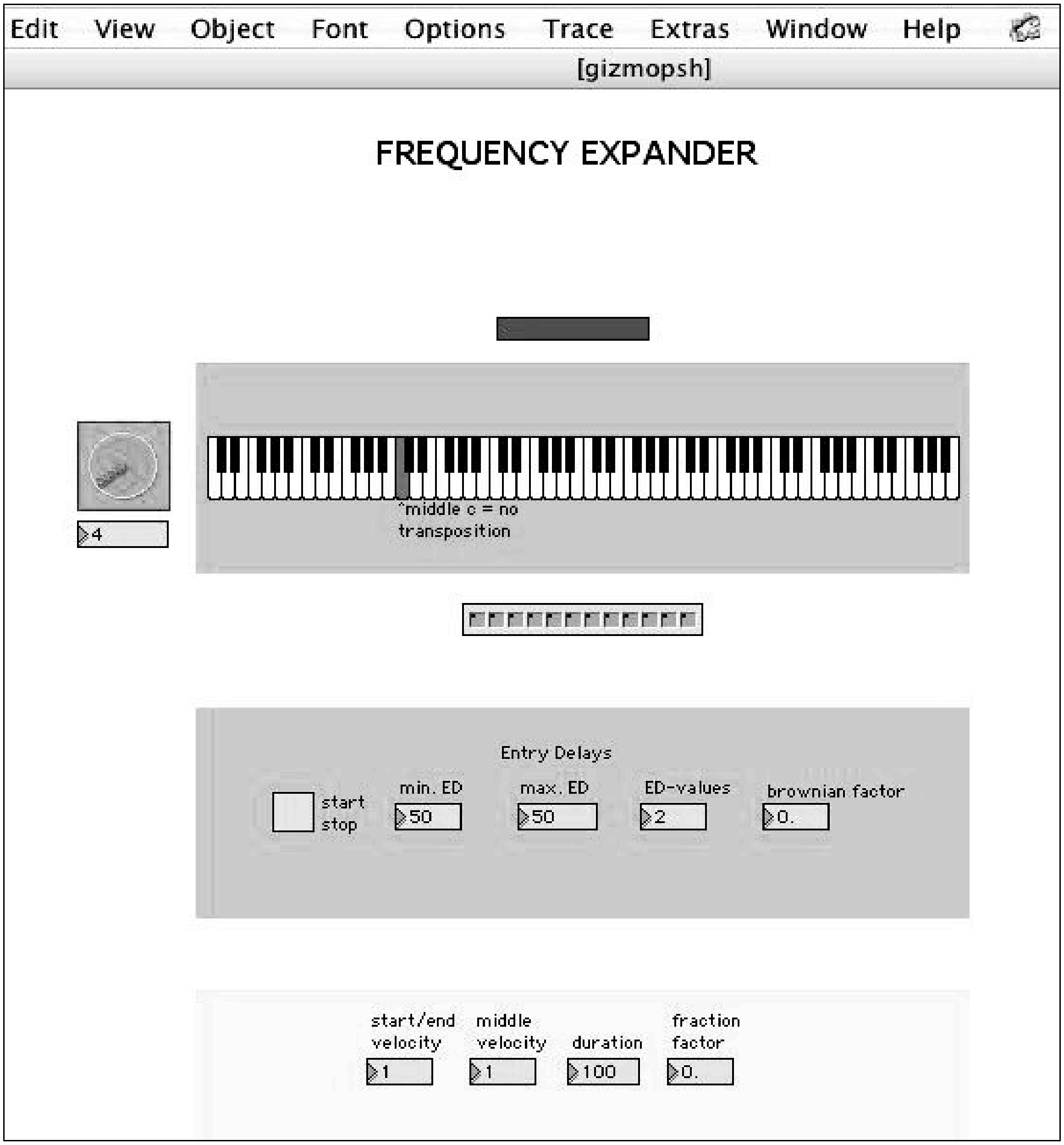
Objective: Expand the guitar’s reduced frequency range.

Brief description:

The Frequency Expander application (See Figures 16 and 17) doubles each played note to a particular interval. Furthermore, the performer/composer can design his own delays ideas through the input of several entry delays values, as well as changing the Brownian factor. Specifically, the Brownian factor parameter manages the distance between two succeeding rhythmical values.

Moreover, the composer is able to apply an automatic dynamic curve on top of the transposed notes choosing a start, middle, and end velocity. In effect, via the fraction factor, the composer can determine the proportion between the crescendo and decrescendo part.

As an extension of the previous techniques, the composer is able to use two instances of this patch, one per stereo guitar output, and assign similar or contradictory settings (i.e. assigning an octave lower to the electric output, and an octave higher to the acoustic output). In that way, the composer could extend the frequency range of a particular chord voicing. *See DVD Chapter 4, ex.8.*



FREQUENCY EXPANDER

4

^middle c = no transposition

Entry Delays

□ start stop

min. ED 50

max. ED 50

ED-values 2

brownian factor 0.

start/end velocity 1

middle velocity 1

duration 100

fraction factor 0.

Fig. 16. Frequency Expander. Locked patch.

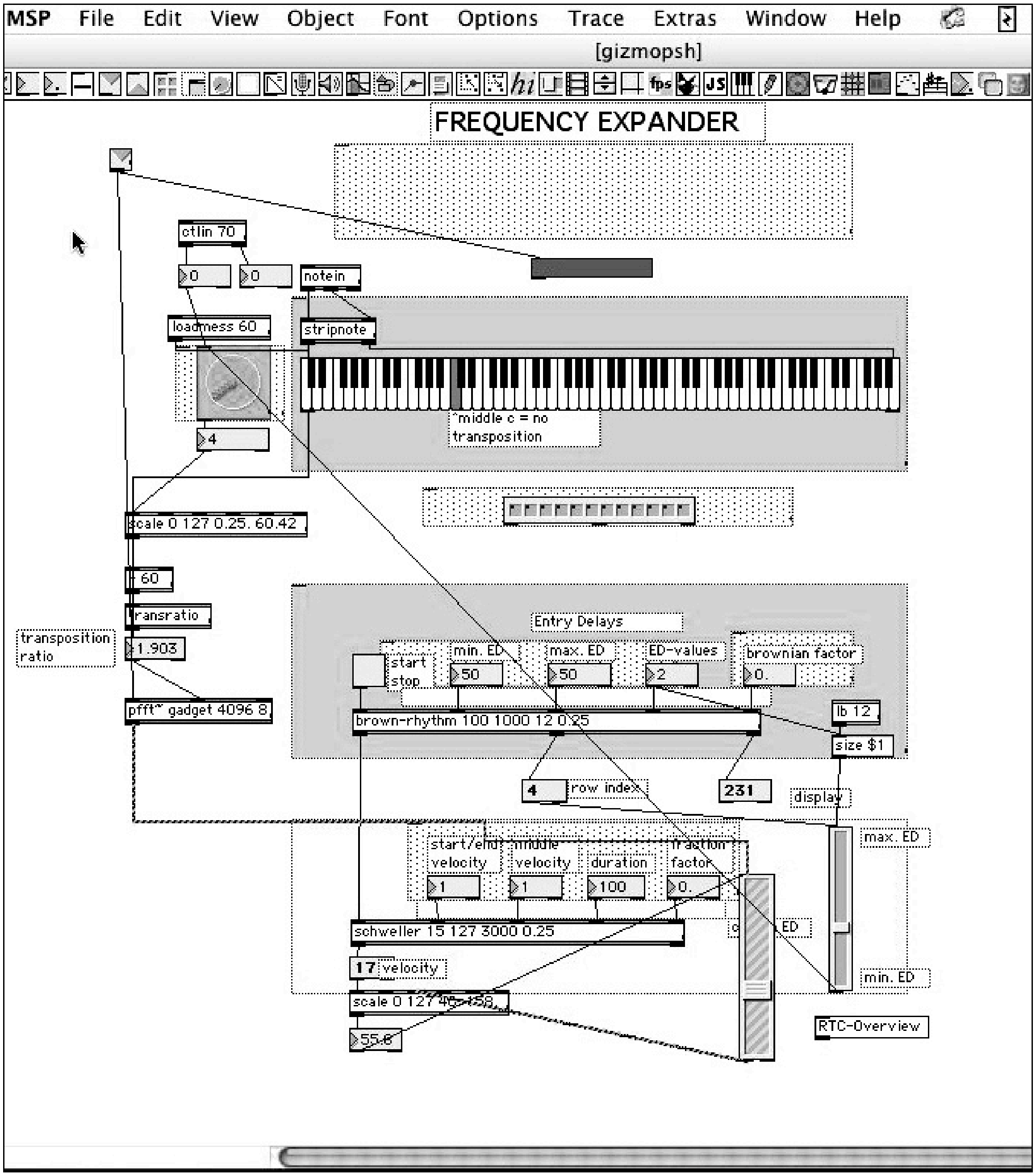


Fig. 17. Frequency Expander. Unlocked patch.

The Frequency Expander offers the following features:

- Visual representation of the transposed note through the virtual keyboard.
- Mono or Stereo operation. (Via matrix selection).

- The actual register can be lowered three octaves down and two octaves up.
- Individual operation: Acoustic output two octaves up, Electric output three octaves down and vice versa.
- Transposition rate dial via mouse or MIDI controller's knob.
- Selectable transposition interval via preset or pressing the desired distance in the virtual keyboard.
- Manual or Virtual mode: (Manual via MIDI Foot Controller or Virtual via software).

Name: Virtual Sustainer

Objective: Extend the guitar's sustain.

Brief description:

Using the Virtual Sustain application (See Figures 18 and 19) the user will be able to emulate the active pads of hardware and virtual synthesizers to produce unique soundfiles to be used as a pad or background. An interesting use of the Virtual Sustain application is to record a series of natural harmonics series and select the loop on option. The aforementioned idea could be also applied to different extended guitar techniques. Additionally, the performer can play a single note or chord and capture the sustain segment creating the fundamental layer for future improvisations. *See DVD Chapter 4, ex.9.*

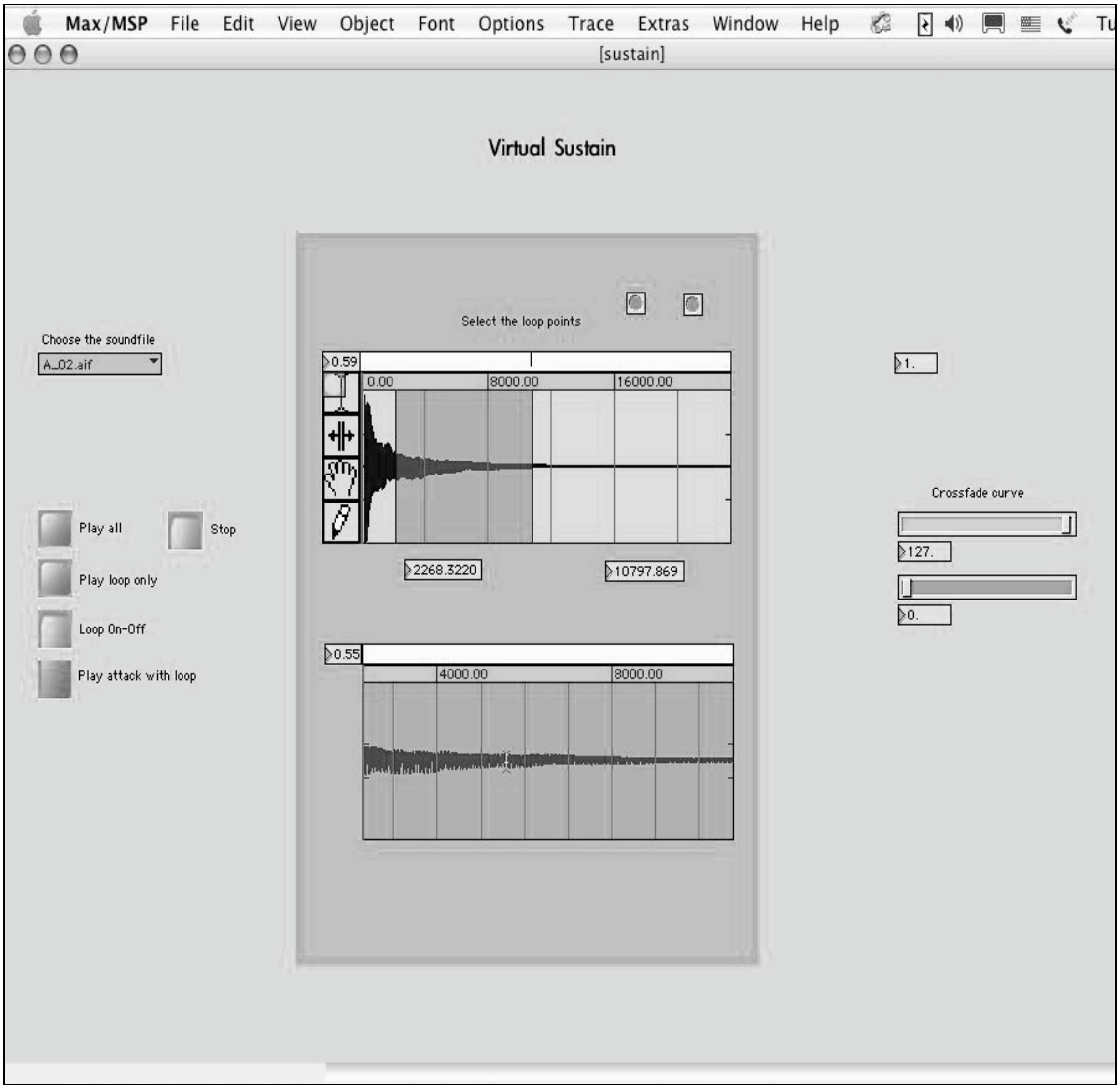


Fig. 18. Virtual Sustain. Locked patch.

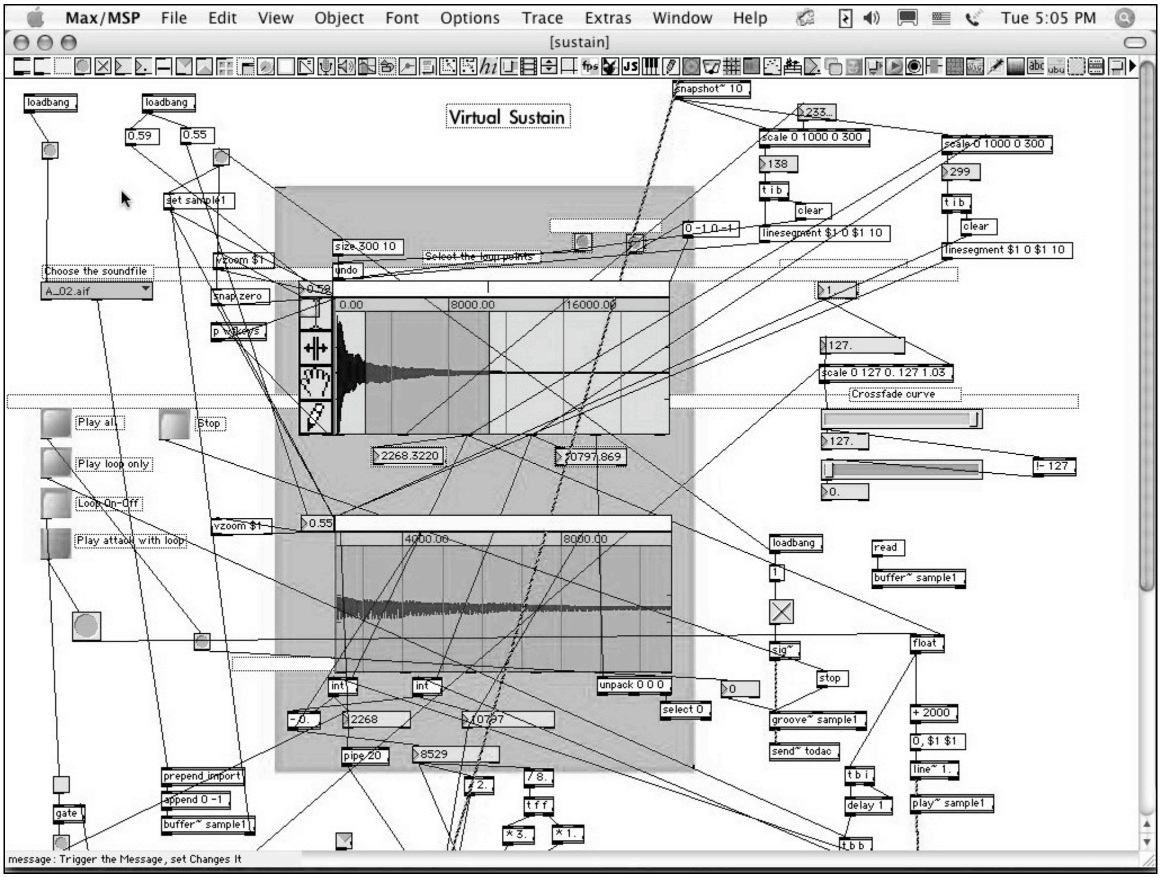


Fig. 19. Virtual Sustain. Unlocked patch.

<u>The Virtual Sustain offers the following features:</u>

- Playback options: play all, stop, play loop only, loop on/off, and play attack with loop.
- Waveform options: full sample and loop portion view. Four different elements for selecting, grabbing and editing the waveform. Vertical zoom, timeline in milliseconds and slider playback. Numeric box reference of the start and end loop points.
- Visual crossfade reference.
- Assignable exponential curve value.
- Individual channel operation.
- Manual or Virtual mode: (Manual via MIDI Foot Controller or Virtual via software).

Name: Polyphonic Creator

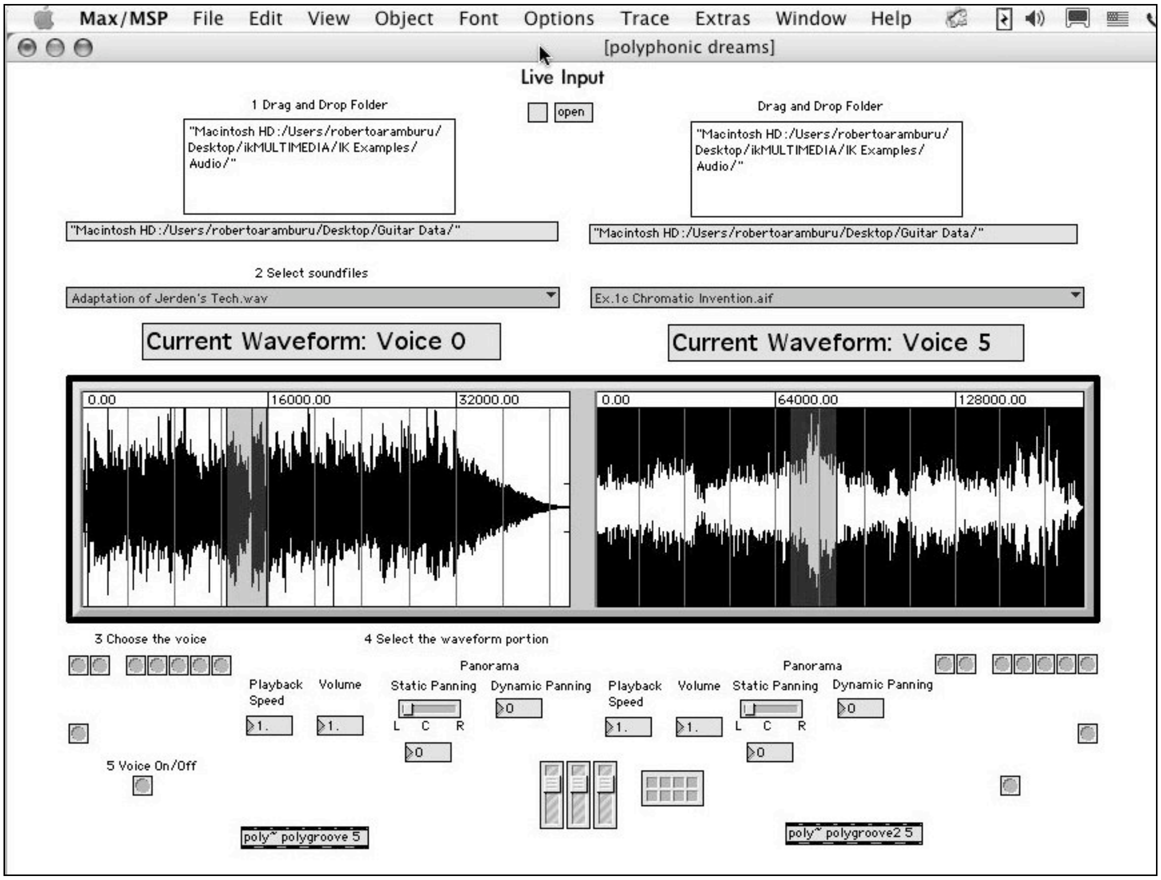
Objective: Extend the actual guitar's polyphony through the recording and processing of the musical content generated in a live performance situation.

Brief description:

With the Polyphonic Creator application (See Figures 20 and 21) the user is able to record each part, playback individually or simultaneously and be able to improvise on top. In addition, the composer is able to create dynamic soundscapes via the juxtaposition of multiples parts. Also, this application provides individual static or active panning settings options. Specifically, the performer/composer has ten independent positions from left to right on each part or voice.

Moreover, the composer is able to organize and create dynamic groups, for instance, assigning a *pp* to the part 1 and 2, a *mf* to part 3, a *f* to part 4, and a *ff* to part 5. Furthermore, the performer/composer is able to distribute each part in a particular frequency range via the playback speed option.

Another interesting improvisational and compositional approach is to analyze and design different relationships between the content of the two waveform objects, for instance, considering the waveform B as an extension of the waveform A settings (thinking A-B as a whole entity), or B as an opposite settings, and/or using A and B as a two independent instruments, etc. Expanding the previous idea, the composer could define each type of relationship per parameter to control (e.g. regarding speed playback, A-B could be used as one instrument; in relation to the amplitude, A's settings could be the opposite of B, etc.). *See DVD Chapter 4, ex.10.*



"Macintosh HD:/Users/robertoaramburu/Desktop/ikMULTIMEDIA/IK Examples/Audio/"

"Macintosh HD:/Users/robertoaramburu/Desktop/Guitar Data/"

Adaptation of Jerden's Tech.wav

Ex.1c Chromatic Invention.aif

Playback Speed 1. Volume 1. Static Panning 0 Dynamic Panning 0

Fig. 20. Polyphonic Creator. Locked patch.

<u>The Polyphonic Creator offers the following features:</u>

- Drag and drop option.[76]
- Visual representation of the soundfiles' content and its loop points via each of the waveforms objects.
- Five parts polyphony per waverform display, total 10. Modifying the poly object the number of parts could be expanded.

[76] This means that the user is able to drag soundfiles from the desktop and drop it into the application, allowing him to work not only with precomposed material but also with the data recently captured in a live performance situation.

- Individual static or dynamic panning, volume, and DSP options for each part.
- Individual channel operation.
- Switch each part on/off.
- Manual or Virtual mode: (Manual via MIDI Foot Controller or Virtual via software).

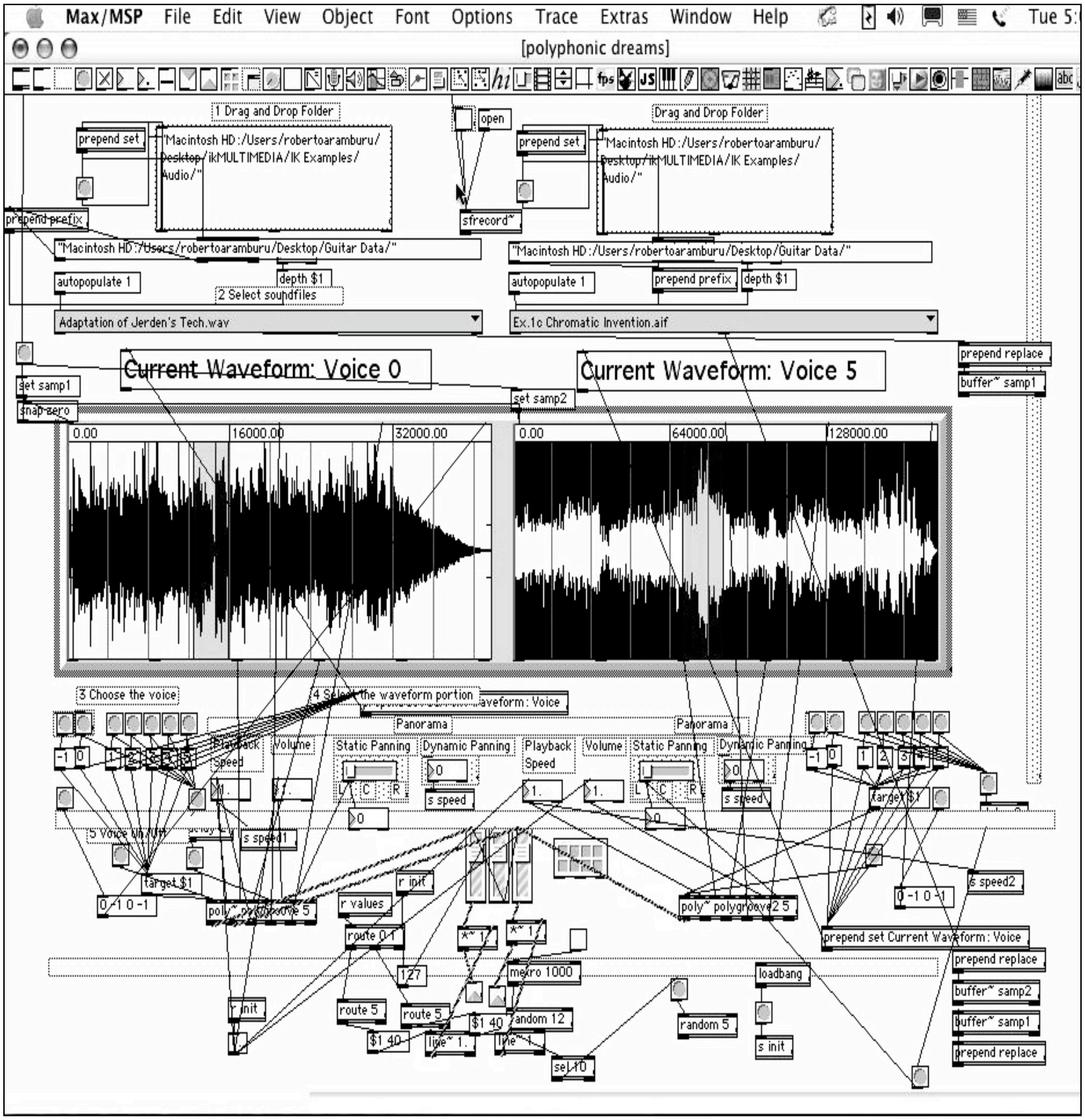


Fig. 21. Polyphonic Creator. Unlocked patch

Name: Stereo Guitar Performance Interface

Objective: Integrate and control each of the custom applications.

Brief description:

The goal of the SGPI (See Figures 22 and 23) is to offer a custom-made performance interface with enough flexibility to redesign a particular configuration (software and hardware) per each musical context or project.

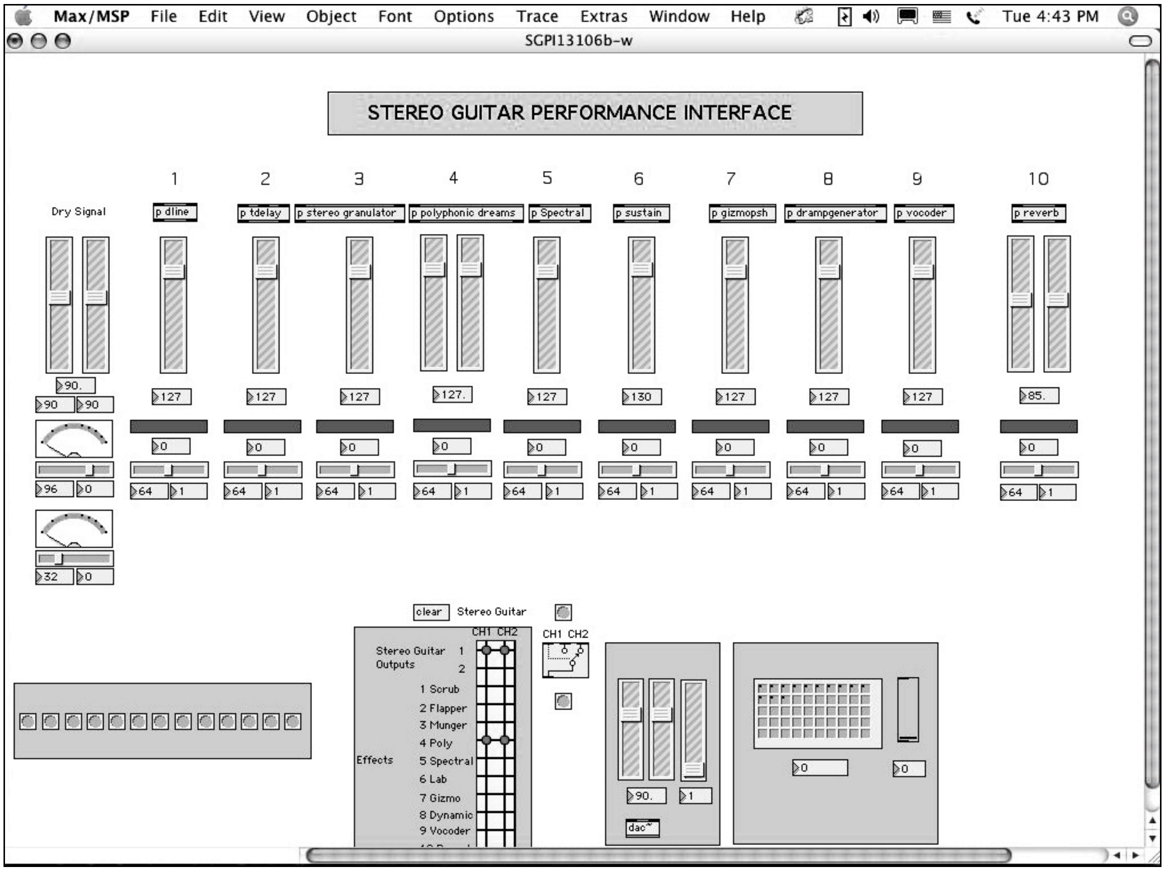


Fig. 22. Stereo Guitar Performance Interface. Locked patch.

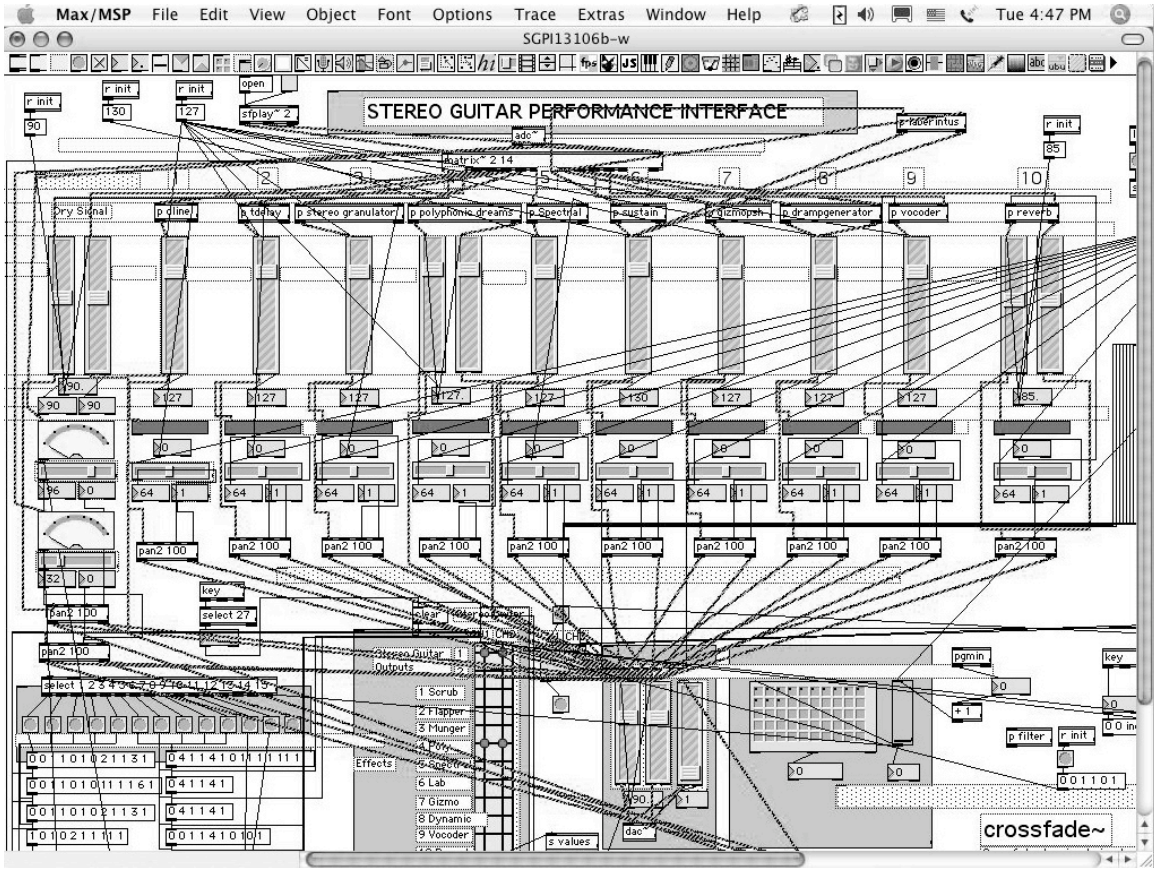


Fig. 23. Stereo Guitar Performance Interface. Unlocked patch.

<u>The Stereo Guitar Performance Interface offers the following features:</u>

- Stereo Live Input with independent volume, panning, and meters. Designed to control the stereo guitar direct signal.
- Individual volume, panning, and meter per application.
- Matrix option: to visualize each sound configuration. For instance, preset one could be the sum of patch one, two and three.
- Macropreset: saves all the information available in the main window such as volume, panning, levels, internal patch presets, etc.
- Stereo Crossfade.
- Controllable via any MIDI foot controller or keyboard.
- Flexible architecture, the user is able to use his own patches and insert them in the SGPI.
- Originally created for the stereo guitar but could be used for any stereo or mono instrument.

# CONCLUSIONS

My motivation related to the *Expanding Guitar Production Techniques: Building the Guitar Application ToolKit (GATK)* thesis was to create an innovative project that expanded the guitar player's options regarding production techniques to be applied in the studio or in a live-performance situation. My goal was to enhance the qualities of the guitar and address the previously mentioned issues. In addition, the intent was to augment the improvisational and compositional strategies through the computer software interaction.

Frequently, the usual electric guitar player setup implies a mono electric guitar, a few effects stomp boxes, and an amplifier. All of those hardware devices depend on physical operation (manual control) to change their parameters. This approach has several obstacles including portability, functionality, customization, and finances. For example, in order to change the settings between songs, the guitarist needs to press several stomp boxes and manually modify the parameters via knobs or switching on/off effects. Moreover, this technique is extremely limited because the setup works serially not in parallel, meaning one effect follows another. Besides that, in a fixed configuration there can be no sonic evolution over time.

Based on the previous scenario, analyzing actual trends regarding interactive music systems and considering my experience as a professional guitar player, the purpose of this thesis was to offer another option through the GATK. As mentioned in chapter four, my personalized electro-acoustic performance setup consists of a stereo hybrid guitar, several controllers, an audio interface, and the laptop. This configuration works in conjunction with custom-made computer performance interfaces, interactive compositional strategies, and unique sound design options. Therefore, this perspective offers unlimited options regarding customization, functionality, automation, and real-time control of music parameters using the Max/MSP programming environment.

If we make a simple comparison between the aforementioned usual electric guitar player setup and my personalized electro-acoustic performance configuration, we will find several advantages in favor of my proposed alternative. Essentially, option one is hardware based and another option makes extensive use of custom applications. For example, the first option is based in physical (i.e. hardware) devices such as combo amplifiers, foot pedals, and multi-effects processors units. This set-up has limited DSP options because it is restricted by the number of stomp boxes units or external sound modules, and is further restrained by the musician’s financial situation. In addition, the sound palette is often more limited because of the stylistic requirements of the music market (e.g. the types of effects that are commonly used in a Top-40 songs chart), by the external multi-effects’ default presets, and by the devices’ features (two or more knobs).

Conversely, the second option offers a more flexible combination of hardware and software. Moreover, the DSP options available are in direct relation to the programmer’s skills in developing his unique DSP applications. Regarding the sound design alternatives, the musician/programmer has the opportunity to create his own original sounds combining all types of synthesis techniques. In short, using the GATK the performer/composer is able to develop his own sonic personality and musical vision.

Quite recently, the hardware-software interaction applied to the electric guitar has been commercially implemented. In just the last two years, the Native Instruments' Guitar Rig 1 was released, and will be quickly followed by the IK Multimedia's Amplitube 2, with a release date of February 6, 2006. Essentially, each product has its own characteristics but none of them has the wide spectrum that the GATK has, mainly because they are not based on the Max/MSP programming language, quite likely because Max/MSP is a complex programming language that was not originally designed for commercial purposes or the average user.

Several internationally recognized composers/performers/educators are leading the interactive music field using this software including Curtis Bahn, Perry Cook, Luke Dubois, George Lewis, Robert Rowe, Dan Trueman, Todd Winkler, and David Wessel. Max/MSP had become one of the core subjects of music technology and electronic music programs offered by university music programs worldwide. For that reason, the user's community has grown exponentially, increasing the speed of the learning curve through the open sharing of knowledge.

Considering the constant and radical changes in the music technology field, it is not always an easy task to predicting the future. Music and technology, however, are soon expected to advance in several areas such as mLAN, remote recordings, modeling guitar technology, USB guitar, plug-ins compatibility, expansion and customization, voice commands to control hardware, and the crossbar computing technology. Regarding the future of the GATK project, it is expected to increase the number of applications, constantly improve the hardware and software interaction and extend its use to other instruments.

Expanding Guitar Production Techniques: Building the Guitar Application ToolKit (GATK) is the first research project of its kind. It is expected that this thesis will be useful as a reference paper for electronic musicians and music technology students (especially guitar players), as a product demonstration for companies that manufacture the relevant software, and as a personal portfolio for future technology-related jobs.

# LIST OF REFERENCES

Books

Battino, David and Kelli Richards. *The Art of Digital Music: 56 Visionary Artists and Insiders Reveal Their Creative Secrets.* San Francisco: Backbeat Books, 2005.

Collins, Mike. *Professional Guide to Audio Plug-ins and Virtual Instruments*. Oxford: Focal Press, 2003.

Cook, Perry. *Real Sound Synthesis for Interactive Applications*. Natick, Mass.: AK Peters, Ltd., 2002.

Dean, Roger. *Hyperimprovisation: Computer-Interactive Sound Improvisation.* Middleton, Wis.: A-R Editions, 2003.

Droney, Maureen. *Mix Masters: Platinum Engineers Reveal Their Secrets for Success.* Boston: Berklee Press, 2003.

Franz, David. *Producing in the Home Studio with Pro Tools*. 2d ed. Boston: Berklee Press, 2003.

Huber, Davis Miles and Robert E.Runstein. *Modern Recording Techniques*. 5th ed. Boston: Focal Press, 2001.

Katz, Bob. *Mastering Audio: The Art and the Science.* Amsterdam: Focal Press, 2002.

Lansdale, Mark and Thomas C. Ormerod. *Understanding Interfaces: A Handbook of Human-Computer Dialogue*. 2d ed. San Diego, Calif.: Academic Press, 2004.

Leider, Colby N. *Digital Audio Workstation*. New York: McGraw-Hill, 2004.

Owsinski, Bobby. *The Mixing Engineer's Handbook*. Vallejo, Calif.: MixBooks, 1999.

Pejrolo, Andrea**.** *Creative Sequencing Techniques for Music Production: A Practical Guide to Logic, Digital Performer, Cubase and Pro Tools*. Boston: Focal Press, 2005.

Roads, Curtis. *The Computer Music Tutorial.* 2d ed. Cambridge, Mass.: MIT Press, 1996.

Rowe, Robert. *Interactive Music Systems*. Cambridge, Mass.: MIT Press, 1993.

________. *Machine Musicianship*. Cambridge, Mass.: MIT Press, 2001.

Russ, Martin. *Sound Synthesis and Sampling*. 2d ed. Amsterdam: Focal Press, 2004.

Shneiderman, Ben. *Designing the User Interface: Strategies for Effective Human Computer Interaction*. 2d ed. Menlo Park, Calif.: Addison-Wesley, 2004.

Winkler, Todd. *Composing Interactive Music: Techniques and Ideas Using Max*. Cambridge, Mass.: MIT Press, 2001.

## Periodicals

Curtis Bahn, Dan Trueman and Perry Cook, "Alternative Voices for Electronic Sound." *The Journal of the Acoustical Society of America* 108, no. 5 (2000): 25-38.

David Wessel and Matthew Wright, "Problems and Prospects for Intimate Control of Computers." *Computer Music Journal* 26, no. 3 (2002): 11-22.

Garnett, Guy E. "The Aesthetics of Interactive Computer Music." *Computer Music Journal* 25, no. 1 (2001): 21-33.

Jordá, Sergi. "Improvising with Computers: A Personal Survey (1989-2001)." *Journal of New Music Research* 31, no. 1 (2002): 1-10.

Lewis, George. "Too Many Notes: Computers, Complexity and Culture in Voyager." *Leonardo Music Journal* 10 (2000): 33-39.

## Dissertations and Papers

Bahn, Curtis and Dan Trueman. "Interface: Electronic Chamber Ensemble." ACM CHI Workshop in New Interfaces for Musical Expression, Seattle, April 1-2, 2001.

Ciufo, Thomas. "Design Concepts and Control Strategies for Interactive Improvisational Music Systems." Brown University, Providence, R.I. 2003.

David Wessel, Matthew Wright and John Schott. " Situated Trio: An Interactive Live Performance for a Hexaphonic Guitarist and Two Computer Musicians with Expressive Controllers." University of California, Berkeley, Calif. 2002.

DuBois, Roger L. "Applications of Generative String Substitution Systems in Computer Music." Ph. D. diss., Columbia University, N.Y. 2003.

Jehan, Tristan and Bern Schoner. “An Audio-Driven Perceptually Meaningful Timbre Synthesizer.” *Proceedings of the International Computer Music Conference*, *Havana, Cuba. 2001.*

Jordá, Sergi. “New Musical Interfaces and New Music-Making Paradigms.” Pompeu Fabra University, Barcelona, Spain. 2001.

Manuals

Behringer. *MIDI Foot Controller FCB1010 User's Manual*. Germany: Behringer Spezielle Studiotechnik GmbH, 2004.

Digidesign. *What's New in Pro Tools LE 6.7: For LE Systems on Windows or Macintosh.* Daly City, Calif.: Digidesign, 2004.

Glyph Technologies. *Gt 050 User's Manual*. Ithaca, N.Y.: Glyph, 2005.

Midiman. *Radium USB MIDI Keyboard Controller.* Irwindale, Calif.: M-Audio, Inc., 2003.

Native Instruments. *NI Komplete 2: Operation Manual*. Berlin: Native Instruments Software Synthesis GmbH, 2004.

Rogers, Doug., Nick. Phoenix, and Brian. Tester. *Symphonic Orchestra: Operation Manual*. West Hollywood, Calif.: East West Sounds, Inc., 2003, 1-15, 93-97.

Roland Corporation. *Divided Pickup GK-MIDI Interface: Owner's Manual*. Los Angeles, Calif.: Roland Corp, 2003.

Roland Corporation. *GI-20 GK-MIDI Interface: Owner's Manual*. Los Angeles, Calif.: Roland Corp, 2003.

Roland Corporation. *XV-5050 64 Voices 2x Expansion: Owner's Manual*. Los Angeles, Calif.: Roland Corp, 2001.

Synkron. *Reason 2.5: Operation Manual*. Sweden: Propellerheads Software, 2003.

Computer Programs

Max/MSP Ver 4.5, developed by Miller Puckette (MIT) and David Zicarelli during 1995-2005.

PeRColate Ver. 0.9 beta 5 for Max/MSP 4.5 (OSX), created by Dan Trueman at Princeton University and R. Luke DuBois at Columbia University.

Real-time Composition Library Ver. 4.0.2, created by Karlheinz Essl during 1992-2004.

The Synthesis Toolkit in C++ (STK) Ver. 4.2.1, developed by Perry R.Cook (Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University and Princeton University ) and Gary P. Scavone (McGill University) during 1995-2005).

Related Web sites

Software:

*Stanford University*

<http://ccrma.stanford.edu/software/stk/information.html>.

*Cycling 74*

<http://www.cycling74.com/products/max/>.

*Digidesign*

<http://www.digidesign.com/>.

*Karlheinz Essl personal webpage*

<http://www.essl.at/software.html>.

*IK-Multimedia*

<http://www.ikmultimedia.com/>.

*Columbia University*

<http://music.columbia.edu/PeRColate/>.

*Native Instruments*

<http://www.nativeinstruments.de/index.php?home_us>.

*Waves*

<http://www.waves.com/>.

Guitars and Guitar Effects

*Boss*

<http://www.bossus.com/>.

*Carvin*

<http://www.carvin.com/products/guitar.php?ItemNumber=AE185>.

*Digitech*

<http://www.digitech.com/>.

*Line 6*

<http://www.line6.com/products/instruments/

# APPENDIX A: Hardware Setups and Software Settings

*The informal nature of this appendix is designed as an advanced user's manual rather than appearing as a formal presentation of material.*

## CHAPTER 2

### MIDI in Music Production

<u>Example 1</u>: Multiple Destinations: Settings

1. Create 3 stereo aux tracks.
2. Insert a virtual instrument in the aux track 1.
3. Insert a virtual instrument in the aux track 2.
4. Insert an external sound module such as Roland XV 5050 (Select SPDIF in the Aux 3 Input).
5. Press control-click in the MIDI output selector and choose the proper virtual instruments and Roland XV 5050 channels. A + sign has to appear in front of the first aux track name in the MIDI output selector.
6. To record the sum of the three aux channels, send each signal to a stereo audio channel.

### Extending the Current Guitar Production Techniques

<u>Example 2</u>: Adaptation of Dave Jerden's triple tracking concept. Settings:

1. Open a Pro Tools session and create two mono audio tracks, six stereo auxiliary tracks and one stereo master track.
2. Audio track one: Input Acoustic Guitar Output (Ch 3), Output Analog 1-2.
3. Audio track two: Input Electric Guitar Output (Ch 4), Output Analog 1-2.
4. Auxiliary track one: Insert SansAmp, choose a preset that works fine in the low range.
5. Auxiliary track two: Insert Amplitube, choose a preset that works fine in the middle range.
6. Auxiliary track three: Insert Guitar Rig 2, choose a preset that works fine in the high range.

7. Repeat steps 4-6 with the Auxiliary tracks 4, 5 and 6.
8. Audio track one: send the signal via bus to aux track 1, 2 & 3.
9. Audio track two: send the signal via bus to aux track 4, 5 & 6.
10. Explore, pan, and assign volumes until the desired tone is achieved.

> *This is a creative adaptation of Dave Jerden's triple tracking concept. In addition, I can send the signal to 6 or 7 different amplifier modeling plug-ins at the same time and treat them as individual amplifiers, applying in this way the splitter technique.*

Example 3: Adaptation of Alan Moulder's technique. Settings:

1. Open a Pro Tools session and create one mono audio track, one stereo audio track three stereo auxiliary tracks and one stereo master track.
2. Audio track one: Input Acoustic Guitar Output (Ch 3), Output Analog 1-2
3. Audio track two: Input JMP Analog 5-6, Output Analog 1-2.
4. Auxiliary track one: Insert an amplifier mod plug-in of your choice that has contrasting or complementary sound to the auxiliary track two.
5. Auxiliary track two: Insert an amplifier modeling plug-in that emulates tube preamps.
6. Auxiliary track three: Insert an amplifier modeling plug-in that emulates solid-states preamps.
7. Find an exciting virtual cabinet-speaker match for each aux track.
8. Audio track one: send the signal via bus to aux track 1.
9. Audio track two: send the signal via bus to aux track 2 & 3.
10. Explore, pan and assign volumes until the desired tone is achieved.

> *In this case, I am adapting the Alan Moulder's technique. One step further will be to send the Digi 002 outputs to two different guitar amplifiers and use the reamping technique, recording the amplifiers outputs and returning the signal to Pro Tools.*

## Digital Signal Process and Audio Plug-ins

Example 4: Plug-In Automation: Settings:

A) To enable plug-in controls for automation:

1 Open the selected plug-in window.
2 Click the automation button in the plug-in window.
3 Choose the controls to automate and click add.
4 Click ok to close the plug-in automation dialog.

B) To automate a plug-in:

1 In the automation enable window, enable plug-in automation.
2 Select an automation mode for each track and choose auto write for the initial automation pass.
3 Click play to begin writing automation, and move the desired controls.
4 After the completion of step 3, click stop.

Example 5: Side-Chain Processing

From the key input menu, choose the selected input or bus carrying the audio that will be use to trigger the plug-in and then:

1. Click external key to activate side-chain processing.
2. Press key listen in order to hear the audio source you have selected to control side-chain input.
3. To filter the key input so that only specific frequencies trigger the plug-in, use the Key HPF and Key LPF controls (if available) to select the frequency range.
4. Begin playback. The plug-in uses the selected input or bus as a key input to trigger its effect.
5. Adjust the plug-ins threshold control (if available) to fine-tune key input triggering.

Example 6: Plug-In in the Way In

In order to avoid clipping on/or saturation the engineer could setup the Pro Tools session like this:

1. Open one stereo auxiliary track, one stereo audio track and a master fader.
2. Auxiliary track input: select the proper channel in which you are plug-in the guitar. Auxiliary output: bus 1-2.
3. Stereo audio track input: bus 1-2, output: analog 1-2.
4. Auxiliary track insert: Select a compressor suitable for acoustic guitar strumming parts.
5. Press rec in the audio track and test the levels and common compressor’s settings.
6. Hit 3 and start recording. Press 0 to stop. Repeat the steps 5-6 or save the take.

Example 7: Midi Guitar Setup

In order to use an electric guitar (with the Roland GK 3 installed) as a MIDI controller and be able to route the signal to an external sound module, the musician has to follow the next instructions:

1. Connect the GK cable from the MIDI guitar to the Roland GI-20.
2. Connect from the Roland GI-20 MIDI out to the Roland XV-5050 Sound Module MIDI In.
3. Connect from the Roland XV-5050 MIDI Thru to the Digi 002 MIDI In.
4. In Pro Tools hardware setup: clock: internal, digital input: optical.
5. In Pro Tools mixer window: MIDI track in: Digi 002 Port 1, out: Roland XV 5050 Aux Track in: S/PDIF, out: analog1-2.
6. Click rec in the MIDI track.
7. Change the output of the aux track to bus 1-2.
8. Open a stereo audio track and set its input to bus 1-2 and its output analog 1-2.
9. Click rec in the audio track and press 3 to start recording and 0 to stop it.

Example 8: Playing Reason's Synthesizers with the MIDI Guitar

In order to be able to play with the MIDI guitar Reason's synthesizers the musician should repeat steps 1 to 4 and then continue:

5. In the Pro Tools mixer window: aux track insert: Reason Mix L-R, in: M1/2, out: analog 1-2.
6. MIDI track in: Digi 002 Port 1, out: Subtractor 1 ch 8.
7. Click rec in the MIDI track.
8. Change the output of the aux track to bus 3-4.
9. Open a stereo audio track and set its input to bus 3-4 and its output analog 1-2.
10. Click rec in the audio track and press 3 to start recording and 0 to stop it.

Example 9: MIDI Guitar: Orchestration Example

In order to orchestrate a particular composition, the composer has to keep the same settings aforementioned above and continue:

- Go to Reason, Create Malstrom.
- In Pro Tools open a new MIDI track, MIDI in Digi 002 Port 1, out Malstrom.
- In the audio track press rec.
- Pressing rec in the MIDI track 2, 3 and audio track 2 the performer will be able to record the Subtract or and Maelstrom synth in one stereo track.

Example 10: Special Settings

If the musician wants to bring 2 mono outputs from Reason via ReWire into a Stereo Audio track in Pro Tools using a multi-mono plug-in he/she needs to follow the instructions:

- Create a stereo aux track in Pro Tools (LE).
- Initiate the multi-mono Reason ReWire plug-in on the insert of that track.
- In the plug-in window click on the 'link' button to disable it.
- Click on the box to the left of the link button. This will allow the musician to select the left or right side of the stereo track and assign a ReWire input to it.

Example 11: Individual Mixer Settings

If the musician needs to make individual mixer settings for a separate device - e.g. add a RTAS effect to a Subtractor sound and/or add some compression or other effect to an individual Redrum channel, he/she should proceed as follows:

1. In Pro Tools, activate ReWire channels 3 and 4.

2. In Reason, flip the rack around by pressing [Tab].Locate the Subtractor device and disconnect it from the Reason mixer. In its place, connect it to input nr 3 on the Hardware Interface at the top of the rack.

3. Find the Redrum device and the desired drum sound channel in which the effect will be applied. Connect the individual output of that drum channel to input nr 4 on the Hardware Interface.

4. Press Rec in Pro Tools in channels 3-4 and monitor the Reason Hardware Interface. Now the sound of the Subtractor device appear in the channel 3 and the individual drum sound on channel 4.

Example 12: Recording automation from a ReWire application

If the musician wants to record automation from the ReWire application into Pro Tools (actually, only Ableton Live supports this feature) the next instructions must to be followed:

1. In Pro Tools: Enable Pro Tools virtual MIDI inputs (MIDI>Input Devices.

2. Create a MIDI track and select one of the virtual MIDI inputs.

3. In your ReWire application: route MIDI data into Pro Tools via the virtual MIDI input bus. Then the musician will be able to record MIDI data, including automation, from your ReWire application.

# CHAPTER 3

# Max/MSP-MUSIC PRODUCTION SOFTWARE INTERACTION

Example 1: Sending Audio from Max/MSP to Pro Tools via ReWire

In order to be able to fuse both programs, the following instructions have to be completed:

1. Open Pro Tools, start a session. Create one stereo aux track and one stereo audio track.
2. Route the output of the aux track to the input of the stereo audio track via bus 1-2.
3. Insert Max/MSP in the aux track. For example, choose 1-0.
4. Open your Max/MSP patch.
5. Go to the DSP status and set it in the following way:
   Audio: On,
   Driver: ad_rewire,
   Output channel 1: 1 ReWire Out 1
   Output channel 2: 1 ReWire Out 2
6. Playback the proper file or patch and check the output levels.
7. Go back to Pro Tools and check the aux and audio tracks inputs levels.
8. Press rec in the audio track and hit the number 3 (numeric pad) to begin recording.
9. Press the spacebar to stop the recording. Check the result and decide if you want to keep the take or go back and record again.
10. Close the VST Plug-in Controller Interface first and then close your Pro Tools session.

Example 2: How to use VST Plug-ins in Pro Tools via Max/MSP

1. Open Pro Tools, start a session. Create one stereo aux track and one stereo audio track.
2. Route the output of the aux track to the input of the stereo audio track via bus 1-2.
3. Insert Max/MSP in the aux track. For example, choose 1-0.

4. Open the VST plug-in controller interface (Max/MSP patch).
5. Go to the DSP status and set it in the following way:
   Audio: On,
   Driver: ad_rewire,
   Output channel 1: 1 Rewire Out 1
   Output channel 2: 1 Rewire Out 2
6. Press plug and select your desired VST plug-in.
7. Click open in order to select the proper patch, bank or custom preset.
8. Click params and select the parameter that you want to control in each of the eight menus. Visually check the correspondence between the movements of your physical knob or fader and the software synthesizer.
9. Play and check the output levels.
10. Go back to Pro Tools and check the aux and audio tracks inputs levels.
11. Press rec in the audio track and hit the number 3 (numeric pad) to begin recording.
12. Press the spacebar to stop the recording. Check the result and decide if you want to keep the take or go back and record again.
13. Close the VST Plug-in Controller Interface first and then close your Pro Tools session.

Example 3: VST Plug-In Controller Interface v2

For instance, repeat the aforementioned steps 1-9 and then:

10. Open the slider automation patch, select the proper slider (1-16), select record from the menu and move the proper fader/know in the MIDI controller.
11. Click stop and then play. The fader/know movement has to reflect the performer previous actions.
12. Go back to Pro Tools and check the aux and audio tracks inputs levels.
13. Press rec in the audio track and hit the number 3 (numeric pad) to begin recording.
14. Press the spacebar to stop the recording. Check the result and decide if you want to keep the take or go back and record again.
15. Close the VST Plug-in Controller Interface first and then close the Pro Tools session.

# APPENDIX B: DVD CONTENT

## CHAPTER 2

### MIDI in Music Production

Example1: MIDI Guitar Techniques: Poly and Mono Mode

Example 2: MIDI Interface Controllers Functions

Part a MIDI Guitar GK3

Part b Auxiliary Controllers

Example 3: MIDI Network

Example 4: MIDI Guitar Techniques: Multiples destinations

### Extending the Current Guitar Production Techniques

Example 5-8: Extended Stereo Guitar Techniques (Table 6 options A, O, U, W)

Ex.5 A

Ex.6 O

Ex.7 U

Ex.8 W

Example 9: Extended Stereo Guitar Techniques: Adaptation of Dave Jerden's Triple Tracking Concept

Example 10: Extended Stereo Guitar Techniques: Adaptation of Alan Moulder's Technique

Examples 11-14: Extended Stereo Guitar Techniques:

Ex.11 Pass Ideas

Ex.12 Contemporary Jazz

Ex.13 Scofeel

Ex.14 Fussion Lead

Digital Signal Process and Audio Plug-ins

Examples 15-17: RTAS-DSP

Ex.15 Mondo Enigmatico

Ex.16 Spektral Delays

Ex.17 Granular Rain

Example 18: Plug-In Automation

Example 19: Side Chain Processing

Example 20: Plug-In in the Way In

Example 21: Multiple Outputs

ReWire: Definition and Examples

Examples 22-23: ReWire examples

Ex.22 Rewire

Ex.23 Multichannel and Multimono

Example 24: Rewire Multiple Channels

# CHAPTER 3

## Max/MSP-MUSIC PRODUCTION SOFTWARE INTERACTION

Example 1: How Max/MSP handles MIDI

Part a: Max/MSP and MIDI

Part b: Max/MSP Automation

Part c: Chromatic Invention

Example 2: How Max/MSP manages Audio

Part a: African Elements Patch

Part b: African Elements (Fragment of the live performance)

Example 3: Sending Audio from Max/MSP to Pro Tools via ReWire

Example 4: How to use VST Plug-ins in Pro Tools via Max/MSP

Part a: VST Plug-In Controller Interface v1. (Absynth patch)

Part b: Absynth's textures

Example 5: VST Plug-In Controller Interface v2

Part a: VST Plug-In Controller Interface v2. (Reaktor patch)

Part b: Fugitive Scene

# CHAPTER 4

# THE GUITAR APPLICATION TOOLKIT (GATK)

## Background: Interactive Computer Music

Example 1: AE185Acoustic/Electric Stereo Guitar

Example 2: Stereo Guitar Performance Interface (SGPI)

*GATK: Description of each application*

Example 3: Dynamic Ramp Generator

Part a: Working with soundfiles

Part b: Live Input

Examples 4-7: Sound Design Set

Ex.4: Delay Line
Improvisation A
Improvisation B
Improvisation C

Ex.5: Triggered Delay Line
Improvisation A
Improvisation B
Improvisation C

Ex.6: Stereo Granulator (Description)
Improvisation A
Improvisation B
Improvisation C

Ex.7: Sound Design Set in the FEAST

Example 8: Frequency Expander

Example 9 Virtual Sustainer (Description)

Part a Guitar Samples

Part b Live Input

Example 10: Polyphonic Dreams (Description)

Part a Polyphonic Dreams (Pro Tools)

Part b Polyphonic Dreams (Max/MSP)

# APPENDIX C: FREE/SHAREWARE VST PLUG-INS

| Plug-In name | Plug-In Type/Comments | Where to find it |
|---|---|---|
| Ambience | Reverb | www.smartelectronix.com/~magnus/ |
| The Fish Fillets | Dynamics Processing | www.digitalfishphones.com |
| Ruby Tube | Overdrive | www.silverspike.com |
| The Claw | OSCar-style overdrive and built-in delay effect. Great for bass lines | www.refx.net |
| Supaphaser 2.0 | Phase-shifting | www.smartelectronix.com/~bram/ |
| MFM (More Feedback Machine) | Delay | www.u-he.com |
| Frohmage and NorthPole | Unusual Filters | www.ohmforce.com (Frohmage)<br>www.prosoniq.com (NorthPole) |
| Cyanide | Reshaping, tuned distortion. | www.smartelectronix.com/~bram/ |
| Bouncing Ball Delay | Delays | www.smartelectronix.com/~bram/ |
| Madshifta | Pitch Shift | www.smartelectronix.com/~bram/ |
| Rebuilder | Sound Reconstruction (Robotic sounds) | www.plogue.com/e_soft.html |
| RoboBear | Granular sonic decomposer | www.tobybear.de |
| Crazy Ivan | Combination of distortion, pitch- and time-shifting, and granular delay | www.smartelectronix.com/~bram/ |