Reconfigurable Autonomous Novel Guitar Effects (RANGE)

Duncan MacConnell
MISTIC
Computer Science
University of Victoria
Victoria, BC
duncanmacconnell@gmail.com

Shawn Trail
MISTIC
Computer Science
University of Victoria
Victoria, BC
trl77@uvic.ca

George Tzanetakis
MISTIC
Computer Science
University of Victoria
Victoria, BC
gtzan@cs.uvic.ca

Peter Driessen
MISTIC
Electrical Engineering
University of Victoria
Victoria, BC
peter@ece.uvic.ca

Wyatt Page
MISTIC
Electrical Engineering
Massey University, NZ
Victoria, BC
w.h.page@massey.ac.nz

ABSTRACT

The RANGE guitar is a minimally-invasive hyperinstrument incorporating electronic sensors and integrated digital signal processing (DSP). It introduces an open framework for autonomous music computing eschewing the use of the laptop on stage. The framework uses an embedded Linux microcomputer to provide sensor acquisition, analog-to-digital conversion (ADC) for audio input, DSP, and digital-to-analog conversion (DAC) for audio output. The DSP environment is built in *Puredata (Pd)*. We chose *Pd* because it is free, widely supported, flexible, and robust. The sensors we selected can be mounted in a variety of ways without compromising traditional playing technique. Integration with a conventional guitar leverages established techniques and preserves the natural gestures of each player's idiosyncratic performing style. The result is an easy to replicate, reconfigurable, idiomatic sensing and signal processing system for the electric guitar requiring little modification of the original instrument.

Keywords

hyperinstrument, guitar, autonomous music-computing, DSP

1. INTRODUCTION

Electric guitar players have utilized audio effects since their inception. An extensive variety of DSP guitar effects are offered commercially, some of which even provide a code environment for user modification of DSP algorithms [1]; however, in most cases the functionality of these devices is specific and their programmability is limited. These commercial audio effects are typically implemented either as foot pedals or as separate hardware devices. An alternative is the use of a laptop and audio interface to replace the dedicated guitar effects. This approach is generic in the sense that any audio effect can be implemented as long as the computer is fast enough to calculate it in real-time. Using a laptop is also completely open, flexible, and programmable. However such a setup requires more cables, more power, and is cumbersome to transport and awkward on stage [13]. In both of these cases (dedicated hardware or laptop) the control of the effects is separated from the actual guitar playing as shown in Figure 1.

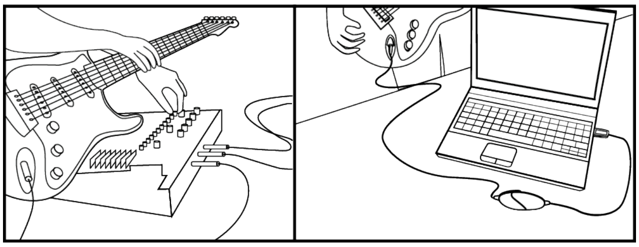

Figure 1: Typical guitar effect Interaction - note the separation of guitar playing vs. effect manipulation

Historic examples include the *Guitarorgans*, the analog synth controller *Stepp DGX MIDI guitar* [2], and the *Roland G 303*. Various systems for mobile devices acting as the DSP host have also become common [3].

Augmented guitars have also been explored. MIT′s *Chameleon Guitar* has multiple soundboards, each equipped with piezo sensors and DSP filtering to simulate the guitar tones offered from different wood [15]. Another example, the *Moog Guitar*, is an electric guitar with onboard sliders that control augmentation of the guitar′s traditional sound by sending electro-magnetic energy into strings. This allows for infinite note sustain, while similarly pulling energy from the strings creates short staccato sounds. Edgar Berdahl introduced a similar idea in his *Feedback Guitar* [2].

The *DUL Radio* [4] from the Center for Digital Urban Living at Aarhus University, Denmark is a wireless accelerom-

[1]`http://line6.com/tcddk/`

[2]`http://www.stepptechnologies.co.uk/`
[3]`http://www.incidentgtar.com/`, `http://www.misadigital.com/`

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

eter sensor package designed for artist's to use with *Pure-data (Pd)* [11] or *MAX/Msp* [4]. The group demonstrates the device by attaching an accelerometer to the headstock of the guitar for 3D gesture tracking. This makes for an easily removable, modular solution for 3D gesture tracking. Unfortunately we were not able to integrate it with the RANGES system at this time because of our LINUX requirement which isn't supported by the DUL drivers at this time.

2. MOTIVATION

In designing an augmented guitar instrument, consideration must be taken to ensure the extensions do not inhibit traditional guitar technique. Effort should be made to create intuitive control interfaces that take advantage of the guitar player's natural performance technique. Traditional audio effect units and commercial DSP solutions tend to disregard this, forcing the musician to interact with musical parameters by way of non-musical gestures: turning a knob or adjusting a fader [7]. This conflicts with the guitarist's normal gestural interaction, fails to convey any meaningful event information, and can even act as a distraction for the audience [9].

There has always been a union of guitar and effect despite a separation of guitar playing and effect control. To address this issue, we have integrated minimally invasive sensors on the body of the guitar to allow natural and intuitive DSP control. The RANGE system was designed for use in performance contexts to allow guitar players more expressivity in controlling DSP effects than conventional pedal controllers provide.

The proximity of the sensors to the guitarist's natural hand position is important, as it allows the guitarist to combine DSP control with traditional guitar playing technique. Like the *Moog Guitar*, the sensors sit flat on the guitar body, eliminating any interference with a guitarist's performance technique. Further, we have reduced the hardware dependencies, cabling, and power requirements to a minimal footprint. Design goals were motivated by the desire to shift away from the cumbersome and distracting laptop on stage in exchange for a smaller, open architecture. This framework is designed to take advantage of low-cost electronic components and free open-source software, facilitating reconfiguration and adaptation to the specific needs of different instruments and musicians.

3. SYSTEM DESCRIPTION

The RANGE system is based on a framework for developing robust hyperinstrument prototypes, which provides audio input, output, and sensor acquisition. The framework itself provides a completely open platform for designing and testing hyperinstruments. The hardware and software components that comprise this system are modular in nature and the configuration is designed so that any user can adapt this work for their own use. For this implementation, the sensor interface consists of three membrane potentiometer strips mounted on the body of the guitar which feed into the analog inputs of the embedded Linux computer. We have selected the *Beaglebone* [5] for it's flexibility and low cost. The guitar's audio signal goes directly into the *Beaglebone* for analysis and processing using an *Audio Cape* for ADC/DAC. Pitch tracking is performed on the incoming audio signal using Fiddle [10] and used to generate control data. The control data and DSP is managed in *Pd*. The potentiometer outputs are mapped to continuous controller values that modify the parameters of effect parameters, oscillators, and filters. These potentiometers offer the guitarist a broad range of interface solutions and sound design possibilities in a small embedded format that has previously been only possible with a laptop. The system has relatively low cost (all prices in US dollars): *Beaglebone* (89), audio cape (58), and membrane sensors ($3 \times 13 = 39$) for a total of 186 USD.

Figure 2 shows a schematic diagram of the system.

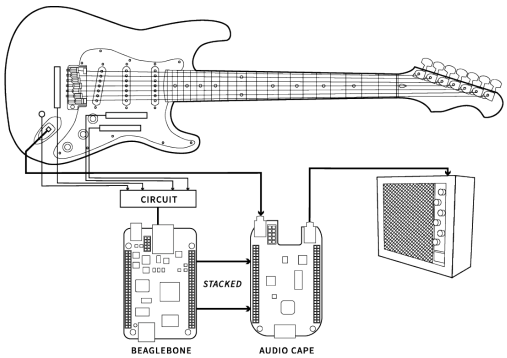

Figure 2: Schematic of RANGE

3.1 Analog Sensor Input

Membrane potentiometers are a common sensor for capturing musical data, and are often incorporated into hyperinstrument design. Adrian Freed [6] provides a detailed look at force sensing resistors, membrane potentiometers, and other sensors. His "Many and DuoTouch Augmented Guitar Prototype" [6] provides simple and elegant circuit solutions to achieve desired sensor behaviour for musical applications. The RANGE guitar is equipped with three 50*mm SoftPot* membrane potentiometers. These sensors are arranged on the body of the guitar, near the volume and tone controls. This arrangement allows the guitarist to easily access the sensors, and the orientation affords comfortable interaction. The sensors are limited to the body of the guitar corresponding to the expressive hand of the guitar player. The expressive hand, responsible for the rhythm and dynamics of the guitar, is most suited for acute sensor control. In addition to the three touch sensors, toggle switches are also mounted to the guitar to provide a simple method for switching software state.

Traditional potentiometers use the position of a sliding wiper to determine resistance. Membrane potentiometers function similarly, providing a variable resistance level based on the position of the user′s finger. The main difference is that membrane potentiometers only allow current to flow when the membrane is pressed, and so the value is lost when the user′s finger is removed. The RANGE′s sensor input behaviour must be consistent and stable in order to be used musically. Specifically, the instrument design requires that the analog input values remain when the membrane is not pressed.

In order to secure a stable and usable signal from the membrane potentiometers, pull-up resistors are used. This forces the potentiometer to open circuit when the finger is removed. A simple software solution is used within *Pd* for detecting when the membrane is forced open, and the previous buffered value is retained. Figure 3 shows the corresponding circuit.

[4]http://cycling74.com/

[5]http://elinux.org/BeagleBone

[6]http://cnmat.berkeley.edu/user/adrian_freed/blog/2009/05/09/AugmentedGuitar

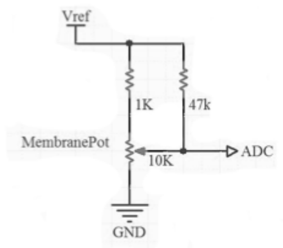

Figure 3: Membrane Potentiometer Circuit

3.2 Hardware, Software, and Latency

The RANGE uses a *Beaglebone* microcomputer, which provides on-board GPIO and ADC pin access, as well as UART, I2C and SPI. It features an ARM 600 MHz Cortex-A8 core using the ARMv7-Architecture, as well as full USB and Ethernet support. The *Beaglebone* is becoming widely supported in the embedded computing community [1], and many expansion "Capes" are being developed to provide an array of hardware interaction opportunities. The *Beaglebone Audio Cape* provides audio input and output by way of two 3.5 mm connectors, and supports sampling rates up to 96 kHz for capture and playback by way of the cape's TLV320AIC3106 codec. The system described provides a complete DSP platform, allowing users to connect to the *Beaglebone* via ethernet for rapid interface prototyping.

The *Beaglebone* hardware ships with the *Linux Angstrom Cloud9* operating system, however many users have experienced unsatisfactory audio output quality. For this framework, *Ubuntu 12.04* was used, which facilitates *Pd* installation and interfaces well with the audio codec provided by the *Beaglebone Audio Cape*. To provide access to the *Beaglebones* GPIO and ADC pins, a *Pd* external has been developed. The ADC provides 12-bit values, which are accessed by the external by directly reading the corresponding files in the userspace. The external is designed to report analog and digital pin values each time the object receives a "bang" message. In this way, pin values can be obtained at any rate, and can be coupled with other musically timed events within the patch. The control data is not altered in any way by the external that retrieves it, as it is meant to make any sensor's data available within *Pd*.

Control of digital audio effects has been a desired function of the RANGE from its inception. The stable sensor values allow for reliable control, while placing the sensors directly on the body allows the guitarist faster and more intuitive interaction. Analog values obtained by the *Pd* external can be scaled and mapped to any control. Therefore specific effect parameters (delay length, feedback level, filter frequency, etc.) are adjusted by the touch potentiometers. This simple prototyping system is robust but offers a lot of flexibility and potential. The potentiometers can also provide an intuitive control interface for synthesis applications. Some novel applications include controlling oscillator frequency, filter frequency/bandwidth, MIDI note attribute, and envelope values (attack, decay, sustain, release). This application allows the RANGE to be used as a versatile synthesizer controller while the guitar can still be played as usual. Figure 4 shows an example mapping, with sensor input controlling a typical electric guitar effect chain on the left and common DSP applications on the right.

In contrast to pure controller approaches that utilize a laptop for DSP, our goal is to use the *Beaglebone* for both control and DSP. Many modern guitar effects are actually internally implemented using a dedicated embedded DSP

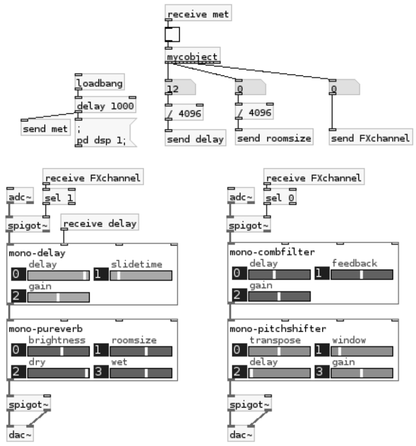

Figure 4: Common Guitar Effect Setup Built in Pd

chip even though to a guitar player they appear similar to traditional analog pedals. RANGE makes this DSP functionality accessible providing a wide range of possibilities for both digital audio effects and their control. In order to be a viable platform for this purpose it is critical that the overall system latency is appropriate for music applications. The framework provides simple sensor and audio throughput, using *Pd* with the ALSA API. For latency tests, it is important to perform measurements under different system (CPU/DSP) load applications [14]. For our system, all tests were performed in the "normal state" (audio throughput, no effects processing) as well as the "use state" (audio throughput, effects processing). In the normal state, audio latency for the system corresponds to the audio delay set by *Puredata*. With *Puredata* set to 10 ms audio delay, we measure a total system delay of 10 ms. With effects engaged, the use state latency is measured at 12 ms. This difference can be attributed to the system load increase from the signal processing. For pitch analysis, normal and use state behaved the same, measuring a total pitch analysis latency of 15 ms. All results reflect usable latency levels, for audio applications, as they approach the general latency goal of 10 ms [14, 5]. Figure 5 shows the audio and pitch-tracking latency.

3.3 String Actuation

A system for actuating the guitar strings has been developed, but has not been implemented into the RANGE guitar. Initial tests were done controlling the magnets remotely from Pd using digital oscillators from the output of the audio cape on the *Beaglebone* driving strings on an autoharp. These experiments are based on previous work by Berdahl[2], McPherson[8], Britt and Snyder[3], on electromagnetic actuation of stringed instruments.

As suggested, a low-cost transconductance amplifier based on a "gainclone" power amplifier IC is used. This design ensures input voltage controls the output current. In this way, the electro-magnet′s output is proportional to the input audio signal′s change in voltage. Using the embedded DSP framework described herein, the magnets can be driven from tuned oscillators or from an acquired audio signal, both within *Puredata*.

This design presents issues with integration directly on the instrument. The magnetic pickups on the guitar present an issue, as interference will occur with the electro-magnet. Though the magnet's required for guitar string actuation do not need to be large, they still present a design issue

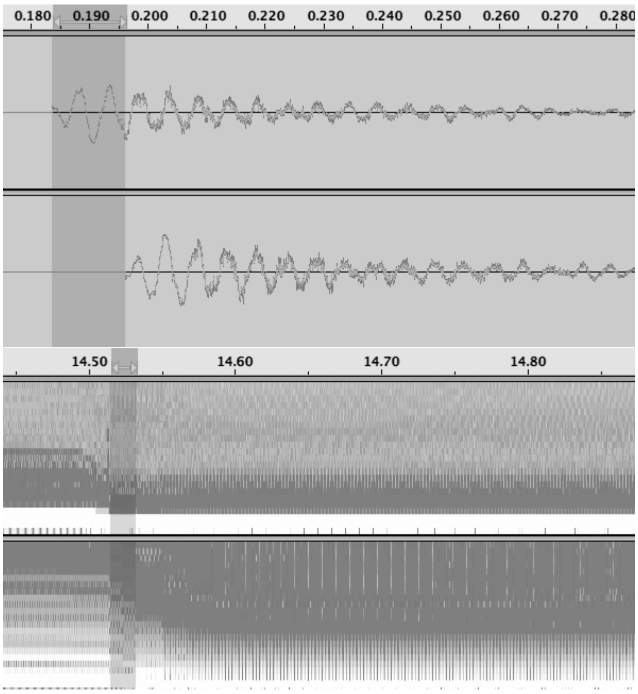

Figure 5: Audio Latency Measurements

when trying to preserve natural playing technique. Additionally, while the amplifier is relatively simple, integration of an amplifier-per-string also presents significant design constraints, contrary to the goal of the RANGE instrument design.

4. CONCLUSIONS AND FUTURE WORK

RANGE successfully presents a reconfigurable, autonomous, and novel DSP and sensing framework for the guitar. The RANGE system has been used in concert several times, and has proved to be a novel and enjoyable method of DSP interaction. During a MISITC collective performance with robotic prepared pianos designed by TRIMPIN, the RANGE sensors were used to interface with various actuators on the piano [7]. The touch sensors have also been used in performance to intuitively control synthesis and DSP effect parameters regularly by the author who is a touring guitar player. There is a website with further documentation and media related to this work [8]. Moving forward, we plan to conduct a user study contrasting the proposed approach with a traditional electric guitar effect setup. We also plan to investigate polyphonic transcription using a surrogate sensing approach [12]. In addition, we plan to extend the platform to support solenoids, motors and magnets for robotic guitar string actuation.

5. ACKNOWLEDGMENTS

We'd like to especially thank: Calum MacConnell for the the graphics; Stephen Harrison, Leo Jenkins, and Gorkem Cipli for their technical support in the lab; Pat Metheny, Mike Gamble, Kevin Patton, Edgar Bergdahl, and Dan Overholt for their inspiration and support regarding extended guitar and string based hyperinstrument expertise and inspiration. NSERC and SSHRC provided funding.

6. REFERENCES

[1] S. Barrett and J. Kridner. *Bad to the Bone Crafting Electronic Systems with BeagleBone*. Morgan and Claypool, 2013.

[2] E. Berdahl, G. Niemeyer, and J. O. S. III. Active control of a vibrating string. In *Acoustical Society of America (ASA) and European Acoustics Association (EAA)*, 2008.

[3] C. Britt, J. Snyder, and A. McPherson. The emvibe: An electromagnetically actuated vibraphone. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2012.

[4] M. Brynskov, R. Lunding, and L. S. Vestergaard. Dul radio: A light-weight, wireless toolkit for sketching in hardware. In *Tangible and Embedded Interaction Work-in-Progress Workshop Proceedings*, pages 67–72, 2011.

[5] P. F. Driessen, T. E. Darcie, and B. Pillay. The effects of network delay on tempo in musical performance. *Computer Music Journal*, 35(1):76–89, 2011.

[6] A. Freed. Novel and forgotten current-steering techniques for resistive multitouch, duotouch, and polytouch position sensing with pressure. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2009.

[7] S. Gelineck. A quantitative evaluation of the differences between knobs and sliders. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2009.

[8] A. McPherson. Techniques and circuits for electromagnetic instrument actuation. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2012.

[9] J. Murphy. The helio: A study of membrane potentiometers and long force sensing resistors for musical interfaces. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2010.

[10] M. Puckette. Real-time audio analysis tools for pd and msp. In *Proc. Int. Computer Music Conf. (ICMC)*, 1998.

[11] M. S. Puckette. Pure data: another integrated computer music environment. In *Proceedings Second Intercollege Computer Music Concerts*, pages 37–41, 1996.

[12] S. Trail. Direct and surrogate sensing for the gyil african xylophone. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2012.

[13] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

[14] M. Wright, R. J. Cassidy, and M. F. Zbyszynski. Audio and gesture latency measurements on linux and osx. In *Proc. Int. Computer Music Conf. (ICMC)*, pages 423–429, 2004.

[15] A. Zorana and J. A. Paradiso. The chameleon guitar—guitar with a replaceable resonator. *Journal of New Music Research*, 40(1):59–74, 2011.

[7] `http://mistic.uvic.ca/assets/TRIMPIN/1_Thome.html`
[8] `http://webhome.csc.uvic.ca/~trl77/RANGE.html`