StrumBot – An Overview of a Strumming Guitar Robot

Richard Vindriis
Victoria University of Wellington
School of Engineering and Computer Science
Wellington, New Zealand
richard.vindriis@gmail.com

Dale Carnegie
Victoria University of Wellington
School of Engineering and Computer Science
Wellington, New Zealand
dale.carnegie@ecs.vuw.ac.nz

ABSTRACT
StrumBot is a novel standalone six stringed robotic guitar consisting of mechanisms designed to enable musical expressivity and minimise acoustic noise. It is desirable for less than 60 dBA of noise at 1 m to be emitted to allow StrumBot to play in intimate venues such as cafés or restaurants without loud motor noises detracting from the musical experience.

StrumBot improves upon previous robotic musical instruments by allowing additional expressive opportunities for a composer to utilise. StrumBot can perform slides, vibrato, muting techniques, pitch bends, pluck power variances, timbre control, complex chords and fast strumming patterns.

A MIDI input allows commercial or custom controllers to operate StrumBot. Novel note allocation algorithms are created to allow a single MIDI stream of notes to be allocated across the six guitar strings.

Latency measurements from MIDI input to string pluck are as low as 40 ms for a best case scenario strum, allowing StrumBot to accompany a live musician with minimal audible delay.

A relay based loop switcher is incorporated, allowing StrumBot to activate standard commercial guitar pedals based on a MIDI instruction.

Author Keywords
Musical Robotics, Electric Guitar, Robotic Manipulators

ACM Classification
• Computer systems organization~Robotics
• Applied computing~Sound and music computing
• Hardware~Sensors and actuators.

1. INTRODUCTION
StrumBot is a new robotic musical instrument (RMI) based on an electric guitar which complements and builds on previous robotic instruments made at Victoria University, specifically BassBot [1], Swivel [2] and MechBass [3]. These have a physical presence and can play music outside the capabilities of a human, however they lack the ability to emulate the level of expressivity a human can produce. This project seeks to create a new six stringed robotic musical instrument that builds upon the experience gained from these earlier devices. This new device will generate less acoustical noise than its predecessors and will enable enhanced levels of expressive functionality.

The overarching goals for StrumBot are as follows:

- Total system noise to be less than 60 dB at 1 m. (typical talking volume [4]])
- Pitch to be accurate to within ±8 cents, (the just noticeable difference (JND) for pitch difference [5]).
- Play fast enough that it does not lag when playing in real time, i.e. respond within 100 ms [6].
- Strum at a rate of at least six strums per second.
- Built as a single instrument and be sufficiently intelligent to take a single input and play across all strings.
- Incorporate features that allow for additional musical expressivity.

NIME'16, July 11-15, 2016, Griffith University, Brisbane, Australia.

2. BACKGROUND
Previous musical robots have a problem with the acoustic noise created from mechanical sources, such as a carriage rubbing against a slide. This noise is not musical and interferes with the enjoyment of listening to the robots perform. While this does not matter in a concert setting, it is not suitable for playing in a home or café type environment. A normal speaking volume (measured at 1 m) is approximately 60 dB and it would be preferred if the robotic noise was below this level.

MechBass, BassBot and Swivel have all been designed for modularity as can be seen in Figure 1 in a similar way to Eric Singer's GuitarBot [7]. Each string comprises its own self-contained robot. To increase the string count an additional module can simply be added. While this approach leads to a scalable instrument, it does make the instrument difficult to program as each string must be sent its own control signal. This makes programming songs a tedious process as the user effectively has to treat each string as an individual instrument and ensure they are aligned.

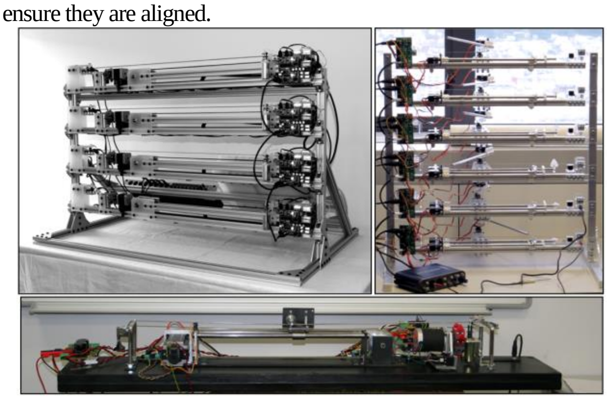

Figure 1 MechBass (TL), Swivel (TR), BassBot (Bot)

An alternative approach is to build the entire robot as one instrument, programming enough intelligence on the robot to choose which string it needs to play and how to strum it. An ideal situation would be to allow a user to write a song without having to think about the instrument that will be playing it, rather the instrument will take the song file and determine how to implement the composition. This instrument agnostic approach would also allow different instruments to all play a song from one MIDI score. This is very difficult to do, however, steps can be taken to get closer to this goal such as implementing an algorithm which receives a series of notes and allocates them into a chord shape across the strings.

The above instruments all make use of a mechanical variable pitch shifter system as opposed to fixed solenoid fretters as seen on instruments such as EMMI's PAM [8] or Compressorhead's "Fingers" [9]. This allows for microtonal changes or use of different musical scales. StrumBot also makes use of this variable pitch shifting technique.

The large majority of guitar robots utilise individual plucking systems, one motor and plectrum per string. MechBass uses two motors per string, a motor to pluck and a motor to raise or lower the plectrum, changing the pluck power. A notable exception to individual pluckers is Baginsky's Aglaopheme [10] which uses a linear motor that drives a durable cable tie (acting as a guitar pick) across all six of its strings allowing for a full strum rather than having individual pluckers per string. A solenoid raises or lowers the pick allowing for strings to be skipped. This does not allow for plucks of different power as solenoids do not allow variable positioning.

3. STRUMBOT SYSTEM

3.1 Overview

StrumBot is a standalone musical robot based on an electric guitar. As can be seen in Figure 2, it has six strings arranged in a fan shape attached to a rigid chassis, a strumming arm to pluck the strings and six individual pitch shifters to act as guitar frets, specifying the notes to be played. A control centre is also included to hold the electronics required to drive StrumBot.

A fan shape is used for the guitar strings to minimise the distance the strumming arm has to move, while leaving space for the pitch shifter motors at the opposing end.

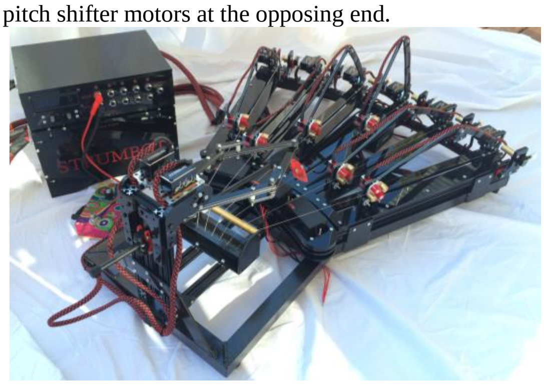

Figure 2 StrumBot

3.2 Strumming Arm

A parallel selective compliant articulated robotic arm (pSCARA) is used on StrumBot to strum the strings as shown in Figure 3. Using a robotic arm allows the pick to pluck at different points along the guitar strings' length, changing the timbre of any played notes. Playing closer to the bridge of the guitar results in a punchy, treble based note while playing farther from the bridge results in a warmer, bassier note.

The pSCARA linkages are made using Aluminium Composite Material (ACM) due to its lightweight, stiff composition. Clevis type joints are used with bearings to reduce sag between the upper and lower arms. Two high speed RC servos (ProTek 270S) provide the motion, allowing for up to 9.3 strums per second.

Figure 3 StrumBot's Strumming Arm

An end effector is attached to the end of the pSCARA to engage and disengage two guitar picks as shown in Figure 4. Two picks are used so that StrumBot can quickly transition from up-strums to down-strums with minimal motion. The pick angle is servo controlled, allowing control of the plucks' amplitude.

When the pSCARA changes angle, the end effector needs to be kept parallel to the fan shaped guitar strings. To achieve this, a carbon fibre rod is connected between the end effector and a bearing loaded pivot point mechanism located at the centre of the radius used by the strings' fan shape. This is an alternative to a second servo motor on the end effector for alignment, reducing weight and simplifying the control system.

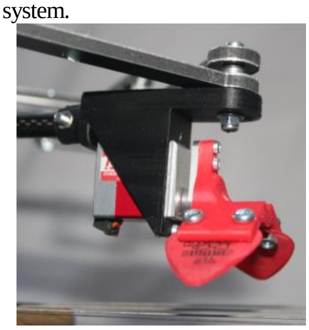

Figure 4 End Effector

Figure 5 Pivot Point

3.3 Pitch Shifters

It is desired for StrumBot to have a sliding pitch shifter, similar to that on GuitarBot or MechBass as opposed to individual solenoids for fretting the guitar string. It has been found in previous research that this allows for less robotic sounding playback as the played notes are not always identical. However, a belt drive system cannot fit, due to the string spacing not allowing space for pulleys at the bridge end of the guitar.

A linkage assembly is used as shown in Figure 6. A single servo motor changes the angle of the upper arm (shown right) which in turn moves the lower arm (left), connected via a bearing assembly. The lower arm is connected to a carriage which rolls up and down an acrylic plate. The carriage holds a clamp which is used to fret the guitar string.

Figure 6 Overhead Arm

The Pitch Shifter is capable of moving up and down the string length at a maximum speed of 1.73 m/s when using a Protek 170S RC servo motor. On StrumBot this allows for the longest octave to be traversed in 0.16 s. Tests showed that this set-up is capable of a pitch accuracy of ±8 cents, the just noticeable difference between two notes.

The clamps carriage consists of a plate of ACM with eight bearing wheels attached as shown in Figure 7. Each wheel has an O-ring acting as a tyre to reduce the noise from rolling on the acrylic plate. Four wheels are placed on top and four wheels below the plate, ensuring that it does not lift off from the plate when being pushed or pulled.

The clamp is driven by a micro scale RC servo (MKS DS-95i). This servo allows the clamp to engage or disengage in less than 0.05 seconds. Two pairs of lugs are included on the clamp. The brass pair is used for standard note playing. Brass was chosen for its good acoustic properties, most notably allowing the guitar string to have a long sustain time when plucked. The servo is used to change the pressure the lugs place on the string allowing for string bends of up to two semi-tones.

The second pair of lugs are wrapped in felt to allow for a palm muting effect. These lugs are made from aluminium to save on weight. The felt absorbs the strings' vibrations, allowing for dynamic control of the plucks' sustain by varying the pressure of the lugs on the string.

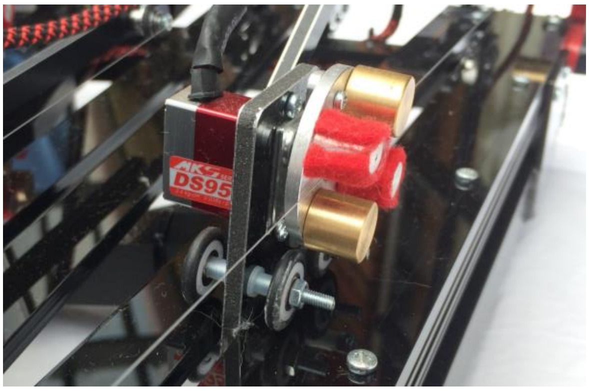

Figure 7 Clamp Carriage

3.3.1 String Selection Algorithm

StrumBot should be able to play guitar chords without large movements between different fingering positions in the same way that a human guitarist can play a set of chords with minimal hand movements. Various algorithms were designed and tested in a Java simulator to allow StrumBot to receive a set of notes (or a chord) via MIDI and allocate them to specific strings. Other robotic instruments such as Swivel and MechBass do not implement this type of algorithm, rather each string requires MIDI notes allocated specifically to it.

The first algorithm developed is a simple sort based strategy. A chord is inputted to the system, sorted and then the first note is placed on the first string that can play it. The first note is then removed from the system and it iterates to the next string. This algorithm works well for chords that use all the strings such as the 'F major' chord. However when a string is not typically used then the simulated chord will be played higher up the neck. It should be noted that musically the simulated chord is correct, however it leads to a large movement when transitioning between chords.

Common guitar chords can generally be played below the 5th fret. If the available range is restricted to this area then the problem with large movements is reduced. With this modification, any notes which would be played beyond the 5th fret are pushed to the next string.

The obvious problem with this methodology is that a chord will not be able to use a note from above the 5th fret. Chords can be played in different positions of the neck to accentuate different frequencies. For example, if a chord is played higher on the neck it will use higher voiced notes. This problem can be solved by relaxing the 5th fret restriction into a preference. The algorithm is modified so that it checks if the note will be going over the 5th fret before moving on. It also checks to see if there are enough strings to play the remaining notes in the chord. If there is not enough space then it will break the 5th fret restriction rule. This allows it to successfully play higher voiced chords.

The final algorithm developed uses an average weighted moving window. The pitch shifters default to the third fret position. The first chord tries to find a way it can be played by moving each pitch shifter a maximum of one fret position, if this is not possible it retries with a maximum movement of two fret positions. This is repeated with a growing window until the chord can be played. The pitch shifters' position is then saved to a weighted average of the previous ten chords. This allows the pitch shifters to stay in a similar location if possible, thus reducing pitch shifter movements.

3.4 Electronics

The majority of the electronics used to control StrumBot are found in a standalone box shown in Figure 8. The top compartment includes a Teensy 3.1 Microcontroller board, MIDI input, Human-Machine interface (HMI) and an effect pedal switcher. The lower compartment includes the switch-mode power supplies used to power the motors and electronics.

The Teensy 3.1 is chosen due to its high power in a small package. It has enough I/O and UARTs to control everything required and compatibility with the Arduino language. It also includes a DAC which could be used for future work such as auto tuning which has not yet been implemented.

Instructions are given to StrumBot via a MIDI input. A standard circuit is implemented in line with the CA-033 MIDI electrical specification [11]. An HMI is implemented using a 16×2 LCD screen and two encoders. This allows the operator to change parameters or test StrumBot without an external controller.

A relay based effect pedal switcher is also included. Five loops are implemented, allowing a MIDI message to active or deactivate them. Placing a guitar effect pedal into a loop (as can be seen in Figure 2) allows StrumBot to significantly change the output sound mid-song, for example, enabling a distortion pedal for a guitar solo.

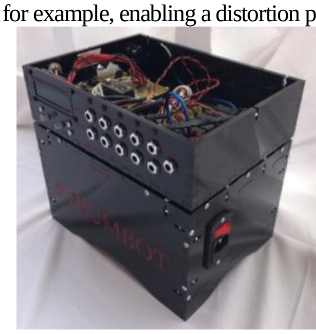

Figure 8 Control Centre **Figure 9 Servo Controller**

In addition to the main control centre, a Pololu Mini Maestro 24 is housed in a small breakout box mounted to the bottom of StrumBot. This allows the Teensy microcontroller to control all 15 servos with one serial pin, leaving I/O free for the other components. The Maestro also controls acceleration and top speed settings for the servo motors. This is located on StrumBot to allow tidy wiring. All servo wires are wrapped with cable sleeving and fixed to the frame to prevent tangles.

4. EVALUATION

4.1 Emitted Noise Results

A major goal of StrumBot is to operate at low acoustic noise levels. This was identified to be a problem on previous RMI's, where high noise emissions reduced listening enjoyment. Throughout the development of StrumBot, components such as motors and mechanical parts were chosen for their low noise emissions. Tests were performed to compare the emitted noise output of StrumBot with MechBass, Swivel and against the target of 60 dBA at 1 m.

These tests included moving a single pitch shifter at a speed similar to MechBass and at maximum speed, moving all pitch shifters, oscillating a single clamp and all clamps, moving a pitch shifter while the clamp is engaged on a wound and a unwound string, plucking all the strings at one pluck/strum per second, plucking all strings at maximum speed and playing a typical song. Results and comparisons to other RMI's MechBass and Swivel can be seen in Table 1. All measurements have an error range of ±1.5 dBA.

Table 1 Noise Emission Results

Test	MechBass (dBA at 1 m)	Swivel (dBA at 1 m)	StrumBot (dBA at 1 m)
Single pitch shifter move (Time to move one octave)	67.2 (341 ms)	53.8 (82 ms)	43.9 (355 ms) 51.8 (144 ms)
Multi pitch shifter move (Time to move one octave)	72.9 (341 ms)	63.8 (82 ms)	48.9 (355 ms)

			59.0 (144 ms)
Single PS clamp	48.3	37.3	32.2
Multi PS clamp	60.0	47.5	40.0
Single PS move (clamped on wound string)	N/A	61.9 (82 ms)	62.3 (144 ms)
Single PS move (clamped on unwound string)	N/A	54.1 (82 ms)	59.2 (144 ms)
Plucking noise multi-string	64.7 (1 PPS)	65.2 (1 PPS)	46.9 (1 SPS)
Pluck noise multi-string (Top speed)	71.8 (8.7 PPS)	61.4 (6.8 PPS)	60.0 (9.3 SPS)
Typical song test	69.2	60.4	56.3

It can be seen that acoustically, StrumBot outperforms the other RMI's. This is to be expected as neither of the other RMI's were designed to minimise mechancial noise. It can also be seen that StrumBot remains under its 60 dBA goal at 1 m in all but the most extreme situations (very high strumming speeds and moving at maximum speed with the clamp engaged).

4.2 Musical Expressivity Results

A number of mechanisms were designed to allow StrumBot to implement features that enable musical expressivity. A robotic arm was built to adjust the position of the pick placement along the length of the strings, modifying the timbre of the output signal. An end effector is implemented that permits the pick angle to be changed, increasing or decreasing the pluck's power. The arm can change speed, enabling the string to be hit slowly or quickly. A servo controlled clamp carriage is designed to allow the string tension to be adjusted while the string is sounding, modifying the pitch. A mute was also built into the clamp for variable dampening control.

4.2.1 Timbre Control

An advantage of having a robotic arm strumming the string as opposed to individual pluckers is that the pick can be moved closer or further from the bridge to adjust the timbre of a plucked note.

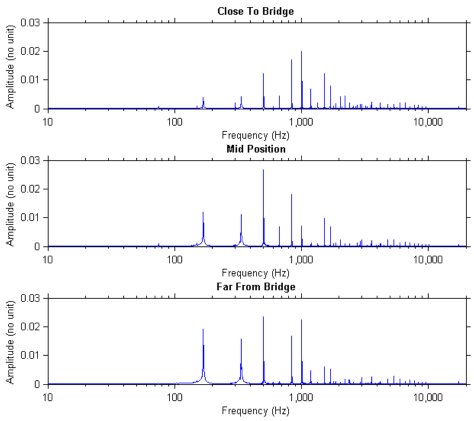

Figure 10 Strumming Arm Reach Frequency Response

The pickup output signal for three single plucks are shown in the frequency domain in Figure 10. The fundamental frequency (pitch shifter position) of each pluck is kept constant at 164 Hz (E3). The strumming arm is moved to three set positions, 10 mm from the bridge ("Close To Bridge"), 50 mm from the bridge ("Mid Position") and 90 mm from the bridge ("Far From Bridge"). It can be seen that the fundamental frequency and lower harmonics are very small when playing close to the bridge and increase as the pick is moved away from the bridge. This means that the plucks far from the bridge sound bassier, whilst the plucks close to the bridge sound thinner, as expected. It should be noted that the difference between plucking very close to the bridge and far from the bridge is audibly significant, matching the below results.

The above measurements can be expressed as a single number by using ChucK [12] and performing a Spectral Centroid analysis. A test was performed on a variety of notes where the arm would pluck the string in different positions and the centroid frequency was recorded. The test results for MIDI note 60 is shown in Figure 11. It was found that the centroid could be adjusted by at least ±18 % over the range of StrumBot by moving the position of the strumming arm.

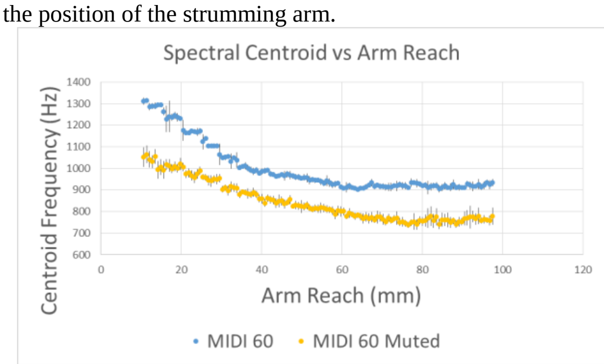

Figure 11 Centroid Frequency vs Distance From Bridge

4.2.2 Peak Loudness and Decay Control

The manipulators on StrumBot allow the attack of a plucked note and its decay to be controlled. The attack of a note can be measured by recording a pluck and using ChucK to output the peak RMS power output. A test was performed adjusting the pick angle and plucking the same note repeatedly.

The dynamic range between two signals is defined by Equation 1. It can be seen in Figure 12 that StrumBot can pluck notes with an RMS amplitude between 0.017 and 0.186 (relative units), producing a dynamic range of 20.1 dB without changing the strumming speed or pitch shifter clamp pressure.

$$DNR = 20 * \log_{10}\left(\frac{A_{louder}}{A_{Quieter}}\right) \qquad \textbf{Equation 1}$$

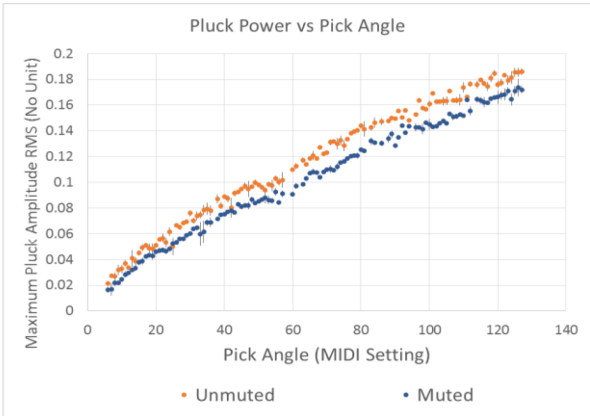

Figure 12 Pluck Power vs Pick Angle

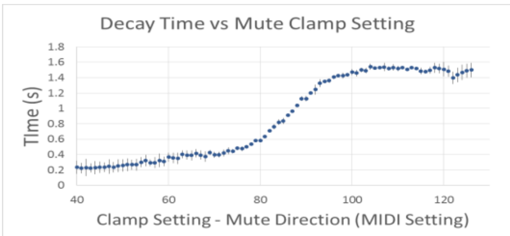

Figure 13 Decay Time vs Mute Clamp Setting

The decay of the note can be adjusted by using the muting lugs on the pitch shifter's clamp. A test was performed by adjusting the pressure the clamp applies to the string and measuring the time a pluck takes to decay from peak power to 1/10th peak as shown in Figure 13. It can be seen that the sustain time can be adjusted from 0.2 s to 1.5 s. This is an extremely useful element to be able to modify. A note can be played and almost instantly stop, or it can be played with a longer decay time, depending on the operator's instruction.

4.2.3 *Pitch Control*

Changing the clamp pressure also changes the string's pitch as clamping tighter increases the string tension. This allows StrumBot to vary the output pitch without moving the pitch shifter. This is used for pitch bending or the vibrato technique (quickly adjusting pitch in an oscillating manner).

A test is performed five times, recording the fundamental frequency while tightening the clamp from the lightest clamp setting to a point where tightening the string no longer had an effect on pitch (i.e. the servo was not strong enough to clamp tighter). The results from this test are shown in Figure 14. StrumBot can bend almost 100 cents (one semi-tone) in either direction from a given note. This is sufficient to perform vibrato or small pitch bends. If larger pitch bends are required then clamping softly and moving the pitch shifter is the preferred method (although, this technique is acoustically louder than moving the clamp in place).

StrumBot defaults to a MIDI control setting of 75, allowing pitch bends in both directions. This is different to a normal guitar, where a guitarist can only tighten a string by bending. To loosen the string, some guitars are equipped with a floating tremolo bridge. Figure 4-5 illustrates that StrumBot can emulate this hardware feature as well. If this is not desired then the default setting for engaging the clamp can be set to 10 and pitch bends of two semi-tones are achievable.

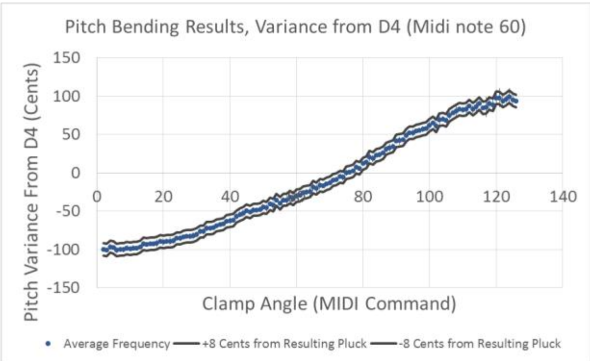

Figure 14 Pitch Bending Results

4.3 Latency

The main disadvantage of having moving components such as the strumming arm and linear pitch shifters is that there is inherently a lag between when a message is received and the mechanisms are moved to their required starting position compared to static mechanisms such as individual plucking mechanisms or discrete solenoid based pitch clamps.

It was desired for StrumBot's' lag to be under 100 ms due to the precedence effect, the defined time between two notes that is observable. To test if this goal was achieved a number of tests were performed with key points timed. The following scenarios were tested with results shown in Figure 15.

1. A full strum across all six strings – this is the most typical usage of StrumBot
2. Repeatedly playing the same note
3. Moving the pitch shifter five fret positions and then playing
4. Playing the external strings (string 1 and 6) without moving the pitch shifters
5. Playing the external strings (string 1 and 6) while moving the pitch shifters five fret positions
6. Playing neighbouring strings (string 3 and 4) without moving the pitch shifters

Four key points have timers attached to them within StrumBot's firmware. 50 plucks of each scenario were performed and sent through the microcontroller serial port to a PC. The first measured point is how long the MIDI command takes to process (blue section). This takes 5 ms due to the programming on the MIDI receiver. The strumming arm and pitch shifters then move into position (orange section). This is only 5 ms for the full strum as it is already in the starting position (5 ms is an overhead ensuring everything is in position). This stage takes the longest for the large string changes, where the strumming arm has to start moving from the first string, move to the gap between the 5th and 6th, stop and then engage the pick before moving again to pluck. This is slower than when simply strumming across all strings when the strumming arm does not need to stop along the way. This stage is faster when the pitch shifter is repeatedly playing the same string, but is still the most significant stage when not moving the pitch shifter.

The third stage is engaging the guitar pick (grey section). This is set to a constant 20 ms to ensure that the pick is engaged properly. This stage is not observable in the scenario where the pitch shifter is moving and the strumming arm's target string does not change. This is because the pick can be engaged once the strumming arm is in position, while waiting for the pitch shifter to complete its movement.

The final stage is plucking the guitar string (green section). This is generally quick, as everything is in position and the arm simply has to move to the string. This test measures when the string is plucked, not when the pitch shifter comes to a rest. This time dominates when strumming across all strings as the strumming arm movement takes time to traverse the strings. Although the full strum takes around 140 ms to complete its strum, strings begin to sound at around 40 ms, significantly lower than the 100 ms goal. Plucking the same note also sits below the 100 ms target. The other scenarios do exceed this time limit, although they could be improved with the use of pre-positioning the strumming arm intelligently (or via the use of pre-positioning MIDI commands).

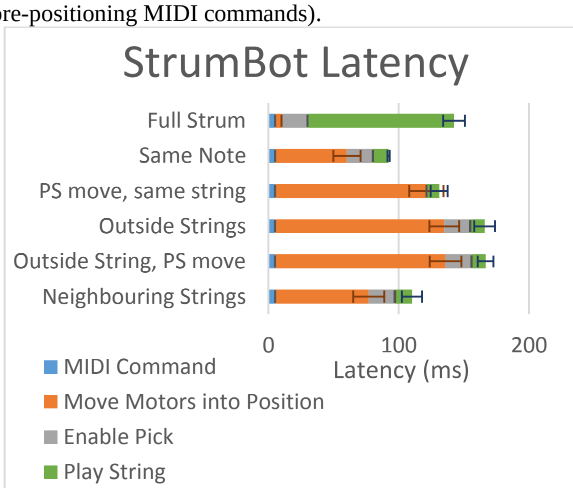

Figure 15 Response Time of StrumBot

5. CONCLUSION

StrumBot is a new type of robotic instrument with multiple musical expressivity control options and low acoustic noise emissions. Musical expressivity options were enabled by using continuous systems such as a robotic arm for strumming instead of discrete pluckers, a slot guided variable positioning pitch shifter instead of discrete solenoids and a variable pressure clamp instead of a system which is always engaged or only has the binary options, engaged and disengaged.

A parallel SCARA strumming arm was designed to emulate a human's strumming motion. The parallel SCARA has a maximum speed of 9.3 strums per second, allowing for 55 notes per second to be played across StrumBot's six strings. An end effector was implemented that passively aligns a guitar pick with the fan shaped string layout. Two guitar picks are mounted to the end effector servo, allowing for fast directional changes. This end effector can adjust its pick angle, changing the RMS pluck power with a dynamic range of 20.1 dB with a precision of 0.79 dB, allowing for quiet and loud plucks to be performed.

The main reason for using the strumming arm as opposed to individual pluckers was to allow for the pick to hit at different points along the string, adjusting the timbre of the resulting signal. This was highly successful, allowing the centroid frequency to be adjusted by at least ± 18 % for any note.

The strumming arm emits up to 60 dBA at 1 m when moving at the maximum speed (9.3 strums per second). It reduces to 55.4 dBA at six strums per second and outputs 46.9 dBA at one strum per second.

A pitch shifter system was designed to work in conjunction with the strumming arm. An overhead arm design was implemented, allowing for relatively bulky motors to be placed at the neck end while minimising the space between the strings at the bridge end. This minimises the strum distance required. A servo controlled clamp was mounted to a carriage designed to pinch the string with brass lugs and mute the string with felt padded lugs.

This pitch shifter is capable of very high speeds, moving one octave in 144 ms. For comparison MechBass can move an octave in 341 ms and Eric Singer's GuitarBot moves at a rate of 250 ms per octave. The pitch shifter is accurate to ±8 cents or better.

The adjustable clamp is a novel mechanism which is not used on any other RMI. Adjusting the target angle of the clamp allows the pitch of the note to be modified with a range of two semi-tones. It can also turn the opposite direction to act as an effective mute, allowing the sustain time (defined as the time from peak to 1/10th power) to be controlled between 0.2 to 1.5 seconds.

Previous RMIs from Victoria University (MechBass and Swivel) utilise a separate MIDI channel for each string. This requires significant effort from the operator to program a score of music. StrumBot implements a number of algorithms to predict the next note and choose which string a note should be played. This simplifies the operation of sending MIDI notes significantly as the operator does not need to visualise exactly how StrumBot will implement the input.

A gap in previous robotic guitar designs is the ability to activate guitar pedals during playback. Effect pedals are one of the most important factors when creating a sound and adding expressivity to a song. For example a distortion pedal may be switched in for a guitar solo, allowing the single notes to stand out. StrumBot solves this problem by implementing a relay based effect switcher, allowing standard effect pedals to be switched in and out of the guitar signal via a MIDI command with no audible delay.

A Teensy 3.1 microcontroller controls StrumBot with the help of a Pololu Mini Maestro 24 Servo controller. A human machine interface was implemented for diagnostic and setting adjustment. The MIDI protocol was implemented to allow control signals to be sent to StrumBot. Active guitar pickups are used to transform the string's vibrations into an audio signal with minimal cross-noise from the servo motors.

6. REFERENCES

[1] R. G. Vindriis, D. A. Carnegie and A. Kapur, "A Comparison of Pick-Based Strategies for Robotic Bass Playing," in *Electronics New Zealand Conference*, Palmerston North, 2011.

[2] J. McVay, D. A. Carnegie, A. Kapur and J. W. Murphy, "Designing and Building Expressive Robotic Guitars," in *New Interfaces for Musical Expression*, Daejeon, Korea, 2013.

[3] J. W. Murphy, J. McVay, D. A. Carnegie and A. Kapur, "MechBass: A Systems Overview of a New Four-Stringed Robotic Bass Guitar," in *Enzcon 2012*, Otago, 2012.

[4] The Engineering Toolbox, "Voice Level and Distance," [Online]. Available: http://www.engineeringtoolbox.com/voice-level-d_938.html.

[5] T. D. Rossing, F. R. Moore and P. A. Wheeler, The Science of Sound, 3rd Edition, Addison-Wesley, 2001.

[6] H. Wallach, E. B. Newman and M. R. Rosenzweig, "The Precedence Effect in Sound Localization," *The American Journal of Psychology*, vol. 62, no. 3, pp. 315-336, 1949.

[7] E. Singer, K. Larke and D. Bianciardi, "LEMUR GuitarBot: MIDI Robotic String Instrument," in *Conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003.

[8] EMMI: Expressive Machines Musical Instruments, "PAM: Poly-tangent Automatic multi-Monochord," 2014. [Online]. Available: http://expressivemachines.com/dev/wordpress/?page_id=10. [Accessed 25 02 2015].

[9] M. Kolb, "Fingers," Compressorhead, 2009. [Online]. Available: http://compressorhead.rocks/Fingers.html. [Accessed 26 02 2015].

[10] Baginsky, "The Three Sirens - A self learning robotic rock band," 29 07 2002. [Online]. Available: http://www.the-three-sirens.info/binfo.html. [Accessed 26 2 2015].

[11] AMEI Association of Musical Electronics Industry, *(CA-033) MIDI 1.0 Electrical Specification Update [2014]*, MMA Technical Standards Board/ AMEI MIDI Committee , 2014.

[12] G. Wang, A. Misra, A. Kapur and P. R. Cook, "Yeah, ChucK it!⇒ dynamic, controllable interface mapping," in *New interfaces for musical expression*, Vancouver, 2005.

[13] J. Murphy, D. A. Carnegie and A. Kapur, "Little Drummer Bot: Building, Testing, and Interfacing With a New Expressive Mechatronic Drum System," in *International Computer Music Conference*, 2014.

[14] J. Long, J. Murphy, A. Kapur and D. A. Carnegie, "A comparative evaluation of percussion mechanisms for musical robotics applications," in *Automation, Robotics and Applications (ICARA), 2015*, Queenstown, 2015.

[15] A. Kapur, J. Murphy and D. A. Carnegie, "Kritaanjli: A Robotic Harmonium for Performance, Pedogogy and Research," in *International Conference on New interfaces for Musical Expression*, Michigan, 2012.