4.1. The Anatomy of a Module

The API portion of the timelab framework, given by the files api.h and gapi.c, provides data structures and functions for dealing with items and DSP functions. Each module class must contain a pointer to a DSP function and (if necessary) to any number of items. These pointers correspond to inlets and outlets in Pd. TimeLab doesn’t distinguish between input pointers and output pointers since they are all of the same data type. The management of these pointers is left to the user.

Thus, a module class called ‘lookup’, which is a saw-tooth oscillator intended for doing table lookups, will have a data structure with at least the fields:

```c
struct _item **signals;
```

Methods for construction, destruction, and the DSP loop itself are also part of the class.

These variables are initialized in the in_lookup_init function as follows:

```c
lookup->signals = signals_new(lookup->signal_count);
```

```c
lookup->signals[0] = signal_setup(0, IT_NUMBER);
lookup->signals[1] = signal_setup(1, IT_SAMPLES);
```

Now we have a pointer to a DSP function that takes as arguments a number of samples to compute (generally this variable is labeled ‘s’ in practice) and a pointer to the class itself, which we can use at any point as a hook into the lookup_class data structure itself. Thus data can be gotten from any module with the usual C-code syntax. ‘Signals[0]’ is an ‘inlet’ that takes a number (type ‘IT_NUMBER’ — ‘IT’ stands for ‘item’) as input and ‘signals[1]’ is an ‘outlet’ that spits out a vector of samples (type ‘IT_SAMPLES’) that is ‘s’ long.

4.2. The Anatomy of a Patch

Since ‘signals’ are actually pointers to ‘items’ (items can be samples, floating point numbers, and other data types), their data can be passed around from one module to another simply by equating pointers. For example, in this bit of a patch, an array of lookup module objects (called `osc_lookup[i]`) do their sample crunching and then the table module object points to the data that `osc_lookup[i]` just got done computing, and we have a lookup oscillator:

```c
osc_lookup[i]->freq = freq[i];
osc_lookup[i]->
    dsp_func(s, osc_lookup[i]);
```

```c
table->signals[0] = osc_lookup[i]->signals[1];
table->dsp_func(s, table);
```

This example is just a small part of a patch that generates Shepard tones. The whole patch lives within a single C function that we call a ‘register function’. A pointer to each register function is then registered in the ‘function register’ (timelab’s DSP engine). At runtime, the scheduler, cued by the DAC, calls the function register (which can hold an arbitrary number of patches) whenever a new block of samples is needed.

The programmer must also deal with setting up the function register, the user interface (if any), and the memory needed for each patch. Utility functions for these tasks are provided by the API and again, it’s all in C.

5. ACKNOWLEDGEMENTS

Thank you very much to Miller Puckette who continues to advise me in this project. Thank you also to Kevin Larke whose knowledge of scheduling has lead me to certain (I hope good) design choices. As timelab goes on, I shall call on their help again, I’m sure.

6. REFERENCES


Figure 1. Trimpin’s Krautkontrol (top) and Eric Singer’s GuitarBot (bottom). [11]
sound in the advent of the phonograph and, subsequently, the loudspeaker largely led to the demise of the increasingly complex automated instruments. During the 1970’s, artists created the field of musical robotics by integrating the pre-loudspeaker automatic music techniques with contemporary electronics. These artists, including still-active workers Trimpin and Godfried-Willem Raes, found in musical robots a way to express computer music through media other than loudspeakers. This interest in utilizing computer music techniques with mechatronic apparatus is still active in contemporary musical robotics who use state-of-the-art techniques to endow their machines with AI-driven musicianship [1]. While much musical robotic research focuses on percussive instruments such as drums and marimbas [5, 10], the past two decades have seen several notable robotic guitar systems. These instruments served as key inspirations to the new work described in this paper. William Baginsky’s The Three Sirens ensemble uses a mechatronically complex robot guitar, dubbed Aglaopheme, which plays music based on the output of an artificial neural network.

A second influential robotic guitar is GuitarBot, built by Singer in 2003. GuitarBot features four vertically mounted strings that remain in permanent contact with their sliding fretter. GuitarBot, shown in Figure 1, was most notably used in concert with guitarist Pat Metheny in 2010.

Groundbreaking sound sculptors Trimpin and Godfried-Willem Raes have also contributed to the subfield of robotic guitars: Trimpin’s 2001 sculpture Krautkontrol features an array of solenoid-actuated guitars (shown in Figure 1) capable of floating bridge-mediated pitch bends [2]. The Logos Foundation, created by Godfried Willem Raes [7], built the Synchrochord monochord string instrument in 2012.

While existing robotic guitars have varying degrees of musical expressivity available to performers and composers, a goal in the design of Swivel was to create a system that affords artists high-resolution control over parameters such as pitch and loudness. The next section details the design, fabrication, and evaluation of these subassemblies.

3. SWIVEL: A SYSTEMS OVERVIEW

The following subsections focus upon the design, implementation, and construction of Swivel’s subassemblies. The mechanisms of action of each subassembly are discussed, and design and construction techniques are detailed.

3.1. Fretting Mechanism

Swivel’s fretting assembly consists of a fretter arm, which is attached to a stepper motor via a pivot arm. The system is diagrammed in Figure 3. The stepper motor positions the fretter arm along the string; upon reaching the desired position, the solenoid-actuated arm clamp brings the fretter arm into contact with the string. By eschewing traditional linear motion-based fretter systems in favor of a rotary mechanism, it was hoped that higher speed could be attained and simpler construction techniques used. This rotary mechanism has been built using rapid prototyping techniques: 3D printing techniques were used to create the pivot arm.

3.2. Plucking Mechanism

A rotary plucking mechanism, diagrammed in Figure 4, is used for string actuation. Guitar picks are mounted radially around a laser cut acrylic clamp. The pick clamp is attached to the shaft of a stepper motor: upon actuation, the stepper motor rotates and brings the picks into contact with the string. To vary the power of the picking event, the stepper motor is attached to a servo-controlled cam. The cam raises and lowers the pick-wheel, allowing composers to choose the loudness of the picking event and provide audiences with a visual cue as the robot’s loudness varies.

3.3. Damping Mechanism

A servo-based damping arm is employed to stop string vibrations; the damper functions by bringing a felt pad into contact with the string. Previous systems have used solenoid-based dampers with only two states: fully damped or undamped. Swivel employs a variable-position damper, allowing for different degrees of damping to be employed. The use of a variable damper instead of a simple bi-state damper allows composers and performers more control over the dynamic response of the robot.

3.4. Electronics, Pickup, and Communication

Swivel communicates with a host PC via the MIDI protocol. Upon receipt of a MIDI message with noteOn information, Swivel’s microcontroller converts the pitch information and the instructions for the fretter stepper to step to a predefined point. Once the fretter is in position, the fret clamp is actuated by a power MOSFET connected to a digital output pin on the board’s microcontroller; concurrently, the plucker cam lifts the plucker stepper into a position corresponding to the MIDI event’s velocity.

After the fretter is in contact with the string and the plucker activated, the plucker actuates and brings a guitar pick into contact with the string. The string’s vibrations are transmitted to the host computer and/or speakers by an optical pickup. Swivel’s electronics are modular, allowing for multiple Swivel devices to be connected on a bus: each Swivel device could then be assigned a MIDI channel, allowing MIDI notes to be directed to each discrete string. Figure 5 shows a block diagram of Swivel’s communications and actuator control electronics.

4. ANALYSIS AND EVALUATION

After the construction of Swivel, the fretter’s performance was evaluated. The purpose of the evaluation was to determine the capabilities of each of the robot’s components. An awareness of the system’s inter-note speed, for example, will provide composers with knowledge of the musical limitations and abilities of Swivel. The subsequent subsections detail the evaluation of the fretting mechanism, a system unique to Swivel. See [13] and [9] for more detailed analysis of the damping and plucking mechanisms.

4.1. Analysing and Evaluating the Fretter

To find the resolution of Swivel’s fretting mechanism, its step size at various microstepping resolutions was determined. Swivel uses a direct drive fretting mechanism: its stepper motor has a resolution of .9 degrees. The microcontroller board allows for multiple microstepping modes in addition to full stepping. Table 1 shows steps per revolution of the stepper motor at different microstepping settings.

To allow for accurate acceleration and deceleration with no slippage or missed steps, a stepper motor acceleration library is used on the microcontroller. The speed required to traverse the string is shown in Table 2.

Table 1: Angles and steps per revolution for Swivel’s microstepping modes.

<table>
<thead>
<tr>
<th>Microstep Mode</th>
<th>Step Angle</th>
<th>Steps Per Revolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Step</td>
<td>0.9 degree</td>
<td>400</td>
</tr>
<tr>
<td>1/2 Step</td>
<td>0.45 degree</td>
<td>800</td>
</tr>
<tr>
<td>1/4 Step</td>
<td>0.25 degree</td>
<td>1600</td>
</tr>
<tr>
<td>1/8 Step</td>
<td>0.12 degree</td>
<td>3200</td>
</tr>
<tr>
<td>1/16 Step</td>
<td>0.06 degree</td>
<td>6400</td>
</tr>
</tbody>
</table>

To determine the resolution of Swivel’s plucking mechanism:

<table>
<thead>
<tr>
<th>Microstep Mode</th>
<th>Traverse Time</th>
<th>Steps Per String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Step</td>
<td>14.6 s</td>
<td>1.050</td>
</tr>
<tr>
<td>1/2 Step</td>
<td>7.3 s</td>
<td>252</td>
</tr>
<tr>
<td>1/4 Step</td>
<td>3.7 s</td>
<td>525</td>
</tr>
<tr>
<td>1/8 Step</td>
<td>1.9 s</td>
<td>132</td>
</tr>
<tr>
<td>1/16 Step</td>
<td>1.0 s</td>
<td>66</td>
</tr>
</tbody>
</table>

Table 2: Time and number of steps required to traverse string at different microstepping settings.

From Tables 1 and 2, conclusions can be reached about Swivel’s fretting capabilities: while full stepping allows for high-speed transitions between notes, its relatively low resolution results in low pitch accuracy. Increased degrees of microstepping result in slower note transition speeds but higher accuracy and the ability for increased microtonality between chromatic notes.
sound in the advent of the phonograph and, subsequently, the loudspeaker largely led to the demise of the increasingly complex automated instruments.

During the 1970’s, artists created to the field of musical robotics by integrating the pre-loudspeaker automatic music techniques with contemporary electronics. These artists, including still-active workers Trimpin and Godfried-Willem Raes, found in musical robotics a way to express computer music through media other than loudspeakers. This interest in utilizing computer music techniques with mechatronical apparatus is still active in contemporary musical roboticists who use state-of-the-art techniques to endow their machines with AI-driven musicianship [1].

While much musical robotic research focuses on percussive instruments such as drums and marimbas [5, 10], the past two decades have seen several notable robotic guitar systems. These instruments served as key inspirations to the new work described in this paper. William Baginskys’s The Three Sirens ensemble uses a mechatronically-complex robot guitar, dubbed Agalopheme, which plays music based on the output of an artificial neural network.

A second influential robotic guitar is GuitarBot, built by Singer in 2003. GuitarBot features four vertically mounted strings that remain in permanent contact with their sliding fretter. GuitarBot, shown in Figure 1, was most notably used in concert with guitarist Pat Metheny in 2010.

Groundbreaking sound sculptors Trimpin and Godfried-Willem Raes have also contributed to the subfield of robotic guitars: Trimpin’s 2001 sculpture Krautkontrol features an array of solenoid-actuated guitars (shown in Figure 1) capable of floating bridge-mediated pitch bends [2]. The Logos Foundation, created by Godfried Willem Raes [7], built the Synchrochord monochord string instrument in 2012.

While existing robotic guitars have varying degrees of musical expressivity available to performers and composers, a goal in the design of Swivel was to create a system that affords artists high-resolution control over the dynamic response of the robot.

### 3. SWIVEL: A SYSTEMS OVERVIEW

The following subsections focus upon the design, implementation, and construction of Swivel’s subassemblies. The mechanisms of action of each subassembly are described, and design and construction techniques are detailed.

#### 3.1. Fretting Mechanism

Swivel’s fretting assembly consists of a fretter arm, which is attached to a stepper motor via a pivot arm. The system is diagrammed in Figure 3. The stepper motor positions the fretter arm along the string; upon reaching the desired position, the solenoid-actuated fretter arm clamp brings the fretter arm into contact with the string. By eschewing traditional linear motion-based fretter systems in favor of a rotary mechanism, it was hoped that higher speed could be attained and simpler construction techniques be used. This rotary mechanism has been built using rapid prototyping techniques: 3D printing techniques were used to create the pivot arm.

![Figure 3. Swivel’s fretting mechanism](image)

#### 3.2. Plucking Mechanism

A rotary plucking mechanism, diagrammed in Figure 4, is used for string actuation. Guitar picks are mounted radially around a laser cut acrylic clamp. The pick clamp is attached to the shaft of a stepper motor: upon actuation, the stepper motor rotates and brings the picks into contact with the string. To vary the power of the picking event, the stepper motor is attached to a servo-controlled cam. The cam raises and lowers the pick-wheel, allowing composers to choose the loudness of the picking event and provide audiences with a visual cue as the robot’s loudness varies.

![Figure 4. Swivel’s plucking mechanism](image)

#### 3.3. Damping Mechanism

A servo-based damping arm is employed to stop string vibrations; the damper functions by bringing a felt pad into contact with the string. Previous systems have used solenoid-based dampers with only two states: fully damped or undamped. Swivel employs a variable-position damper, allowing for different degrees of damping to be employed. The use of a variable damper instead of a simple bi-state damper allows composers and performers more control over the dynamic response of the robot.

#### 3.4. Electronics, Pickup, and Communication

Swivel communicates with a host PC via the MIDI protocol. Upon receipt of a MIDI message with noteOn information, Swivel’s microcontroller converts the pitch information to instructions for the fretter stepper to step to a predefined point. Once the fretter is in position, the fretter clamp is actuated by a power MOSFET connected to a digital output pin on the board’s microcontroller; concurrently, the plucker cam lifts the plucker stepper into a position corresponding to the MIDI event’s velocity.

After the fretter is in contact with the string and the plucker raised to the predefined height, the plucker actuates and brings a guitar pick into contact with the string. The string’s vibrations are transmitted to the host computer and/or speakers by an optical pickup. Swivel’s electronics are modular, allowing for multiple Swivel devices to be connected on a bus: each Swivel device could then be assigned a MIDI channel, allowing MIDI notes to be directed to each discrete string. Figure 5 shows a block diagram of Swivel’s communications and actuator control electronics.

![Figure 5. A block diagram of Swivel's control and communications electronics](image)

### 4. ANALYSIS AND EVALUATION

After the construction of Swivel, the fretter’s performance was evaluated. The purpose of the evaluation was to determine the capabilities of each of the robot’s components. An awareness of the system’s inter-note speed, for example, will provide composers with knowledge of the musical limitations and abilities of Swivel. The subsequent subsections detail the evaluation of the fretting mechanism, a system unique to Swivel. See [13] and [9] for more detailed analysis of the damping and plucking mechanisms.

#### 4.1. Analysing and Evaluating the Fretter

To find the resolution of Swivel’s fretting mechanism, its step size at various microstepping resolutions was determined. Swivel uses a direct drive fretting mechanism: its stepper motor has a resolution of .9 degrees. The microcontroller board allows for multiple microstepping modes in addition to full stepping. Table 1 shows steps per revolution of the stepper motor at different microstepping settings.

<table>
<thead>
<tr>
<th>Microstep Mode</th>
<th>Degree per Step</th>
<th>Steps Per Revolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Step</td>
<td>0.9 degree</td>
<td>400</td>
</tr>
<tr>
<td>1/2 Step</td>
<td>0.45 degree</td>
<td>800</td>
</tr>
<tr>
<td>1/4 Step</td>
<td>0.26 degree</td>
<td>1600</td>
</tr>
<tr>
<td>1/8 Step</td>
<td>0.11 degree</td>
<td>3200</td>
</tr>
<tr>
<td>1/16 Step</td>
<td>0.06 degree</td>
<td>6400</td>
</tr>
</tbody>
</table>

#### 4.2. Analysing and Evaluating the Plucker

The speed required to traverse the string is shown in Table 2.

<table>
<thead>
<tr>
<th>Microstep Mode</th>
<th>Traverse Time</th>
<th>Steps Per Revolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Step</td>
<td>14.6 s</td>
<td>1050</td>
</tr>
<tr>
<td>1/2 Step</td>
<td>7.2 s</td>
<td>2525</td>
</tr>
<tr>
<td>1/4 Step</td>
<td>3.7 s</td>
<td>5050</td>
</tr>
<tr>
<td>1/8 Step</td>
<td>2.9 s</td>
<td>1300</td>
</tr>
<tr>
<td>1/16 Step</td>
<td>1.0 s</td>
<td>660</td>
</tr>
</tbody>
</table>

From Tables 1 and 2, conclusions can be reached about Swivel’s fretting capabilities: while full stepping allows for high-speed transitions between notes, its relatively low resolution results in low pitch accuracy. Increased degrees of microstepping result in slower note transition speeds but higher accuracy and the ability for increased microtonality between chromatic notes.
A musical composition was written to highlight Swivel's microstepping capabilities; the piece, called "Spatial String," consisted of Swivel's output processed by digital audio effects and used as an input into a live diffusion performance system built by sonic artist Bridget Johnson. Using the fretting mechanism's high-precision 1/16 microstepping mode (which divides the string into 1650 steps), loops of microtonal content were played by the instrument and placed in space by the diffusion performance system.

The piece was performed at Wellington's Adam Art Gallery on October 25, 2012 to an audience which sat in close proximity to Swivel: the audience was clearly able to see the kinetic actions of the robot, shown in performance in Figure 7, and equate them with the diffusion system's sonic output. Swivel's first performance served as a testbed for the ease with which the robot could be used in a live context. It was found that the optical pickup system required extensive pre-performance calibration and will thus be simplified in future iterations of the system.

6. CONCLUSIONS AND FUTURE WORK
Swivel serves as a testbed for rotary motion string fretting, a novel technique previously unexplored. While Swivel's note transition times in microstepping mode are quite high, the speed with which notes can be fretted in full stepping mode highlights the promise of such fretting techniques. Future iterations of Swivel will use either selectable microstepping modes, which can be adjusted online, or a feedback-equipped DC servomechanism, allowing for closed-loop control and potentially higher fretting speeds. Qualitatively, it is felt that Swivel's highly kinetic modes of action provide audiences with a visceral, fulfilling experience wherein the robot's motions are quite pronounced. While the current generation of linear motion-based robotic guitars, such as [9], outperform Swivel in some aspects, the ease of assembly, low cost, and potential for future boosts in operating speed and precision warrant further work on rotary-motion slide guitars.

7. REFERENCES
Graduate School of Fine Arts, Tokyo University of the Arts
5000 Omonma Toride Ibaraki Prefecture Japan
t-tsusbasa@y4.dion.ne.jp

REINFORCEMENT LEARNING OF MUSICAL SCALES FROM HUMAN EMOTIONS

Music has a deep relationship with emotions and the connection between musical scales and emotions is especially strong. That has been pointed out since the age of ancient Greece. However, it is not necessarily clear which scale is associated with which emotion, and it is not easy to find scales that represent certain emotions well by psychological experiments. In this paper, we propose a method to find scales that represent emotions well with an engineering approach that uses reinforcement learning, not by a psychological approach. Since this method takes an adaptive approach in which with a scale is changed to adapt to a target emotion, we can expect to acquire a desirable scale without enumerating all scales one by one. As a result of a pilot experiment, we could acquire scales that represent “high” representational power of happiness and sadness and the scales that represent “a little high” representational power of anger and fear.

1. INTRODUCTION
There are strong relationships between musical scales and emotions. If we think of bright major scale and dark minor scale, that is obvious. Those relationships have been recognized since the age of Plato. In Plato’s “The Republic” [1], Socrates argues, from an educational viewpoint, that “sad” Lydian scale and “loose” Ionian scale should not be used, and that only Dorian and Phrygian scales, which represent “courage” and “moderation” respectively should be used.

In recent decades, empirical researches about correspondences between scales and emotions have been conducted. Especially many studies about major and minor scales exist. Kastner et al. [2] performed an experiment about major and minor scales that used illustrations of facial expressions as emotion labels for children. All of the 38 subjects of the experiment, we could acquire scales that represent “high” representational power of happiness and sadness and the scales that represent “a little high” representational power of anger and fear.

The scales mentioned here are different from church modes, though the names are identical.