Cyther: a human-playable, self-tuning robotic zither

Scott Barton
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
sdbarton@wpi.edu

Ethan Prihar
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
ebprihar@wpi.edu

Paulo Carvalho
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
pacarvalho@wpi.edu

ABSTRACT
Human-robot musical interaction typically consists of independent, physically-separated agents. We developed *Cyther* - a human-playable, self-tuning robotic zither – to allow a human and a robot to interact *cooperatively* through the same physical medium to generate music. The resultant co-dependence creates new responsibilities, roles, and expressive possibilities for human musicians. We describe some of these possibilities in the context of both technical features and artistic implementations of the system.

Author Keywords
Musical robotics, human-robot interaction, musical instrument design

ACM Classification
J.5 [Computer Applications] Arts and Humanities – Music, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing – Systems.

1. INTRODUCTION
We often think of an instrument and a performer as separated by an inflexible boundary that defines means and ends. What if this boundary is made porous, allowing a human to play the role of pseudo-static sound shaper while an instrument voices rhythms and pitch sequences? By integrating robotic actuation into a human-playable instrument, a *cooperative*, multi-agent system is created where performer and machine interact with each other through a shared medium. The robot inspires the performer with machine expressions and the performer transforms these gestures by physically manipulating the instrument. Reciprocally, the performer can affect how the robotic system both interprets and generates statements. The results illuminate the expressive spaces that are human, that are mechanical, and that emerge as these worlds synthesize.

2. PRIOR WORK
In order to properly contextualize *cooperative* robotic instruments, we identify a number of categories of prior art. *Mechanical* and *mechatronic* instruments are capable of some degree of autonomous musical functionality. *Robotic* instruments incorporate feedback that allows a machine to interact with its environment. These categories lead us to *cooperative* instruments, actuated by both human and machine, which offer inspiring possibilities for creativity and performance.

2.1 Autonomous Mechanical and Mechatronic Instruments
Mechanical instruments are typically controlled by predetermined sequences stored on physical media, such as a pinned barrel or a perforated roll. These instructions are performed by actuators powered by pneumatic, hydraulic, or spring-based sources, which exert forces upon a system in order to create movement.

Mechatronics is a word that is generally understood to describe systems that have mechanical, electronic, and control elements [1]. Here, we adapt the term to denote mechanical instruments that incorporate electronic elements that allow for computer control. A survey of these instruments can be found in [2] and [3].

There are many examples of mechatronic string instruments (our focus here) that can be distinguished according to how they change pitch. One may effectively change the length of the string either via a sliding bridge (e.g. LEMUR's *GuitarBot* [4]) or a bar positioned perpendicular to the string that can rotate (e.g. *Swivel* [5]). These designs can produce any pitch along a given continuum and thus can perform portamenti. A disadvantage is that they create a proportional relationship between pitch interval size and production time, thus rapid large sequential pitch intervals are difficult to realize. In addition, unless there is a system that either damps the string or releases the stopper, movement between pitches is audible, which at times might not be desired.

Another method is to fix tangents at specific locations along the string that correspond to desired pitches (e.g. EMMI's PAM [6]). Such configurations can produce rapid sequences of notes regardless of pitch interval size, and thus allow a composer to explore patterns that are not idiomatic to human performers. Disadvantages are that tuning is discrete and difficult to change, portamenti are not possible, and the visual experience may be underwhelming in performance because of the small distances between the actuators and the strings.

Other instruments combine elements of the previous two categories. MechBass [7] uses linear solenoids that are housed in carriages that can be moved along the length of a string. The linear solenoids can articulate discrete notes, and the traveling carriages allow for continuous pitch production. EMMI's AMI is designed with both fixed tangents and a moving bridge so that both rapid sequential pitch intervals and portamenti can be achieved [8].

2.2 Robotic Instruments and Self-Tuning
While there is some ambiguity in the definitions of the terms *mechatronic and robotic*, here we understand the latter to indicate systems that have mechatronic components as well as sensors that provide feedback, which allows them to interact with their environment. We can further distinguish between *low-level* and *high-level* feedback. *Low-level* feedback gives information about the state of a mechanical or electrical component. An example would be GuitarBot's bridge

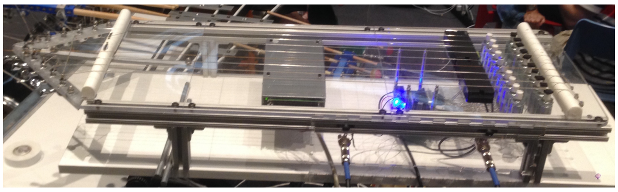

Figure 1. *Cyther* v2.

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

positioning system, which compares actual and specified location using a rotary potentiometer. If there is a difference between these values, the bridge is moved appropriately [4].

Low-level feedback has enabled the development of autonomous tuning systems. Some of these are intended to tune human-playable instruments prior to a performance [9 - 12]. Autonomous tuning has also been implemented in mechatronic instruments [5, 10, 11]. Typically, the frequency of the vibrating string is analyzed as input to the tuning system. This approach is problematic for an instrument that purports to tune dynamically (while it is being played) as it 1) requires that the string is vibrating and 2) may produce errors if a human performer stops the string during tuning. Lookup tables that store correspondences between pitches and motor positions, such as used by the AxCent Tuning System [12], address these issues, but alone can't ensure proper intonation in performance without some other form of feedback.

High-level feedback is the kind that allows a machine to perceive its musical environment, which subsequently affects the expressions it generates. Georgia Tech's Shimon, a robotic marimbist, is an example. Shimon analyzes input MIDI data, which it uses in a number of interaction modules that generate musical responses [13]. High-level feedback allows machines to improvise with humans in conventional ways, but it also allows for new kinds of musical interactions. These interactions typically occur between independent agents (either human or machine) but they can also occur in a context where human and machine *cooperate* via same physical medium.

2.3 Cooperative Musical Machines

A *cooperative* musical machine, an idea we introduce here, requires both human and machine input as parts of a symbiotic whole. Such a system embraces what machines do well, such as complex polyphony and temporal precision. Simultaneously, it alters the affordances available to a human performer, who may subsequently direct her attention towards timbral, articulatory and gestural nuance. Together, *cooperative* instruments enable new kinds of musical interaction and expression.

2.3.1 Cooperative (Electro)Mechanical Instruments

Cooperative musical machines have been with us for centuries, although examples of them are relatively few. The pianola, which came into prominence at the turn of the 20th century, automatically plays the keys of a piano according to a predetermined score, thus the machine assumes the significant responsibility of producing pitches and rhythms. The human must continually pedal the bellows, which creates the suction required for the machine to operate and affects dynamics, but is also free to control a number of other parameters (depending on the instrument) such as tempo (via the Metrostyle), foregrounding / backgrounding (via the Themodist), and sustain [14]. More recently, Wintergatan's *Marble Machine* plays vibraphone bars, drums and an electric bass via a combination of human and mechanical efforts [15]. Gurevich's STRINGTREES allows a human performer to move rotating strings in and out of the path of an automatic picking mechanism to create rhythmic and harmonic sequences [16].

While a human can be inspired by the products of a *cooperative* (electro)mechanical instrument, the latter has no knowledge of a human performer's actions. If we accept Winkler's proposition of interaction as a "two-way street" [17], then the kind of interaction that cooperative mechanical instruments provide is of a modest quantity.

2.3.2 *Cooperative Robotic Instruments*

Cooperative robotic instruments have the same basic qualities as *cooperative* mechanical instruments, but they also incorporate sensing capabilities and artificial intelligence (AI). The latter creates potential for more significant interactions between humans and machines. These instruments can perceive the musical content of a phrase and respond with a complementary idea voiced on the same acoustic instrument.

Examples of *cooperative* robotic instruments are rare. A number of software-based systems have been developed in the last 30 years that provide automatic accompaniment in some form [18], [19], but these typically create a multi-voice situation where an autonomous human and an autonomous machine interact, as opposed to a *collaborative* scenario where human and machine cooperate to generate a single musical voice. Sheffield and Gurevich [20] developed a percussion system that allows a human performer to enable mechatronic actuation through a capacitive touch sensor, but the machine primarily reacts in a one-to-one way to human input: it doesn't interpret or generate new ideas. (This is not a shortcoming of the system, rather, it is a purposeful design choice.) François Pachet's *Continuator* is a software-based automatic collaborator, which, when matched with a Yamaha Disklavier, allows a human pianist to trade ideas with a computer-based system on the same acoustic piano [21]. Georgia Tech's Haile can improvise with a human performer on the same drum [22]. Neither of these truly satisfy the definition of *cooperative* instrument though, for while the human and the machine voice ideas through the same medium, their actions are not parts that require the other to form a unified whole.

The creative potential of *cooperative* robotic instruments is promising, and was the motivation for the development of *Cyther*.

3. DESIGN

3.1 Specifications and Requirements

Cyther was imagined as a board zither that could be played by both a human and a machine. For it to be human-playable, there should be few physical obstacles in the way of the performer. Strings should be able to be struck by a percussionist but also plucked, damped, and stopped as consistent with string instrument technique. The machine should be able to strike and damp any combination of strings autonomously. It should be able to dynamically change the pitch of any string to create portamenti and new tunings. It should be able to change pitches quickly, at least as fast as 100 msec, given this interval is cited as the smallest found in human rhythm production [23]. It should be portable in regard to weight and size: less than 50 pounds and able to fit in a medium-sized keyboard or guitar case. The instrument and its electromechanical systems should be self-contained, so that using it is a matter of plugging it in. Communication should occur via serial commands over USB, and input power should be from a standard 120 VAC outlet.

3.2 *Cyther* v1 and v2

Two versions of *Cyther* have been created. *Cyther* v1 was a hand-made, wooden prototype that was designed and constructed in the summer and fall of 2015. In order to make the components visible, better fit the tuning machines, increase manufacturing precision, and realize visual aesthetic preferences, *Cyther* v2 was V-shaped on one end and made out of aluminum and laser-cut acrylic. The other electromechanical components were largely the same between the two versions.

3.2.1 Structure

To make *Cyther* human-playable, all of the components are mounted underneath the strings. This allows a performer to pluck, bow, strike, stop, damp, or otherwise access any part of any string. Strings are spaced ~21mm apart, which allows a percussionist to strike the strings with a variety of mallets, but at the same time allows a performer to play multi-string monophonic lines and chords comfortably with each hand. The frame is 12” wide X 36” long X 4” high. This allows the instrument to fit snugly inside a standard mid-sized keyboard case for easy transport. Given the string spacing, the width of the instrument allows for 10 strings. *Cyther* v1 was made out of wood while *Cyther* v2 is made out of t-slotted aluminum (frame) and sheets of ¼” acrylic, which provides a translucent resonating top that also houses the electromechanical systems and the pickups. The ball-ends of the strings pass through holes on one side of the instrument, pass over two custom 3D-printed bridges (one on each side of the frame) and then to guitar tuning machines on the other side of the instrument, which are configured in a V-shape (see Figure 1).

3.2.2 Electromechanical Systems

Because the components need to be mounted under the top layer of the instrument, push solenoids were chosen to strike and damp each string for a total of 20 actuators. The actuator chosen has a coil voltage of 12 VDC, a holding force of 12.7 N (2.85 lbs.) and a continuous (100%) duty cycle. The latter is necessary to apply the dampers (and even the strikers) for long periods of time. The holding force enables a reasonable striking dynamic range and ensures that the dampers reduce vibrations quickly and adequately. The actuators are controlled by the *Multi Solenoid Driver Module v1.1 (MSDM)*, a custom PCB designed at the Music, Perception, and Robotics Lab at WPI. The board is based around the ATmega2560 microcontroller and can drive 25 solenoids (each can draw up to 1 amp) and contains FTDI-USB, I^2C and RS485 connectors for communication. Currently, serial data is sent to the microcontroller via a USB to TTL cable, though this is a temporary solution pending further development of a proprietary networking protocol.

3.2.3 Software

The firmware on the boards is composed entirely of non-blocking functions. The non-blocking code helps ensure that a command to play multiple solenoids simultaneously results in individual actuations that are temporally proximal enough to be perceived as a chord. The MSDM works as a slave to a master controller, which allows multiple instruments to be synchronized. Currently, the master sends a packet composed of a start byte, a length, a body and a checksum. The body is composed of a variable number of header-message pairs that contain information about what the robot should do. For example, the header 0x20 (hexadecimal representation for 32), is followed by 32 bits that correspond to the state of the 25 solenoids and 5 reserved bits. The checksum ensures data integrity and reduces the chances of a corrupted package producing messages.

3.3 *Cyther* v3: Robotic Tuning

Pitch changes on a particular string of *Cyther* v1 and v2 are only achievable by a human performer who either stops the string with his fingers or an object (such as a slide), or manually turns the tuning mechanism. Practically, using one’s fingers only allows harmonics. A slide can produce any pitch within a continuum, but intonation is difficult without significant practice. Turning the tuning machines manually is awkward, and presents similar intonation difficulties.

In order to create a more expressive instrument, our goal was to make a system that could autonomously, dynamically and accurately produce 1) a range of pitches not limited to easily-produced harmonics or even a particular scale, and 2) portamenti. It should also allow for cooperative interaction and thus should not provide physical encumbrances to a human performer.

To change a string’s pitch, there are a limited number of variables that one can affect. The frequency (f) that a string will vibrate at is a function of its length (L), tension (T), and linear density (m):

$$f = \frac{\sqrt{\frac{T}{m/L}}}{2L}. \tag{1}$$

Changing the mass of a string dynamically is difficult. Changing the length of a string by stopping it is a common approach, but this isn’t an ideal solution here. A sliding bridge can produce pitches along a continuum, but it may damp partials of harmonics produced by a human performer as it moves. Fixed tangents provide a more limited set of pitches than a sliding bridge, they might potentially interfere with the performer physically, and also are unable to produce the desired portamenti. Fixed tangents also involve numerous actuators that require more in regard to space, wiring, power, cost, and control schemes.

We thus chose to vary tension in order to achieve our musical goals. A variable tension system is able to produce any pitch along a continuum. It is also able to produce portamenti starting from any pitch, including from harmonics and stopped notes produced by human performers. It can function with or without human involvement and it doesn’t present physical encumbrances to human musicians.

With that said, as Bart Hopkin notes, it is difficult to use tension to control string pitch. Because of this, there are relatively few instruments that have utilized this technique to achieve dynamic pitch changes, including the pedal steel guitar, the American washtub bass, the whamola, Indian *ektars* and the Vietnamese *dan bau* [24]. Part of the difficulty lies in the various points of friction that are typically present on string instruments, such as the bridges, which can grip and release strings in undesired ways. A system must have minimal friction in order to achieve accurate pitch variation via tension changes.

3.3.1 Structural Design

The structural design of *Cyther* v3, shown in Figure 2, is based on *Cyther* v2 in regard to its size and materials. The pitch detection hardware is mounted to one acrylic sheet, and the pitch changing hardware is mounted to the other sheet.

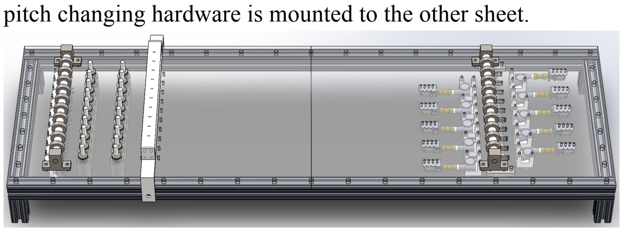

Figure 2. Complete *Cyther* v3 CAD model.

The bridges of *Cyther* v3 are designed to allow the strings to move as freely as possible. Friction between the strings and the bridge can create non-uniform tension throughout the individual strings [25] and add inefficiency to the tuning system resulting in slower, less precise pitch changes. To reduce friction, the bridge, shown in Figure 3, is made from radial bearings mounted around a shaft. The bearings rotate around the shaft as the string is adjusted, which allows for the tension throughout the string to be constant.

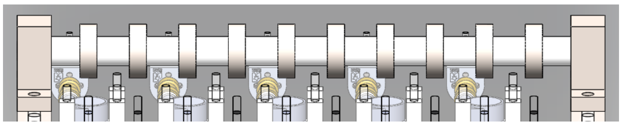

Figure 3: *Cyther* V3 bridge made from a shaft and radial bearings.

Strings were selected so that the highest pitch of one corresponded to the middle of the range of the next highest string. The range of each string was determined by the pitch it produced at seven pounds of tension to the pitch it produced at fifteen pounds of tension. Together, the ten strings cover three octaves: G2 – G5. Two electromagnetic pickups are mounted under the strings, which allows the sonic output of the instrument to be amplified via standard 1/4" jacks.

3.3.2 Tuning System Actuation

The pitches of *Cyther* v3's strings are changed via a motor-driven tuning machine, shown in Figure 4. Each string is wound around a guitar tuning key that was modified so that it could be rotated by a motor. Each key contains a worm drive with a ratio of 16:1, which increases the torque used to wind the string and prevents any forces from being applied to the motor. The motor thus does not have to be constantly powered, which stabilizes the frequency of the string and increases the lifespan of the instrument.

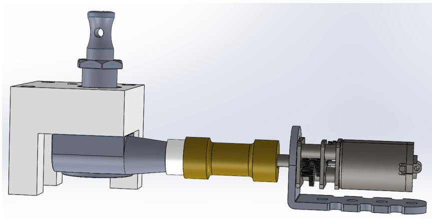

Figure 4: Motor and worm drive used to change string tension.

Each motor requires a stall torque equal to four times the maximum tension, 17 lbs. in this case, to stay at optimal power levels. The Pololu 100:1 Micro Metal Gearmotor was chosen for its size and 30 oz. in. stall torque. The extra torque helps offset friction, gear train inefficiencies, and unpredictable misalignment of parts due to manufacturing inconsistencies, which reduce the amount of force the motor can transfer to the string.

3.3.3 Tension Sensing

As previously mentioned, frequency sensing creates issues when trying to produce dynamic pitch changes. Therefore, *Cyther V3* can estimate the pitch of a string without needing to sense its frequency. Per equation 1, when a string's length and weight are known constants, its vibrating frequency can be determined by sensing its tension. The tension sensor shown in Figure 5 contains a linear potentiometer coupled to a spring. The ball end of the string is pressed against the spring cap. As the tuning machine rotates to tighten the string, the ball end of the string applies force on the spring cap, which compresses the spring. As the spring compresses, the wiper on the potentiometer moves, which, with the appropriate circuitry, produces a varying and measurable voltage. The system is designed to create as little contact as possible between moving parts in order to provide accurate information.

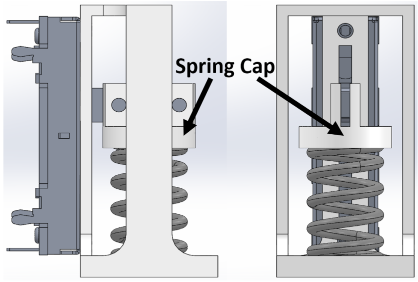

Figure 5: Tension sensor used for pitch estimation.

The tuning system keeps each string at a desired frequency by comparing the current potentiometer value of each tension sensor to a goal value determined by the tuning algorithm. If there is a difference between these values, the motor tightens or loosens accordingly until they are equivalent.

3.3.4 Frequency Sensing

While frequency sensing alone is potentially problematic for a dynamic pitch-changing instrument, it is useful as a complementary system in order to achieve more accurate tuning. Each string on *Cyther* v3 thus has an optical pickup that positions a 650nm laser diode across from a photoresistor. The microcontroller polls the optical pickups and estimates frequency by timing, filtering and averaging the intervals between rising edges. The frequency measurement is then passed to a function that updates the curve relating potentiometer value (tension) to string frequency.

3.3.5 Tension-Frequency Relationship

The potentiometer to frequency relationship curve (PFRC) is determined by combining Equation 1 and Hooke's Law, which states that the force required to deform a spring is proportional to the distance of that deformation. The unit weight and length of each string is constant, and each potentiometer will output values that change linearly with respect to distance traveled. This information can be used to derive the PFRC, which states the relationship between the potentiometer value and the frequency measurement:

$$Potentiometer\ Value = A \times Frequency^2 + B \quad (2)$$

The constants of the PFRC are determined experimentally for each string by measuring the frequency and potentiometer value at three different frequencies that span the range of the string. Linear regression is used to find a best fit curve, and then A and B are set as constants.

When a new frequency measurement is generated, a curve-adjusting algorithm (CAA) is used to adjust the PFRC. The CAA begins by creating a set of points along the current PFRC that are evenly distributed within the playable range of the single string. A new point is created with the coordinates ($frequency^2$, potentiometer value). A new best fit curve is generated using linear regression, and the constants from that curve replace those of the PFRC. The number of points created in step one of the CAA will change how aggressively the PFRC will accommodate new data. With fewer points, the PFRC will quickly move towards new data; with more points the PFRC will make smaller adjustments.

4. RESULTS

4.1 Electro-Mechanical Systems

Speed: a striking solenoid was able to produce a uniform tremolo on the same string at an IOI (inter-onset interval) as low as 43 msec (~23 Hz). *Polyphony*: any combination of strikers and dampers may be used simultaneously. *Dynamics*: the striker on the highest string produced a 10 dB range (corresponding to ontimes, the time that the solenoid is activated, of 8-14 msec). The striker on the lowest string produced a 20 dB range (corresponding to ontimes of 8-22 msec). For both strings, the ramp from quietest to loudest was generally linear, with the amplitude plateauing at longer subsequent ontimes. *Dampers*: to measure damping effectiveness, the string was actuated with an ontime of 22 msec, allowed to vibrate for 500 msec and then the damper was activated. The RMS amplitude for the freely vibrating string was compared to the RMS amplitude of the string 500 msec after the damper was applied. The lowest string damper achieved 14.1 dB of attenuation; the third lowest string achieved 15.5 dB; and the highest string achieved 30 dB.

4.2 Self-tuning Systems

4.2.1 Hardware

The motors were able to tune the strings from minimum to maximum tension without stalling. They used more current when tuning the strings up (384 mA) than when tuning them down (253 mA). Stall current was measured at 843 mA.

We tested the speed of the tuning mechanisms using a number of different pitch intervals (see Figure 6).

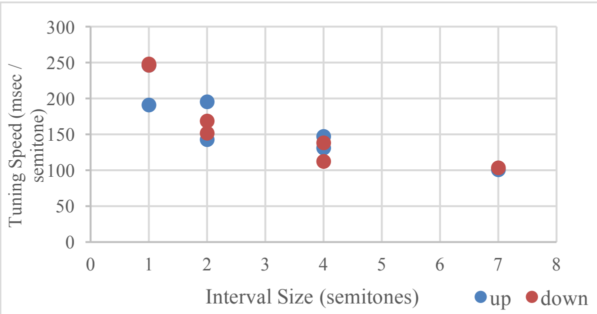

Figure 6. Speed of tuning mechanism performing different pitch intervals.

Durations were longer than the goal speed of 100 msec, which could be improved through more powerful motors, or by reducing friction.

We also tested the accuracy of the tuning mechanisms using the same pitch intervals (see Figure 7).

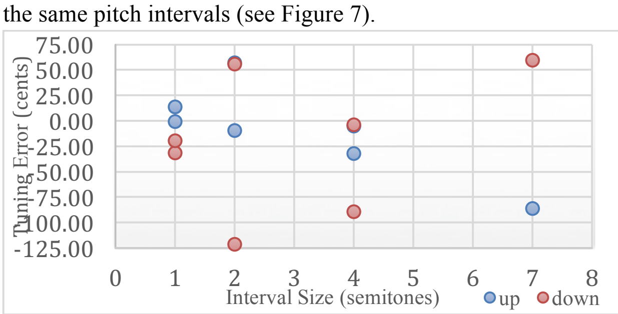

Figure 7. Accuracy of tuning mechanism performing different pitch intervals.

Humans are able to perceive a difference as small as 3 cents between two frequencies (though this number varies over the frequency range) and the JND (just-noticeable difference) for musical intervals ranges from 13 to 26 cents [26]. About half the intonation errors produced in our experiment were within or below this range. The fact that results that vacillated around zero suggests that errors were not compounding sequentially. This supports the notion that the CAA was successfully adjusting the PFRC for intonation discrepancies

The motors generate acoustic noise when they are both holding and rotating that can interfere with the sound produced by the string. We measured SPL from a distance of 2 ft.: the turning motor was measured at 58 dBA, the holding motor at 51.5 dBA and a moderate string pluck at 56.1 dBA. With that said, this isn't a significant problem because the strings are amplified with electromagnetic pickups.

The laser diodes used in the pickups were not uniform and as a result, photoresistors output non-uniform resistances. This problem was solved by measuring each photoresistor's resistance when the laser was on and picking a resistor equal to the measured resistance (instead of using one resistor value for every pickup).

4.2.2 Frequency Analysis

The time needed to determine a string's frequency ranged from 101-349 msec, with an average of 183.8 msec. There is a relationship between how hard a string is plucked and the measurement delay (when the string was manually plucked very hard, the measurement delay was around a second).

The accuracy of the frequency analysis algorithm was tested using a waveform generator. A sine wave with a magnitude of 4V and offset of 2.5V was input directly into the microcontroller (for these tests, we used an Arduino mega, which features the same ATmega2560 as the MSDM). The algorithm produced an average error of 2.81 cents.

We then tested the accuracy of the frequency analysis algorithm using the output of *Cyther*. Using an audio recording of one of *Cyther's* strings, we compared the results of a FFT analysis to our frequency detection algorithm. The average error was -15.28 cents, which while higher than the previous test, is still within the aforementioned goal range.

5. IN PRACTICE

A number of musical works have been composed for and performed with *Cyther*. The first was *Life's Node* (2015), a collaborative improvisation between the first author and percussionist / composer Nate Tucker (who was a collaborator in the original ideation process for *Cyther* v1). In this work, one performer composes and sequences the machine's gestures in real-time while the other shapes those gestures and expresses his own ideas on the instrument. The piece explored a mode of cooperation not previously mentioned, namely of two human performers interacting with each other through a human-playable mechatronic instrument. Subsequently in 2016, the first author developed software that allows him to improvise with the instrument as it functions autonomously. In this piece, categories of gestures and a general contour are defined, though the specific musical ideas that articulate such forms are determined by probabilistic and stochastic processes. In both of these works, the human performer can actuate the instrument in traditional ways, but can also act as a sound shaper, creating a variety of rhythms, harmonies and melodies by damping and stopping the strings.

Cyther's mechatronic systems allow one the liberty to experiment with other kinds of sonic manipulation techniques. For example, we have used piezo discs and an EBow to create a variety of timbres by dynamically adjusting their position relative to the instrument as it plays autonomously. This is an example of a *cooperative* interaction: the machine is responsible for pitch and rhythmic content while the human performer shapes the timbre and dynamics of that output. Each is a part of a whole that requires the other.

6. CONCLUSIONS

The iterations of *Cyther* described in this paper largely realize our original goals. The structural design of the instrument provides few physical obstacles, thus traditional (and extended) string and percussion techniques can be used. Actuators can autonomously strike and damp rapidly, and can be used in any combination simultaneously. The self-tuning mechanisms allow the instrument to generate new tunings and portamenti while it is operating autonomously or while a human is interacting with it. The speed and accuracy of the tuning system is reasonable, though it could be improved in the future by reducing friction, using more powerful motors, and increasing sensing precision. The instrument is self-contained, transportable, and requires few connections (120 VAC power, USB, and two ¼" jacks for audio amplification).

Most importantly, *Cyther* provides a physical medium that can be activated by human and machine simultaneously and in *cooperation*. A human performer and *Cyther* can both affect rhythm, pitch, timbre, dynamics, and articulation. Either can play the role of impulse or filter. When both play the role of impulse simultaneously, novel kinds of rhythms and pitch sequences are created. When a human plays the role of filter, full attention can be devoted to timbral and articulatory nuance. In our experience composing for and performing with the instrument, we have found that it helps realize music that we haven't heard before, which is exciting. Future work is directed at exploring more of these musical possibilities and developing the high-level sensing and generation capabilities of the instrument.

7. REFERENCES

[1] A. Brown, "Mechatronics and the Role of Engineers," *ASME*, Aug-2011. [Online]. Available: https://www.asme.org/engineering-topics/articles/mechatronics/mechatronics-and-the-role-of-engineers. [Accessed: 14-Apr-2017].

[2] A. Kapur, "A History of robotic Musical Instruments.," in *in Proceedings of the International Computer Music Conference*, Barcelona, Spain, 2005.

[3] M. Bretan and G. Weinberg, "A survey of robotic musicianship," *Commun. ACM*, vol. 59, no. 5, pp. 100–109, 2016.

[4] E. Singer, K. Larke, and D. Bianciardi, "LEMUR GuitarBot: MIDI robotic string instrument," in *Proceedings of the international conference on New interfaces for musical expression*, McGill University, Montreal, Canada, 2003, pp. 188–191.

[5] J. Murphy, A. Kapur, and D. Carnegie, "Swivel: Anaysis and System Overview of a New Robotic Guitar," in *Proceedings of the 2013 International Computer Music Conference*, 2013.

[6] "PAM | EMMI: Expressive Machines Musical Instruments." [Online]. Available: http://expressivemachines.com/dev/wordpress/?page_id=10. [Accessed: 19-Jan-2017].

[7] J. McVay, D. A. Carnegie, J. W. Murphy, and A. Kapur, "Mechbass: A systems overview of a new four-stringed robotic bass guitar," in *Proceedings of the 19th Electronics New Zealand Conference (ENZCon), Dunedin, New Zealand*, 2011, pp. 145–150.

[8] T. Rogers, S. Kemper, and S. Barton, "MARIE: Monochord-Aerophone Robotic Instrument Ensemble," in *Proceedings of the international conference on New Interfaces for Musical Expression*, Louisiana State University, Baton Rouge, LA, USA, 2015, pp. 408–411.

[9] "Tronical Tune - How it works - Autotunes your guitar in seconds." [Online]. Available: https://www.tronicaltune.com/how-it-works. [Accessed: 25-Jan-2017].

[10] S. Trail, G. Tzanetakis, L. Jenkins, M. Cheng, D. MacConnell, and P. Driessen, "STARI: A self tuning auto-monochord robotic instrument," in *Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on*, 2013, pp. 405–409.

[11] J. W. Murphy, P. Mathews, A. Kapur, and D. A. Carnegie, "Robot: Tune Yourself! Automatic Tuning for Musical Robotics.," in *in Proceedings of the international conference on New Interfaces for Musical Expression*, Goldsmiths, University of London, London, UK, 2014, pp. 565–568.

[12] "AxCent Tuning Systems: The Technology." [Online]. Available: http://axcenttuning.com/theTechnology.php. [Accessed: 25-Jan-2017].

[13] G. Hoffman and G. Weinberg, "Interactive improvisation with a robotic marimba player," *Auton. Robots*, vol. 31, no. 2–3, pp. 133–153, 2011.

[14] "The Pianola Institute - History of the Pianola - An Overview." [Online]. Available: http://pianola.org/history/history.cfm. [Accessed: 18-Jan-2017].

[15] "Wintergatan - Marble Machine." [Online]. Available: http://www.wintergatan.net/#/m.m.machine. [Accessed: 20-Jan-2017].

[16] M. Gurevich, "Distributed Control in a Mechatronic Musical Instrument.," in *Proceedings of the international conference on New Interfaces for Musical Expression*, Goldsmiths, University of London, UK, 2014, pp. 487–490.

[17] T. Winkler, *Composing interactive music*. The MIT Press, 2001.

[18] G. Xia and R. Dannenberg, "Duet Interaction: Learning Musicianship for Automatic Accompaniment," in *Proceedings of the international conference on New Interfaces for Musical Expression*, Louisiana State University, Baton Rouge, LA, USA, 2015.

[19] J. P. Forsyth, R. M. Bittner, M. Musick, and J. P. Bello, "Improving and Adapting Finite State Transducer Methods for Musical Accompaniment," in *in Proceedings of the International Computer Music Conference*, University of North Texas, Denton, TX, USA, 2015.

[20] E. Sheffield and M. Gurevich, "Distributed Mechanical Actuation of Percussion Instruments," in *Proceedings of the international conference on New Interfaces for Musical Expression*, Louisiana State University, Baton Rouge, LA, USA, 2015, pp. 11–15.

[21] F. Pachet, "The Continuator: Musical Interaction With Style," *J. New Music Res.*, vol. 32, no. 3, pp. 333–341, Sep. 2003.

[22] G. Weinberg and S. Driscoll, "The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, New York, NY, USA, 2007, pp. 97–104.

[23] J. London, "Three Things Linguists Need to Know About Rhythm and Time in Music," *Empir. Musicol. Rev.*, vol. 7, 2012.

[24] B. Hopkin, *Musical Instrument Design*. Tucson, AZ: See Sharp Press, 1996.

[25] T. D. Rossing, Ed., *The Science of String Instruments, by Rossing, Thomas D*, 1st ed. New York: Springer-Verlag, 2010.

[26] D. Deutsch, *Psychology of music*. Elsevier, 2013.