Alto.Glove: New Techniques for Augmented Violin

Dr. Seth Dominicus Thorn
Arts, Media + Engineering
Arizona State University
seth.thorn@asu.edu

ABSTRACT
This paper describes a performer-centric approach to the design, sensor selection, data interpretation, and mapping schema of a sensor-embedded glove called the "alto.glove" that the author uses to extend his performance abilities on violin. The alto.glove is a response to the limitations—both creative and technical—perceived in feature extraction processes that rely on classification. The hardware answers one problem: how to extend violin playing in a minimal yet powerful way; the software answers another: how to create a rich, evolving response that enhances expression in improvisation. The author approaches this problem from the various roles of violinist, hardware technician, programmer, sound designer, composer, and improviser. Importantly, the alto.glove is designed to be cost-effective and relatively easy to build.

Author Keywords
violin, bowing, gesture tracking, data gloves, augmentation, mapping

CCS Concepts
• **Applied computing~Sound and music computing** • Applied computing~Performing arts • *Human-centered computing~Gestural input*

1. INTRODUCTION

1.1 Wearables and Data Gloves
Trends toward technical miniaturization and wireless connectivity have resulted in computational ubiquity and a plethora of wearable technologies. Beyond cellular phones and wrist watches, data gloves are more rarefied examples of this technology. The extraordinarily refined movements and subtle expressivity of the human hand call for computational interfacing and experimentation.

An early example of a data glove with iconic status in the public consciousness is the Power Glove, which was produced in the late 1980s for use with the Nintendo gaming console but failed to achieve widespread use due to technical limitations. In the context of live music performance, a highly successful data glove is Sonami's Lady's Glove, introduced in 1991 and progressively refined during the next twenty-five years. Sonami's performances make a notion of distance thematic, a decision reflected in the choice of sensors used in the glove: in conjunction with a set of magnets, hall sensors on the fingertips produce various measures of proximity. A more recent, softly commercialized data glove is the mi.mu promoted by Imogen Heap. Unlike Sonami's glove, the mi.mu is built without a view toward a determined or overdetermined application; instead, it is marketed for generalized use in live musical performance and must likewise anticipate and accommodate a variety of situations. Maximum usability is generated by minimalist design. Generally desirable qualities in this context are highlighted in the mi.mu documentation, such as fingertips that are left uncovered to allow playing of various musical instruments, the open palm that affords skin contact when clapping, RGB LEDs and haptic feedback, and wireless connectivity for untethered operation.

NIME'18, June 3-6, 2018, Blacksburg, Virginia, USA.

1.2 Augmented Violin
In the context of violin playing, transferring any added technical implements from the bodies of the violin and bow to the body of the performer typically yields increased playability, since alterations to the weight of bow in particular—and bearing in mind the expense and fragility of these instruments—produce significant perturbations in playing. Adding sixty grams of material to the bow doubles its weight; by comparison, a similar amount of weight added above the wrist or to the forearm is perceived as a much less dramatic alteration. A violinist chooses among bows that potentially vary in weight by only two or three grams. Moreover, playing characteristics are significantly altered by the distribution of this weight and even the shape of the stick. (The famous French maker François Tourte, the namesake of the modern bow, produced a subtly egg-shaped stick, presumably in order to lower its center of gravity.)

These considerations produce multiple problems for designers and players who seek to computationally extend their performances on violin in a transparent manner. In his seminal paper on design principles for computer musical controllers, Cook points out that string players have very little "bandwidth" to spare due to the full occupation of their hands during performance, so designers in this area must instead try to "exploit interesting remapping of existing gestures" [4]. Both of these points are confirmed by an early revision of the IRCAM Augmented Violin, performed by violinist Mari Kimura, who commissioned the design of a glove that would hold a motion-sensing module that had at first been attached to the frog of the bow but proved to be too cumbersome. The revised design is completely minimalist, using only a six-axis motion module in tandem with the sophisticated MuBu gesture following software also developed at IRCAM.

1.3 Resistance and Novelty
A potential criticism of the desire for "wearability" [9] is that interface transparency is no boon to creative improvisation and gesture invention in responsive media. Discovering material and conceptual resistance to be a primary feature of the musical instrument, Evens argues that these instruments imply fundamental limitations on the cybernetic project, the desire to put human consciousness in unmediated contact with the computer [7]. For the same reason, Sha uses costumes of "fantastical design" in his responsive TGarden in order to both extend *and* constrain the movements of the participants, provoking them to invent gestures rather than habitually repeat familiar ones [16]. This case suggests that, if wearability tends to presuppose and accommodate the environment of the human actor, encumbrance and resistance produce a gently skewed set of affordances that encourage the participants to become something other than the actors they were when they first dawned the costumes and entered the responsive media space. Likewise, Sha promotes use of *computational* media in order to skew the material physics in these responsive spaces toward "pseudo-physical" experiences.

To my mind, the goals for computationally extended violin performance ought to be similar. Rather than aiming for the reproduction of classical performance gestures and bowing styles, how can novel hardware interfaces and their software counterparts disengage habitual performance and reinvent the performer? How can hardware extend performance in spite of the lack of spare bandwidth?

These challenges can be addressed in part by designing systems that are extremely responsive. This requires bracketing the traditional terms of violin performance and its canon of classifications, and using instead a different conceptual model for understanding the fundamental mechanics of violin playing. In this paper I will describe the trajectory of a hardware and software system I designed for this purpose. I call the hardware component the "alto.glove" and the responsive software system for improvisation "Windowless."

2. BACKGROUND

2.1 Classification of Bowing Styles

There are diverse reasons for interest in violin bowing technique. A paragon of continuously nuanced gesture, violin bowing can be inherited in order to produce completely new, "reconstructed" instruments like the BoSSA or other novel friction controllers [5]; violin playing can be physically extended by adding additional components to the violin itself, as in Overholt's Overtone Violin [13]; violin technique can be preserved as much as possible, with sensors selectively added to allow a computational response to certain motions and gestures, as in the IRCAM Augmented Violin [3, 11]; or violin playing (or other string playing technique) can serve as a test scenario—or more likely, limit case—for machine learning, owing to its high-degree of refinement and the unique challenge this poses to recognition/representation [15].

Such research efforts should not be confused with the task of building responsive and dynamic computational responses to the actions of players. As the technology has advanced, so has the differentiation of those tasks. In the earliest version of the IRCAM augmented violin project, the scientific project of classification sometimes remains close to the artistic task of computational response: the authors describe a "selection" mapping mode that identifies bow strokes and alters the sound processing according to which type has been recognized [3]. On the other hand, the "mixed mapping" mode they describe, which produces responses related to the likelihood that a particular stroke is either *detaché, martelé,* or *spiccato,* offers a rich model for a performance-centric system, because it turns a two-dimensional featurespace into a three-dimensional one while making it continuous. In performance and improvisation, I find that the arbitrary localization of nodes in a featurespace creates a multi-dimensional, textured substrate that generates much more compelling material for mapping to sonic responses. Rather than activating responses to *a priori* defined styles of bowing, the system continuously responds to every bowing nuance and gesture.

2.2 Machine Learning and Expression

In *Philosophy of New Music*, Adorno points to, arguably, a single criterion for evaluating new music, namely whether it liquefies or capitulates to the process of reification [1]. This criterion can be brought to bear on the relationship between the computational substrate of novel digital interfaces and the human movements they track or classify. Generally speaking, classification of bowing gestures by violinists is a problem with greater scientific (representational) merit than artistic potential. Classification studies can advance in a variety of ways: by increasing the dimensionality of a featurespace through either adding additional sensors or making available multiple levels of first- and second-order processing from already available sensors [15]; by reducing the "extrinsic" variables of violin playing not directly related to the actions of the left arm [17]; or by increasing the algorithmic sophistication and speed of supervised machine learning [15]. Schedel and Fiebrink overcame the first of these difficulties by utilizing a K-Bow in their demonstration. The K-Bow contains numerous embedded sensors that cannot easily be affixed or integrated into a traditional bow. The eight native features from the K-bow percolate through a feature extraction process that produces a total of eighty features. The second and third problems are addressed by making use of Fiebrink's Wekinator software, and in their demonstration example, by having the participating cellist iteratively train and refine the classification algorithm. Their demonstration ultimately points to a paradigm of machine learning in which the desiderata of reproducibility and universalizability are mediated by those of speed and ease of training by individual musicians, or as they write, the need to "create good classifiers quickly." This problem is also pointed out by the authors of the Augmented Violin project, who found that among different players, "idiosyncratic behaviors were…found, showing that a universal calibration might not be reliable" [3]. Studies aiming for generalizable training sets and feature extraction discover that supervised learning sessions must be performed by the participants with laboratory constraints *cateris paribus*, that is, at various fixed tempi and dynamics, sans vibrato. Even the position of the left hand on the fingerboard has been found to introduce variations in windowed extremes of right hand acceleration values [17]. But the laboratory does not produce the "natural" setting for which it aims, especially if training goals overtake the task of enhancing or "augmenting" expressivity. Mari Kimura, for instance, warns that the performer might be compelled to modify her playing—in effect, a mimesis of the discretization built into the system—to avoid misrepresentations. Instead of hyperbolizing the expressivity of bowing, systems that classify bowing may ultimately produce more leaden performances.

3. HARDWARE

The alto.glove is my response to the limitations—both creative and technical—that I perceived in these feature extraction processes. The hardware answers one problem: how to extend violin playing in a minimal yet powerful way; the software answers another: how to create a rich response that enhances expression in improvisation. I approach this problem equiprimordially from the various possible roles of violinist, hardware technician, programmer, sound designer, composer, and improviser.

Figure 1. Alto.glove.

3.1 Glove Design and Sensor Layout

The alto.glove evolved to include (minimalist) elements of physical extension in addition to its gesture tracking capabilities. In particular, the addition of two force-sensitive resistors (FSRs) on the index finger and thumb allow continuous control of two parameters with only partial release of the bow grip. Along with the two FSRs, the alto.glove uses three flex sensors, a nine-axis motion sensor, and three momentary contact buttons. The flex sensors are distributed on the MCP joint of the index finger and the PIP joint of the fourth finger, while a bidirectional sensor accounts for both flexion and extension of the wrist. The FSRs are located on the inside of the proximal phalanx of the index finger (for actuation by the pad of the thumb) and on the anterior of the MCP joint of the thumb (for actuation by the pad of the index finger). The motion sensor is built into a circuit board that rests proximal to the wrist, on top of which are three momentary contact buttons, arranged linearly.

The top layer of the alto.glove is constructed from two layers of Powermesh fabric sewn together at different points to create channels for the flex sensors and FSRs. Fabric on the fingers is cut short to avoid contact with any part of the bow. Sensors are secured in these channels

by adhering small pieces of neoprene to the ends of the sensors in order to stich them in place, a technique documented on the mi.mu design blog. Silicone-coated stranded wires pass between the two top layers of fabric and project out the rear of the glove, inserting into a ten-pin Molex Microblade connector attached to the PCB. The PCB is fixed to the glove by an elastic harness attached to a silicone-coated metal buckle (see Figure 2). The far end of the elastic harness has a small piece of Velcro sewn to it, allowing the strap to be secured and its tightness adjusted for individual fit. The metal clip fits neatly over the row of three momentary contact buttons located on top of the PCB. These buttons realize a "techneme" in Baudrillard's sense [2], owing to the dual purpose they serve vis-à-vis their placement: by serving as a latching point for the Velcro strip that wraps around the wrist, they help secure the PCB in place. An 850 mAh lithium polymer battery is held in a small pocket on the posterior of the wrist, with additional security provided by the elastic strap that wraps around it.

The location of the FSRs allows them to be actuated without completely releasing the grip on the bow, providing for quick triggers between musical phrases. The flex sensor on the fourth finger accounts for the horizontal position of the bow in the hand, while the sensor on the wrist is a good proxy for the distance of the hand from the strings. The accuracy of these metrics, however, depends to a certain extent on aspects of the performer's technique and overall development.

Figure 2. Alto.glove elastic harness and buckle.

3.2 Cost-Effective Hardware

The form-factor of the PCB was dictated by the choice of microcontroller. The alto.glove uses a Feather M0 WiFi development board manufactured by Adafruit, which contains a WINC1500 chipset for wireless capability, a 48 MHz SAMD21 processor, as well as on-board lithium polymer charging through a micro USB port. Initial experiments were done utilizing a flexible PCB material (Pyralux), but a hard PCB solution was settled on after successful prototyping with the elastic strap and buckle. Ease of production and cost-efficiency were important factors in the development of the alto.glove, which is why I chose the Adafruit board ($35) over the faster and more capable—but much more expensive—X-OSC board ($200). UDP packets are steadily streamed at ~155 Hz to the PC. The motion sensor is embedded in the PCB at the wrist, but was placed on top of the hand in an earlier version, secured to the fabric with E6000 adhesive. (This version ultimately failed due to continuous stress on the projected wires.) If the haptic feedback motor and addressable RGB LED are used minimally, I estimate that the glove can be used for upwards of six hours on a single charge.

4. COMPUTATION AND MAPPING

While I make use of the accelerometer to determine features of the hand such as roll and pitch as well as general motion, I find the gyroscope to be much more useful for extracting salient features of bowing. In my review of the literature on bowing parameter extraction, I found that most trials relied on just an accelerometer (likely due to the prohibitive cost of these sensors until only a few years ago). One exception is the Mini-MO sensor in the 2011 version of the IRCAM Augmented Violin, which contains both an accelerometer and a gyroscope [11]. I tried several approaches to feature extraction, including the use of supervised learning with the Wekinator software. While such an approach eliminates the need to procedurally think through the relationship of sensor positions and values to various features of bowing—and indeed, the purpose of machine learning is to articulate relationships that resist procedural codification—the goal of supervised learning is typically classification, and I realized early on in my project that I was not interested in using classification of bowing styles to generate variation or structural changes in my performances.

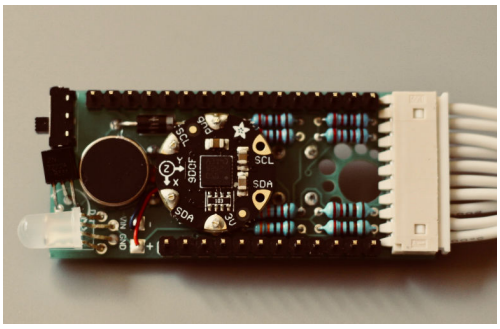

Figure 3. Alto.glove PCB shield.

4.1 "The System of Springs"

Just as data must be thoughtfully "prepared" for supervised learning by going through additional layers of selection and windowing, skillful placement of the motion sensor on the body of the performer, especially for a well-defined musical activity such as violin playing, increases the initial power of the system. A gyroscope is an obvious candidate for motion interpretation in violin playing because these motions can be accurately described and segmented according to changing angular velocities and zero-crossings, respectively. The great violin pedagogue Ivan Galamian described violin playing as a "system of springs" distributed among bow hair tension, bow stick flexibility, and "the joints of the shoulder, elbow, wrist, fingers, and thumb" [8]. Galamian elucidated "three stages of the stroke," which he called the "triangle," "square," and "point" positions. When the bow is set on the string at the frog, the shape of the arm resembles a triangle; when set at the midpoint, it looks like a square; and when set at the tip or "point," the arm is more or less extended, depending on the length of the player's arm. These and other settings are normative because they reduce tension and strain in the player's body while maximizing the use of gravity for applying pressure to the string and drawing out a full tone. Bow motion from the frog to the middle of the bow should come from the shoulder rather than the elbow, whereas bowing between the middle and the tip of the bow is controlled by the elbow rather than the shoulder. This explains why using the angle of the elbow would not be a useful metric for determining bow position. On the other hand, the z-axis of a gyroscope mounted flat above the wrist provides a very good indication of bowing velocity across the strings. And because string crossings correlate to pronation and supination of the forearm, the x-axis of the gyroscope is a superior metric for these motions as well.

This mechanistic account of violin playing is more primordial than first thinking bowing in terms of high-level semantic categories such as the *detaché, martelé,* or *spiccato* strokes. This is because a mechanical view exposes the more basic elements of these movements: *detaché* strokes smoothly transition into each other with no pause in the stroke and include a mild, vertical springing movement that does not cause the bow to leave the string. *Martelé*, by contrast, involves a full stop between each stroke without the bow leaving the string; if it does, it becomes *spiccato*. Each of these strokes extends a simpler stroke, which is why D.C. Dounis, another highly influential pedagogue, explains that the *detaché* is "the basis of the whole bow technique" [6]. The list of possible classifications goes on: *legato*,

accentuated legato, *thrown bow*, *flying staccato*, *thrown staccato*, etc. A potentially significant complication is the possibility of subtle differences in technique among players due to different schools of training. Louis Kievman, for instance, taught that *parlando* (*accentuated legato*) should be achieved by pulsating the thumb and keeping the other fingers still, but other players may have different means of accentuating the stroke [10]. Mounting sensors directly to the bow abstracts from these technical differences, but at the cost of obliterating the sensitive dynamics of the bow. Therefore, I propose working exclusively with body-mounted hardware in the context of violin playing. In addition, rather than using high-level semantics as the minimum units in supervised machine learning, thinking about the constituent motions of bowing—full stops versus continuity, vertical springing versus stasis—should be used to produce a much thicker set of basic motions that a violinist might then explore through motion tracking hardware and highly responsive software. At the limit, thinking about these mechanics may even suggest that the specifics of violin playing can be altogether "bracketed" in designing the responsive system. Returning to Sha's TGarden, holding this knowledge in abeyance would be tantamount to putting an exaggerated "costume" on the violinist, who now explores the possibilities of the responsive media system without feeling compelled to capitulate to canonistic demands of "correct" playing. Practical implementation of this principle would simply mean adhering to these more pre-semantic metrics of gesture tracking at the level of signal processing.

4.2 Data and Signal Processing

After floating point conversion, digital filtering, and calibration, data streams from x- and z-axes of the gyroscopic pass through initial "first-order" processing in Max MSP. Positive and negative values are split into two independent data streams. For each stream, running total, running average, and peak (or trough) values are calculated. A "peak on change" value outputs the peak (or trough) value of a stream only when the sign has changed. In the case of the x-axis, the resulting metric can be used to determine whether one, two, three, or all four strings are being played across. Clockers keep track of the time spent in either the positive or negative ranges of the gyroscopic axes, correlating to the duration of individual bow strokes. This processing was prepared rather arbitrarily (experimentally), with the intention of visualizing the resulting data while playing a variety of bow strokes and styles to look for interesting, continuous metrics that might be mapped to sound processing elements—"interesting" in the strict sense of being between entities (*inter esse*) or categories, *a priori* defined markers of violin technique that conceal latent, interstitial behaviors.

A layer of "second-order" processing reconstructs higher-level semantic features of violin playing. But here, too, a *poetic* extraction scheme—by which I mean that strict goals of the process are not set out in advance, but rather left to emerge out of the process itself—is used to produce novel metrics. For instance, the amplitude envelope of the violin is multiplied by the value of the gyroscopic z-axis generalized to a value of "1," during up-bows, or "-1," during down-bows. By merging sensor metrics with a continuous audio envelope, the problem of setting a determinate threshold for an audio gate is avoided. In other instances, it is decided on a case-by-case basis if and how the audio envelope should gate incoming sensor data, e.g., if a value freezes when the gate is closed, or ramps to "0" over a short period of time, etc. The "running total" or "accumulation" of gyroscopic values between sign changes is a particularly interesting metric, since it continuously describes both duration and speed of individual strokes. A metric combining the three axes of the accelerometer is used to determine stroke "intensity."

4.3 Mapping

As Magnusson has pointed out, the software substrates of digital instruments make these instruments highly mutable [12]. In the alto.glove software, several features are made available through the processing schemes outlined above. When mapping these features to sound processing elements, I make use of some of these second-order, higher-level metrics, but often find the need to come up with different solutions to be coded extemporaneously with other available features. If the results are not highly idiosyncratic to a particular project—or even if they are—I name the result and embed the solution in a dedicated data processing/communication application used with the alto.glove, which makes these features available to other applications via the OSC networking protocol.

Clever solutions are required to make compelling connections between bowing and sonic results. To this end, it has been useful for me to think of sonic processing in terms of multiple layers of sound taking place in distinctive temporalities. During improvisation, triggering sound recording and remodeling processes according to changes in bowing direction is compelling, since this provides an immediate and dramatic response to a discrete change in motion. But on the other hand, "triggered" sounds are deprived of the finesse that is achievable in continuously nuanced, "gestured" sounds—that is, in the friction process of bowing itself.

4.4 "Windowless"

Cook's "fifth principle" in his paper on designing music controllers is: "make a piece, not an instrument or controller" [4]. This principle can be understood as the corollary to Magnusson's observations about the essential *tabula rasa* of the digital controller. To the extent that mappings between digital controllers and their software substrates result from decisions that could have, in principle, been made otherwise, these mappings are *arbitrary* in a strict sense. I leave it undecided whether Windowless is a particular composition for the alto.glove or an instrument paradigm with tighter coupling between a substrate of sonic material and specific features of the hardware controller—although clearly it is not and could not be the latter. What is presented is rather like a hyperbolic acoustics, a term perhaps more apropos in the context of augmentation than *composition*, *instrument*, etc. I have simply provided myself with a dense palette of sonic affordances tied to various levels of pre-semantic signal processing and higher levels of gesture tracking salient to the violin. I will explain these affordances and some aesthetic principles they point to in the sections that follow.

4.4.1 Modularity

Windowless is organized into three standalone applications: an application that performs feature extraction information derived from the alto.glove and two additional applications that receive that information via OSC and "map" it to a variety of sonic processes. Fourteen sound "modules" are available in total from both of these applications. Many of the modules have a large number of parameters that can be finely tuned to alter its overall behavior and sonic output. To abstract from this complexity, I gave myself the constraint of limiting each module to a total of ten possible presets. Each module is, moreover, very idiosyncratic, even "overdetermined"—they are not intended to be abstract synthesizers, but very concrete effects. For instance, a two-voice, polyphonic FFT resynthesis module exists with two layers of harmonization mixing, tape-warble, and a reverberation module. The order of these processes is fixed. The module is mapped to the duration of individual bow strokes. Example presets for this module include: "up-bow / 500 ms / minor," which "freezes" the sound after 500 milliseconds during up-bows only, and pitch-shifts the output to create a minor harmony. A more agitated setting is "bidirectional / 25 ms / fifth," since the frozen sound now rapidly responds to changes in bowing direction. ("Fifth" is shorthand here for a pitch-shifting scheme that uses only octaves and fifths.) Working with this shorter

"trigger" time narrows the temporal stratum of improvisation and musical invention, since there is less room for deliberate control of temporally extended responses. By contrast, longer triggers of at least one or two seconds allow for methodical capture and harmonic layering. Although not implemented in this module, continuous control of this triggered process might be recuperated by creating a relationship between the amplitude of the violin input and the amplitude of the module, whether direct or inverted. These are decisions to be made empirically based on experimentation with the system in real-time. An additional refinement might be to scale the relationship between violin amplitude and module amplitude according to a different parameter on a larger time-scale, which would make the system tend even more toward "semiotic" (as opposed to "symbolic") behavior [7].

4.4.2 Experimentalism

Reducing the trigger time in the freeze module to a limit that produces spastic output points to a more general principle affirmative of an experimental ethos. This principle is: try setting parameters beyond the range that would produce normal operation or (more) predictable behavior. For instance, another sound processing module I created and call "Arvo" records microphone input from the violin into sixteen individual buffers, one after the other. Sequential buffer selection and recording are controlled by a noise gate that receives microphone input. With a "hold" time of 150 milliseconds, individual "notes" can be deliberately recorded into the system. A hold time of 10 milliseconds, on the other hand, creates a rapid, granular-like response. Each of the buffers is algorithmically set to a different playback speed that effectively pulls the playback up by an octave, down by either one or two, or leaves it unchanged. In addition, the various playback rates can be correlated with particular audio effects, such as distortion, low-pass filtering, or reverberation. The FSR on the index finger is mapped to provide global control over this module: when actuated, a gate is opened briefly that plays all of the buffers into a long reverberation unit before clearing them. This allows for the deliberate use of "reverberant space as a cadence," an effective means of phrasing in electroacoustic composition [14].

4.4.3 "Subtractive" Synthesis, Delay

Another feature in the Arvo module is the ambisonic panning of the sixteen voices. While this module was originally built for use in an eight-channel space, selecting only two of the eight channels for use with stereo diffusion reveals a compelling imperative: increase the textural subtlety of your sonic environment by decreasing its symmetry and spatial-temporal roundedness. In the case of Arvo, listening to only two channels of ambisonic output that, over eight channels, would create a revolving "circle" of sixteen captured loops, reveals a much more dynamic sonic environment.

Other features in Windowless follow this principle of system perturbation, whether by removing audio components or introducing subtle delays that disrupt otherwise immediate relationships. For instance, grasping the bow during *pizzicato* playing causes the microphone input to be passed through a long reverberation unit, but a short delay is added to the gating logic in order to slightly skew this relationship. The result is that some initial plucks may not be passed through the reverberation unit, but other sounds from the violin may be passed through even after the bow grip is reset for *arco* playing. In general, obfuscations of this type have the potential to break one-to-one mapping dynamics, leading to richer sonic results and performances.

4.4.4 Instruments, Machines

Windowless blends instrumental and machining dynamics. The modules described thus far capture and modulate instrument dynamics of the violin. There are still more of these modules, including one I call "Holst" that chorographically modulates the amplitude and panning of an FFT resynthesized sound, with each component of the trajectory—the initial rise in amplitude, the spatial movement across the field, and the release of the sound at the end of this trajectory—modulating the output chain in a particular way. This module is triggered by *tremolo* playing, modeled by having changes in bowing direction increment an *incdec* object in Max MSP that is simultaneously decremented at a fixed rate. Another module creates a rush of sustained reverberation during prolonged drawing of the bow.

The most obvious and present machining dynamics in Windowless are produced by a fifty-voice polyphonic pulsar synthesis engine I built. Randomization of upper and lower limits for frequency, duty cycle, pulse probability, duration, and amplitude, along with global control of density, produce a full range of rich textures, from scintillating electricity-like noises, to grunge, to more gurgling ambiances suggestive of fluid dynamics or even swarming insects. The control structure for this module is complex and resists procedural explication (see Figure 4). I call this module "Stockhausen." Bowing intensity is combined with detection of microphone audio signal to drive randomized parameter selection in the pulsar module. Bowing duration drives multiple parameters, including the amplitude modulation of pitch-shifted sound that varies according to changes in bowing directing, and delay modulation of the pulsar output. The FSR on the index finger triggers a pulsar train but also affects reverberation added to the signal. There are still many other correlations. In general, "Stockhausen" hyperbolizes the principle of one-to-many mapping that characterizes highly interdependent, semiotic systems [7].

A second machining dynamics module tracks the rate of string crossings to drive a form of frequency modulation synthesis. The string crossing metric is calculated by adding the peak and trough values of the x-axis of the gyroscope between sign changes, summing the current and two previous values, then comparing this result to a predetermined threshold. The time interval between string crossings is also calculated. When the

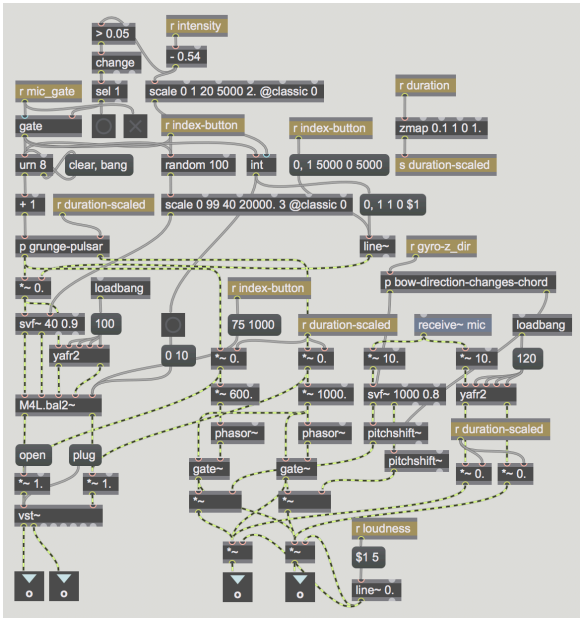

Figure 4. "Stockhausen" module.

string crossing "mode" is detected, the crossing time interval is multiplied by the amplitude of the signal coming from the violin to drive a *phasor~* that sweeps a wide frequency range to modulate and drive a *rect~* oscillator. Loudness values are multiplied into integer range, then rounded off to create stepwise modulation. In addition, sequential triggering of the string crossing mode toggles the mapping mode between this FM module and a live granulator modulated with pink noise.

4.4.5 Substrate, Scale vs. Zmap

It should now be clear that the sonic substrate of Windowless is highly idiosyncratic in terms of the sound modules that are driven and the way in which glove/violin metrics are calculated and combined. Many layers of sound processing occur at once, yielding rich textures and ambience. Multiple functions are associated with the FSR on the index finger. The bow is spatially "thickened" by pitch-shifting the dry sound of the violin down an octave as the tip is approached. Likewise, I amalgamate multiple metrics throughout the software in order to make the system richer and more semiotic. For example, I have discovered that rapid, exaggerated movements of the bow can trigger the string crossing mode even when (deliberate) string crossings are not being performed. Rather than mitigate this error by elongating the minimum period for string crossing detection, I tend to leave such "loose" calculations in place. In general, the ethos I am affirming here—a hermeneutics of misinterpretation, the fertility of error, and the unique affordances of ambient rather than procedural programming more generally—might be represented by the difference between the *zmap* and *scale* objects in Max MSP: while the former clips the range of input values to established limits and permits only direct scaling, the latter allows input values to spill past the anticipated limits, thickening the system and enriching the possible responses. The procedure is: build a loose system, discover its affordances, tighten it up—but never too much. I use pitch-tracking in a similar manner, which is still a difficult operation in the context of string instruments. Instead of suppressing the *errors* (*err*: to wander, to stray), I control and exaggerate them by lowering the input amplitude to increase these generative and fertile meanderings.

5. CONCLUSION

My goal in this paper was to articulate an approach to hardware, sound, and mapping design for the violin that neither reifies the tradition of its performance practice nor brushes it aside. This is accomplished by designing relatively minimalist hardware with added affordances, and by use of a dense, highly responsive computational substrate which I explore improvisatorially. Rather than engage with bowing at a high semantic, discrete, classificatory level, the system observes these dynamics at a more rudimentary, mechanical one. This increases the responsivity of the system to my playing and allows for exploration of interstitial and exploratory gestures, ranging from very fine, muted motions to more hyperbolic, ecstatic, and extreme possibilities. This framework is, moreover, guided by a reflection on fundamental aspects of violin bowing technique relayed by the tradition of its performance practice. Bowing technique consists of a set of basic movements, e.g., if strokes are smoothly or abruptly terminated, the amount of vertical springing action, the lifting of the bow from the string, and other motions. The use of machine learning to abstract from these features also abstracts from the expressive possibilities rooted in the material of "singular" strokes. Likewise, my system is designed to respond to all movements, even and especially when they exceed anticipated input limits. The result is a palette of possibilities that work together in a "rich" but—from the point of view of the performer—"not complicated" way [16].

6. ACKNOWLEDGMENTS

My thanks to Sha Xin Wei, whose approach to media was greatly influential on the development of my system and saturates its elaboration in this paper; to my advisor, Butch Rovan; and to the Brown Arts Initiative, for its generous financial support of this project.

7. REFERENCES

[1] T. W. Adorno, *The Philosophy of New Music.* Translated by Robert Hullot-Kentor. Minneapolis: University of Minnesota Press, 1998.

[2] J. Baudrillard, *The System of Objects.* Translated by James Benedict. New York: Verso, 2005.

[3] F. Bevilacqua et al., "The Augmented Violin Project: Research, Composition and Performance Report." In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, 402–406. Paris, France: IRCAM — Centre Pompidou, 2006.

[4] P. Cook, "Principles for Designing Computer Music Controllers." In *Proceedings of the CHI'01 Workshop on New Interfaces for Musical Expression (NIME-01), Seattle, USA*, 2001.

[5] P. Cook, "Remutualizing the Musical Instrument: Co-Design of Synthesis Algorithms and Controllers." In *Proceedings of the Stockholm Music Acoustics Conference*, 33:1–4. Stockholm, 2003.

[6] D.C. Dounis, *The Artist's Technique of Violin Playing*. New York: Carl Fisher, 1921.

[7] A. Evens, *Sound Ideas*. Minneapolis: University of Minnesota Press, 2005.

[8] I. Galamian, *Principles of Violin Playing and Teaching*. Edited by Sally Thomas and Stephanie Chase. Mineola, New York: Dover Publications, 2013.

[9] F. Gemperle et al., "Design for Wearability." Digest of Papers, Second International Symposium on Wearable Computers (Los Alamitos, CA: IEEE Computer Society, 1998), 116-122.

[10] L. Kievman, *Virtuoso Violin Technique*. Brooklyn: Kelton Publications, 1979.

[11] M. Kimura et al., "Extracting Human Expression for Interactive Composition with the Augmented Violin." In *Proceedings of the 2012 Conference on New Interfaces for Musical Expression*, 2012.

[12] T. Magnusson, "Of Epistemic Tools: Musical Instruments as Cognitive Extensions." *Organised Sound* 14, no. 2 (August 2009): 168.

[13] D. Overholt, "The Overtone Violin." In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, 34–37. Vancouver, BC, 2005.

[14] C. Roads, *Composing Electronic Music: A New Aesthetic*. New York: Oxford University Press, 2015.

[15] M. Schedel and R. Fiebrink. "A Demonstration of Bow Articulation Recognition with Wekinator and K-Bow." In *Proceedings of the 2011 International Computer Music Conference*. Ann Arbor, 2011.

[16] X. W. Sha, *Poiesis and Enchantment in Topological Matter. Cambridge: MIT Press, 2013.*

[17] D. Young, "Classification of Common Violin Bowing Techniques Using Gesture Data from a Playable Measurement System." In *Proceedings of the 2008 International Conference on New Interfaces for Musical Expression*, 44–48. Genova, Italy: Citeseer, 2008.

8. APPENDIX

A performance with the alto.glove and Windowless can be viewed here: https://vimeo.com/251840089.