Touch Responsive Augmented Violin Interface System II: Integrating Sensors into a 3D Printed Fingerboard

Chantelle Ko
University of Calgary
Calgary, Alberta, Canada
chantelle.ko@ucalgary.ca

Lora Oehlberg
University of Calgary
Calgary, Alberta, Canada
lora.oehlberg@ucalgary.ca

ABSTRACT

We present TRAVIS II, an augmented acoustic violin with touch sensors integrated into its 3D printed fingerboard that track left-hand finger gestures in real time. The fingerboard has four strips of conductive PLA filament which produce an electric signal when fingers press down on each string. While these sensors are physically robust, they are mechanically assembled and thus easy to replace if damaged. The performer can also trigger presets via four FSRs attached to the body of the violin. The instrument is completely wireless, giving the performer the freedom to move throughout the performance space. While the sensing fingerboard is installed in place of the traditional fingerboard, all other electronics can be removed from the augmented instrument, maintaining the aesthetics of a traditional violin. Our design allows violinists to naturally create music for interactive performance and improvisation without requiring new instrumental techniques. In this paper, we describe the design of the instrument, experiments leading to the sensing fingerboard, and performative applications of the instrument.

Author Keywords

Violin, touch sensor, FSR, fingerboard, augmented, 3D printing, conductive filament, interactive

CCS Concepts

• **Hardware → Communication hardware**, interfaces and storage → Sensors and actuators;
• **Hardware → Communication hardware**, interfaces and storage → Sound-based input/output;

1. INTRODUCTION

The violin's design continues to develop into the digital age, with the introduction of commercially available electric and MIDI violins. Augmented violins combine traditional instruments with computer programming, and thus offer new expressive possibilities for performers and composers. Some augmented violins also offer new ways for violinists to improve their technique [3, 6, 7, 21, 22, 25]. The Touch Responsive Augmented Violin Interface System (TRAVIS) II presents a new method of designing an augmented violin in which conductive strips on the fingerboard detect contact with the strings.

Both authors are classically trained violinists, and the first author has an extensive background in electroacoustic and interactive music composition and performance. As such, the TRAVIS project follows an autobiographical design approach where the instrument was built to serve the first author's creative needs [18].

TRAVIS II is an iteration on a prior project, TRAVIS I [11]. TRAVIS I is an acoustic violin with two linear SoftPot sensors placed under the G and E strings, and two FSRs (Force Sensitive Resistors) clamped to the right upper bout. The SoftPots in TRAVIS I do not cover the areas on the fingerboard below high second position; to sense in all positions, the nut and therefore the string height would need to be raised to prevent SoftPots from contacting the vibrating strings. Also, the player can only use the SoftPots on the G and E strings; there is no sensing on the D and A strings. Furthermore, the aesthetic of TRAVIS I makes it difficult to use in traditional contexts. Its SoftPots and wiring are permanently attached and cannot be removed from the violin when stored in its case. Finally, TRAVIS I has only two FSRs to trigger presets, which limits its compositional potential.

Figure 1. TRAVIS II with electronics setup.

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

The TRAVIS II design seeks to address these limitations in TRAVIS I and increase the expressive flexibility of TRAVIS as an augmented instrument. TRAVIS II uses four strips of conductive PLA in a custom 3D printed fingerboard, which can sense the violinist's left-hand gestures on all four strings and for the full length of the strings. TRAVIS II includes a set of four FSRs that can select programmed presets for both sound and visuals. We also designed the violin to be continually useable in a traditional context such as an orchestra ensemble, rehearsal, or sight-reading session. We added all additional technology via removeable, 3D printed clamps, and the sensing fingerboard does not alter the instrument's original dimensions (e.g. fingerboard thickness, string height). The instrument is physically robust and has easily replaceable sensing elements. In this paper, we describe the instrument design, experiments in 3D print filament, printer settings, string selection, and musical applications of the final instrument.

2. Related Work

Miranda and Wanderley [17] proposed four categories of new digital instruments: alternative gestural controllers, instrument-inspired gestural controllers, instrument-like gestural controllers, and augmented musical instruments. We position our work as a new *augmented musical instrument*. Augmented instruments are broken down into two groups [19]. The first group are instruments that track traditional playing gestures and techniques, such as IRCAM's Augmented Violin Glove [2, 9]. The second group are instruments that require the invention of new extended gestures, such as Overholt's Overtone Violin [20].

Many augmented violins and cellos have focused on tracking the bow or the right hand: IRCAM's Augmented Violin Glove [2, 9], Guettler et al.'s Electronic Violin Bow [8], McMillen's K-Bow [16], and Young's Hyperbow [25]. Moreover, Machover's Hypercello [13, 14] and Pardue et al.'s Tracking System for Violin [22] contain components for tracking both left and right hands.

Another method of capturing violin gestures was through optical tracking, such as Schoonderwaldt and Demoucron's research in tracking both the left and right hands [23]. Dalmazzo and Ramirez's Air Violin also tracked the left hand with optical tracking and an electromyographic (EMG) sensor [3]. Thorn's Transference used an EMG and gyroscope to track the left hand finger and arm gestures, as well as an IMU and flex sensors to track the right hand [24].

The preference for augmenting the bow or tracking the right hand is because, in traditional playing, the bow is where the majority of the expression is produced: it determines the timbre, articulation, and dynamics of the music. In addition, if permanent physical alterations need to be made, it is more cost efficient to alter the bow than it is to alter the instrument. However, to capture and map the full expression of the player, both hands need to be taken into consideration. Attaching electronics to the bow may also add extra weight and change the balance point [25].

Augmenting the fingerboard of string instruments is an attractive approach. However, this can present many physical and technical challenges. In working on the Hypercello [13, 14], Machover found that the cello strings were two abrasive for the thermoplastic attached to the fingerboard, and the instrument's calibration changed over time. In the Augmented Cello project [4, 5], Freed et al. tried adding touch sensors to the fingerboard, but found that the resulting data was non-linear, and the data varied between strings, so they abandoned it. There are one augmented bass project and two augmented violin projects that successfully added touch sensors to the fingerboard. Bahn added a mouse touch pad to the fingerboard of the SBass [1], while Grosshauser and Troester's sensor fingerboard had embedded FSRs [6,7]. Grosshauser and Troester's instrument was used to study how violinists play and when they were pressing too hard with their fingers. However, the copper coloured sensors aesthetically stood out, and we speculate that violinists would not be able to continue to use the instrument in a traditional context. Pardue et al. sought to solve this problem with a low-cost real-time tracking system for violin [22]. The system intended to help students improve their intonation by providing feedback based on the sensors that track left hand finger position, combined with pitch tracking. The touch sensitive component of the system was made with velostat on top of the fingerboard and a voltage running down the strings. While the technology could be attached and removed from any violin, the string height had to be raised to ensure that they were clear of the velostat when playing open strings. It is important to note that in a later iteration of the project, Pardue et al. [21] made a new violin for beginner violinists that no longer relied on touch sensors to detect intonation errors. Instead, it used digitally automated pitch and tone correction, so the violin always played in tune.

Beyond improving upon the TRAVIS I design, our approach to augmentation is inspired by both Pardue et al. [22] and Grosshauser and Troester [6, 7]. We seek to address design challenges presented by the violin's geometry with a novel solution: use strips of conductive 3D print filament embedded into a 3D printed fingerboard to create touch sensors. These conductive strips are more resistant to damage than thermoplastic or velostat, and individually replaceable if damaged. The instrument is aesthetically and geometrically close to a classical violin such that it can be played in a traditional context. TRAVIS II is primarily used to explore new creative, expressive, improvisational, and compositional possibilities in interactive music.

3. TRAVIS II IMPLEMENTATION

TRAVIS II is a design iteration of the TRAVIS I concept. In this section, we discuss the technical implementation of TRAVIS II and how it improves upon the TRAVIS I design. In addition to starting from better violin model than TRAVIS I, the augmentation includes full-length sensors on all four strings, and four FSRs instead of two.

3.1 The Fingerboard and Sensors

The primary feature of TRAVIS II is its sensing fingerboard. The main fingerboard is 3D printed with black PLA; its geometry includes four slots underneath each of the strings. Due to the build volume of the 3D printer, the fingerboard was printed in two parts. Four sensor strips made from ProtoPasta's black conductive PLA slide into the slots.[1] The strips are designed to slide in and are secured with a press fit. If they become damaged, they can slide out to be replaced. The strips are flush with the surface of the fingerboard, so the strings do not need to be raised. Within close proximity the differences in texture between the strips and the PLA fingerboard are visible (see Figure 3). However, from the audience's more distant perspective this difference is not noticeable.

For testing, we attached different PLA fingerboards to the neck of the violin with double-sided tape. In the first test, the PLA fingerboard was too flexible and, when played in high positions, it bent from regular finger pressure. Open source designs for 3D

---

[1] ProtoPasta's conductive PLA: https://www.proto-pasta.com/pages/conductive-pla

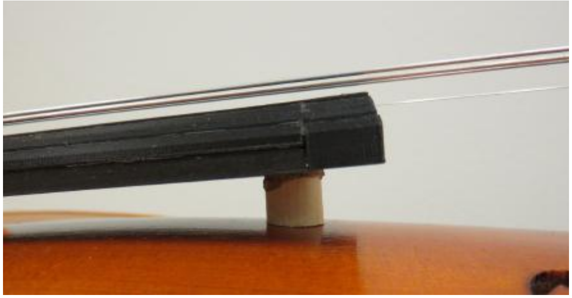


Figure 2. Side view of the end of the fingerboard and support.

Figure 3. Front view of the end of the fingerboard without the Arduino and FSRs.

printed violins have addressed this problem with support rods.[2] However, the slots for the conductive strips leave very little room to include a support rod. Therefore, we worked with a luthier who placed a thin, flat piece of ebony onto the neck of the violin to support the PLA surface. After a few weeks, while the fingerboard continued to support the weight of the finger pressure, we found the natural placement of the fingerboard started to sag. In response, the luthier also placed a small piece of wood underneath the fingerboard to prop it up into the correct position (see Figures 2 and 3). A copy of the fingerboard was made from PETG filament. It was not noticeably stronger than the PLA version. Aesthetically, the PLA version is preferred because it is not as shiny as the PETG.

The aesthetic of the sensing fingerboard more closely resembles the aesthetic of a traditional fingerboard; both the regular and conductive PLA are black. While some layer lines from FDM printing are visible on the PLA surface, the texture of these lines against the fingertips do not distract from playing.

As documented by Leigh et al. [12] and McGhee et al. [15], conductive 3D printer filament works well as the resistive component of touch sensors. Similar to Pardue et al.'s design [22], TRAVIS II runs a 3.3V voltage down the strings; when a metal string comes in contact with a conductive strip, the string's voltage changes.

We also placed four small FSRs to a 3D printed clamp and mounted it on the right upper bout of the violin (see Figure 1). Here they are easily accessible to the performer; violinists rest their left hand on the right upper bout of the violin when not playing, and the FSRs are directly below the left hand when playing in high positions.

TRAVIS II uses an Arduino MKR1000 to send data to Max MSP/Jitter via OSC messaging. TRAVIS II's Arduino is multiplexed for eight sensors and sits inside a custom 3D printed case mounted on the left upper bout of the violin. This placement is based on the TRAVIS I design. We considered alternative placements for the Arduino. The Arduino could clamp onto the left lower bout, where the weight would be less noticeable on the shoulder. However, this placement would obstruct the performer from bowing behind the bridge; an extended bowing technique. Alternatively, the Arduino could attach underneath the shoulder rest, as in Grosshauser and Troester's violin [6, 7]. However, shoulder rests are not very secure and many accidentally fall off mid-performance. Placement near the shoulder rest would also limit how the shoulder rest can be adjusted; some performers would not be able to use their preferred shoulder rest height. By placing our Arduino and battery on the left upper bout of the violin, our added electronics do add some weight to the violin. However, we have found that this added weight is not uncomfortable and keeps all components safely out of the way while playing.

The wires, the Arduino case, and the FSR clamp easily disassemble from the violin, both for storage in the case and when playing the violin in a traditional context. The wires connect to the Arduino via an IDC connector, to the 3D conductive strips with Swiss machine headers, and to the strings with a JST connector. The JST connector sits underneath the chin rest and the wires are soldered onto the balls of the strings. Strings are easily replaceable by cutting off the wires, then re-soldering the wires to the new strings.

4. RESULTS

Our initial tests ensured that TRAVIS II could work sufficiently as an augmented instrument, by producing a distinct range of values from the 3D printed touch sensors. We then tested the 3D printing settings, resistors, and strings to ensure that the instrument could produce reliable and stable data for the Arduino with the largest range of values between 0-1023. However, we placed higher priority on the physical design. This included the sensor strips having dimensions that fit into the fingerboard, are durable so they do not break when assembling them, and they have a smooth, high-quality finish on their surface.

We printed different dimensions of the conductive strips and used different 3D printing settings to see how these variables affect the strip's resistance. We tested the resistance with a multimeter at the 20kΩ setting. We recorded the resistance at three points along the strips. First, we measured the resistance across 1 cm of the strip at one end. Then, we measured the resistance from one end to the approximate center of the strip (appx. 13.2 cm). Finally, we measured the resistance at either end of the strip (appx. 26.4 cm). After we finalized the dimensions and 3D printer settings, we tested each of the final strips for the fingerboard one last time before integrating them with the violin.

Afterwards, we tested the range of analog data values read into the Arduino with various resistors in the circuit (1kΩ, 3.3kΩ, 4.7kΩ, 10kΩ, 22kΩ, and 47kΩ). We setup the circuit with the test strips, the strings, and resistors; we then recorded sensor values resulting from pressing the string at each end of the conductive stip. We selected final resistors based on which provided the largest range of sensor data.

4.1 The Conductive 3D Print Filament

The dimensions of the strips are tightly constrained, particularly when fitting four into the geometry of a standard fingerboard. If strips are too wide, the slots are too large to fit into the fingerboard. If strips are too narrow, they could break while sliding them into the fingerboard, or the string may miss the strip surface when pressed.

We initially tested a conductive strip that was 265 mm long, 2.5 mm tall, 5 mm wide at the end closest to the bridge, and

---

[2] Form Labs violin: https://formlabs.com/blog/designing-a-3d-printed-acoustic-violin/

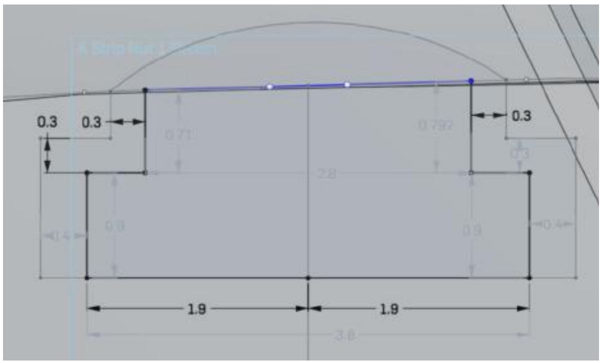


Figure 4. Screenshot of the face of the A strip (dark outline) from the nut end. The faded outline is the sketch for the face of the slot. Measurements are in millimeters.

4 mm wide at the end closest to the nut. However, when measuring its resistance, the resultant readings were not stable, and we could not record approximate measurements. The next test strip was narrower; consistently 3 mm wide at both ends, and produced a stable, measurable resistance value.

We tested different cross-sectional geometries to identify a series of four strips that both fit within the physical constraints of the fingerboard and had a relatively consistently varying resistance across the strip. We tested the resistance with a multimeter (20kΩ setting) at three distances from the nut: 1 cm, appx. 13.4 cm (middle), and appx. 268 cm (end to end). In our final strips (cross-sectional geometry in Figure 4), the resistance changed proportionally across the length of the strip (see Table 2). We tested the resistance again after 10 months and found values close to our readings from our initial testing; thus, the conductive materials in the filament had not degraded over this time span.

We also fine-tuned the length of the strips to ensure that they could easily be installed; if the end of the strip fit too firmly with the nut end of the slot, it was difficult to later remove. The final length was 268.5 mm; we are able to slip a sewing needle behind the strips at the nut end of the slot to push them out.

4.2 3D Printing Settings

Once we had finalized the physical dimensions, we tested various slicing settings in Cura to see whether printing parameters affected resistance. We used the same process as above, measuring the resistance at three points along each strip. First, we varied the layer heights from 0.1 mm – 0.5 mm. As the layer height of each test strip increased, the approximate resistance decreased (see Table 1). Also, the taller the layer height, the lower the quality of the print. We decided to keep the layer height at 0.1 mm so that the header holes of the strips had the best print quality possible (Figure 3), and so that the surface of the strips could be as smooth as possible.

We tested three different infill patterns – Triangles, Lines, and Cubic – with a 70% infill density. The resistances of these patterns were not significantly different from one another. We did not test different infill densities because of strength concerns; the strips needed to slide in and out of the fingerboard without breaking.

By trial and error, we discovered that sanding the top surface of the conductive filament ruined its performance. In an effort to minimize the layer lines from 3D printing, we placed the strips into their slots in the fingerboard and then sanded the entire surface. This damaged the conductivity/resistivity of the sanded surfaces and there were barely any readings from the multimeter. We re-printed the fingerboard and all strips after this discovery, and no longer attempted post-processing the surface finish.

Table 1. Test strip resistance relative to print layer height

Layer Height	1 cm (Close)	~13.2 cm (Middle)	~26.4 cm (Ends)
0.1 mm	2.55 kΩ	6.1 kΩ	11.6 kΩ
0.2 mm	3.38 kΩ	6.5 kΩ	8.66 kΩ
0.3 mm	1.27 kΩ	5.8 kΩ	7.7 kΩ
0.5 mm	1.64 kΩ	4.06 kΩ	6.82 kΩ

Table 2. Approximate resistance of each of the final conductive strips (measured 20kΩ setting).

String	1 cm (Close)	13.2 cm (Middle)	~26.4 cm (Ends)
G	2.19 kΩ	6.67 kΩ	10.89 kΩ
D	1.97 kΩ	8.25 kΩ	16.7 kΩ
A	1.64 kΩ	6.63 kΩ	12.3 kΩ
E	3.07 kΩ	7.55 kΩ	10.49 kΩ

Table 3. Summary of tested strings and their materials.

String Brand	String Name	Core	Winding
Pirastro Gold	E	-	Tin-plated Carbon Steel
Dominant (135)	E 130	Steel	Aluminum
	A 131	Synthetic	Aluminum
	D 132	Synthetic	Aluminum
	G 133	Synthetic	Silver
Evah Pirazzi	E	-	Silvery Steel
	A	Synthetic	Aluminum
	D	Synthetic	Silver
	G	Synthetic	Silver

Table 4. Range of values received from the sensors when the violin is completely setup.

		Dominants with Pirastro Gold E		Evah Pirazzi	
String	Resistor	Values	Range	Values	Range
G	3.3kΩ	260-720	460	255-685	430
D	3.3kΩ	255-645	390	265-640	375
A	3.3kΩ	245-630	385	235-490	255
E	4.7kΩ	315-765	450	320-660	340

4.3 Violin String Sensor Testing

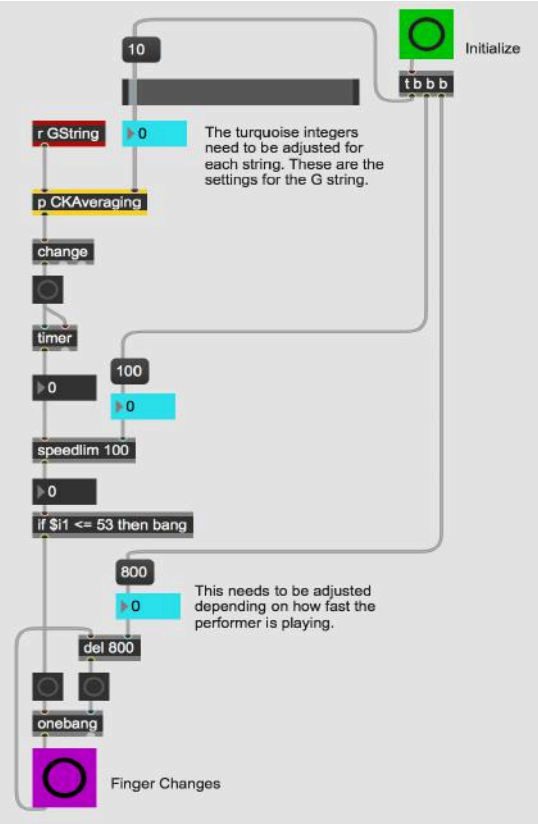
Each of the four strings on violins are made of different metals, and different brands of strings vary in the materials and manufacturing of the cores and windings. We had full sets of Dominant[3] and Evah Pirazzi[4] strings and a Pirastro Gold E. The metals that they are made from are in Table 3.

We setup these strings in the sensor circuit to test the range of data received from them with different resistor values (1 kΩ, 3.3 kΩ, 4.7 kΩ, 10 kΩ, 22 kΩ, and 47 kΩ). The Evah Pirazzi's range of data was sufficient, but we concluded that the Dominant pack, with the Pirastro Gold E, had the largest range of data (see Table 4). In its final configuration, the E string uses a 4.7 kΩ resistor, and the rest of the strings have a 3.3 kΩ

---

[3] Dominant Strings: http://www.thomastik-infeld.com/family-detail/Dominant%20Violin

[4] Pirastro Strings:https://www.pirastro.com/public_pirastro/pages/en/index.html



The turquoise integers need to be adjusted for each string. These are the settings for the G string.

This needs to be adjusted depending on how fast the performer is playing.

Figure 5. Example Max MSP patch, programming finger change recognition.

resistor. None of the tested completed sensors reached their full range of 1023 values. This is not necessarily a limitation, because the values are re-scaled in Max MSP.

5. APPLICATIONS

TRAVIS II is currently used as a controller for interactive compositions made with Max MSP/Jitter. At the time of writing, the first author has composed and performed one piece for solo TRAVIS II, *Dream State,* and one duet with both TRAVIS I and II, *Kindred Dichotomy.*[5] The duet was performed in the concert works category at AWMAS 2020 [10]. The first author also composed and performed five pieces with the initial TRAVIS I prototype; these compositions can still be performed on TRAVIS II by ignoring the D and A strings in the Max MSP patch. The four FSRs are primarily used to trigger presets for different sections of the composition. However, since they give a range of data based on pressure, it is possible to use them to scrub audio samples as well.

In addition to using finger position to control sound processing parameters, different processes can be assigned to each individual string sensor; when pressed, it routes the sound to its assigned process. This method of mapping different processes to each string was employed in both of the pieces for TRAVIS II.

TRAVIS II works particularly well with several specific violin techniques, such as double stops. If the process routing described earlier is implemented, where a different effect is assigned to each string, double-stops would route the sound to two effects, or scrub through two pre-recorded samples, simultaneously. This was employed at the end of *Kindred Dichotomy.* To achieve combined effects, the violinist also does not necessarily have to play both strings with the bow; they could keep their fingers on multiple strings while playing a melody with their bow on only one string. Keeping this in mind opens a wide range of possibilities with this one technique, and compositionally this method is more interesting than to simply assign two processes to one string.

The software can also recognize when fingers are either changing quickly or playing fast gestures, such as trills, vibrato, glissandos, and shifts (see Figure 5). These fast techniques are recognized by setting a threshold on the [timer] object in Max MSP. However, the program is unable to differentiate these techniques from one another; the composer or performer would need to take this limitation into consideration during the creative process. The sensitivity of this kind of gesture tracking also must be pre-adjusted to the speed of the violinist's technique.

6. DISCUSSION

TRAVIS II achieved its goal in augmenting a violin and improved upon TRAVIS I's design. In comparison to TRAVIS I, the violin used to make TRAVIS II is a much higher quality violin model. Before augmentation took place, based on the first author's subjective standards, the violin would have been considered an intermediate to lower-advanced level instrument. It would not have previously been used for situations such as a solo recital, audition or high-level examination. TRAVIS II has more FSRs available to change settings and all four strings can be tracked. It expands the composer's available palette of timbres, effects, and recorded samples.

One of the main limitations of augmenting a cello with touch sensors is the strings are large and thick, therefore any resistive film placed on top of the cello fingerboard is more receptive of damage than on a violin. We speculate if this method of augmentation with conductive 3D print filament can be successfully applied to a cello, and other instruments of the violin family. A much larger 3D printer would be needed.

There are some limitations to the sensor results and TRAVIS II's design. Only packs of Dominant and Evah Pirrazi strings, plus a Pirastro Gold E string, were tested. Also, only one brand of conductive PLA, by ProtoPasta, was in stock at the time of purchase. A more thorough study would include testing multiple brands of strings and conductive filament. Other brands of conductive filament have different magnitudes of resistivity, and therefore could provide different sensor data [15].

Printing only the top half of the fingerboard and sitting it on top of an ebony piece greatly improves the overall strength. PETG is a filament for FDM printers that is stronger than PLA or ABS.[6] We found the PETG version was not noticeably stronger on the violin, it did not hide the layer lines more than the PLA, and it was much shinier. Therefore, we continue using the PLA model. Layer lines are visible, and it is speculated that a resin filament would make a cleaner looking surface. Unfortunately, we did not have a resin 3D printer available and ordering models online in resin are quite expensive.

The objective of the project was to be able to remove all of the technology, with exception of the 3D printed fingerboard and hidden JST connector for the strings, in order to still be able to play the violin in a traditional context. There are still a few subtle variations that differentiate TRAVIS II from a traditional violin geometry. For example, the traditional fingerboard has an undercurve. The small PLA header piece does have an undercurve (Figure 2 and 3), but the rest of the ebony piece is flat. Visually, this difference is not noticeable from the audience perspective, so the violin can continue to be played as a

---

[5] Videos: https://www.chantelleko.com/travis-videos.html

[6] PETG and ABS comparison: https://all3dp.com/2/petg-vs-abs-3d-printing-filaments-compared/

conventional acoustic violin. However, the flat ebony piece does affect the tone of the violin and makes it sound “tinnier”. The tinny quality contributes to why we would not recommend playing this violin at a solo recital.

While the Max MSP patch in Figure 5 can measure the speed of data to recognize when fingers are playing fast gestures, it cannot differentiate vibrato from other fast gestures (e.g., shifts, glissandos, trills). It also only recognizes the location of where it is pressed, and not which finger is pressing down. Finger recognition could be achieved with EMG sensors [3, 24], or with a fingertip-less glove that has flex sensors on each finger. Vibrato recognition may also be more accurately achieved with an IMU on the hand [24], or flex sensors on the wrist, and elbow. These design concepts may be considered for future research.

Overall, TRAVIS II achieved its design goals, and at the time of writing this paper, it has been used in two successful interactive compositions. It is an example of a new method to track the left-hand finger gestures through conductive 3D print filament and offers speculation on how this method could be applied to other string instruments.

7. ACKNOWLEDGEMENTS

We would like to recognize the luthier, Aaron Pratte, who detached the original fingerboard and made the ebony piece.

The wireless version of TRAVIS I was a collaboration between the first author, Robert Pritchard, and the engineering team of WiRED: Jin Han, Esther Mutinda, Carol Fu, and Lily Shao.

8. REFERENCES

[1] Bahn, C. & Trueman, D. 2001. Interface: Electronic Chamber Ensemble. In *Proceedings of the CHI'01 Workshop on New Interfaces for Musical Expression (NIME01)*, Seattle, WA. 19-23.

[2] Bevilacqua, F., Rasamimanana, N., Fléty, E., Lemouton, S., & Baschet, F. 2006. The Augmented Violin Project: Research, Composition and Performance Report. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06),* Paris, France. 402-406.

[3] Dalmazzo, D., & Ramirez R. 2017. Air Violin: A Machine Learning Approach to Fingering Gesture Recognition. In *Proceedings of ACM SIGCHI International Workshop on Multimodal Interaction for Education (MIE’17).* ACM Press, NewYork, NY. 63-66.

[4] Freed, A., Uitti, F.M., Mansfield, S., & MacCallum, J. 2013. “Old” is the new “New”: A fingerboard case study in recrudescence as a NIME development strategy. In *Proceedings of the 2013 International Conference on New Interfaces for Musical Expression (NIME13)*. Daejeon, Korea. 442–445.

[5] Freed A., Wessel, D., Zbyszynski, M., & Uitti, F. M. 2006. Augmenting the Cello. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06),* Paris, France. 409-413.

[6] Grosshauser, T., & Gerhand T. 2014. Musical instrument interaction: development of a sensor fingerboard for string instruments. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction (TEI'14).* New York, NY. 177-180.

[7] Grosshauser, T., Großekathoefer U., & Hermann, T. 2010. New Sensors and Pattern Recognition Techniques for String Instruments. In *Proceedings of the 2010 International Conference on New Interfaces for Musical Expression (NIME10)*. Sydney, Australia. 271-276.

[8] Guettler, K., Wilmers, H., & Johnson, V. 2008. Victoria Counts -- a Case Study With Electronic Violin Bow. In Proceedings of the 2008 International Computer Music Conference. 569-662.

[9] Kimura, M., Rasamimanana, N., Bevilacqua, F., Zamborlin, B., Schnell, N., & Fléty, E. 2012. Extracting Human Expression for Interactive Composition with the Augmented Violin. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME12),* University of Michigan, Ann Arbor.

[10] Ko, C. 2020. Kindred Dichotomy. Composition performed at the *Alliance for Women in Media Arts and Sciences Conference (AWMAS2020),* UC Santa Barbara, 7 February 2020. https://awmas2020.wixsite.com/home

[11] Ko, C. 2018. TRAVIS: Touch responsive augmented violin interface system. Poster presented at the *Conference for Interactive Art Science and Technology (IAST),* University of Lethbridge. http://iast.ca/tag/research-creation/

[12] Leigh S.J., Bradley R.J., Purssell C.P., Billson D.R., & Hutchins D.A. 2012. A Simple, Low-Cost Conductive Composite Material for 3D Printing of Electronic Sensors. Public Library of Science (PLoS ONE), 7, 11.

[13] Machover, T. 1992. Hyperinstruments: A progress report, 1987–1991. MIT Media Laboratory, 53-78.

[14] Machover, T., & Chung, J. 1989. Hyperinstruments: Musically intelligent and interactive performance and creativity systems. In *Proceedings of the 1989 International Computer Music Conference*. 186-190.

[15] McGhee, J., Sinclair, M., Southee, D., & Wijayantha, K. 2018. Strain sensing characteristics of 3D-printed conductive plastics. *Electronics Letters* 54, 9, 570-571.

[16] McMillen, K.A. 2008. Stage-Worthy Sensor Bows for Stringed Instruments. In *Proceedings of the 2008 Conference on New Interfaces for Musical Expression (NIME08)*. Genova, Italy 347-348.

[17] Miranda, E., & Wanderley, M. 2006. New digital musical instruments: Control and interaction beyond the keyboard (*Computer music and digital audio series*; v. 21). Middleton, Wis.: A-R Editions.

[18] Neustaedter, C. & Sengers, P. 2012. Autobiographical design in HCI research: designing and learning through use-it-yourself. In *Proceedings of the Designing Interactive Systems Conference (DIS’12)*. ACM, 514–523.

[19] Overholt, D. 2011. Violin-Related HCI: A taxonomy elicited by the musical interface technology design space. In *Arts and Technology: Second International Conference (ArtsIT) 2011*, Esbjerg, Denmark, 80-89.

[20] Overholt, D. 2005. The Overtone Violin. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*. Vancouver, BC, Canada. 34-37.

[21] Pardue, L. S., Buys, K., Edinger, M. I., McPherson, A., & Overholt, D. 2019. Separating Sound from Source: Sonic transformation of the violin through electrodynamic pickups and acoustic actuation. In *Proceedings of the 2019 Conference on New Interfaces for Musical Expression (NIME19*). Porto Alegre, Brazil. 278–283.

[22] Pardue, L. S., Christopher H., & McPherson, A. P. 2015. A Low-Cost Real-Time Tracking System for Violin. *Journal of New Music Research* 44, 4, 1-19.

[23] Schoonderwaldt, E., & Demoucron, M. 2009. Extraction of bowing parameters from violin performance combining motion capture and sensors. *The Journal of the Acoustical Society of America*. 126, 5, 2695–2708.

[24] Thorn, S. 2019. Transference: A Hybrid Computational System for Improvised Violin Performance. In *Proceedings of the 13th International Conference on Tangible, Embedded and Embodied Interaction (TEI'19)*. Tempe, AZ, USA. 541-546.

[25] Young, D. 2002. The Hyperbow: A Precision Violin Interface. In *Proceedings of the 2002 International Computer Music Conference*. 489-492.