T-Patch: a software application for T-Stick Digital Musical Instruments

Takuto Fukuda IDMIL, CIRMMT, McGill University
550 Rue Sherbrooke Ouest, Suite 500
Montreal, QC, H3A 1E3, Canada
takuto.fukuda@mail.mcgill.ca

Marcelo M. Wanderley
IDMIL, CIRMMT, McGill University
550 Rue Sherbrooke Ouest, Suite 500
Montreal, QC, H3A 1E3, Canada
marcelo.wanderley@mcgill.ca

ABSTRACT

This paper introduces the T-Patch, a software application that streamlines the use of T-Stick Digital Musical Instruments (DMIs). It offers a user-friendly interface for gesture extraction, mapping, signal conditioning, sound synthesis, and sequencing with cues, enabling composers to create music without programming, therefore lowering the entry fee associated with the use of the T-Stick. Our main contribution is two-fold: (1) providing a versatile software solution to address the current lack of music-making support for T-Stick DMI, and (2) highlighting the importance of demonstration content, such as a video, to showcase the instrument's capabilities and inspire new users. The T-Patch provides an accessible and friendly way for using the T-Stick, while offering a shared software solution for various music-making scenarios.

Author Keywords

T-Stick, Max, software, DMI, education

CCS Concepts

•**Applied computing** → **Sound and music computing;** Performing arts; •**Information systems** → *Music retrieval;*

1. INTRODUCTION

Digital Musical Instruments (DMIs) [9] have the potential to revolutionize the way we create and perform music, but the longevity of these instruments is often uncertain. The lack of performers and technical issues can make these instruments disappear quickly. Ferguson and Wanderley claim that "The current scene is peppered with unique and fascinating digital instruments with a performer base of one" [2]. McPherson and Kim have identified the problem of the "second performer," in which a new instrument becomes extinct as its inventor/designer stops using it [10]. Adnan and Juan suggest that the prolonged use of a DMI can be achieved through musical pedagogy that focuses on both the development of the instrument/interface and its use in musical performance [7]. Fukuda et al. argue that the existence of compositions that are idiomatic to a new instrument is crucial in achieving its long-term usage [3]. These statements highlight that the generation of repertoires through widespread usage of the new instrument is critical in achieving instrument longevity. An important part of the issue is the substantial barrier faced by one willing to start using DMIs, as formalized by Wessel and Wright when mentioning the need for a "low entry fee with no ceiling in virtuosity", stated another way: "Getting started with a computer-based instrument should be relatively easy but this early stage ease-of-use should not stand in the way of the continued development of musical expressivity" [12].

To address these challenges, the T-Stick Music Creation Project (TMCP) was launched. The project includes workshops, mentorship, and technical support to help participating composers generate new compositions for the sopranino T-Stick DMI—a cylindrical musical interface that includes a 9-DoF inertial measurement unit (IMU), Force-Sensing Resistors (FSR), capacitive sensors and a tact push button, shown in Figure 1. While TMCP has succeeded in generating five new works for the T-Stick, it has also revealed the need for a dedicated software application that allows composers to easily try various playing techniques to generate sounds [3].

Figure 1: Sopranino T-Sticks.

The T-Patch is an educational software application that addresses the problem of the lack of a software application. Providing non-technical users with functionalities for receiving OSC data from the T-Stick, gesture extraction, mapping, signal conditioning, sound synthesis, sequencing with cues, and GUI, this application aims to lower the entrance effort for using T-Sticks and welcomes novice users such as composers who have minimal computer programming skills.

The paper is organized as follows: first, we present our motivation, followed by a description of the design and implementation of the T-Patch. Second, we introduce our

NIME'23, 31 May–2 June, 2023, Mexico City, Mexico.

early observations on user experience, and our findings and implications. Finally, we discuss limitations and future work.

2. MOTIVATION

The T-Stick is a family of gestural controller interfaces that utilizes various sensors [6]. These interfaces allow musicians and composers to control sound and music generation through physical gestures, making them a powerful tool for creative expression. However, despite a robust communal effort to ease the use of this DMI such as embedding gesture extraction algorithms into its firmware [8], there is still an obstacle for users who have minimum technological expertise in programming: the lack of dedicated software application allowing for immediate use of the interface. The conventional T-Stick package includes a simple diagnostic Pure Data patch that displays the data from the T-Stick interface without any sound generation capabilities.

This limitation presents a challenge for composers who are not proficient in programming, as there is a lack of a universal and shared software implementation that allows composers to try playing techniques to generate sounds. In the past, composers such as Stewart [11] have addressed this issue by developing patches customized specifically for their compositions. However, this approach is not feasible for composers with limited programming skills. To fully leverage the creativity of composers with minimum technological skills in an educational context, and to overcome this obstacle, we have developed the T-Patch application. Its goal is to make the T-Stick more accessible and user-friendly for composers and musicians of all skill levels, and to contribute to the longevity and widespread use of the T-Stick.

3. DESIGN

The design of the T-Patch strikes a balance between flexibility and usability for composers with limited technical expertise. We acknowledge that high flexibility in software can often lead to a complex design and a steep learning curve for users. To address this challenge, we have developed a GUI-based software for T-Stick. Users can simply set parameters to achieve their desired output without the need for programming, allowing for ease of use and minimal effort to get started.

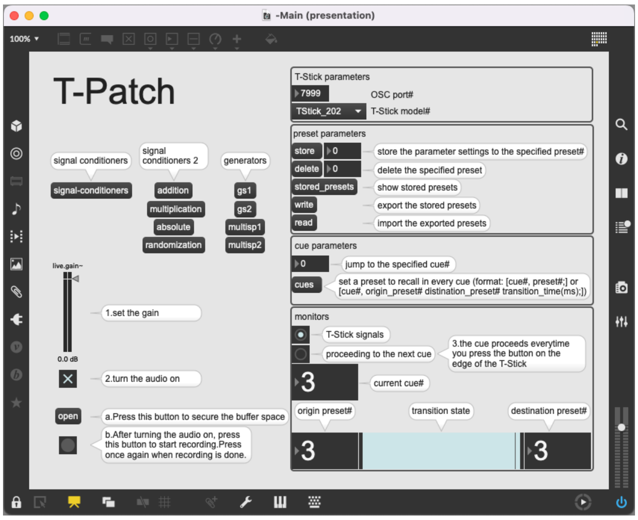

Figure 2: GUI of the T-Patch main window

The T-Patch offers a comprehensive set of functionalities that allow users to easily create and perform music with the T-Stick. These functionalities include OSC receiver, gesture extraction, signal conditioning, mapping, sound synthesis, and sequencing.

3.1 OSC receiver

The OSC receiver functionality receives and processes OSC data [13] from the T-Stick. The T-Stick sends sensor and gesture data in OSC format, which the OSC receiver converts into usable data. The receiver's GUI allows users to set the T-Stick model number and port number.

Figure 3: GUI for the OSC receiver

3.2 Improved Gesture extraction

The gesture extraction functionality detects the "jab" gesture, which is a common and important gesture for composers such as [11]. The current T-Stick firmware already detects several gestures such as "touch" "rub" "brush" and "shake" from the raw sensor data, however, the "jab" gesture detection algorithm has not yet been embedded. By including this feature in the T-Patch, composers can take full advantage of the T-Stick's capabilities. No GUI is provided for this functionality since there exists no changeable parameter.

3.3 Signal conditioning

The T-Patch includes a feature we call signal conditioning, which allows users to modify sensor and gesture data to create musical information. This includes modules such as smooth, function, scale, randomization, and arithmetic operations like addition, multiplication, absolute, and modulo. These conditioners are modular, allowing users to combine and chain them in any order for maximum creative control. They provide users with a wide range of options to modify data.

In addition, the signal conditioning functionality includes the use of 32 bus channels (see Section 3.4), which allows users to chain multiple conditioners in both serial and parallel ways. This added flexibility allows users to tailor the T-Stick to their specific needs. For example, a user could create a chain of smooth, function, scale, and randomization conditioners, or a chain of smooth, function, scale, addition, multiplication, and absolute conditioners to modify the data in various ways.

In contrast to the Digital Orchestra Toolbox [4], which provides a set of Max abstractions for signal conditioning, the T-Patch integrates the signal conditioning functions into its software, eliminating the need for programming skills.

Signal conditioners like the addition and multiplication conditioners are essential for creating many-to-one mappings. These conditioners take in two input sources, for instance, one serving as the multiplier and the other as the multiplicand in a multiplication conditioner, making them particularly useful in scenarios where multiple signals converge in a many-to-one mapping (as depicted in Figure 4).

3.4 Mapping

The mapping functionality enables users to try different mapping types, such as one-to-one, one-to-many, many-to-

Figure 4: The figure illustrates the GUI for the multiplication signal conditioner.

one, and many-to-many mapping. The software provides 32 bus channels that allow users to transfer data between the different signal conditioners (see Section 3.3) in a desired order. This flexibility allows for a wide range of creative possibilities and allows users to tailor the T-Stick to their specific needs. For even more advanced mappings, libmapper can be utilized [5].

3.5 Sound synthesis

The sound synthesis functionality of the T-Patch allows users to generate a wide range of sounds with the T-Stick. The software includes 2 granular synthesis modules and 2 sample playback modules, each with a variety of customizable parameters such as on/off, grain rate, pitch transposition, onset time, grain size, loudness, pan, window type, and sound file import. These parameters can be set directly using the preset layer in the sequencing functionality (introduced in 3.6), or they can be mapped to sensor or gesture data, allowing for dynamic changes in sound based on sensor input and gesture detection. The granular synthesis modules and the sample playback modules allow users to import their own sound samples and manipulate them in real time, offering a wide range of possibilities for sound generation.

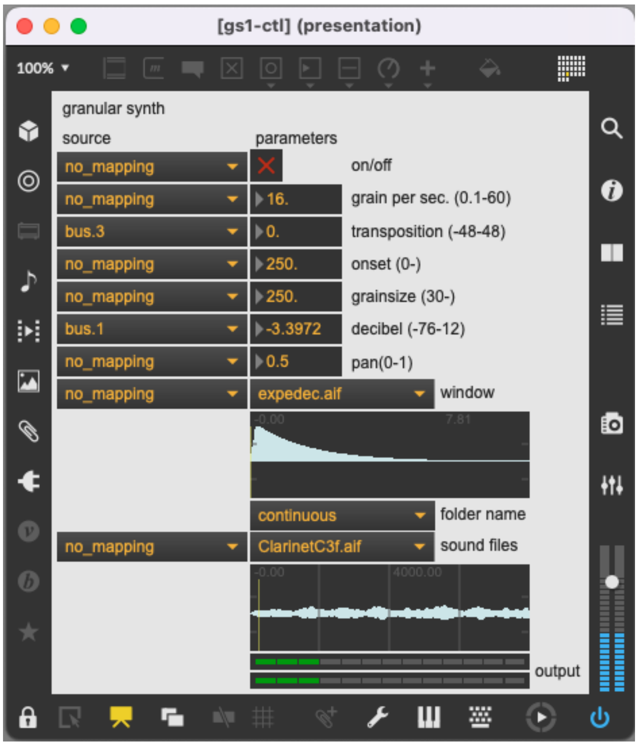

Figure 5: The figure illustrates the GUI for a granular synthesis module. The GUI includes parameters for mapping input sources and parameters for synthesis. These settings can be saved as presets and recalled later, but will be overwritten if the module is receiving input values from a specified channel.

3.6 Sequencing

The sequencing functionality allows users to create dynamic mappings and create music whose response to their gestures changes over time. It consists of two layers: the cue layer (figure 6) and the preset layer (figure 7). The cue layer allows the user to specify which presets should be recalled at a certain time, enabling transitions between presets. The preset layer stores the settings of various parameters, such as mapping, signal conditioning (e.g., multiplier value), and sound synthesis (e.g., transposition and onset values), allowing users to flexibly design a sequence of musical events.

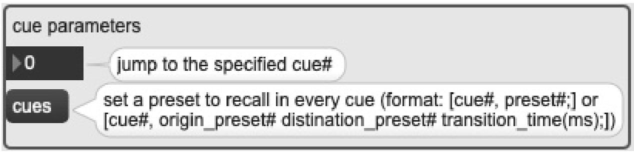

Figure 6: GUI for the cue layer in the sequencing functionality

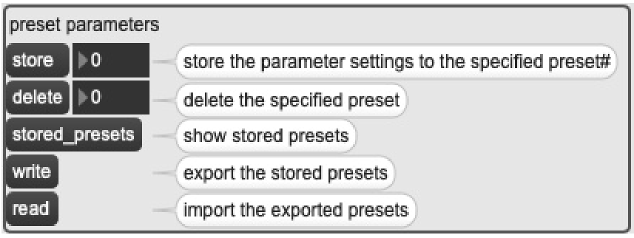

Figure 7: GUI for the preset layer in the sequencing functionality

3.7 Scaled Values

To ensure flexibility in mapping and signal conditioning, all sensor and gesture data values are scaled to a range between 0 and 1. This allows for interchangeable use of different gesture types within the T-Patch and enables users to easily switch between different gestures.

3.8 Implementation

The T-Patch was implemented using the Max visual programming environment, which is a commonly used tool in the digital music field. To simplify the user experience and ensure compatibility, the software does not require any external dependencies such as libraries or Max objects. The package also includes a launcher that automatically sets the file path, making it easy for users to import the T-Patch abstractions. T-Patch was optimized for both Windows and Mac and works with the sopranino T-Sticks.

The T-Patch's modular design makes it simple for instrument designers to adapt it to other interfaces such as the Karlax[1] and the eigenharp[2]. For instance, to use the Karlax with the T-Patch, the designer only needs to modify the OSC namespace and the setting of the scale Max object, converting the Karlax sensor values (0 to 127) to the T-Patch's required scaled value range (0 to 1) as mentioned in Section 3.7. This adaptability of the T-Patch increases productivity for DMI designers.

[1]http://www.dafact.com/
[2]http://www.eigenlabs.com/

4. TUTORIAL VIDEOS

The T-Patch comes with accompanying tutorial videos that guide users through the setup and operation of the T-Stick, as well as the creation of custom mappings between sensor/gesture data and sound parameters. These tutorials aim to provide a quick and easy way for users to familiarize themselves with the T-Patch application and start creating music with the T-Stick.

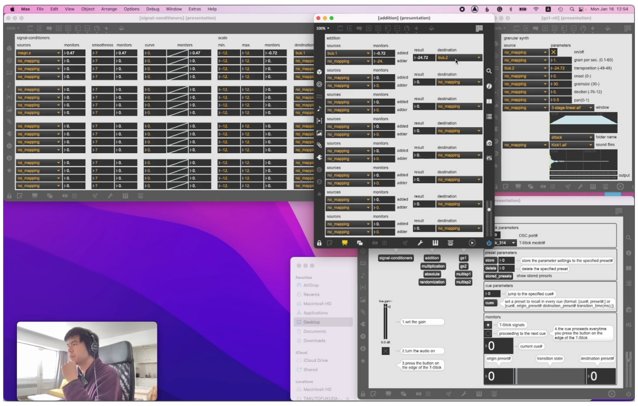

Figure 8: The figure depicts the tutorial video.

5. EARLY OBSERVATIONS ON USER EXPERIENCE

In a series of weekly workshops, three students with no programming skills and no prior experience with digital musical instruments were introduced to the T-Patch and the T-Stick. They were given the option to compose using either the T-Stick or the Karlax DMI. At the time of writing this paper, the workshops have only been held five times and the project is still in its early stages.

The T-Patch software application was developed and introduced to composers with an elementary level of technical expertise during a series of weekly workshops. The early observation implies that, although its impact on music creation is still an open question, the application significantly reduces the entrance effort for testing the T-Stick.

Along with the in-person introduction to the software, video tutorials were provided to explain how to set up the T-Stick, produce sound, and map gestures to synthetic parameters. Initial observations showed that the students were able to successfully set up and play the T-Stick and create their desired mappings.

6. DISCUSSIONS

However, the results of the workshop series indicated that the three students favored the Karlax over the T-Stick for their compositions. This choice was driven by the requirement that their compositions be performed by other musicians, who preferred the form and mechanism such as the keys of the Karlax, and the saxophonist and guitarist they asked to perform for them were more familiar with it. Additionally, one student reported difficulties in producing the desired sound on the T-Stick due to its unique shape and sensor configuration, which differs from traditional acoustic instruments.

The preference for the Karlax highlights the need for a performance demo video of the T-Stick. The lack of dedicated T-Stick performers and demonstrations showcasing the instrument's capabilities in actual compositions made it challenging for the students to envision how to compose for the T-Stick. While the video tutorial that demonstrated playing techniques and sounds helped to lower the barrier of entry for new users, there is a need for videos that illustrate the T-Stick's potential in compositional contexts, especially for an instrument that requires a different performance practice from existing acoustic instruments. This insight resonates with Perry's Principle 5, which suggests to prioritize creating a music composition over building an instrument or controller [1]. In retrospect, showcasing existing compositions for the T-Stick at the start of the workshop series would have given the composers a better understanding of how to perform the instrument.

7. LIMITATIONS

The T-Patch project is not without limitations. Firstly, to use the T-Patch effectively, a certain amount of learning is required. The developers have attempted to minimize the learning curve for users by providing in-person workshops and instructional videos. However, users still need to invest time and effort in learning the software. Secondly, the software is implemented in the Max environment, which may cause compatibility issues if the Max version is updated. To address this limitation, the development team is exploring options to compile the T-Patch as a standalone application to ensure compatibility and minimize disruptions.

In terms of functionality, the T-Patch is specifically designed for the sopranino T-Stick and is not compatible with other T-Stick variations such as soprano, alto, and tenor. Additionally, the T-Patch currently only offers granular synthesis and sample playback modules for sound synthesis, limiting the creative options for users who might want to experiment with other synthesis techniques. Furthermore, the T-Patch only supports preset recall through button presses and does not currently allow for changing presets using gestures.

Lastly, the T-Patch package, as an educational tool, lacked a video that demonstrated the instrument's potential in a compositional setting. This was noted as a shortfall after the workshop series, where the three composers ultimately chose to compose using the Karlax instead of the T-Stick. The lack of a video showcasing the compositional possibilities of the T-Stick may have contributed to the composers' decision. To address this, we now recognize the importance of including a video that demonstrates the T-Stick's compositional capabilities.

8. FUTURE WORK

Future work aims to evaluate the accessibility of the T-Stick with the T-Patch application. Three factors will be investigated: (1)the motivations for composers choosing the Karlax over the T-Stick, (2)the extent to which T-Patch has reduced the entrance effort for using the T-Stick and (3)the impact of T-Patch on the users' creativity. To identify the composers' motivations, the project envisions an interview with them after the end of the workshop series. To measure the users' experience, the project will conduct questionnaires to gather feedback on the ease of use of T-Patch and T-Stick. To understand the limitations of T-Patch in terms of sound creation, the project will compare the versatility of gesture-sound mappings between a group using T-Patch and another group creating their own patch. Additionally, a comparative study with lib mapper-compatible mapping interfaces is within the scope. Combined with qualitative research analysis, these approaches will facilitate to reducing the researchers' biases.

T-Patch will be expanded to meet the needs of composers. More functions will be added to enhance the signal conditioning functionality, such as a quantizer that can convert sensor values into pitches on musical scales. The sound synthesis capabilities will be expanded by incorporating more synthesis techniques, allowing for a wider range of sounds and textures. Audio effects, such as filters and reverberation, will also be added. The possibility of making the T-Patch compatible with other T-Stick models, such as soprano and tenor, will also be explored.

Support and workshops for T-Patch users will continue to be provided, along with improved documentation and tutorials. Redesign of the GUI from the spreadsheet paradigm will be considered to make the interface design more intuitive and enhance the learnability of the software with or without our guidance. A data-logging feature will be included to allow users to record and save their performances. Demonstration videos will be created to illustrate the mapping between playing techniques and sounds, providing performers and composers with a comprehensive understanding of the instrument's capabilities and potential.

The goal is to continue to improve the T-Patch and make the T-Stick more accessible to a wider range of users. The project aims to provide a flexible software application that addresses the lack of music-making support in the existing T-Stick package and inspire new users.

9. CONCLUSION

The T-Patch addresses the lack of dedicated software for the sopranino T-Stick by offering a user-friendly GUI and programming-free controls. This allows users to easily adjust and fine-tune the T-Stick to achieve their desired sound and expression, and opens up new creative possibilities for composers. The signal conditioning functionality, with modules for smooth, function, scale, randomization, and arithmetic operations, provides users with a wide range of options to modify sensor and gesture data in ways that are idiomatic to the T-Stick. The use of 32 bus channels for chaining multiple conditioning functions in both serial and parallel ways enhances the expressiveness of the instrument and allows users to tailor it to their specific needs.

However, our early observations of the users indicate the need for demonstration content, such as a video, to showcase the instrument's possibilities in a compositional context and inspire new users. This is especially important for performers who are accustomed to playing acoustic instruments, as there are currently few dedicated T-Stick performers. To make the T-Stick more accessible and versatile for musicians, we recommend that educational packages for DMIs include hardware, software, tutorial, and demo video content. We hope that the T-Patch will contribute to the longevity and widespread use of the T-Stick, making it a valuable tool for composers and musicians alike.

10. ACKNOWLEDGMENTS

This research is supported by Input Devices and Music Interaction Laboratory (IDMIL), The Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT), and McGill University.

11. ETHICAL STANDARDS

There is no funding source for this research and are no observed conflicts of interest. Composers' comments on why they chose to compose for the Karlax rather than the T-Stick were stated in a voluntary manner during the workshops in their knowledge that the feedback might be used to improve our research.

12. REFERENCES

[1] P. R. Cook. Principles for designing computer music controllers. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 3–6, Seattle, WA, 2001.

[2] S. Ferguson and M. M. Wanderley. The McGill Digital Orchestra: An Interdisciplinary Project on Digital Musical Instruments. *Journal of Interdisciplinary Music Studies*, 4(2):17–35, 2010.

[3] T. Fukuda, E. Meneses, T. West, and M. M. Wanderley. The T-Stick Music Creation Project: An approach to building a creative community around a DMI. In *NIME 2021*, apr 29 2021. https://nime.pubpub.org/pub/7c4qdj4u.

[4] J. Malloch, M. M. Schumacher, S. Sinclair, and M. Wanderley. The digital orchestra toolbox for max. In T. M. Luke Dahl, Douglas Bowman, editor, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 255–258, Blacksburg, Virginia, USA, June 2018. Virginia Tech.

[5] J. Malloch, S. Sinclair, and M. M. Wanderley. Libmapper: (a library for connecting things). In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 3087–3090, New York, NY, USA, 2013. Association for Computing Machinery.

[6] J. Malloch and M. M. Wanderley. The T-Stick : From Musical Interface to Musical Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 66–69, New York City, NY, United States, 2007.

[7] A. Marquez-Borbon and J. P. Martinez-Avila. The Problem of DMI Adoption and Longevity: Envisioning a NIME Performance Pedagogy. In T. M. Luke Dahl, Douglas Bowman, editor, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 190–195, Blacksburg, Virginia, USA, June 2018. Virginia Tech.

[8] E. A. L. Meneses, T. Fukuda, and M. M. Wanderley. Expanding and embedding a high-level gesture vocabulary for digital and augmented musical instruments. In S. Yamamoto and H. Mori, editors, *Human Interface and the Management of Information. Interacting with Information*, pages 375–384, Cham, 2020. Springer International Publishing.

[9] E. R. Miranda and M. M. Wanderley. *New Digital Musical Instruments: Control and Interaction beyond the Keyboard*. A-R Editions, 2006.

[10] A. P. McPherson and Y. E. Kim. The Problem of the Second Performer: Building a Community Around an Augmented Piano. *Computer Music Journal*, 36(4):10–27, 12 2012.

[11] D. Stewart. *Catching Air and the Superman*. PhD thesis, McGill University, 01 2009.

[12] D. Wessel and M. Wright. Problems and Prospects for Intimate Musical Control of Computers. In *Proceedings of the CHI'01 Workshop on New Interfaces for Musical Expression (NIME-01)*, pages 11–14, 2001.

[13] M. Wright, A. Freed, et al. Open soundcontrol: A new protocol for communicating with sound synthesizers. In *ICMC*, 1997.