HarMIDI: Sensor System To Read MIDI from Indian Harmoniums

Suraj Jaiswal
Indian Institute of Technology Kanpur
India
jsuraj@iitk.ac.in

Vipul Arora
Indian Institute of Technology Kanpur
India
vipular@iitk.ac.in

ABSTRACT

While digital music technologies are rapidly growing, music communities using traditional acoustic instruments are sometimes unable to take full advantage of all of the digital processing techniques available to electronic musicians. One way to include them in the latest developments is to develop interfaces connecting non-electronic instruments to the digital world. This paper presents HarMIDI, a sensor system to convert keystrokes on an Indian harmonium to MIDI. The paper presents the sensor assembly, calibration methods, and the algorithm to output MIDI. The calibration methods calibrate the notes and temporally synchronize the MIDI stream with audio. The system has been evaluated for time synchronization of onsets and offsets. The sensor setup is affordable, portable and can be used with any existing harmonium.

Author Keywords

Harmonium, MIDI, Indian Classical Music, Sensors

CCS Concepts

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

1. INTRODUCTION

The technology that is utilised in a wide variety of musical instruments has recently undergone significant development. There are many different electronic instruments that can interface with the most recent softwares, and a great deal of music synthesis can be accomplished by using those instruments. MIDI is a standard protocol that connects a wide variety of electronic musical instruments to computers and other audio devices for the purpose of performing a variety of tasks, including playing, editing, recording, and synthesising. Moreover, state of the art machine learning based algorithms need large data for training. This need can be fulfilled by the MIDI data gathered from electronic music instruments, along with the audio data.

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

Indian musical instruments are utilised extensively in a wide variety of musical styles. However, despite the significant progress in digital music, Indian music is somewhat marginalized due to lack of digitization and the inability of Indian instruments to interface with any other electronic devices. Harmonium [1], which is considered to be an instrument of Indian classical music, is utilised frequently in Indian music, such as in Hindustani, Sikh music and folk music. It is necessary to have a system that can read the MIDI information of the playing of the harmonium in order to increase its capabilities to interact with digital music world. This will not only assist in getting data from the harmonium for computational analyses, but it will also make it possible for the harmonium to interact with other digital instruments and software without altering its original traditional method of playing or its design. This system has applications in a variety of areas, including music composition, music pedagogy, and music performance.

In this paper, we present HarMIDI, which is a sensor-based system that reads the key press gestures from a harmonium and converts it into MIDI.

2. RELATED WORK

A variety of sensor systems have been developed by researchers to digitize the analog instruments in order to facilitate their interaction with digital electronics dedicated to music. McPherson [9] has developed a portable sensor system that can sense the motion of keys on a piano and map those movements to MIDI. For percussion instruments, Sokolovskis *et al.*[13] uses optical sensors for localization of the strike on an acoustic drumhead.

There have been very few works on using sensors to digitize harmonium playing. A robotic mechanism, named Kritaanjali, has been developed for playing a harmonium [7, 10]. It has been used for musical robotics. Another piece of work that was developed by Guin *et al.* [4] demonstrates a crank-rocker linkage system with a dc motor that is used to automate the bellow mechanism of the harmonium. This makes it easier for beginners to learn how to play the instrument.

Apart from designing for traditional analog musical instruments, designing their digital equivalents is also common. Yunik *et al.* [17] developed a microprocessor-based digital flute. Pardue *et al.* [12] have developed a digital tabla that is capable of producing sounds comparable to that of a traditional tabla by capturing multiple gestures with the use of sensors. In classical Indian music, the digital version of drone instrument, *tanpura*, has become more popular than its traditional analogue. It is well accepted by musicians even for live concerts. There are digital versions of a large number of traditional Indian instruments available commercially . A range of digital harmoniums developed by

Radel Electronics [3] are available where a complete digital harmonium is made named Maadhurium and its newer version Maadhurium+.

When the instruments are digitised, there are many applications and uses for them; one of these areas is education. This may be seen by looking at the work that was done by Pardue [11], which involved the digitization of a violin, which is used to teach others how to play the violin. In the context of piano teaching and learning, many specialised apps using augmented reality and artificial intelligence technologies have already been developed, and are found to be effective [8, 2].

While developing sensor based systems to digitise musical instruments one prominent issue is latency. The effect of latency on the performances can be observed in the work done by Jack *et al.* [5]. It is essential to ensure that the action-to-sound latency is either low or stable. It is recommended by Wessel and Wright [16] that digital musical instruments should have a latency less than 10ms. In order for a hardware-based system to be widely accessible to the majority of users, it is crucial that the system's price be reasonable. In the work of Vieira *et al.* [14], a low-cost MIDI controller was created, despite the fact that there are already numerous MIDI controllers available in the market.

3. METHODOLOGY

The system is made up of two basic components: the first is a sensor-based system that is used to read the key press gestures from the harmonium, and the second is algorithms that are used to process the sensor data and turn it into MIDI. There are also a few different calibration approaches that are used in conjunction with the algorithm in order to calibrate the sensors.

3.1 System Overview

HarMIDI consists of a array of sensors that are placed over the *kamani* (spring) of the harmonium. These sensors read the key presses on the harmonium. Serial communication is used to send the sensor data to a computing device through a microcontroller. The data so gathered is processed and is converted into a MIDI file. For the purpose of calibration and validation of the system, the audio generated by harmonium can also be recorded by a microphone in the computing device.

3.2 Hardware Description

The hardware mainly consists of an array of sensors and a microcontroller. The sensors used in this work are TCRT5000 Infrared sensors which measure the distance between an object and the sensor. The sensors are attached to Arduino Uno, an 8-bit microcontroller which transfers the analog signal from the sensor to the computing device using the serial communication. The hardware is designed so that it can easily be installed over the harmonium without any hassle. It integrates seamlessly without altering the traditional way of playing the harmonium. Moreover, the setup is portable and works with existing harmoniums in a non-invasive fashion, without a need to re-design the harmonium. The setup of the hardware can be seen in the Fig. 2.

3.3 Calibration

Before we can use the harmonium to create a MIDI file, the sensors that have been installed on it will need to have their calibration adjusted. The IR sensor estimating the position of the harmonium keys, gives a continuous reading. This reading depends on different factors, such as, the height of sensor placement, the distance between two sensors (inter-sensor interference), and the individual sensor characteristics. Calibration is needed standardize the output invariant to these factors. We call it the amplitude calibration. It sets a threshold θ to distinguish the ON and OFF states of a key.

Another calibration that is necessary is to determine which note (*svara*) is pressed. The identification of the note N at which the sensor is placed is accomplished through frequency calibration. Yet another calibration needed is that of the time delay $\hat{\tau}$ between the sensor signal and the audio signal. This is necessary for synchronizing the audio recordings of the harmonium with the MIDI file generated by the proposed system. We also use it for experimental evaluations.

The amplitude, frequency and time-delay calibrations give threshold (θ), note value (N), and time-delay ($\hat{\tau}$), respectively. These calibration methods are detailed below. The overview of calibration method can be seen in the Fig. 3.

3.3.1 Time-Delay Calibration

The time-delay between the recorded audio and the sensor data is computed using cross-correlation. For two discrete-time signals, $x_1[t]$ and $x_2[t]$, the cross-correlation is computed as

$$R_{x_1,x_2}[\tau] = \sum_{t=0}^{N-1} x_1[t]x_2[t+\tau] \tag{1}$$

The time delay between the two discrete-time signals is estimated as

$$\hat{\tau} = \arg\max_{\tau} R_{x_1,x_2}[\tau] \tag{2}$$

Note that the audio signal lags behind the sensor signal by 40-50 ms. This delay is attributed to the microphone setup used.

3.3.2 Amplitude Calibration

We use two different methods for amplitude calibration as explained below.

Method 1.

The sensor is positioned so that it is above the *kamani* of the harmonium key. Initially, the user is asked not to press the key, and the reading from the sensor reading is recorded. Next, the user is asked to press the key, and the sensor reading is again recorded. These readings are then saved on the computing device as $y_{\min}$ and $y_{\max}$, corresponding to the unpressed and pressed states, respectively. The amplitude threshold is then computed as

$$\theta_1 = \frac{y_{\min} + y_{\max}}{2} \tag{3}$$

Method 2.

It may be more robust to define the thresholds in terms of the audio recorded from the harmonium. For this, we synchronize the audio and the sensor readings using the delay $\hat{\tau}$ estimated between the two.

A user is asked to unpress and press the key to get $a_{\min}$ and $a_{\max}$, respectively. A hyperparameter β is defined to determine the ON and OFF state of audio. The switching time t^* is computed as

$$t^* = \arg\min_{t} |a[t] - (\beta a_{max} + (1-\beta)a_{min})| \tag{4}$$

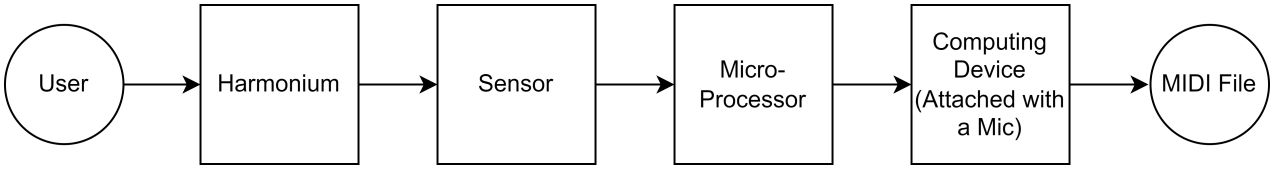

Figure 1: System Overview

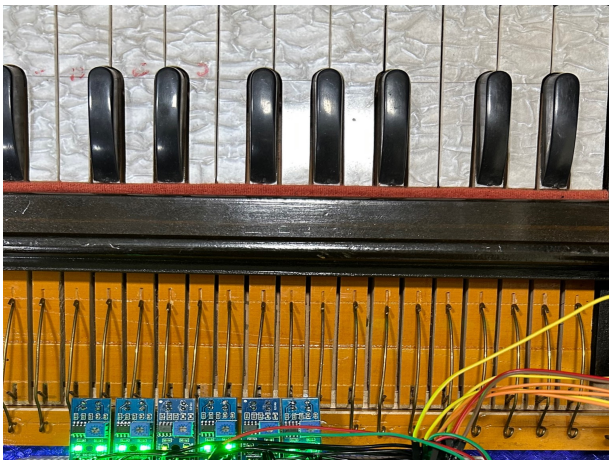

Figure 2: Hardware Setup

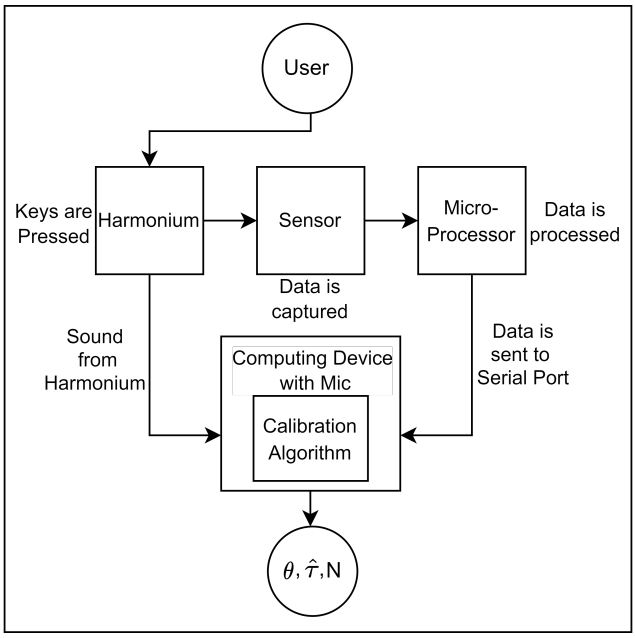

Figure 3: Calibration Method Overview

Using t^*, the threshold in the sensor signal is estimated as

$$\theta_2 = \frac{y[t^*] - y_{min}}{y_{max} - y_{min}} \tag{5}$$

Fig. 4 illustrates the procedure by plotting the amplitudes of the synchronized sensor and audio signals.

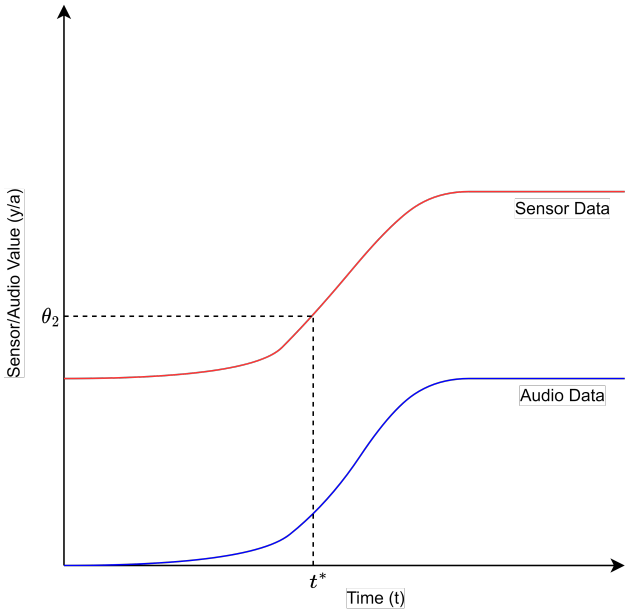

Figure 4: Calibration Method 2

3.3.3 Frequency Calibration

The note corresponding to the key, where the sensor is placed, is done by the audio recorded when the user presses the key. Pitch is computed from the recorded audio using the parselmouth python library [6]. The pitch value is then use to determine the MIDI note number N.

Calibration is only required once for the same instrument and set of keys. Sensor re-calibration may be required due to sensor drift over time or due to sensor positioning errors. The sensor and audio reading can be used to calculate when the system should be re-calibrated.

3.4 Reading and Processing Sensor & Audio Data

The analog readings captured by the sensor are sent to the microcontroller. The microcontroller then transfers the data to the serial port of the computing device using serial communication. The baud rate of the above serial communication is 9600 bits per second. Timestamps are also recorded along with the sensor values. The recorded data is then saved on the computing device.

To write the MIDI file, the analogue readings from the sensor are converted into digital ON/OFF values, with the

help of the amplitude threshold (θ_1 or θ_2). The MIDI number comes from the note value N associated with the pressed key. The timestamps are changed to the absolute duration of the key pressed.

In parallel to this the audio data is recorded by the computing device at a sampling frequency of 44.1 KHz. This audio data is recorded only to evaluate the performance of the system or for calibration. To optimize the performance and reduce the latency in reading the data of the sensor, the recording of the audio and the reading of the sensor data are done in parallel using multi threading. Without parallelization, the system misses the data received from the microcontroller.

The volume of MIDI note is set to a constant value in the current implementation. The air pressure inside the bellows of the harmonium could be used to set the volume but requires installing the sensor inside the harmonium through an invasive process. One may also use audio amplitude to set the MIDI note volume.

4. EXPERIMENTS

A prototype of HarMIDI, consisting of an array of six sensors, is experimentally evaluated to compare its performance with conventional MIDI keyboards.

The first experiment assesses the ability of the system to synthesize music in real time. We use python-rtmidi package to synthesize sound from the MIDI signal generated due to the key-presses. We find that using multi-thread algorithm, with a thread dedicated to each key, optimizes the performance as compared to using a single thread to handle all the keys. Using a single thread increases the latency of sound synthesis and reduces the fidelity of sound generated, i.e., sound synthesized may not be exactly same as the keys played. Using multi-threads ameliorates both these issues. Incidentally, the problem of fidelity is not observed in case of MIDI reading for offline processing. This could be due to lower computational needs there as compared to those for synthesizing sounds.

Another experiment aims at assessing the ability of the HarMIDI system to connect with existing MIDI softwares, which generate and receive MIDI events. We use VMPK (Virtual MIDI Piano Keyboard) [15] for this experiment. The MIDI signal generated by the multi-thread algorithm is fed to a virtual MIDI port. The same is then redirected to VMPK which synthesies the sound. The latency observed in the sound synthesized with VMPK is slightly more than that observed with python-rtmidi, but the fidelity is identical. This indicates that, in principle, any digital audio workstation software capable of reading virtual MIDI port can be connected with HarMIDI.

5. EVALUATION AND DISCUSSION

The system is evaluated by comparing the note onsets and offsets in the MIDI with those in the audio recording. Figs. 5-6 plot the audio and the MIDI signals together without and with time delay calibration, respectively. The onset and offset times of the two signals match well when the calibrated $\hat{\tau}$ is used. Even without calibrations, their note durations match well.

We also evaluated the system with subjective tests with musicians, who played on the harmonium having the HarMIDI system. They listened to their audio recordings and the audio synthesized from the MIDI data, and confirmed the validity of the system output.

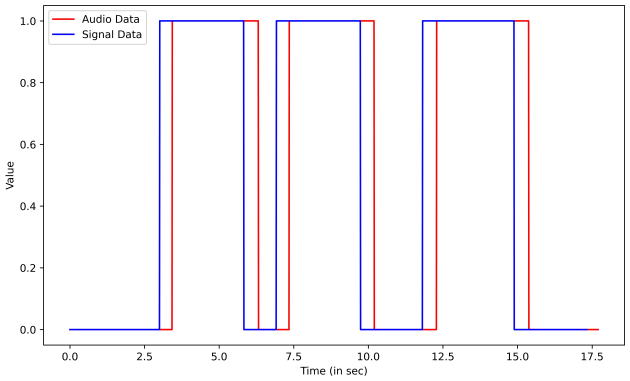

Figure 5: Plot without Time-Delay Calibration

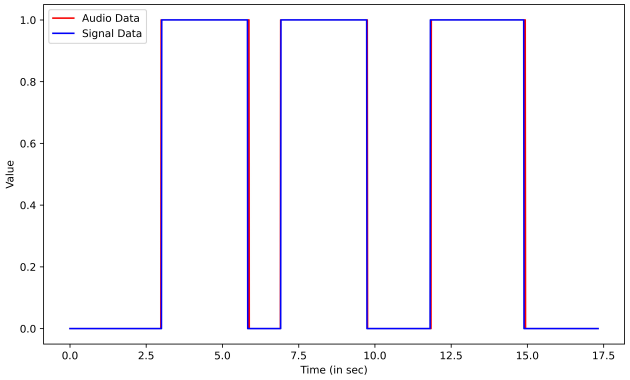

Figure 6: Plot with Time-Delay Calibration

5.1 Cost Analysis

When it comes to its development, HarMIDI is quite frugal. The components used to build the prototype cost approximately US$ 37 only. This prototype serves 32 keys on the harmonium. The breakup of the cost can be seen in Table 1. This cost is expected to further come down as production scales up. On the other hand, a commercial MIDI keyboard with 32 keys costs approximately US$ 95. The price range for a fully functional digital harmonium is from US$ 342 to US$ 440 [3].

6. APPLICATIONS & FUTURE WORK

Composing and synthesising music are two areas that benefit greatly from the proposed system. The traditional harmonium users can connect their harmoniums to popular MIDI interfaces. The proposed system is also useful for harmonium education, where computer assisted tools for teaching and learning the harmonium could be built. Our current efforts are on making the system scale up to cover all of the keys of the harmonium and also to digitalize the expression of hand pumping of the harmonium. We are also working towards optimising the system for real-time control of MIDI interfaces using harmonium.

Table 1: Cost Breakup

Component	Cost (US$)
Arduino Mega	15
Sensors	12
Miscellaneous	10
Total	37

7. CONCLUSIONS

HarMIDI is a sensor system that can be simply added to a harmonium that converts the sound produced by the instrument into MIDI. It is portable and has a quick installation process. In terms of the development, this is a frugal approach. It will bring in an extremely sizable community of harmonium players into the realm of digital MIDI. The findings demonstrate that the performance of the system is reliable, and that the MIDI signal obtained from the harmonium through the proposed system shows fidelity to the audio signal recorded from the same.

8. ETHICS STATEMENT

This work was financially supported by a grant from the Prasar Bharati. The system was used by trained music specialists, who provided informed consent. There are no conflicts of interest.

9. REFERENCES

[1] V. Brinda and M. Bhageerathi. A study on design and development of harmonium and its types. *International journal of economic perspectives*, 16(5):151–159, 2022.

[2] K. Cui. Artificial intelligence and creativity: piano teaching with augmented reality applications. *Interactive Learning Environments*, pages 1–12, 2022.

[3] R. Electronics. Digital Harmonium Archives - Radel India - Digital Indian Musical Instruments. https://www.radel.in/product-category/digital-harmonium/. Accessed online on 2023-01-30.

[4] B. Guin, K. Ghar, and N. Modak. Design and development of an indian harmonium with automatic bellow mechanism. In *Journal of Physics: Conference Series*, volume 1950, page 012021. IOP Publishing, 2021.

[5] R. H. Jack, T. Stockman, and A. McPherson. Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument. In *Proceedings of the Audio Mostly 2016*, AM '16, page 116–123, New York, NY, USA, 2016. Association for Computing Machinery.

[6] Y. Jadoul, B. Thompson, and B. de Boer. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, 71:1–15, 2018.

[7] A. Kapur, J. W. Murphy, and D. A. Carnegie. Kritaanjali: A robotic harmonium for performance, pedogogy and research. In *NIME*, 2012.

[8] K. Lei. The effectiveness of special apps for online piano lessons. *Interactive Learning Environments*, pages 1–12, 2022.

[9] A. P. McPherson. Portable measurement and mapping of continuous piano gesture. In *NIME*, pages 152–157, 2013.

[10] J. Murphy, A. Kapur, and D. A. Carnegie. Mechatronic keyboard music: Design, evaluation, and use of a new mechatronic harmonium. In *ICMC*, 2014.

[11] L. S. Pardue. *Violin augmentation techniques for learning assistance*. PhD thesis, Queen Mary University of London, 2017.

[12] L. S. Pardue, K. Bhamra, G. England, P. Eddershaw, and D. Menzies. Demystifying tabla through the development of an electronic drum. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 596–599, 2020.

[13] J. Sokolovskis and A. P. McPherson. Optical measurement of acoustic drum strike locations. In *NIME*, pages 70–73, 2014.

[14] R. Vieira and F. Schiavoni. Fliperama: An affordable arduino based midi controller. In *Proceedings of NIME*, volume 2020, 2020.

[15] Virtual MIDI Piano Keyboard. https://vmpk.sourceforge.io/. Accessed online on 2023-02-11.

[16] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer music journal*, 26(3):11–22, 2002.

[17] M. Yunik, M. Borys, and G. Swift. A digital flute. *Computer Music Journal*, 9(2):49–52, 1985.