Acquisition, processing and data analysis of piezoelectric sensors for training musical robots in a didactic model

Alan M. Marotta[1], Emerson S. Costa[1], Erick N. M. Alves[1], Thiago V. A. Abreu[1], Luan C. Marotta [1], Cauan C. Marotta[2]

[1] *Dept. Eng. Mecatrônica, Centro Federal de Educação Tecnológica de Minas Gerais, Rua Alvares de Azevedo, 35503-822 Divinópolis/MG, Brasil*
alanmarotta@cefetmg.br, emerson@cefetmg.br, ericknathancoro@hotmail.com, thiagotb79@gmail.com, luanmarotta@gmail.com
[2] *Dept. Medicina, Universidade Federal de Minas Gerais, Avenida Presidente Antonio Carlos, 31270-901 Belo Horizonte/MG, Brasil*
cauanmarotta@gmail.com

Abstract. This paper introduces a developed system tailored for processing and analyzing data acquired from piezoelectric sensors. The system's objective is to detect rhythmic patterns in percussive music, generating a database to train musical robots. This training enables collaborative interaction among musical robots, thereby nurturing human musical advancement. The proposed approach involves assessing the musician's performance by installing piezoelectric cells affixed to rubber pads corresponding to each instrument. Processing techniques applied to the piezoelectric sensor signals facilitate system implementation through accessible didactic hardware, which is more user-friendly compared to alternatives such as frequency spectrum processing. Analyzing parameters based on rhythmic beat intensity and timing leads to the establishment of a precise database characterizing musician dynamics. The progression from a simplified to a more intricate data model explores intensity and data structure within the database. Testing employed well-known basic rhythms to construct a rhythm repository, demonstrating the system's adeptness in microcontroller-based data processing and analysis. The system showcases benefits like compactness, energy efficiency, and reduced space and weight, favorable for constructing robotic frameworks. The proposed approach supports real-time and embedded system applications, extending a multitude of possibilities and applications within the realm of music and robotics.

Keywords: piezoeletric sensor, data processing and analysing, musicial robots, didatic model

1 Introduction

In the context of training and human interaction with robots, this paper addresses the processing of data acquired from piezoelectric sensors with the purpose of analyzing and identifying musical rhythms in percussion instruments. As examples of notable musical robots, we can mention Uchiyama et al. [1] (saxophone), Wei et al. [2] (ocarina), Marynowsky et al. [3], Zhao et al. [4] (flute) and Hoffman and Weinberg [5] (marimba). We explore the elements comprising a system consisting of the identification stage (capture and processing), recording stage (data structure), and automated reproduction stage.

In addition to enabling the development of interactive musical robots, this work facilitates the replication of the proposed system using didactic and accessible components. The concept revolves around obtaining signals from musical instruments, which in most cases involve musical rhythms generated by percussion instruments. The initial stage of the system thus includes a network of sensors capable of acquiring information that allows for the identification of the moment and intensity at which the instruments are played.

The subsequent discussion delves into the adopted methodology and the components constituting the system for capturing, processing, and learning percussive rhythms, aiming to serve as a foundation for applications involving musical human-*robot* interaction.

2 Methodology

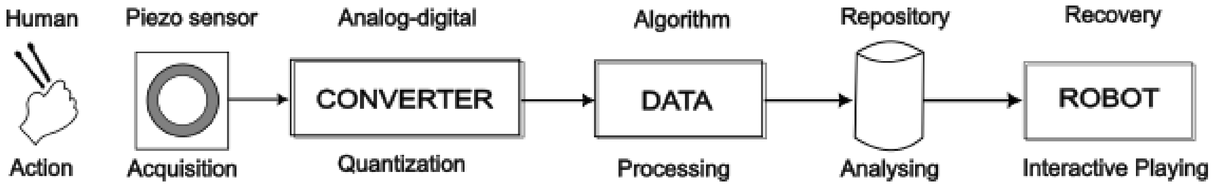


Figure 1. System block diagram

In robot training, a well-known technique is the "teaching and playback", usually incorporated in industrial manipulators, see [6]. Similarly, this work follows the same idea. The proposed system is composed of the stages outlined by the blocks depicted in Figure 1. The block titled ***Acquisition*** features a piezoelectric sensor capable of generating electrical signals based on motion caused by human action, herein referred to as the musician. In this stage, the signals are captured.

Following the ***Acquisition*** block is the ***Quantization*** block, tasked with converting the electrical pulses from the piezoelectric sensor into digital values associated with the intensity of the musician's actions. Subsequent to the ***Quantization*** block, the ***Processing*** block is responsible for identifying the peaks corresponding to the moments of action and their respective intensities. In the ***Analyzing*** block, the peaks identified in the ***Processing*** block result in a vector of instances when each instrument undergoes the musician's performance. The data is stored in a repository as part of the rhythm database, which can be retrieved and serve as a dataset for the subsequent training of the AI of the musician robot.

The final ***Robot*** block pertains to the reproduction of the rhythm, achieved through robotic execution on percussion instruments. Collectively, all the blocks form a complete system for capturing, processing, identifying, storing, and retrieving the rhythms performed by musicians. In the following section, the proposed system is developed, with each individual block being elaborated and illustrated.

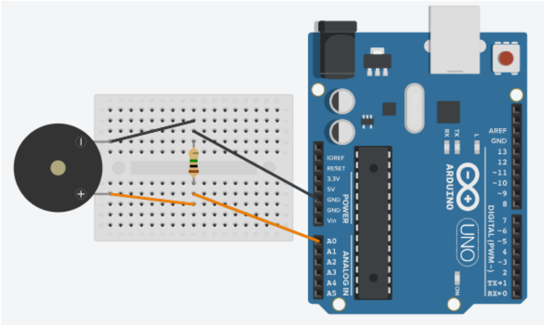
3 Development

In this section, we embark on the development of the blocks illustrated in Figure 1. Commencing the sequence is the initial block denoted as "acquisition," in which the sensor interfaces with the signal quantization block. Following this is the subsequent stage, encompassing analog-to-digital conversion. Subsequently, the phase of data processing ensues, aimed at discerning the salient peaks attributed to the actions of the musician. Moving forward, the fourth block engages in the identification and archival of instances of actions along with their corresponding magnitudes. Concluding the progression, the recovery of patterns takes place—either in digital manifestation or, as is pertinent to our context, in an acoustic manifestation. Further elucidation regarding the aforementioned blocks is expounded upon in the subsequent subsections.

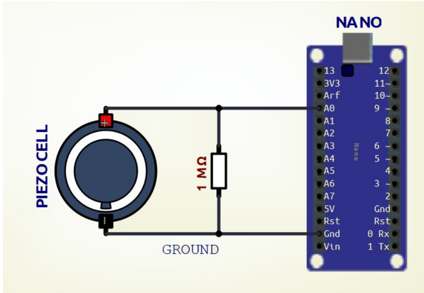
3.1 Action and acquisition

This subsection addresses the initial stage, the capture of the moments and intensities at which the instruments are played. An evident strategy involves the analog-to-digital conversion of the generated sound and its subsequent processing. This procedure serves as an option for static processing purposes. However, in real-time applications, the computational effort is amplified, requiring additional time and thereby constituting a constraining factor for real-time implementations.

The sensing element employed in this context is a piezoelectric cell, for which further details can be found in [7] (pages 337-338) and [8]. The prototype utilized in this study is illustrated in Figure 3. The piezoelectric sensor is affixed using adhesive directly onto the rubber pad (5mm thickness). The additional components comprise a 1 $M\Omega$ resistor connected to the microcontroller (Arduino Uno). Connection can be established via USB cable for data storage on a computer or, as in the prototype, for portable application. For portability, components such as a HC-12 radio module and power supply derived from a lithium battery (with charge controller and on/off switch) were integrated. The overarching aim is to encase the entire prototype within a suitably insulated sardine can, isolating metallic components effectively.



(a) Integration of piezoelectric sensor connections with the Arduino Uno model



(b) Integration of piezoelectric sensor connections with the Arduino Nano model

Figure 2. Integration of the piezoelectric sensor with a widely recognized didactic development platform for enhanced connectivity.

Figure 3. The prototype of the pad comprises a piezoelectric sensor affixed to a rubber surface, an Arduino Nano microcontroller, a lithium battery, and an RF communication module.

3.2 Quantization, analog to digital convertion

This section addresses the stage of converting the analog voltage signal from the piezoelectric sensor into a corresponding digital value. A primary concern of this study is to enable a didactic replication within a microcontrolled framework. Thus, among various available options, a widely adopted platform found in academic laboratories is the Arduino. The most common board is the Arduino Uno, featuring the *ATmega328* microcontroller, which is also present in the compact version, Arduino Nano.

The analog signals generated by the piezoelectric cells are quantized with a resolution of 10 bits per sample. The standard serial channel employs 8-bit data along with 2 signaling bits. To maintain quality and utilize all bits of the transmitted characters, the maximum sampling frequency of 10 kHz is supported by the serial communication channel.

The created test pad consists of a sardine can, onto the lid of which the piezoelectric sensor is affixed using adhesive tape onto a rubber pad made of EVA (Ethylene Vinyl Acetat) material. This pad is designed to fit into the relief of the can's lid.

Figure 4 depicts an illustrative example of a signal acquired using the rubber pad. For a maximum sampling frequency, it becomes necessary in this case to employ binary operations within the computer program. However, for didactic purposes in visualizing the transmitted data through a text data terminal, a sampling rate of 1000 samples per second is utilized in this study. Therefore, each sample number is associated with a 1 ms interval.

The magnified view of a peak in the captured signal, corresponding to the impact action on the sensor, is illustrated in Figure 5, wherein an oscillation around sample number 2500 is observed. This instability could potentially stem from a reverberation caused by the drumstick used in the impacting action on the pad with the sensor.

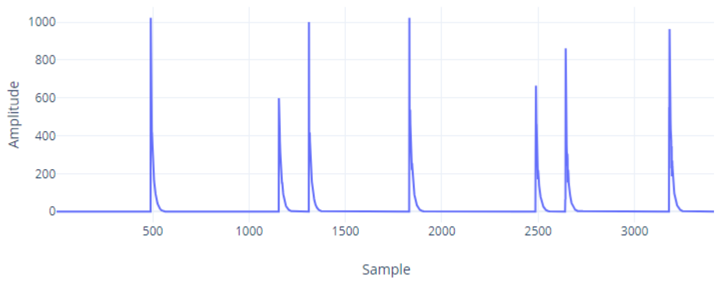


Figure 4. Wave form example on EVA rubber

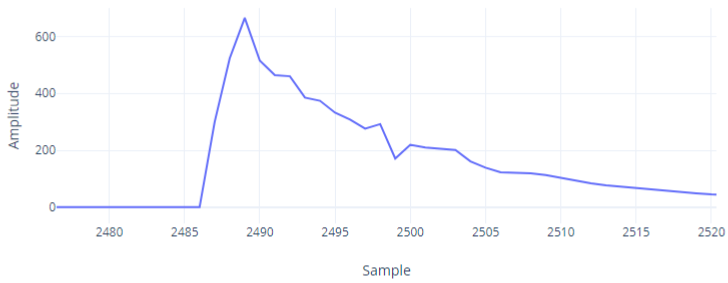


Figure 5. Wave Form Detail from EVA material, peak zoom

3.3 Data processing

Algorithm 1 Peak intensity detection with dead time algorithm

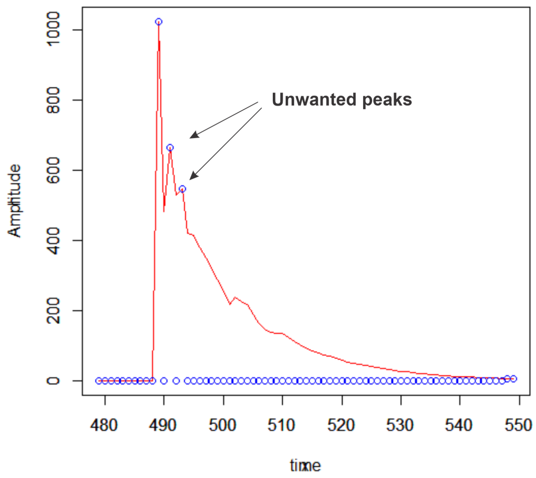
procedure PEAKDETECT(y) ▷ The vector denoted as y encompasses the acquired signal samples.
    $i \leftarrow 2; p[1] = 0;$
    while $i < lenght(y)$ **do** ▷ Processing samples of vector y
        if $y[i] > limiar$ **then** ▷ Considering a limiar value
            if $i - lastPeak > deadTime$ **then** ▷ Considering the dead time
                if $y[i] > y[1+1]$ **then** ▷ Monitors the increasing values
                    $p[i-1] \leftarrow 0;$ ▷ The previous peak is no longer the highest
                    $p[i] \leftarrow y[i];$ ▷ Establishes the present value as the updated peak
                else ▷ In the event that the current value is lower than the assumed peak
                    $p[i] \leftarrow 0;$
                    $lastPeak \leftarrow i;$
        $i \leftarrow i + 1;$
    return p ▷ Returns p as the peak vector, with its index associated with the time of the sample

The data processing stage is conducted through the implementation of Algorithm 12. This algorithm serves to identify the instances and intensities of peaks associated with the musician's actions. In the didactic context, with samples taken every 1 millisecond, the sample index itself signifies the occurrence moment of the sample.

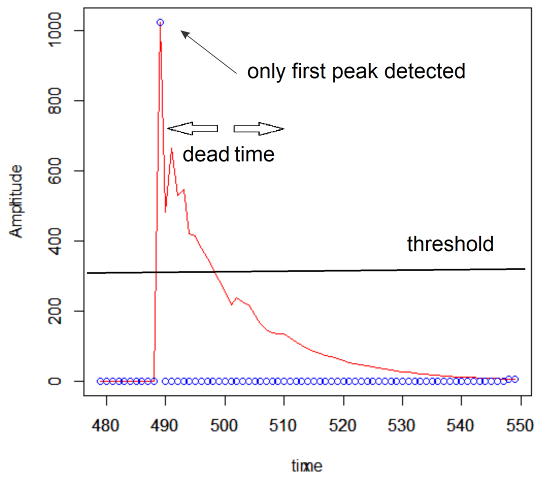
Figure 6a illustrates the identification of peaks above a minimum threshold. It can be observed that two secondary peaks are identified. By incorporating a dead time after detecting the first peak, as shown in Figure 6b, the complete algorithm successfully identifies only the desired peak. The timestamp of the sample corresponding to the last peak is stored to prevent the identification of new peaks during the mentioned "dead time." A dead time of 30 ms was employed in the system.

3.4 Data Analysis and data base

In the simplest system, each peak can indicate the act of playing the instrument. The amplitude of the peak is linked to the intensity of the action. The utilized prototype features a quantization stage with a quality of 10 bits,



(a) Peak detection algorithm illustration without considering a dead time



(b) Peak detection illustration with a dead time considering algorithm

Figure 6. Peak detection processing examples

and a sampling time of 1 millisecond.

For the proposed scenario, four sensors are addressed, corresponding to the percussive instruments: snare drum, bass drum, hi-hat, and crash cymbal.

Table 1. Drum items

PART	N.BIT	WEIGHT
Snare drum	1	1
Bass drum	2	2
Hi-hat	3	4
Crash cymbal	4	8

Considering the tempo of the music in "bpm," beats per minute, we can, in a didactic model, work with the downbeat and the upbeat, occurring within the beat time. Therefore, we have 4 bits corresponding to the 4 instruments. With the intention of using only one character to indicate all execution possibilities, the characters were processed as text using their respective hexadecimal representation. This is organized in Table 1 along with the associated weights of each instrument in the composition of the hexadecimal character.

Data analyzed from different standard rhythms, considering the note duration, as well as the upbeat (between one note and another), and determining a rhythm with a constant beat, the result is a structure of 32 bytes in hexadecimal text format for a pattern of 4 bars with 4 notes/4 upbeats. The sequences resulting from peak analysis are presented in the results section (Table 2).

3.5 Recovery Playing

To complete the system, the final stage involves recovery, reproducing the musician's actions by a robot. For experimental purposes, computer applications that replicate instrument sounds can be utilized. However, for this work, which intends to recover sounds generated from the actual instruments, the test prototype named "DrBot" is built using a drum and a snare drum, acrash cymbal, and a hi-hat. Solenoid valves are affixed to a mechanism that moves the drumstick, with its impact imitating the musician's action.

The reproduction mechanism, dubbed "DrBot," is illustrated in Figure 7. Figure 7a depicts the side of the bass drumskin, while Figure 7b showcases the side of the snare drum and the mechanisms. The hi-hat reproduction is accomplished using the blue tambourine, with the crash cymbal positioned above it.

(a) Mechanism of striking the bass drum's side using a nylon drumstick.

(b) Mechanisms employed for the crash cymbal, tambourine, and snare drum.

Figure 7. DrBot, drummer robot prototype

4 Results

With the intention of applying the findings of this study to the processing of signals captured by piezoelectric sensors from the impact action generated by musicians on pads, the achieved outcomes consist of a comprehensive and didactic system that enables the reproduction of real-time data processing experiments. Several processed rhythms were captured, resulting in a table containing data in hexadecimal format, as presented in the following table.

Table 2. Rythms patterns examples

Id	Name	Original Name	Data Recognized of Rythms Pattern	BPM
1	Basic Rock	Rock Base	E444546464445464E444546464445464	133
2	Rock 2	Rock 2	E040504141605040604050414160 5040	144
3	Double Rock	Rock Dobrado	64445444646454446444544474556554	144
4	Marcation	Marcação	10221022102210201022102210221020	150
5	Salsa	Salsa	20020012002020102002001200202010	150
6	Mambo	Mambo	11201120102010001120112010201000	144
7	Brazilian Rythm	Axé	10010022100100221001002210010022	166
8	Bass Drum	Surdo	00102010001020000010201000102000	166
9	Peaked Bass Drum	Surdo Repicado	12121000102010001212100010201000	150
10	Brazilian Rhythm	Baião	64445444646454446444544474556554	166

In Table 2, there are examples of rhythms along with the recognized data structure. The ”bpm” column represents the approximate velocity in ”bpm,” which stands for ”beats per minute.” Also, to ensure cultural accuracy and respect, the names of each rhythm were presented in their original language (Portuguese), accompanied by a free translation into English.

5 Conclusions

Known rhythms were performed by human musicians, and the processed signals formed a database containing representations of rhythmic patterns. A dataset with some example rhythms was generated through the processing and analysis of signals obtained from the piezoelectric sensor, as demonstrated. The intention is to employ the generated data in future projects involving the training of musical robots with AI for human-machine interaction activities. Thus, the aim is to offer musicians the possibility of musical interaction using AI. The system offers benefits such as compact size, energy efficiency, and reduced space and weight, which are advantageous for constructing robotic structures.

In its current form, the work presents a didactic case for introducing signal processing for real-time acquisition and processing. This has applications in the musical domain for accompanying human musicians as well as for practical training with other instruments.

The implementation of an automated structure for acoustic reproduction of rhythmic patterns has proven highly interesting. This secondary outcome complements the system, enabling its implementation across all phases. From this work, other algorithms and tools can be developed; one example is the assessment of the musician’s accuracy based on reference, aimed at rhythmic training.

Finally, it is worth emphasizing that the piezoelectric sensor processing approach proposed in this study can find application in contexts beyond music. Another example of application could involve counting impacts generated by machines on a production line, thus being applicable to manufacturing systems.

Acknowledgements. This work is powered by CEFET-MG (Centro Federal de Educação Tecnológica de Minas Gerias), UFMG (Universidade Federal de Minas Gerais), FAPEMIG (Fundo de Ampara à Pesquisa) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J. Uchiyama, T. Hashimoto, H. Ohta, Y. Nishio, J.-Y. Lin, S. Cosentino, and A. Takanishi. Development of an anthropomorphic saxophonist robot using a human-like holding method. In *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1–6, 2023.

[2] J. Wei, Z. Liu, F. Sun, and B. Fang. Robot learning from human demonstration for playing ocarina. In Z. Yu, X. Hei, D. Li, X. Song, and Z. Lu, eds, *Intelligent Robotics*, pp. 171–182, Singapore. Springer Nature Singapore, 2023.

[3] W. Marynowsky, J. Knowles, O. Bown, and S. Ferguson. *Sonic Robotics: Musical Genres as Platforms for Understanding Robotic Performance as Cultural Events*, pp. 219–235. Springer International Publishing, Cham, 2023.

[4] F. Zhao, M. Li, C. Zhou, G. Chen, and Y. Lou. Music melody extraction algorithm for flute robot. In *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 38–43. IEEE, 2020.

[5] G. Hoffman and G. Weinberg. Shimon: an interactive improvisational robotic marimba player. In *CHI EA ’10: CHI ’10 Extended Abstracts on Human Factors in Computing Systems*, pp. 3097–3102, 2010.

[6] Y. Maeda and Y. Moriyama. View-based teaching/playback for industrial manipulators. In *2011 IEEE International Conference on Robotics and Automation*, pp. 4306–4311, 2011.

[7] S. Soloman. *Sensors and Control Systems in Manufacturing*. McGraw Hill, 2nd edition, 2010.

[8] multiple. Ieee standard on piezoelectricity. *ANSI/IEEE Std 176-1987*, vol. , 1988.