Intelligent drummer module based on beat-tracking

Módulo baterista inteligente basado en beat -tracking

Módulo baterista inteligente com base em batida de batida

Juan Camilo Gómez Villamil[1]
Juan Pablo Hernández Mejía[2]
Luis Miguel Capacho Valbuena[3]

Received: September 5th, 2020
Accepted: November 20th, 2020
Available: January 11th, 2021

How to cite this article: J.C. Gómez Villamil, J.P. Hernández Mejía, L.M. Capacho Valbuena, Intelligent Drummer Module Based on Beat-Tracking. *Revista Ingeniería Solidaria*, vol. 17, no. 1, 2021, doi: https://doi.org/10.16925/2357-6014.2021.01.08

Research article. https://doi.org/10.16925/2357-6014.2021.01.08

1 Programa de ingeniería electrónica. Facultad de Ingeniería. Universidad de Quindío.
Email: jcgomezv@uqvirtual.edu.co
Orcid: https://orcid.org/0000-0002-3941-7407

2 Programa de ingeniería electrónica. Facultad de Ingeniería. Universidad de Quindío.
Email: jphernandezm@uqvirtual.edu.co
Orcid: https://orcid.org/0000-0003-4941-0767

3 Programa de ingeniería electrónica. Facultad de Ingeniería. Universidad de Quindío.
Email: lmcapacho@uniquindio.edu.co
Orcid: http://orcid.org/0000-0002-4109-2533

Abstract

Introduction: This article is the product of the research "Intelligent drummer module based on beat-tracking", developed in the Universidad del Quindío in 2020.

Problem: Most of the beat-tracking research tends to work on the exploration of theoretical strategies and not on the development of automatic devices that can be functional in real musical environments. As a consequence of the above, there is a scarcity of electronic devices for musical backing based on the beat-tracking technique.

Objective: To develop an automatic musical backing device based on beat-tracking with real-time operation.

Methodology: To achieve the general objective, the cascade development methodology is applied to address the process in three large phases: Development of the algorithm, implementation, and design of the prototype.

Results: The beat tracking algorithm is evaluated with 179 music excerpts of various musical genres and tested on the evaluation toolbox afterward. It is found that the drummer module algorithm has an AMLt (Allowed Metrical Levels, continuity not required) measure of 74.32%.

Conclusion: The algorithm provides the drummer module a solid operation and the multithreaded programming grants the embedded system real-time playback capability without lag problems.

Originality: Design of a functional product based on beat-tracking that provides three uses for a musician: musical accompaniment, musical practice, and Home-Studio recording.

Limitations: The implementation of this prototype within an embedded system with a lack of computational resources and the complexity of the analysis of audio signals.

Keywords: Algorithm, Beat-tracking, Onset, Embedded system, Multithreading, Drums.

Resumen

Introducción: Este artículo es el producto de la investigación: "Módulo baterista inteligente basado en beat-tracking", desarrollado en la universidad del Quindío en 2020.

Problema: La mayor parte de las investigaciones sobre beat-tracking tienden a trabajar en la exploración de estrategias teóricas y no en el desarrollo de dispositivos automáticos que pueden ser funcionales en entornos musicales reales. Como consecuencia de lo anterior, hay escasez de dispositivos electrónicos para acompañamiento musical basados en la técnica de beat-tracking.

Objetivo: Desarrollar un dispositivo de acompañamiento musical automático basado en beat-tracking con operación en tiempo real.

Metodología: Para lograr el objetivo general, se aplica la metodología de desarrollo en cascada abordando el proceso en tres grandes fases: Desarrollo del algoritmo, implementación y diseño del prototipo.

Resultados: El algoritmo de beat-tracking se evalúa con 179 muestras musicales de varios géneros y luego se prueba con el toolbox de evaluación. Se encuentra que el algoritmo del módulo baterista tiene una medida AMLt (niveles métricos permitidos, no se requiere continuidad) del 74,32%.

Conclusiones: El algoritmo proporciona al módulo baterista una operación sólida y la programación multiproceso otorga al sistema integrado la capacidad de reproducir secuencias en tiempo real sin problemas de retraso.

Originalidad: Diseño de un producto funcional basado en beat-tracking que proporciona tres usos para un músico: acompañamientos musicales, práctica musical y grabación Home-Studio.

Limitaciones: La implementación de este prototipo dentro de un sistema embebido con falta de recursos computacionales y la complejidad del análisis de las señales de audio.

Palabras clave: Algoritmo, Beat-tracking, Onset, Sistema embebido, Multiproceso, Batería.

Resumo

Introdução: Este artigo é o produto da pesquisa "Módulo baterista inteligente baseado em beat-tracking", desenvolvida na Universidad del Quindío em 2020.

Problema: a maior parte da pesquisa de beat-tracking tende a trabalhar na exploração de estratégias teóricas e não no desenvolvimento de dispositivos automáticos que podem ser funcionais em ambientes musicais reais. Como consequência do exposto, há uma escassez de dispositivos eletrônicos para backing musical baseados na técnica de beat-tracking.

Objetivo: Desenvolver um dispositivo musical automático de fundo baseado no rastreamento de batidas com operação em tempo real.

Metodologia: Para atingir o objetivo geral, a metodologia de desenvolvimento em cascata é aplicada para abordar o processo em três grandes fases: Desenvolvimento do algoritmo, implementação e design do protótipo.

Resultados: o algoritmo de rastreamento de batida é avaliado com 179 trechos de música de vários gêneros musicais e testado posteriormente na caixa de ferramentas de avaliação. Verificou-se que o algoritmo do módulo do baterista tem uma medida AMLt (Níveis Métricos Permitidos, continuidade não exigida) de 74,32%.

Conclusão: O algoritmo fornece ao módulo drummer uma operação sólida e a programação multithread garante ao sistema embarcado capacidade de reprodução em tempo real sem problemas de lag.

Originalidade: Projeto de um produto funcional baseado em rastreamento de batida que fornece três usos para um músico: acompanhamento musical, prática musical e gravação em estúdio doméstico.

Limitações: A implementação deste protótipo dentro de um sistema embarcado com falta de recursos computacionais e a complexidade da análise de sinais de áudio.

Palavras-chave: Algoritmo, Beat-tracking, Onset, Sistema embarcado, Multithreading, Bateria.

1. Introduction

The recognition of the beat of a song as a cognitive process is developed by human beings intuitively without any musical training [1]; Nowadays, the outstanding and probably only human capacity to naturally perceive the beat in music, continues to provoke cognitive scientists to strive to understand its brain processes [2]. At the same time, this concern has permeated other areas of knowledge such as electronics and prompting extensive research into the description of algorithms that apply different methodologies for the development of Beat-tracking [3].

Based on the literature review, it is clear that there is a low interest in designing new technologies related to the use of beat-tracking. This denotes a high lack of

knowledge of the market that could include this type of product; In other words, the shortage in the development of automatic devices for musical backing with percussion unleashes a clear carelessness of concern in a market space with great potential.

Accordingly, a prototype is designed with a real application based on beat-tracking, making a significant contribution in this area since many works that point to the theoretical analysis of beat-tracking do not materialize as a functional product. Likewise, it leaves a precedent of an engineering project to promote applicative orientations that innovate the world of science applied to digital signal processing and consumer electronics in music.

In this project, a beat-tracking algorithm is deployed in real-time and integrated into a system for playback of rhythmic sequences, focused on determining the Onsets and the characteristics of the musical audio signals to be processed; Computationally condensing the aforementioned algorithm, and the construction of the module with the vital components for its correct operation.

1.1. Literature review

Music, without any objection, produces physical and psychological reactions for most people [4]. When humans listen to music they try, in one way or another, to predict its rhythm. This process of rhythm perception encompasses the inference of the pulse, but also its continuity [2] [5]. In other words, music is a phenomenon based on temporal events that imply satisfying two restrictions: first, the instants where time is indicated, such as the beginning of a note played by one of the instruments; and second, this set of rhythms must reflect a constant interval between pulses [6]. Equally, music analysis is a growing area for world research; Beat-tracking systems are widely used for music information retrieval (MIR) [7].

In the 90s, D. Rosenthal modeled rhythm perception processes to find rhythmic structures of MIDI streams [8]. Afterward, Goto Masataka used multiple variable tempo and phase agents that help in predicting the beginning of a beat [9]. Likewise, he developed a software that implements real-time computer graphics synchronized with the musical audio signal. This program shows virtual dancers and several graphic objects whose movements and positions change over time (the time of each movement in the selected sequence is determined automatically based on the results of rhythm tracking) [10].

Furthermore, in 2001, Simon Dixon, including the agents in his work, processed a sequence of beats that were derived from the grouping of the intervals between each Onset [11]. Besides this, Hainsworth in 2004 fitted particle filters as an extension

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

of the Monte Carlo sequential estimation method to increase the performance of the problems of tracking musical signals and the extraction of time initiations [12].

In 2005, Bello, et al. presented possible enhancements for onset detection strategies as a result of the comparison of the most commonly used techniques [13], whereas Sethares presented and compared two methods for beat tracking: the first, based on the Bayesian work environment and the other, on a gradient strategy. These techniques can be applied to a digitized performance that does not require a musical score or MIDI transcription [14].

During the following year, Klapuri proposed to measure the degree of musical accent as a temporal function in four different frequency ranges, adding a bank of resonant filters that extract characteristics of the musical signal to finally analyze and estimate the probabilistic measurement of the pulse [15].

Then, in 2007, Davies and Plumbley designed a beat tracking method that recovers the beat times passing the output of an onset detection function through adaptively weighted comb filter bank matrices to separately identify the beat period and alignment [16]. In the same year, Ellis described a beat-tracking system that calculates global time and builds a transition cost function. This system applies dynamic programming to find the set of beats closest to the actual tempo and it uses 12-dimensional "Chroma" feature vectors that collect spectral energy to handle variation in musical instrumentation [6] [17].

Shiu, in his work (2008), proposed a tracking algorithm based on the Phase-lock-loop (PLL) technique which in real-time calculates the position of the pulses and the time signature of the music. Likewise, it has an algorithm based on the Kalman filter (KF) associated with enhanced probability data (EPDA). However, the use of Kalman filtering is not reliable in the presence of tempo fluctuations or expressive time deviations [18].

Four years later, Degara suggested the nearest neighbor regression algorithm to predict pulse accuracy that statistically evaluates the performance of the tracking system [19]. Furthermore, in 2018, Al-Hussaini presented a beat tracking algorithm in real-time, light enough to be implemented in an embedded system that predicts the pulse with a variant of dynamic programming; including a home memory algorithm to reduce the necessary computational costs. Based on experimental results, the pulse positions of a musical signal in real-time have been satisfactorily demonstrated [20].

Finally, in 2019, Ramírez developed a musical transcription system for stringed instruments based on the detection of the Onset, which proposes to generate the envelope of the signal and from this, determine the highest peaks as the instant of

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

time that the note is played; Obtaining, through a visualization interface, a score with the musical figures corresponding to the interpretation of the instrument [21].

2. Materials and methodology

2.1. Methodology

To achieve the general objective, the cascade development methodology proposed in [22] was applied to three specific objectives.

Objective 1: To develop a beat-tracking algorithm capable of identifying the pulse of a musical audio signal in real-time.

- **Phase 1: Requirements analysis**

Beat-tracking algorithm development projects were investigated. In addition to this, for the development of the algorithm, it was essential to clarify the construction needs. The following list shows the requirements:

- Active microphone streaming for audio capture
- Energy spectral analysis of music signal
- Onset calculation function
- Beat calculation function
- Tempo estimation based on beat intervals
- Future beat estimation function
- Rhythmic sequence playback function

- **Phase 2: Design and specifications**

The essential aspects of the algorithm design were identified based on the needs and in comparing different pulse monitoring methodologies. According to this, there are several useful libraries for this project such as Numpy, Time, Math, PyAudio, Scipy, SoundFile, and SoundDevice. All of them are open-source libraries that are provided for the Python community.

- **Phase 3: Implementation**

Taking into account the previous requirements, they were integrated into an optimized script that executes the algorithm and guarantees its operation over time.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

- **Phase 4: Verification**

Successive tests were performed to identify the pulse of different rhythmic patterns, corroborating that each musical audio signal is properly mapped in terms of time and frequency components.

- **Phase 5: Operation and maintenance**

Based on the results, to make improvements and add functionalities to the algorithm, which allowed troubleshooting that arose due to temporary inaccuracies that were found.

Objective 2: To implement the beat tracking algorithm and rhythmic sequence playback.

- **Phase 1: Requirements analysis**

The demand for computational resources required by the developed algorithm was identified. Based on this, different development cards have been found in the market which have similar computational characteristics. In order to choose the right board, a comparison is made between the Arduino Due, Beaglebone Black and Raspberry Pi 3b+.

From the above it was found: first, the Arduino Due has a processor that lacks computational capability and also does not have standardized audio output. Second, the Beaglebone Black has a number of GPIO pins that significantly exceed the need for the project, and its cost is the highest of the three cards. Finally, the Rasperry Pi 3b+ has the necessary resources for digital signal processing, it has standardized audio output and its cost-benefit ratio is the highest. In this way, the Raspberry Pi 3b+ is chosen as the development card, due to the convergence between what it offers and what the project requires.

On the other hand, there are development cards similar to the Raspberry Pi 3b+ on the market. For example, ASUS Tinker Board, OrangePi, Raspberry Pi 4, among other cards that due to the characteristics they share: Multi-Core processor, RAM of 1GB or more, and the operating system for Python execution, allow for the implementation of the system. However, it was decided to deploy the project on the Raspberry Pi 3b+ due to its ease of access in commercial terms.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

- **Phase 2: Design and specifications**

It was found that the computational and electrical characteristics demanded by the algorithm have been met by the Raspberry Pi 3b+, these specifications are shown in Table 1.

Table 1. Raspberry Pi 3b+ specifications.

Component	Description
Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Access	Extended 40-pin GPIO header
Connectivity	4 × USB 2.0 ports
sound	4 pole stereo output
SD card support	Micro SD format for loading operating system and data storage
Input power	5V/2.5A DC via micro USB

Source: own work based on [23]

- **Phase 3: Implementation**

The previously designed algorithm was loaded onto the selected development board. The Raspberry Pi 3b+ characteristics support the algorithm demands in real-time

- **Phase 4: Verification**

Repetitive beat tracking tests of different music signals were executed, checking the algorithm performance in real-time and its fidelity over time.

- **Phase 5: Operation and maintenance**

The need to make changes to the algorithm or migrate to a different technology, based on the operation of the acquired development board and its behavior in a real environment, was analyzed and discarded.

Objective 3: To develop a prototype of the device which integrates the algorithm on a development board with the power stages.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

- **Phase 1: Requirements analysis**

The essential power electronic components for the operation of the system were determined.

- **Phase 2: Design and specifications**

The output power required by the audio system was established in 15W and the electrical characteristics of the power supply fixed at 5V for the development board and 12V for the power stage.

- **Phase 3: Implementation**

Each of the electronic components was integrated into a box, making connections among all the stages according to the defined requirements.

- **Phase 4: Verification**

Cyclical tests of different musical signals were carried out and the correct operation of the device with regard to its design specifications were guaranteed. In addition, the power stage for the audio amplification performed as expected.

- **Phase 5: Operation and maintenance**

The system, composed of the algorithm, the power supply, the development board, and the amplification stage, worked properly. Specific results are shown in section three.

2.2. Main objective

Figure 1 shows the implementation process of the drum module prototype based on beat-tracking, the processing of the audio signal, and the most relevant aspects of the algorithm.

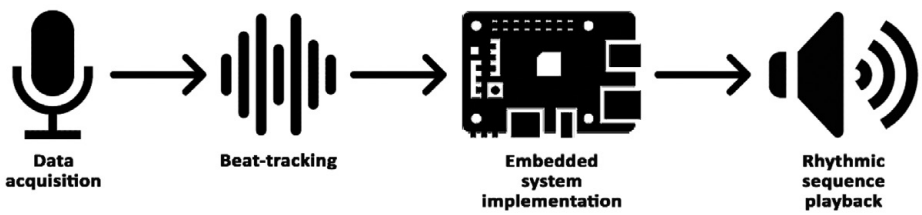

Figure 1. Implementation process
Source: own work

2.3. Data acquisition:

Conventionally, the collection of information from audio signals is carried out using microphones. These make it possible for high-performance sensor systems to generate high-quality electrical audio signals that are understandable to digital systems [24]. In other words, music signals can be converted into discrete information through the use of transducers (microphones) [25].

In the case of the project, a condenser microphone is used which provides a high quality in the perceptible bandwidth; it was connected to an audio interface that allows for the conversion of the analog signal to a digital signal that can be read via USB.

2.4. Beat tracking:

Beat tracking is defined as the determination of time instances in an audio signal allowing for metric analysis. Besides this, the goal of beat-tracking is to automatically track all beat locations in a collection of sound files and generate these note onset times for each signal [26].

Beat-tracking as an algorithm is made up of several processes that are the result of different state-of-the-art methods and can be divided into the following three programming stages:

2.4.1. Onset detection

The Onset is a single instant chosen to mark the temporarily extended transient. In most cases, it will coincide with the onset of the transient or the earliest time the transient can be reliably detected [13].

Notwithstanding, only a spectral analysis oriented on the magnitude of the spectrogram is necessary without making use of any phase or tone information; in this way, the improved function of the Onset detection of spectral flux through the suppression of the vibrato (quasi periodic change in the frequency of a note) proposed by [27] is used, where they apply a trajectory tracking approach based on maximum filters. This function is given by:

$$SF'(n) = \sum_{K=1}^{k=\frac{N}{2}} H(|X(n,k)| - |X(n-\mu,k)|) \qquad (1)$$

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function, n the frame number, and k the frequency interval index; and the signal is divided into overlapping chunks of length *N =2048sample*s. Furthermore, determines the offset and is given by:

$$\mu = máx\left(1, \left[\frac{\left(\frac{N}{2} - min\{w(n) > r\}\right)}{h} + \frac{1}{2}\right]\right) \quad (2)$$

Where *r* is a parameter that defines the ratio of the height of the window function *w(n)* with the length, and the hop size measured between two frames.

2.4.2. Selection of beats

Taking into account Equation (1), the corresponding Onsets of the signal are found. However, not all Onsets are constituted as real beats and an Onset must fulfill the three conditions proposed by [28] to be considered as a beat:

1. SF∗(n)=max(SF∗(n - pre_max:n+post_max))
2. *SF∗(n)≥mean(SF∗(n - pre_avg:n+post_avg))+ δ*
3. *n -n_(previous onset) >combination_width*

Where δ is the tunable threshold and the other parameters are taken from a large-scale investigation, which finds the best performance of the whole data set [29], obtaining the following values:

- *pre _ max*= 30 *ms*
- *pre _ avg*= 100 ms
- *post _ max*= 0 ms
- *post _ avg*= 0 ms
- *combination _ width*= 30 ms

The *post _ max* and *post _ avg* parameters are set to zero milliseconds due to the real-time application in this project..

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

2.4.3. Beats prediction

Considering that the algorithm does not have future information, all calculations are carried out causally (information from the present and the past); it is necessary to predict the position of the adjacent beats, allowing the algorithm to take the information and process it in real-time.

To address the problem mentioned above, the weighted moving average method proposed by [20] is used, which averages the time values of the beats of the current window to predict the future positions starting from the last beat calculated and estimate the tempo of future rhythms. The following equation calculates future beats:

$$\frac{new - beat(t)}{beat(t) - beat(t-1)} > \beta \qquad (2)$$

where β is the rejection threshold for the selection of future beats.

2.5. Algorithm operation:

Initially, the libraries are imported, global variables are created, and an instance is defined to control the audio. Subsequently, the audio files with .WAV extension, which contain the characteristic sounds of an acoustic drum kit, are stored in numerical arrangements. After these settings, the A pushbutton gives a signal to move forward into the sequence selector function in which the initial rhythm must be chosen. In the selector function, the B pushbutton allows the user to move among the rhythmic sequences and the A pushbutton establishes the desired pattern and starts the audio stream in order to capture the microphone signal and store the captured signal by itself.

A window of six seconds is used as a build-up period of signal information to start the process of calculating the onsets and beats. Thereafter, six-second windows are captured from the data buffer; these windows add one second of new information and reuse five seconds of the past.

On the other hand, information from the present and the past is used to estimate the current tempo and also predict the future beats in each algorithm iteration. With these calculated beats, the previously established rhythmic sequence is played which is adjusted to the calculated tempo and a musical metric of 4/4. Also, it is possible to switch to another rhythmic sequence by pressing the B pushbutton and stop the playback and audio streaming (includes ending all underlying processes) by pressing the A pushbutton during the streaming function. The operation of the entire algorithm is shown in Figure 2.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

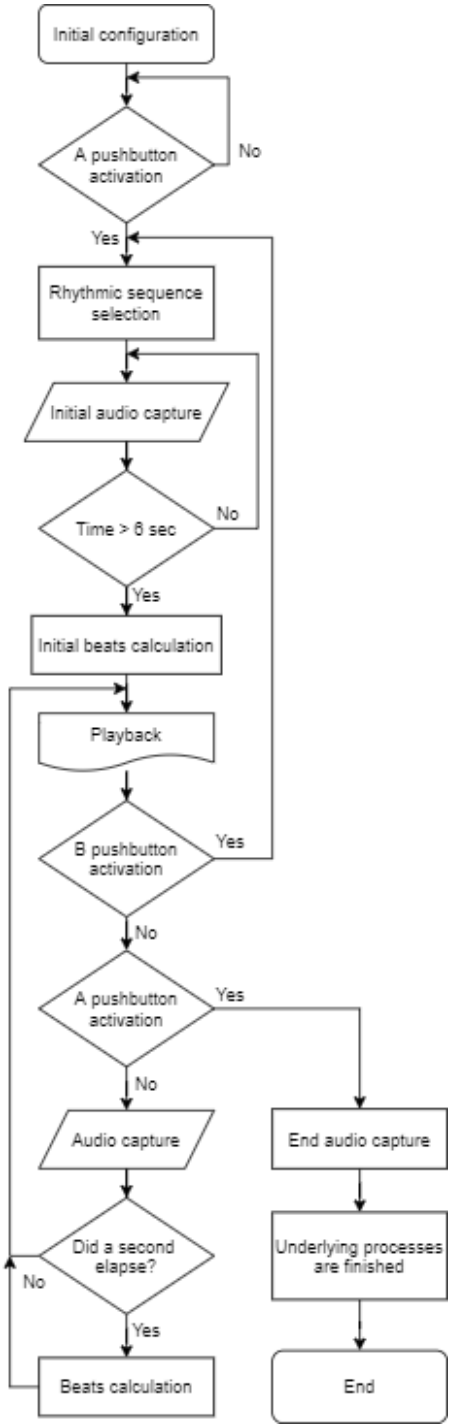

Figure 2. Beat-tracking algorithm flow diagram
Source: own work

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

2.6. Embedded system implementation:

Embedded systems are electronic devices designed to perform one or more functions in a complete system [30]. A relevant aspect is that they have both computational and power limitations, which optimize existing resources and reduce implementation costs [31].

Based on the above limitations, it is necessary to use the multithreaded programming model which can have multiple flows of execution; for example, a sequential part is created and subsequently, a series of tasks are created that can be executed in parallel. In that way, it is very important to manage the synchronization between threads and the programmer must prevent multiple threads from updating the same data at the same time [32].

Consequently, a Raspberry Pi + 3b is used, which is a development board based on a Debian GNU / Linux 10 operating system, with a Unix distribution and an ARM architecture [33]. This board supports and executes the beat-tracking algorithm developed in real-time. Also, this module allows for the integration of the microphone directly via USB and the audio output to the corresponding power stage.

2.7. Rhythmic sequence playback:

"Repetition is a part and parcel of symmetry—and of establishing motifs and hooks. You find a melodic or rhythmic figure that you like, and you repeat it throughout the course of the melody or song. This sort of repetition…helps to unify your melody; it's the melodic equivalent of a steady drumbeat, and serves as an identifying factor for listeners…." [34]

In this paper, the rhythmic sequences are structured as quarter-note levels (beat times) and the metric is based on a 4/4 time-signature. Furthermore, to cover as many musical genres as possible, six different basic rhythmic sequences are chosen: Rock, Pop, Funk, Disco, Punk, and Electro. These are capable of encompassing a large portion of current commercial music. From Figure 3 to Figure 8, each rhythmic sequence is drawn in a musical measure.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

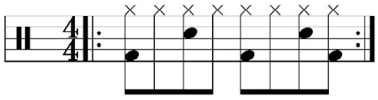

Figure 3. Rock rhythmic sequence
Source: own work

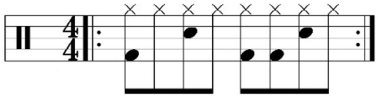

Figure 4. Pop rhythmic sequence
Source: own work

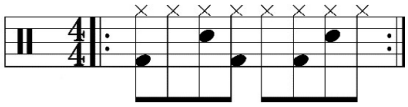

Figure 5. Funk rhythmic sequence
Source: own work

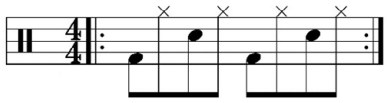

Figure 6. Disco rhythmic sequence
Source: own work

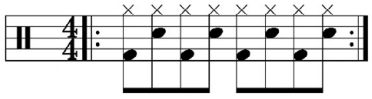

Figure 7. Punk rhythmic sequence
Source: own work

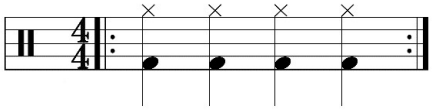

Figure 8. Electro rhythmic sequence
Source: own work

Another key element is that, according to the way the backing process is taking in this algorithm, it is mandatory to use the multithreaded programming model to play the rhythmic sequences in real-time without any overlapping event during the playback of each track.

Finally, knowing the low power of the audio signals delivered by the development board, it is necessary to amplify them for later emission into perceptible audio ranges. This work is performed by electronic audio amplifiers that are responsible for raising the intensity levels of the signals that correspond to the rhythmic sequences.

2.8. Drum Module Design:

The structure of the designed device is shown in Figure 9. The main container is made of wood and has dimensions length x width x height of 300 x 200 x 150 (mm). The module is powered with 110VAC and internally the conversion to 12 and 5 VDC is done by voltage adapters in order to supply the development board and the power stage.

Sound capture is performed with a Neewer NW-800 condenser microphone connected to a Behringer U-Phoria UM2 audio interface that provides signal digitization.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Also, the audio interface is connected to the development board via USB. The Raspberry Pi, the voltage adapters and the sound power stage will be all be located within the box.

The power boost of the audio signal coming from the development board is done through a TechMan ASM-7297 card that allows for an amplification of 15W. Two speakers of equivalent power and 8 ohms of impedance are connected to the amplification card for the playback of the rhythmic sequences and the complete operation of the system.

Figure 9. Structure of the drummer module
Source: own work

3. Results

The evaluation of the beat tracking algorithms is done by comparing the output beat times against annotated beat times. Nevertheless, the expected result of the annotation is an unequivocal depiction of the beginning of the musical events. Therefore, the beat tracking evaluation method for locating actual beat times will hinge upon whether the purpose is to identify descriptive beat locations or to replicate the human behavior when tapping their foot along with the music [35].

Over time, the timing error between found beats and annotated beats must be stable and slight. Thus, to guarantee the above condition, there are four continuity-based evaluation methods:

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

- CMLc: Correct Metrical Level, continuity required.
- CMLt: Correct Metrical Level, continuity not required.
- AMLc: Allowed Metrical Levels, continuity required.
- AMLt: Allowed Metrical Levels, continuity not required.

These are proposed in [35] and all of them are focused on regions of nonstop properly tracked beats.

In this project, the above beat tracking criteria are implemented on the beat tracking evaluation toolbox developed by [35], which is used to evaluate the algorithm's performance: accuracy and continuity. For this purpose, the beat tracking algorithm is tested with 179 music excerpts of various musical genres. Then, the results are compared to the [6], [15], [35], and [36] algorithm performances in Table 2 in order to quantitatively categorize and identify the performance of the designed algorithm with respect to the existing state of the art algorithms.

Table 2. Results of the [6], [15], [35], [36] algorithms on the beat tracking evaluation toolbox.

Algorithm	CMLc (%)	CMLt (%)	AMLc (%)	AMLt (%)
Davies	63.00	67.61	81.10	87.71
KEA	52.51	68.32	67.32	86.93
Dixon	52.94	65.08	69.45	90.73
Ellis	37.00	46.27	49.19	83.11
Drummer Module	38.37	49.44	56.99	74.32

Source: own work

Furthermore, the global beat error histograms are calculated for each algorithm. These histograms can show how deviated the calculated beats are from the annotated beats. Figure 10 shows the histograms for the [6], [15], [35], and [36] algorithms.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

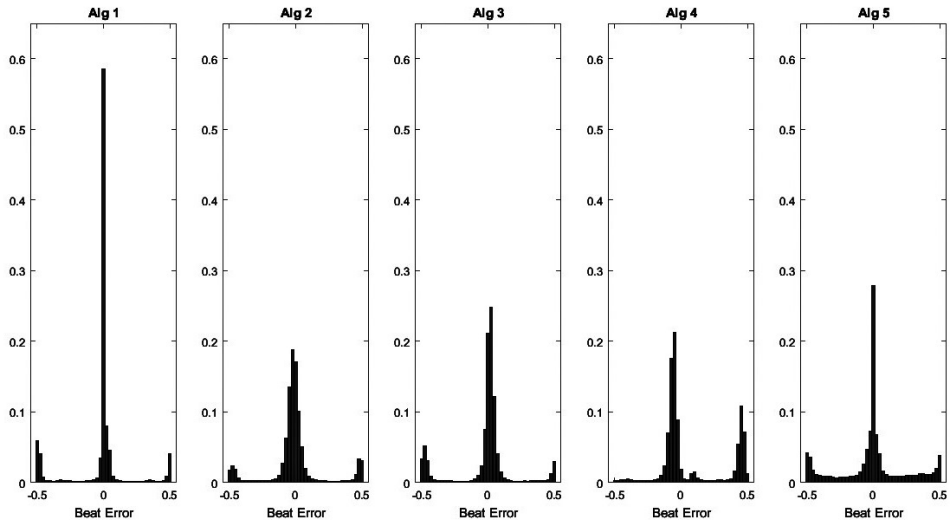

Figure 10. Global beat error histograms.
Alg 1: Davies. Alg 2: KEA. Alg 3: Dixon. Alg 4: Ellis. Alg 5: Drummer module.
Source: Beat tracking evaluation toolbox [35]

For finding continuous mismatches with the algorithm performances, the evaluation toolbox applies an offset analysis, adding ±70ms to all beat locations in 11.6ms intervals. The general idea of this test is to find some steady inconsistency, in phase, in the algorithms. The result of this analysis is shown in Figure 11.

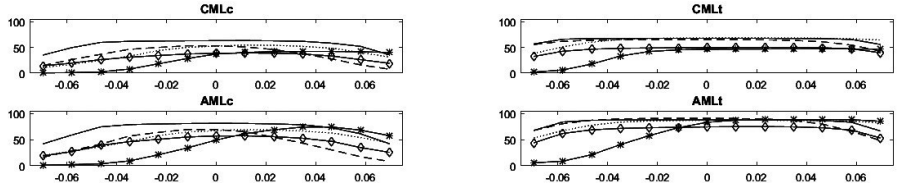

Figure 11. The effect of temporal offset to beat tracking accuracy.
Solid black line: Davies. Dotted line: KEA. Dashed line: Dixon. Line with crosses: Ellis. Line with diamonds: Drummer module.
Source: Beat tracking evaluation toolbox [35]

4. Discussion and conclusions

In the evaluation of the aforementioned beat tracking algorithms, it can be shown that all of them, except for Ellis, have either a nil or negative impact on the continuity-based criteria when adding any offset value to the found beats. On average, in the drummer

module algorithm within the ±70ms boundaries, the performance criteria decrease as follows: the CMLc from 38.37% to 30.29%, the CMLt from 49.44% to 46.32%, the AMLc from 56.99% to 43.57%, and the AMLt from 74.32 to 68.03%.

On the other hand, the drummer module error histogram shows that the algorithm has a strong peak around zero, equal to 27.85%. Moreover, the calculated beats that are in a ±0.1 deviation range are admissible due to the project purpose and the human perception about beats in music, and these are equivalent to 29.88%. In other words, it is possible to claim that the drummer module algorithm achieves 57.73% of suitable performance. Eventually, it is found that KEA, Dixon, and Ellis algorithms have their strongest peaks slightly deviate from the zero value; conversely, the Davies algorithm has its most remarkable peak at zero value exposing the best performance.

Taking into account the selected accuracy criteria, it is found that, for the CMLc measurement, the drummer module has a performance equal to 38.87%. This happens because the measurement only takes the longest segment of continuously correct beat tracking, but when the correct metric is necessary and continuity is not required (CMLt), the algorithms tend to improve their performance because it takes the total number of correct beats; in this case, the drummer module algorithm obtains 49.44%.

Nevertheless, the AMLc and AMLt are based on allowing metrical levels. It means that the annotated beats are halved and doubled to add other correct metrical levels. Thus, the AMLc measurement allows ambiguity in the metrical levels, but strictly requests continuity; for the drummer module this value is equivalent to 56.99%. Lastly, the AMLt measurement is the same as AMLc, but it does not require continuity; the drummer module gets 74.32%. In that way, if the project results are focused on the AMLt measurement, the drummer module greatly improves its accuracy.

In general, it is found that the use of allowed metrical levels and without taking into account the continuity can significantly improve the performance of the algorithms; to be specific the average increase from CMLc to AMLt is about 35.79%. Also, the flexibility provided by the AMLt measurement, based on its characteristics, is the closest metric to human beats perception in music.

Finally, the project algorithm performance provides the drummer module a solid operation in terms of beats calculation of musical audio signals; this includes the accuracy in the tempo estimation and the correct behavior over time. Besides this, the multi-threaded programming provides the embedded system the capability to playback in real time without lag problems. In this way, it is valid to affirm that the designed algorithm meets the scope of the project.

e-ISSN 2357-6014 / Vol.17, n°. 1 / January - april 2021 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

5. References

1. V. Alluri *et al.,* "Musical expertise modulates functional connectivity of limbic regions during continuous music listening," *Psychomusicology Music. Mind, Brain,* vol. 25, no. 4, pp. 443–454, 2015, doi: https://doi.org/10.1037/pmu0000124.

2. P. Toiviainen, I. Burunat, E. Brattico, P. Vuust, and V. Alluri, "The chronnectome of musical beat," *Neuroimage,* no. September, pp. 1053–8119, 2019, doi: https://doi.org/10.1016/j.neuroimage.2019.116191.

3. A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Trans. Audio, Speech Lang. Process.,* vol. 20, no. 9, pp. 2539–2548, 2012, doi: https://doi.org/10.1109/TASL.2012.2205244.

4. W. T. Fitch, "The biology and evolution of music: A comparative perspective," *Cognition,* vol. 100, no. 1, pp. 173–215, 2006, doi: https://doi.org/10.1016/j.cognition.2005.11.009.

5. H. Honing, "Without it no music: Beat induction as a fundamental musical trait," *Ann. N. Y. Acad. Sci.*, vol. 1252, no. 1, pp. 85–91, 2012, doi: https://doi.org/10.1111/j.1749-6632.2011.06402.x.

6. D. P. W. Ellis, "Beat tracking by dynamic programming," *J. New Music Res.*, vol. 36, no. 1, pp. 51–60, 2007, doi: https://doi.org/10.1080/09298210701653344.

7. M. Istvanek, Z. Smekal, L. Spurny, and J. Mekyska, "Enhancement of conventional beat tracking system using Teager-Kaiser energy operator," *Appl. Sci.,* vol. 10, no. 1, pp. 1–20, 2020, doi: https://doi.org/10.3390/app10010379.

8. D. Rosenthal, "Emulation of Human Rhythm Perception," *Computer Music Journal,* vol. 16, no. 1. pp. 64, 1992, doi: https://doi.org/10.2307/3680495.

9. M. Goto and Y. Muraoka, "A Real-time Beat Tracking System for Audio Signals," *Work. Comput. Audit. Scene Anal. Music*, pp. 68–75, 1995.

10. M. Goto and Y. Muraoka, "Real-time Beat Tracking for Drumless Audio Signals," *Speech Commun.,* vol. 27, no. 3--4, pp. 311–335, 1999.

11. S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Int. J. Phytoremediation,* vol. 21, no. 1, pp. 39–58, 2001, doi: https://doi.org/10.1076/jnmr.30.1.39.7119.

12. S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP J. Appl. Signal Processing*, vol. 2004, no. 15, pp. 2385–2395, 2004, doi: https://doi.org/10.1155/S1110865704408099.

13. J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. Speech Audio Process.,* vol. 13, no. 5, pp. 1035–1046, 2005, doi: https://doi.org/10.1109/TSA.2005.851998.

14. W. A. Sethares, R. D. Morris, and J. C. Sethares, "Beat tracking of musical performances using low-level audio features," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 2, pp. 275–285, 2005, doi: https://doi.org/10.1109/TSA.2004.841053.

15. A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the Meter of Acoustic Musical Signals," vol. 14, no. 1, pp. 342–355, 2006, doi: https://doi.org/10.1109/TSA.2005.854090.

16. M. E. P. Davies and M. D. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE Trans. Audio, Speech Lang. Process.,* vol. 15, no. 3, pp. 1009–1020, 2007, doi: https://doi.org/10.1109/TASL.2006.885257.

17. D. P. W. Ellis and G. E. Poliner, "Identifying ' Cover Songs ' With Chroma Features And Dynamic Programming Beat Tracking Daniel P . W . Ellis and Graham E . Poliner Columbia University, New York NY 10027 USA," *New York,* pp. 1429–1432, 2007, doi: https://doi.org/10.1109/ICASSP.2007.367348.

18. Y. Shiu and C. C. J. Kuo, "Musical beat tracking via Kalman filtering and noisy measurements selection," *Proc. - IEEE Int. Symp. Circuits Syst.,* pp. 3250–3253, 2008, doi: https://doi.org/10.1109/ISCAS.2008.4542151.

19. N. Degara, E. A. Rua, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley, "Reliability-informed beat tracking of musical signals," *IEEE Trans. Audio, Speech Lang. Process.,* vol. 20, no. 1, pp. 278–289, 2012, doi: https://doi.org/10.1109/TASL.2011.2160854.

20. I. Al-Hussaini *et al.,* "Predictive Real-Time Beat Tracking from Music for Embedded Application," *Proc. - IEEE 1st Conf. Multimed. Inf. Process. Retrieval, MIPR 2018*, pp. 297–300, 2018, doi: https://doi.org/10.1109/MIPR.2018.00068.

21. H. Ramírez, "Sistema De Transcripción Automática Musical Para Instrumentos De Cuerda Percutida," vol. 23, no. 3, pp. 35–37, 2019.

22. S. Balaji, “Waterfall vs v-model vs agile: A comparative study on SDLC,” *WATEERFALL Vs V-MODEL Vs Agil. A Comp. STUDY SDLC*, vol. 2, no. 1, pp. 26–30, 2012.

23. RaspberryPi, “Raspberry Pi 3 Model B+,” pp. 1–5, Accessed: Sep. 04, 2020. [Online]. Available: www.raspberrypi.org/products/raspberry.

24. L. Urbansky and U. Zölzer, “A digital radio-frequency condenser microphone with amplitude modulation,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2019-May, pp. 1–5, 2019, doi: https://doi.org/10.1109/ISCAS.2019.8702611.

25. D. Caicedo, J. C. Martínez, and J. Andrade, “Control de iluminación con reconocimiento remoto de voz,” *Ing. Solidar.,* vol. 7, no. 13, pp. 35–45, 2011.

26. F. Shaikh, “Learn Audio Beat Tracking for Music Information Retrieval (with Python codes),” pp. 1. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/02/audio-beat-tracking-for-music-information-retrieval/.

27. S. Böck and G. Widmer, “Maximum filter vibrato suppression for onset detection,” *DAFx 2013 - 16th Int. Conf. Digit. Audio Eff.,* pp. 1–7, 2013.

28. S. Böck, F. Krebs, and M. Schedl, “Evaluating the online capabilities of onset detection methods,” *Proc. 13th Int. Soc. Music Inf. Retr. Conf. ISMIR 2012,* pp. 49–54, 2012, doi: https://doi.org/10.5281/zenodo.1416036.

29. I. de percepción Computacional, “GitHub - CPJKU/onset_db: Onset data set which can be used to tune/evaluate onset detection algorithms,” pp. 1, 2020. [Online]. Available: https://github.com/CPJKU/onset_db.

30. A. Modules, “Embedded Systems - Adaptive Modules,” pp. 1, 2020. [Online]. Available: https://adaptivemodules.com/info/embedded-systems/.

31. C. Koulamas and M. T. Lazarescu, “Real-time embedded systems: Present and future,” *Electron,* vol. 7, no. 9, pp. 10–12, 2018, doi: https://doi.org/10.3390/electronics7090205.

32. G. Zaccone, “Python parallel programming,” pp. 15, 2015.

33. J. H. M. Dassen and C. Stickelman, “The Debian GNU/Linux FAQ,” pp. 1–3, 2019.

34. M. Miller, "The Complete Idiot's Guide to Music History," pp. 1–336, 2008.

35. M.E.P. Davies, N. Degara, and M.D. Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," *Audio*, no. October, pp. 17, 2009, doi: https://doi.org/10.13140/2.1.4703.4568.

36. S. Dixon, "Evaluation of the audio beat tracking system BeatRoot," *J. New Music Res.,* vol. 36, no. 1, pp. 39–50, 2007, doi: https://doi.org/10.1080/09298210701653310.