

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Interactive Tool for Rhythmic Synchronization

Carolynn Francis M.Sc. Thesis October 2021

> Supervisors: prof. dr. ir. Dennis Reidsma BPA Spieker MA

Human Media Interaction Group Faculty of Electrical Engineering, Mathematics and Computer Science University of Twente P.O. Box 217 7500 AE Enschede The Netherlands

Acknowledgments

The thesis marks the end of my Masters in Embedded Systems at the University of Twente. The past two years have been very challenging. The thesis has taught me a lot, it has given me the opportunity to widen my horizons. I gained fundamental and important skills to get through life.

I would like to express gratitude to my primary supervisors Dennis, Benno for the motivation and support through out the entire period. I also would like to thank my external supervisor Andreas for his invaluable feedback.

Lastly, I could not be more grateful to my friends and family for their immense love and support.

Abstract

Interactive technology for music teaching is a developing research topic. This thesis focuses on helping pre-service teachers in music teaching to help students stay in rhythm using interaction technology. The report delves into various literature that helps develop a system starting from sensing the instrument, then analyse elements within the data retrieved from the instrument; such as beat detection and rhythm synchronisation and lastly providing relevant visual feedback to the pre-service teachers and students.

Based on the literature study and the background research, a prototype is developed that reads the instrument data and analyses it. The prototype is later evaluated based on its individual properties and a user test is conducted to evaluate the performance of the audio-visual feedback component of the system.

The system is maximally simple but it should act as a starting point to develop holistic, interactive systems for music teaching. The system is flexible such that the individual components of the end-to-end system can be improved by based on the need.

Contents

Ac	cknov	vledgments	ii
Ał	ostrad	xt	iii
Li	st of a	acronyms	vii
1	Intro	oduction	1
	1.1	Approach	2
	1.2	Research goals	2
2	Lite	rature Study	3
	2.1	Interaction technology in Music teaching	3
	2.2	Measurement of audio output	3
		2.2.1 Piezo Sensor	6
	2.3	Beat Detection	6
	2.4	Rhythm Synchronisation	8
	2.5	Networking	11
	2.6	Feedback	12
		2.6.1 Visual Feedback	12
		2.6.2 Audio Feedback	15
	2.7	Conclusion	17

3	Sys	em Design 18			
	3.1	Hardw	vare Setup	18	
		3.1.1	Microcontroller	18	
		3.1.2	Sensors	19	
		3.1.3	Full Hardware Setup	20	
	3.2	Softwa	are Setup	22	
		3.2.1	Reading Sensor Data	22	
		3.2.2	MQTT	22	
		3.2.3	Beat Detection	22	
		3.2.4	Synchronisation	24	
		3.2.5	Feedback	25	
4	Eva	luation		26	
	4.1	Exper	iment Design	26	
	4.2	Effectiveness of the Network Communication			
	4.3	Effecti	veness of Beat Detection Algorithm	28	
	4.3 4.4	Effecti Effecti	iveness of Beat Detection Algorithm	28 29	
	4.3 4.4	Effecti Effecti 4.4.1	iveness of Beat Detection Algorithm	28 29 30	
	4.3 4.4	Effecti Effecti 4.4.1 4.4.2	iveness of Beat Detection Algorithm	28 29 30 31	
	4.3 4.4	Effecti Effecti 4.4.1 4.4.2 4.4.3	iveness of Beat Detection Algorithm iveness of Feedback First Iteration User Test User Feedback	28 29 30 31 32	
5	4.3 4.4 Con	Effecti Effecti 4.4.1 4.4.2 4.4.3	iveness of Beat Detection Algorithm	28 29 30 31 32 35	
5 Re	4.3 4.4 Con	Effecti Effecti 4.4.1 4.4.2 4.4.3	iveness of Beat Detection Algorithm	28 29 30 31 32 35 37	

Α	Арр	endix /	A : Program Codes	42
	A.1	Progra	am at Client PI	42
	A.2	Progra	ams in the Server PI	45
		A.2.1	Program for real-time visualisation	45
		A.2.2	Master Program	47
в	Арр	endix I	B: Consent Form	50

List of acronyms

- **BLSTM** Bidirectional Long Short-Term Memory
- **BPM** Beats Per Minute
- **DBN** Dynamic Bayesian Network
- **DNN** Deep Neural Networks
- **DP** Dynamic Programming
- **GMM** Gaussian mixture models
- GUI Graphical User Interface
- HMM Hidden Markov Method
- IBI Inter Beat Interval
- LSTM Long Short-Term Memory
- MIDI Musical Instrument Digital Interface
- **MQTT** Message Queuing Telemetry Transport
- **OSC** Open Source Control
- **RNN** Recurrent Neural Networks
- SMI Smart Musical Instruments
- SPI Serial Peripheral Interface
- **STFT** Short Time Fourier Transforms

List of Figures

2.2	Smart Cajon (Turchet, McPherson, & Barthet, 2018)	4
2.1	Architecture of SMI (Turchet, 2019)	5
2.3	Sensus Smart Guitar developed by MIND Labs (Turchet, McPherson, & Fischione, 2016)	6
2.4	Illustration of beat and downbeat in a metric structure	6
2.5	Pipeline for beat/downbeat tracking	7
2.6	Metronomes out of phase	10
2.7	Metronomes in-phase	11
2.8	The visualisation display prototype developed by (Ferguson, Moere, & Cabrera, 2005)	13
2.9	Digital Violin Tutor by (Percival, Wang, & Tzanetakis, 2007)	14
2.10	Birch Lab system (Sain, Leinweber, Mendler, & Hast, 2019)	14
2.11	Visualisation concepts developed by (Kruijshaar, 2020)	15
2.12	Interactive sonification by (Ferguson, 2006)	16
2.13	Ipalmas (Jylhä, Ekman, Erkut, & Tahiroğlu, 2011)	16
2.14	Workflow	17
3.1	Raspberry Pi 4 model B	19
3.2	Piezo Sensor	19
3.3	MCP3008 ADC from Microchip	20

LIST OF FIGURES

3.4	Circuit design for the client pi	21
3.5	Hardware Setup	21
3.6	Flowchart of Beat Detection	23
3.7	In Sync	24
3.8	Out of Sync	25
4.1	Latency check	26
4.2	Distribution of Communication Latency between publisher and sub- scriber	27
4.3	Distribution of Latency observed right after visualisation	28
4.4	BPM check	29
4.5	Auto-scale "ON"	30
4.6	Auto-scale "OFF"	31
4.7	Warm up	32
4.8	Perfect Sync	33

Chapter 1

Introduction

Music in early childhood helps with children's development in reading, language skills, cognitive ability, social skills and emotional development. So it is important to include music as part of the educational program (Barrett, Flynn, & Welch, 2018). In the Netherlands, the Dutch government has instituted "Méér Muziek in de Klas"¹ foundation program that promotes music education in primary schools within the country.

Pre-school or primary school teachers provide music education. Who in some cases have no music major. They deem music and music teaching essential but they do not perceive themselves to be competent in this field for many reasons (Burak, 2019). The main reason is their lack of musical knowledge and perceived limitations in areas such as playing an instrument, vocal training, rhythm synchronisation and melody. Hence pre-service teachers have a negative belief on their self-efficacy in music training (Burak, 2019). On the other hand it has been noted that pre-service teachers have a positive attitude towards technology in music education (Atabek & Burak, 2020).

This research aims to provide a technology based integrated education system that will help pre-service teachers gain their confidence in music teaching by providing feedback to the teachers about the performance of individual students and also their level of group coordination in producing rhythmic music. There is also an added feature of real-time audio-visual feedback where the feedback is directly given to the student by the system itself with no human interference.

¹More music in the class

1.1 Approach

This research focuses on developing an integrated system that includes a smart instrument that detects beats per minute, synchronisation of the beats between other instruments and provides audio-visual feedback on the synchronisation. The entire process runs in real-time.

The thesis report structure is as follows: Chapter 2 covers the background research on the different parts of the system as well as technology driven music education. In Chapter 3, the implementation of the system is explained. Chapter 4, evaluates the system and answers the constructive research question. The future work of the project is discussed in Chapter 5 and the conclusion is Chapter 6.

1.2 Research goals

The research questions for the project can be divided in two sections; constructive and empirical.

- Constructive: How to integrate such a system? A system that performs instrument reading, detects beats, perform synchronisation and provide feedback to the players?
- 2. Constructive: How good is the integrated system?
 - · How effective is the beat detection algorithm?
 - · How effective is the networking?
 - · How effective is the feedback module?
- 3. Empirical: How suitable is it for follow up research?
 - Does the effectiveness of the system from previous section affect the possibility of follow up?
 - Is the system suitable for real-time use or offline(data collection for the educator?
 - Is the system flexible to incorporate other instruments other than percussion?

Chapter 2

Literature Study

This section mentions the various literature present for the different research questions mentioned in the previous section.

2.1 Interaction technology in Music teaching

Using interaction technology in music education helps children relate more closely to school music and facilitates students' creation of connections between the music they listen to and their real lives. The use of technology in music education increases the musical performance, participation, interest and motivations of students as well as their musical perceptions (Atabek & Burak, 2020).

2.2 Measurement of audio output

To sense the audio signal produced by the instrument, we should first consider the type of instruments we are going to use. In our project we will be using percussion instruments (Triangle, Drums, Boomwhackers, etc). We further proceed on to the introduction to smart musical instruments and about the different such instruments present. (Turchet et al., 2016) introduces us to something called Smart Musical Instruments (SMI) which refers to the network of inter operable devices dedicated to the production and/or reception of musical content. Here they explain how implementing IoT technologies to musical instruments have the potential to enable new musical experiences. To make a SMI however, we need various technologies including wireless sensor networks, networked musical performance systems

CHAPTER 2. LITERATURE STUDY

(Rottondi, Chafe, Allocchio, & Sarti, 2016), sensors and actuators (Turchet et al., 2018) and embedded boards that support real-time audio and sensor processing (Gonzalez Sanchez et al., 2018). The architecture of SMI is shown in Figure 2.1

Since our instrument is percussion based, the sensors used for it can be a piezoelectric or a condensor microphone or both. (Turchet et al., 2018) developed a smart Cajon seen in Figure 2.2 using both the microphones to capture the sound.

Another smart instrument is the Sensus Smart guitar seen in Figure 2.3 developed by (Turchet et al., 2016). It has multiple sensors and an interoperable wireless communication using Bluetooth and Musical Instrument Digital Interface (MIDI) protocols.



Figure 2.2: Smart Cajon (Turchet et al., 2018)



Figure 2.1: Architecture of SMI (Turchet, 2019)



Figure 2.3: Sensus Smart Guitar developed by MIND Labs (Turchet et al., 2016)

2.2.1 Piezo Sensor

In the pressure sensor, a thin membrane is placed on a massive base to transfer the applied force to the piezoelectric element. Upon application of pressure on this thin membrane, the piezoelectric material gets loaded and starts generating electrical voltages. The produced voltage is proportional to the amount of pressure applied.

2.3 Beat Detection

After sensing the audio signal, analysis is done to detect beat and rhythmic synchronisation. Beats are fundamental to the perception of timing in music, which usually matches the foot tapping a person does when listening to music. In musical notation, a bar (or measure) is a segment of time corresponding to a specific number of beats in which each beat is represented by a particular note value and the boundaries of the bar are indicated by vertical bar lines. A downbeat is the first beat of a bar (Fuentes, 2019). An illustration of beat and downbeat in a metric structure can be seen in Fig 2.4. In this section, we discuss various methods used to detect beat in audio signals.



Figure 2.4: Illustration of beat and downbeat in a metric structure

The first step for beat tracking is extracting the features from the audio signal. Features include, low-level features such as spectral features as in (Oliveira, Gouyon, Martins, & Reis, 2010); mid-level features like onsets either in discrete (M. E. Davies & Plumbley, 2007) or continuous form (Stark, Davies, & Plumbley, 2009), or higher level features such as rhythmic patterns(Krebs, Böck, & Widmer, 2013) or metrical relations are used or a combination of these features (Böck, Krebs, & Widmer, 2014). In some cases, the audio signal is transformed to the frequency domain using Short Time Fourier Transforms (STFT) (Böck & Schedl, 2011) (Krebs et al., 2013) (Cheng, Fukayama, & Goto, 2018) (Di Giorgi, Mauch, & Levy, 2021).

To determine the periodicity of the extracted features, autocorrelation (Böck & Schedl, 2011) or comb filters (M. Davies, Plumbley, Stark, & Davies, n.d.; Böck, Krebs, & Widmer, 2015) can be used.

The final stage is the post-processing stage where dominant tempo is determined using Dynamic Programming (DP) (Ellis, 2007), Hidden Markov Method (HMM) (Paulus & Klapuri, 2009; Krebs et al., 2013; Peeters & Papadopoulos, 2011), Deep Neural Networks (DNN) (Böck, Davies, & Knees, 2019) or Recurrent Neural Networks (RNN) (Cheng et al., 2018; Böck et al., 2014).

In (Oliveira et al., 2010), they operate on the spectral flux autocorrelation and determine a tempo and period hypothesis from which they further output beats on-the-fly.

The general pipeline for beat/downbeat tracking is displayed in Figure 2.5:



Figure 2.5: Pipeline for beat/downbeat tracking

Most of the literature covers non-real-time and offline approaches. The ones that do proposes real-time offline onset detection methods do not incorporate machine learning techniques or use probabilistic information.

(Eyben, Böck, Schuller, & Graves, 2010) designed an online algorithm that aims to minimise the delay between the occurrence of the onset in the audio signal and its reporting. The method used is Long Short-Term Memory (LSTM) and RNN.

RNN in theory is able to remember past values but in practice however suffers from the vanishing gradient problem (input values decay or blow up over time). (Böck &

Schedl, 2011) implements a Bidirectional Long Short-Term Memory (BLSTM) with RNN to overcome the problem caused by RNN.

Programming languages that can be used for analysis in case of acoustic audio signal are Python, C++. With regards to MIDI, we can use the aforementioned languages and also Max/MSP.

(Fuentes, 2019) gives a detailed review of beat and downbeat estimation, where the author explains both the concepts and also the various techniques that have been employed to identify beat and downbeat over the years.

Section 2.4 further continues the analyses of audio input with regards to rhythm synchronisation and the different techniques used for beat detection and rhythm synchronisation are displayed in Table 2.1

2.4 Rhythm Synchronisation

Rhythm synchronisation between musicians is to maintain the same tempo and alignment. For this purpose we will have to determine the base beat of the audio. It is also called downbeat, determining real-time downbeat from an instrument is a challenge and there are various research going on in that area.

(Böck, Krebs, Durand, Pöll, & Balsyte, 2017) present a real-time online beat and off-beat drummer. In order to find the beat-synchronous features, they first identify the beat positions using RNN and a spectral flux based input feature is scored with Gaussian mixture models (GMM) which is then used as observations for a Dynamic Bayesian Network (DBN).

(Krebs et al., 2013) present a HMM based system that simultaneously extracts beats, downbeats, tempo, meter, and rhythmic patterns. For this, they had annotated 697 ballroom dance pieces with beat and measure information. The result showed that explicitly modelling rhythmic patterns of dance styles reduces octave errors and substantially improves downbeat tracking.

Table 2.1 shows the recent methods used for beat and downbeat tracking of audio signals

Authors	Beat	D.Beat	Features	Likelihood	Post-Proc
(M. E. Davies & Plumbley, 2007)	Yes	No	ODF+AC	comb filter	NA
(Stark et al., 2009)	Yes	No	ODF	comb filter	NA
(Oliveira et al., 2010)	Yes	No	SF+AC	NA	NA
(Böck & Schedl, 2011)	Yes	No	STFT	RNN+BLST	MPeak Pick
(Peeters & Pa- padopoulos, 2011)	Yes	Yes	CH + SB	template	HMMs
(Krebs et al., 2013)	Yes	Yes	STFT	GMM	HMM+DBN
(Böck et al., 2014)	Yes	No	STFT	RNN+BLST	MDBN
(Böck et al., 2015)	Yes	No	STFT	RNN	comb filter
(Krebs, Böck, Dorfer, & Wid- mer, 2016)	No	Yes	STFT+CH	Bi-GRU	DBN
(Böck et al., 2017)	Yes	Yes	STFT	RNN+GMM	DBN
(Cheng et al., 2018)	Yes	No	mel-log STFT	RNN	DBN
(Böck & Davies, 2020)	Yes	Yes	STFT	TCN	DBN
(Di Giorgi et al., 2021)	Yes	No	mel-log STFT	NA	HMMs

 Table 2.1: The table is extended from paper (Fuentes et al., 2019)

Entrainment

The term 'entrainment' refers to the process by which independent rhythmical systems interact with each other. This process of interaction may result in the systems to synchronise with each other, aligning in both phase and period. In music, inter-individual entrainment is the co-ordination between the individuals in the group. This is facilitated by the entrainment of rhythms to auditory and visual information (Clayton, 2012).

(Pantaleone, 2002) performed an experiment where metronomes of different frequencies are placed on a light wooden board. Naturally, they oscillate is different directions as seen in Figure 2.6. Then the board is placed on top of two empty soda cans, it was then noticed that gradually, the metronomes start to synchronise with each other as seen in Figure 2.7.



Figure 2.6: Metronomes out of phase



Figure 2.7: Metronomes in-phase

Entrainment can be relatively symmetrical, for example, in a musical ensemble any individual can influence any other, but in practice some people are more likely to have influence than the others (conductors, soloists, senior musicians).

2.5 Networking

Networking of the system in very important since we are dealing with a real-time system, the delay of response should be reduced to a minimum. Hence, we need to develop a system with low latency. Nowadays, the vast majority of professional audio devices targeting the hard requirements of real-time performance are produced using ad-hoc real-time operating systems or dedicated digital signal processors. These are complex to program, offer very limited support to interface to other hard-ware peripherals, and lack modern software libraries for networking and access to cloud services. Such technical limitations are one of the main reasons for which so few SMIs have been created so far (Turchet, 2019).

Edge computing will be optimal as data pre-processing is decentralised. and then the processed data is sent to the central node; this reduces latency.

To ensure wireless inter-operability, the instruments should conform to standards and protocols that are widely used to enable the exchange and interpretation of shared data. We can use a hardware that supports Bluetooth or Wifi 802.11ac and a software that utilizes protocols such as MIDI and Open Source Control (OSC) (Turchet, 2019).

Message Queuing Telemetry Transport (MQTT) is one of the internet-based communication protocols which is a publish-subscribe lightweight protocol used over the TCP / IP protocol (Naik, 2017). MQTT is open, simple and designed to be easily implemented. MQTT minimizes network bandwidth and device resource requirements when trying to guarantee reliability and delivery. This approach makes the MQTT protocol very suitable for connecting machines to machines (M2M), an important aspect of the concept of Internet of Things (Oklilas, Zulfahmi, Ermatita, & Jaya, 2019). In this protocol the publishers send a message to subscribers via an intermediate server called a broker. Each published message has one topic, which clients can use to subscribe to a broker (Kawaguchi & Bandai, 2020).

(McPherson, Jack, & Moro, 2016a) compares the configurations of various microcontrollers/MIDI protocol and concludes that Bela meets even the most stringent specifications for interactive systems with a latency as low as 1ms and jitter around 20 μ s. Bela is also USB/MIDI compatible.

2.6 Feedback

Real-time feedback to students and teachers about the musical performance will help the students improve their skills. Feedback can be in many forms– visual, audio or physical feedback. In our project, we may use one of the approaches or a combination of feedback approaches.

2.6.1 Visual Feedback

It is important to figure out what elements are to be visualised and to whom. The students should maybe only get a clear picture or a remark on their performance so that it is not too distracting while the feedback given to the teacher can be a little detailed.

In (Ferguson et al., 2005) shown in Figure 2.8 the elements used in the visualisation are harmonic content, noise, pitch and loudness. While the idea is good, it requires

some prior knowledge to understand the visualisation. It is not very clear and it will be difficult for both pre-service teachers and the students to understand the visualisation.



Figure 2.8: The visualisation display prototype developed by (Ferguson et al., 2005)

(Percival et al., 2007) developed a Digital Violin Tutor that provides feedback in the absence of human teachers and it offers different visualisation models as seen in Figure 2.9 - video, piano roll, fingerboard animation and also 3-D animation; although it is unclear if it works for a group setting.



Figure 2.9: Digital Violin Tutor by (Percival et al., 2007)

The Birch Lab system Figure 2.10 developed by (Sain et al., 2019) provides both visual and textual feedback in a group setting to pre-service teachers and students. The pre-service teacher receives feedback on the group performance while the students receive individual feedback.



Figure 2.10: Birch Lab system (Sain et al., 2019)

(Kruijshaar, 2020) focuses solely on visual elements and exclude textual annota-

tions since it maybe be distracting to the pre-service teachers Figure: 2.11. They visualise, pitch, loudness and timing in a group setting.

We can further extend (Kruijshaar, 2020; Sain et al., 2019) to provide visual feedback system that mainly uses visual elements to both the pre-service teachers and the students.



Figure 2.11: Visualisation concepts developed by (Kruijshaar, 2020)

To develop a Graphical User Interface (GUI), there are numerous platforms that can be used . (Sain et al., 2019) use Kivy, a python library for their GUI. (Kruijshaar, 2020) uses Adobe After Effects software and (Nijs, Coussement, Muller, Lesaffre, & Leman, 2010) use OpenGL software. There is also numerous web applications that help with creating a GUI, such as: javascript, HTML and css.

2.6.2 Audio Feedback

Another form of feedback is audio feedback where the teacher can provide some musical cues if they play a beat early/late or if the group tempo is deviating.

(Ferguson, 2006) seen in Figure: 2.12 presents a sonification algorithm that can be used to learn musical instrument skills. It presents a melodic sound representing either a successful or unsuccessful onset. The audio signal from the instrument is analysed using Max/MSP.



Figure 2.12: Interactive sonification by (Ferguson, 2006)

iPalmas is an interactive rhythmic tutor developed by (Jylhä & Erkut, 2011). They use audio-visual feedback for their tool. When a student is off beat, the correct beat of the tutor is given as audio feedback to correct the student. When the student performs well, the tutor sound is further away.



Figure 2.13: Ipalmas (Jylhä et al., 2011)

2.7 Conclusion

In this chapter we have described the various background literature present to answer the research questions needed to develop a low-latency education system that provides visual feedback after analysing the audio elements retrieved from the musical instruments. Based on the choices made from the literature study the next stage of thesis work is to develop a smart instrument with low-latency and determining the beat and rhythm synchronisation between students and also provide audio and visual feedback to the students and pre-service teachers.

This chapter provides the relevant study required for the research.



Figure 2.14: Workflow

Chapter 3

System Design

As mentioned earlier, the novelty of the project is the integration of various components to make a holistic system that detects audio input using sensors, detects beats and the synchronisation between each instrument and the all this is done in real-time.

In order to answer the research questions formulated in Chapter: 1, an experimental setup must be designed and assembled. The setup provides rhythm synchronisation within multiple instruments. It consists of a network of raspberry pi computers on which the sensors attached on the instruments are connected. Wireless communication is used between the devices using MQTT protocol. The system can be divided into two sections, hardware setup which includes the physical components required to build the setup; and the software setup, that helps read sensor input, design beat detection and feedback algorithm and communication within the instruments. Figure: 2.14 displays the workflow of the project.

3.1 Hardware Setup

3.1.1 Microcontroller

For the experimental setup, 4 Raspberry Pis are used where one acts as the server or broker and the rest are the clients. The sensors are connected to the client PIs and the data received from the sensors are collected and analysed in within the client , the calculated BPMs are sent to the server PI where the synchronisation between the clients are calculated. Raspberry Pi was chosen for the experiment for its low-latency performance compared to Arduino Uno (McPherson, Jack, & Moro, 2016b).



Figure 3.1: Raspberry Pi 4 model B

3.1.2 Sensors

The instrument used for the project is percussion based (Drums, bongos,tabla etc). To be able to read data from these instruments, the sensor most apt is a piezo sensor. It provides the electrical voltage proportional to the amount of pressure applied on the membrane. The sensor is place under the membrane of the instrument. Hence when the instrument is struck, the pressure is calculated and its equivalent voltage is sent as output.



Figure 3.2: Piezo Sensor

Raspberry Pi can only read digital inputs, since the piezo sensor used for the experiment is analog, an analog to digital converter is required to read the sensor data. MCP3008 fig: 3.3 is an 8-Channel 10-bit ADC IC, it can measure 8 different analog voltage with a resolution of 10-bit. It measures the value of analog voltage from 0-1023 and sends the value to the Raspberry PI through Serial Peripheral Interface (SPI) communication.



Figure 3.3: MCP3008 ADC from Microchip

3.1.3 Full Hardware Setup

The client device is constructed using Raspberry Pi, piezo sensor and MCP3008 ADC. The output of the piezo sensor can spike upto \pm 30V which is beyond the 5V power supply provided to the Raspberry Pi⁻¹. In order to clamp down the piezo sensor output, a zener diode is used. The sensor is then connected connected to the ADC through a resistor of 1M Ω and a zener diode 3.3V. The schematic diagram of the device can be seen in Figure 3.4. The same connection is done to the other client devices. The full hardware setup can be seen in Figure 3.5. The master pi has no hardware connections.

¹https://raspberrypi.stackexchange.com/questions/103868/piezo-sensor-to-pick-up -acoustic-instrument-signal-using-rpi-and-adc



Figure 3.4: Circuit design for the client pi



Figure 3.5: Hardware Setup

3.2 Software Setup

The experiment is conducted using Raspberry Pi. Most of the programming is in Python language. There are numerous python libraries that are available to the public.

3.2.1 Reading Sensor Data

Once the wiring between the Raspberry Pi and the ADC is complete, to access the GPIO pins of the PI, a python library called Adafruit_GPIO is installed. Here also mention the pins corresponding to the clock, MOSI and MISO in the program to configure the SPI communication.

Next to access the sensor data from the ADC, another python library called Adafruit_MCP3008 is installed. This library has the required functions to read the adc data.

3.2.2 MQTT

The MQTT protocol was explained in Chapter: 2. For the experiment the Master Pi acts as the Broker and the other Pis are the clients. The calculated BPMs from the clients are sent to the broker through a publish command. The broker receives the message payload from the clients through a subscribe command. To reduce latency different topics for different clients are created which will reduce the load on number of subscribers per topic.

3.2.3 Beat Detection

For beat detection, through the literature review process various techniques were found which relied on machine learning. However for implementation a rather straight forward method is incorporated. This was done to reduce amount of processing power required. It also reduces the processing time which in turn further reduces latency.

To check for synchronisation, beat is detected based on the onset on the drums. These beats are then converted to standard tempo units such as Beats Per Minute (BPM) based on Inter Beat Interval (IBI). The complex part lies in the calculation of BPMs. In music, a note played is not always once per beat. So in order to get the accurate BPM, peak irregularities need to be processed. Figure: 3.6 describes the beat detection process.



Figure 3.6: Flowchart of Beat Detection

3.2.4 Synchronisation

To begin, it is assumed that the person playing the first instrument is playing in the correct rhythm. The other players need to be in sync with the first instrument. Therefore, to check if the other two instruments are in sync, the following calculations are carried out.

$$Sync_2 = BPM_1 - BPM_2$$
$$Sync_3 = BPM_1 - BPM_3$$

When the synchronisation values are close to 0, it indicates that the instruments are *in sync* as seen in Figure: 3.7. The greater the synchronisation value the more out of sync the players are from the first instrument.



Figure 3.7: In Sync



Figure 3.8: Out of Sync

3.2.5 Feedback

To provide visual feedback, a real-time graph is used. The real-time graph is plotted with the help of a python library called *matplotlib*. Within the library, the *animation* function is used to plot in the BPMs of the individual instruments on the graph in real-time. The BPM is recalculated every time a beat occurs and the graph is refreshed. The y-axis represents the beats-per-minute and the x-axis represents the time. A legend is placed on the graph to help the participants identify the line that indicates their beat. The legend does not have a fixed location on the graph, it moves around based on the lines plotted to avoid an overlap and ensure the participants can visualise the graph clearly.

For audio feedback, the participants are provided a non rhythmic tone through headphones to indicate the level of their synchronisation. The volume of the audio increases or decreases based on the level of synchronisation. When the synchronisation value between the participants is nearing 0, i.e. "in-sync", the volume increases. When the synchronisation between the participants decreases, the volume of the audio sample decreases in levels of 5. This method is derived from (Ferguson, 2006), where the loudness level of the sound is converted to integers analogous to the familiar *Crescendo*² and *Decrescendo*³.

²Increase in loudness

³Decrease in loudness

Chapter 4

Evaluation

4.1 Experiment Design

An integrated system was developed that can be used in a classroom. To check the effectiveness of the network communication, the beat tracking algorithm and the audio-visual feedback module, various tests are conducted. The results of the tests are then discussed.

4.2 Effectiveness of the Network Communication

Network delay is calculated in order to assess its effect on the performance of the system. There are two integral parts of the program were the network latency is calculated. First is when the payload information is received by the subscribe module and the second is when the data is visualised. The first step to calculate the latency is to ensure that the clients and the subscriber/broker Pis are all time synchronized. For this, the real time clock is configured to have the same value in all the devices.

Shell	Shell
BPM:111.64867882396726 IBI:505 ms Time: 1630943918229 ms	BPM 1: 112 [Timestamp: 1030943917729 ms BPM 1: 111 [Timestamp: 1630943918234 ms
BPM:108.14708002883923 IBI:709 ms Time: 1630943918938 ms	BPM 1: 108 Timestamp: 1630943918942 ms
BPM:100.0032044203744 [161.003 ms [11me: 1030943919541 ms BPM:102.65183917878528 IBI:728 ms Time: 1630943920269 ms	BPM 1: 106 Timestamp: 1630943919546 ms BPM 1: 102 Timestamp: 1630943920273 ms
BPM:101.14632501685772 IBI:557 ms Time: 1630943920827 ms	BPM 1: 101 Timestamp: 1630943920831 ms
BPM:102.6869/586856068 IBI:488 ms IIme: 1630943921314 ms BPM:105.3740779768177 IBI:472 ms Time: 1630943921786 ms	BPM 1: 102 Timestamp: 1630943921318 ms BPM 1: 105 Timestamp: 1630943921792 ms
BPM:105.95090941197246 IBI:506 ms Time: 1630943922292 ms	BPM 1: 105 Timestamp: 1630943922297 ms

Figure 4.1: Latency check

Then a timestamp is taken right before publishing a payload. At the subscriber, again a timestamp is taken right after it receives the payload as seen in Figure: 4.1. The

difference between the two timestamps is the time taken for the data packet to reach the subscriber from the client - *communication latency*. Then another timestamp is placed right after the visualisation module. The difference between the timestamp at the publisher to the timestamp after the visualisation is the time taken between the playing of the instrument to the moment it is visualised - *visualisation latency*.

To calculate the distribution of the communication latency, a sample of size 100 was taken. It consists of time difference between the publisher and the subscriber. The frequency of the values was calculated and plotted, the result of which can be seen in Figure: 4.2. Similarly Figure: 4.3 displays the latency distribution observed during visualisation.



Figure 4.2: Distribution of Communication Latency between publisher and subscriber



Figure 4.3: Distribution of Latency observed right after visualisation

From Figure: 4.2, it can be observed that the communication between client to master PI has a latency of average 3 ms. And from Figure: 4.3, the latency between the publisher to the payload visualisation is an average of 29 ms.

In the future, this can be fixed by calibrating the time lost in the communication within the program.

4.3 Effectiveness of Beat Detection Algorithm

This test is performed to check if the beat detection algorithm is able to calculate the BPM value of the music played by the participants. It also checks if all the instruments calculate the same values for the music played.

To check the effectiveness of the beat detection algorithm, a metronome of 100 bpm is played. All three participants are then instructed to match the instruments' beats to that of the metronome. As can be seen in Figure: 4.4, it indicates that the participants were able to follow the metronome and also that the beat detection algorithm displays the right BPM value as that of the metronome. It was also observed that the participants were in synchronisation with the metronome.



Figure 4.4: BPM check

4.4 Effectiveness of Feedback

In order to check the effectiveness of the system's audio-visual feedback on rhythmic learning, an user evaluation is carried out. For this purpose, an experiment is conducted using the system. The experiment contains 3 instruments connected to piezo sensors. The participants play the instrument and the individual BPMs are calculated and are sent to the broker PI. Here a live graph displays the real time BPMs of all the instruments. Participants see the live graph and their synchronisation values are measured with the help of the audio-visual feedback. Their performance is observed followed by an in-depth interview to get their input on the performance of the feedback module.

4.4.1 First Iteration

A preliminary test was conducted with only two participants. The test is to check the feasibility of the system with regards to collaboration of two instruments. The participants were asked to play the instruments. This is to check if the audio-visual feedback worked as it should with two or more instruments were connected in the network. The participants were to simply tap the sensors on their respective instruments in varying bpms and as they were playing. During this test, an issue was raised with regards to the visual and audio feedback. As seen in Figure: 4.5, the visualisation was not convenient as the autoscale option was ON, which resulted in the graph moving up and down.



To overcome this issue, a y-axis limit was set ranging from 0-200 BPM.

Figure 4.5: Auto-scale "ON"



Figure 4.6: Auto-scale "OFF"

For the audio feedback, a classical piece with varying loudness in itself was used as the audio sample. However, it was observed that this made it difficult for the participants to understand the audio feedback and to synchronise with one another since the loudness of the audio feedback varied even when the participants' synchronisation value was constant. To address this point, a non-rhythmic tone was chosen as it would make it easier for the participants to understand the audio feedback.

4.4.2 User Test

Three participants were chosen to check the effectiveness of the feedback module. The participants were interviewed and their input on the feedback module was noted.

The participants were seated more than 1.5 meters from each other and were given individual headphones. The visual feedback was displayed in a monitor connected to the server PI and the audio feedback was sent to the headphones. The participants could see each other and the volume of the audio sample was set not too loud so that they could also hear the other participants play. But since the audio sample is a non-rhythmic tone, there was clear distinction between the rhythm played by the participants and the audio sample. Initially the three participants where asked

to play any rhythm to warm up to the system. Figure: 4.7 shows the few seconds of the warm up session. Then participants 2 and 3 are asked to synchronise to participant 1 based on the audio-visual feedback provided. The end result can be seen in Figure: 4.8 were the participants were able to synchronise with each other. The user feedback was conducted through an interview where the participants were asked to provide their thoughts on the audio and the visual feedback separately and together. The result of this feedback session was noted and is provided in the next subsection.



Figure 4.7: Warm up

4.4.3 User Feedback

After the experiment was conducted, the users were interviewed regarding their experience of the system. They were asked to note the strong and weak points as well as any other points of interest regarding the audio-visual feedback module.

Audio feedback

The participants stated that the audio feedback was helpful in observing whether they were in sync or not. However it was not as helpful to figure out exactly how



Figure 4.8: Perfect Sync

off-sync they were. Therefore it was useful as a binary or ternary indicator, helping the participants decipher whether they were in perfect sync, some sync or no sync. For eg, when the volume of the feedback is 25% the original volume, it was not as perceptively different from 50%.

With the available equipment, the audio from the instrument was audible in addition to the audio feedback. This caused the participants to unconsciously shift their focus slightly to the audio from the other instrument. This resulted in mixed responses. While some participants found it more helpful to listen to the actual beat from other participants, for another participant it created confusion as they were unable to decipher which of the other beats needed to be followed. However, the audio feedback combined with the instrument audio was enough to get in synchronisation with other players without the aid of visual feedback. When the sound from other instruments were soft/unclear, the audio feedback proved its effectiveness in attaining synchronisation eventually with the other participants.

Visual feedback

The participants found the live graph easy to comprehend with the legends and the BPM indicator. However, one participant found the visual feedback to be affecting

them in a negative way wherein their focus was shifted from playing the instrument to looking at the live graph. This threw them off-sync.

Although, eventually they were all able to synchronise together, because of the drawback due to focus shift from audio to visual, the time it took to reach synchronisation was longer than it would have without any visual feedback. That participant found the absence of visual feedback to help with the synchronisation. This is theoretically mentioned in (Ferguson, 2006), where the author states that asking a musician to concentrate on both a visual source to identify a auditory problem may disconnect the perception of the two.

Just like in the case of audio feedback, the participants found the visual feedback more effective when they had no direct line of sight with the other participants. And visual feedback was much more effective in addition to audio feedback.

Chapter 5

Conclusion & Discussion

This research aimed to provide a base for developing integrated systems for teaching music in classrooms. The integrated system developed during the project has shown potential for further research and improvement in future works.

In order to assess the success of the project, it is necessary to discuss the evaluation results in relation to the research question from Chapter: 1. The constructive questions were answered in Chapters: 3 and 4, whereas the empirical questions are discussed in the below section.

1. Does the effectiveness of the system from previous section affect the possibility of follow up?

Based on the experiment performed, it was noted that although the interaction system is not perfect, it allows participants to synchronise with each other. It is able to achieve the objective that it was set out to do. This system can then be used as a base for future work. By incorporating better feedback systems, or improving the current network model.

2. Is the system suitable for real-time use or offline(data collection for the educator)?

The experiment was conducted in real-time and it showed promising results. The participants were able to visualise real-time feedback as well as listen to the change in the audio volume. An additional module can be programmed where the synchronised data can be collected for offline purposes. For eg, a script can collect the data acquired from each instrument and then run it to display their synchronisation.

3. Is the system flexible to incorporate other instruments other than percussion?

Currently the system uses piezo sensors for analysing the beats. When other instruments are incorporated such as a violin, they do not use piezo energy for beats. So a different set of sensors are required to read the audio from the instruments and use that for analysing the beats. The current beat detection algorithm will not be able to help in this scenario, hence a different technique must be used with regards to beat detection. But the program in the server can be used as such since it collects the BPM from the clients.

The system developed can be the next step for rhythmic learning. It can be improved upon in a few ways. Currently, the system uses four raspberry pis. This can be reduced to one main pi which will act as a broker and the others can be a smaller processing board such as nodeMCU so the system will be compact and the easier to implement the sensors.

Different sensors can be used – instead of a piezo sensor, a microphone can be used to capture the audio signal than the voltage output and processing of the audio can be done using the Bela board for low latency. The existing network delay can be eliminated by compromising wireless communication network and instead use direct wiring between the different instruments. Bela board can be used in the future has it shows promising results in the field of audio processing with low latency.

The beat detection algorithm can be improved by using machine learning techniques although there might be a trade-off with real-time output. But this can be used for offline purpose.

Better audio-visual feedback can be done in the future. The system can take the BPM values played by the instructor and find an audio sample that matches the BPM and feed that as feedback to the students.

Since this is a qualitative research, and the main concern is gathering an in-depth understanding on the performance of the interaction system developed and because of this the sample size was very small. In the future, more tests can be done to see how the system works with more participants.

References

- Atabek, O., & Burak, S. (2020). Pre-school and primary school pre-service teachers' attitudes towards using technology in music education. *Eurasian Journal of Educational Research*, 2020(87), 47–68. Retrieved from www.ejer.com.tr doi: 10.14689/ejer.2020.87.3
- Barrett, M. S., Flynn, L. M., & Welch, G. F. (2018). Music value and participation: An Australian case study of music provision and support in Early Childhood Education. *Research Studies in Music Education*, 40(2), 226–243. Retrieved from https://doi.org/10.1177/1321103X18773098 doi: 10.1177/1321103X18773098
- Böck, S., Davies, M. E., & Knees, P. (2019). Multi-task learning of tempo and beat: Learning one to improve the other. In *Proceedings of the 20th international* society for music information retrieval conference, ismir 2019 (pp. 486–493).
- Böck, S., & Davies, M. E. P. (2020). DECONSTRUCT, ANALYSE, RECONSTRUCT: HOW TO IMPROVE TEMPO, BEAT, AND DOWNBEAT ESTIMATION. IS-MIR, 574-582. Retrieved from https://program.ismir2020.net/static/ final_papers/223.pdf
- Böck, S., Krebs, F., Durand, A., Pöll, S., & Balsyte, R. (2017). ROBOD : a Real-time Online Beat and Offbeat Drummer Department of Computational Perception, Johannes Kepler University Linz, Austria. Retrieved from https://gitlab.cp .jku.at/ROBOD/supplementary/
- Böck, S., Krebs, F., & Widmer, G. (2014). A multi-model approach to beat tracking considering heterogeneous music styles. In *Proceedings of the 15th international society for music information retrieval conference, ismir 2014* (pp. 603–608).
- Böck, S., Krebs, F., & Widmer, G. (2015). Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proceedings of the* 16th international society for music information retrieval conference, ismir 2015 (pp. 625–631).
- Böck, S., & Schedl, M. (2011). Enhanced beat tracking with context-aware neural networks. In *Proceedings of the 14th international conference on digital audio effects, dafx 2011* (pp. 135–140). Retrieved from http://www.music-ir.org/ mirex/wiki/2006:Audio_Beat_Tracking
- Burak, S. (2019, 5). Self-efficacy of pre-school and primary school pre-service teachers in musical ability and music teaching. *International Journal of Music Education*, 37(2), 257–271. Retrieved from http://journals.sagepub.com/ doi/10.1177/0255761419833083 doi: 10.1177/0255761419833083

Cheng, T., Fukayama, S., & Goto, M. (2018). Convolving Gaussian kernels for rnn-

based beat tracking. *European Signal Processing Conference*, *2018-Septe*, 1905–1909. doi: 10.23919/EUSIPCO.2018.8553310

- Clayton, M. (2012). What is Entrainment? Definition and applications in musical research. *Empirical Musicology Review*, 7(1-2), 49–56. doi: 10.18061/1811/ 52979
- Davies, M., Plumbley, M. D., Stark, A. M., & Davies, M. E. P. (n.d.). Real-time beat-synchronous analysis of musical audio Automatic General Audio Signal Classification View project Evolutionary Neural Network View project REAL-TIME BEAT-SYNCHRONOUS ANALYSIS OF MUSICAL AUDIO (Tech. Rep.). Retrieved from https://www.researchgate.net/publication/228988158
- Davies, M. E., & Plumbley, M. D. (2007). Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3), 1009–1020. doi: 10.1109/TASL.2006.885257
- Di Giorgi, B., Mauch, M., & Levy, M. (2021). Downbeat Tracking with Tempo-Invariant Convolutional Neural Networks. Retrieved from http://arxiv.org/abs/2102 .02282
- Ellis, D. P. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, *36*(1), 51–60. doi: 10.1080/09298210701653344
- Eyben, F., Böck, S., Schuller, B., & Graves, A. (2010). Universal onset detection with bidirectional long short-term memory neural networks. In *Proceedings of the 11th international society for music information retrieval conference, ismir 2010* (pp. 589–594).
- Ferguson, S. (2006). Learning musical instrument skills through interactive sonification. (January 2006), 384–389.
- Ferguson, S., Moere, A. V., & Cabrera, D. (2005). Seeing sound: Real-time sound visualisation in visual feedback loops used for training musicians. *Proceedings* of the International Conference on Information Visualisation, 2005(January), 97–102. doi: 10.1109/IV.2005.114
- Fuentes, M. (2019). Thèse de doctorat Thèse de doctorat. (Paris VI).
- Fuentes, M., Maia, L. S., Rocamora, M., Biscainho, L. W., Crayencour, H. C., Essid, S., & Bello, J. P. (2019, 11). Tracking beats and microtiming in Afrolatin American music using conditional random fields and deep learning. In *Proceedings of the 20th international society for music information retrieval conference, ismir 2019* (pp. 251–258). Retrieved from https://zenodo.org/ record/3527792 doi: 10.5281/ZENODO.3527792
- Gonzalez Sanchez, V. E., Martin, C. P., Zelechowska, A., Bjerkestrand, K. A. V., Johnson, V., & Jensenius, A. R. (2018, 6). Bela-Based Augmented Acoustic Guitars for Sonic Microinteraction. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 324–327).

Retrieved from https://www.zenodo.org/record/1302599http://www.nime .org/proceedings/2018/nime2018_paper0068.pdf doi: 10.5281/ZENODO .1302599

- Jylhä, A., Ekman, I., Erkut, C., & Tahiroğlu, K. (2011). Design and evaluation of human-computer rhythmic interaction in a tutoring system. *Computer Music Journal*, 35(2), 36–48. Retrieved from http://www.mitpressjournals.org/ doi/pdf/10.1162/COMJ_a_00055 doi: 10.1162/COMJ{_}a{_}00055
- Jylhä, A., & Erkut, C. (2011). Auditory feedback in an interactive rhythmic tutoring system. ACM International Conference Proceeding Series (September), 109– 115. doi: 10.1145/2095667.2095683
- Kawaguchi, R., & Bandai, M. (2020). Edge Based MQTT Broker Architecture for Geographical IoT Applications. *International Conference on Information Networking*, 2020-Janua, 232–235. doi: 10.1109/ICOIN48656.2020.9016528
- Krebs, F., Böck, S., Dorfer, M., & Widmer, G. (2016). Downbeat tracking using beatsynchronous features and recurrent neural networks. *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016*, 129–135.
- Krebs, F., Böck, S., & Widmer, G. (2013). Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th international society for music information retrieval conference, ismir 2013* (pp. 227–232). Retrieved from https://github.com/
- Kruijshaar, J. (2020). Technology supported music education: Visual feedback support for pre-service teachers in guiding a music class.
- McPherson, A., Jack, R., & Moro, G. (2016a). Action-Sound Latency: Are Our Tools Fast Enough? Proceedings of the International Conference on New Interfaces for Musical Expression, 16, 20-25. Retrieved from http://isophonics.net/latency-measurements.%0Ahttp://www .nime.org/proceedings/2016/nime2016_paper0005.pdf
- McPherson, A., Jack, R., & Moro, G. (2016b). Action-Sound Latency: Are Our Tools Fast Enough? Proceedings of the International Conference on New Interfaces for Musical Expression, 16(July 2016), 20-25. Retrieved from http://isophonics.net/latency-measurements.%0Ahttp://www .nime.org/proceedings/2016/nime2016_paper0005.pdf
- Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. 2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings. doi: 10.1109/SysEng.2017.8088251
- Nijs, L., Coussement, P., Muller, C., Lesaffre, M., & Leman, M. (2010). The music paint machine: A multimodal interactive platform to stimulate musical creativity in instrumental practice. *CSEDU 2010 2nd International Conference on*

Computer Supported Education, Proceedings, 1(June 2014), 331–336. doi: 10.5220/0002859103310336

- Oklilas, A. F., Zulfahmi, R., Ermatita, & Jaya, A. P. (2019). Temperature Monitoring System Based on Protocol Message Queue Telemetry Transport (MQTT). *Proceedings - 1st International Conference on Informatics, Multimedia, Cyber and Information System, ICIMCIS 2019*, 61–66. doi: 10.1109/ICIMCIS48181 .2019.8985356
- Oliveira, J. L., Gouyon, F., Martins, L. G., & Reis, L. P. (2010). IBT: A real-time tempo and beat tracking system. In *Proceedings of the 11th international society for music information retrieval conference, ismir 2010* (pp. 291–296). Retrieved from http://www.music-ir.org/mirex/2009/
- Pantaleone, J. (2002, 10). Synchronization of metronomes. American Journal of Physics, 70(10), 992–1000. Retrieved from http://aapt.scitation.org/ doi/10.1119/1.1501118 doi: 10.1119/1.1501118
- Paulus, J., & Klapuri, A. (2009). Drum sound detection in polyphonic music with hidden Markov models. *Eurasip Journal on Audio, Speech, and Music Processing*, 2009. doi: 10.1155/2009/497292
- Peeters, G., & Papadopoulos, H. (2011). Simultaneous Beat and Downbeat-Tracking Using a Probabilistic Framework: Theory and Large-Scale Evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, *19*(6), 1754–1769. doi: 10.1109/TASL.2010.2098869
- Percival, G., Wang, Y., & Tzanetakis, G. (2007). Effective use of multimedia for computer-assisted musical instrument tutoring. *Proceedings of the ACM International Multimedia Conference and Exhibition*(May 2014), 67–76. doi: 10.1145/1290144.1290156
- Rottondi, C., Chafe, C., Allocchio, C., & Sarti, A. (2016). An overview on networked music performance technologies. *IEEE Access*, 4, 8823–8843. doi: 10.1109/ ACCESS.2016.2628440
- Sain, R., Leinweber, S., Mendler, G., & Hast, F. (2019). Birch Lab Music education assistance tool.
- Stark, A. M., Davies, M. E., & Plumbley, M. D. (2009). Real-time beat-synchronous analysis of musical audio. In *Proceedings of the 12th international conference* on digital audio effects, dafx 2009 (pp. 299–304).
- Turchet, L. (2019). Smart Musical Instruments: Vision, Design Principles, and Future Directions. IEEE Access, 7(January), 8944–8963. doi: 10.1109/ ACCESS.2018.2876891
- Turchet, L., McPherson, A., & Barthet, M. (2018). Real-time hit classification in a smart cajón. Frontiers in ICT, 5, 16. Retrieved from www.cajondg.com/ product/cajon-centaur/?lang=en doi: 10.3389/fict.2018.00016

Turchet, L., McPherson, A., & Fischione, C. (2016). Smart instruments: Towards an ecosystem of interoperable devices connecting performers and audiences. SMC 2016 - 13th Sound and Music Computing Conference, Proceedings, 498–505.

Appendix A

Appendix A : Program Codes

A.1 Program at Client PI

```
1
2 import paho.mqtt.client as mqtt
3 import time
4 import datetime
5 import os
6 import Adafruit_GPIO.SPI as SPI
7 import Adafruit_MCP3008
8 import RPi.GPIO as GPIO
9
10 GPIO.setmode(GPIO.BCM)
11 DEBUG = 1
12
13 # Software SPI configuration:
14 CLK = 18
15 MISO = 23
16 MOSI = 24
17 CS = 25
18
19 # Hardware SPI configuration:
20 SPI_PORT = 0
21 SPI_DEVICE = 0
22 mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))
23
_{24} rate = [0] * 10
_{25} amp = 100
_{26} GAIN = 2/3
27 curState = 0
_{28} stateChanged = 0
29
```

```
30 connected = False
31
32 def on_connect(client, userdata, flags, rc):
     if rc == 0:
33
          connected = True
34
          print("Connected")
35
      else:
36
          print("Not Able To Connect")
37
38
39 broker_address = "192.168.2.48"
40
41
42 client = mqtt.Client("P1")
43 client.on_connect = on_connect
44
45 client.connect(broker_address)
46 time.sleep(0.4)
47 client.loop_start()
48 client.subscribe("test")
49
50 def read_pulse():
      firstBeat = True
51
      secondBeat = False
52
      sampleCounter = 0
53
      lastBeatTime = 0
54
      lastTime = int(time.time()*1000)
55
      th = 525
56
      P = 512
57
      T = 512
58
      IBI = 600
59
      BPM = 0
60
      Pulse = False
61
      while True:
62
63
           Signal = mcp.read_adc(0)
64
           curTime = int(time.time()*1000)
65
           sampleCounter += curTime - lastTime
66
           lastTime = curTime
67
           N = sampleCounter - lastBeatTime
68
69
           if Signal > th and Signal > P:
70
               P = Signal
71
72
           if Signal  (IBI/5.0)*3.0 :
73
               if Signal < T :</pre>
74
                 T = Signal
75
76
```

```
if N > 250 :
77
                if (Signal > th) and (Pulse == False) and (N > (IBI
78
      /5.0)*3.0)
                   :
                  Pulse = 1;
79
                  IBI = sampleCounter - lastBeatTime
80
                  lastBeatTime = sampleCounter
81
82
                  if secondBeat :
83
                    secondBeat = 0;
84
                    for i in range(0,10):
85
                      rate[i] = IBI
86
87
                  if firstBeat :
88
                    firstBeat = 0
89
                    secondBeat = 1
90
91
                    continue
92
                  runningTotal = 0;
93
                  for i in range(0,9):
94
                    rate[i] = rate[i+1]
95
                    runningTotal += rate[i]
96
97
                  rate[9] = IBI;
98
                  runningTotal += rate[9]
99
                  runningTotal /= 10;
100
                  BPM = 60000/runningTotal
101
102
103
                  # ct stores current time
104
                  ct = datetime.datetime.now()
105
                  print("BPM:" + str(BPM), "|IBI:" +str(IBI), "ms","|Time:
106
      ",ct)
                  client.publish("some/topic", int(BPM))
107
108
           if Signal 
109
                amp = P - T
110
                th = amp/2 + T
111
                T = th
112
                P = th
113
                Pulse = 0
114
115
           if N > 2500 :
116
                th = 512
117
                T = th
118
                P = th
119
                lastBeatTime = sampleCounter
120
                firstBeat = 0
121
```

```
122 secondBeat = 0
123 print("no beats found")
124
125 time.sleep(0.005)
126
127 read_pulse()
128 client.loop_stop()
```

Listing A.1: Beat Detection program

A.2 Programs in the Server PI

A.2.1 Program for real-time visualisation

```
1 import time
2 import math
3 from collections import deque , defaultdict
4 import matplotlib.animation as animation
5 from matplotlib import pyplot as plt
7 import threading
8 from random import randint
9 from statistics import *
10
 class DataPlot:
11
      def __init__(self, max_entries = 20):
12
          self.axis_x = deque(maxlen=max_entries)
13
          self.axis_y = deque(maxlen=max_entries)
14
          self.axis_y2 = deque(maxlen=max_entries)
15
          self.axis_y3 = deque(maxlen=max_entries)
16
17
          self.max_entries = max_entries
18
19
          self.buf1=deque(maxlen=5)
20
           self.buf2=deque(maxlen=5)
21
          self.buf3=deque(maxlen=5)
22
      def add(self, x, y,y2,y3):
23
24
25
          self.axis_x.append(x)
          self.axis_y.append(y)
26
          self.axis_y2.append(y2)
27
           self.axis_y3.append(y3)
28
29
30 class RealtimePlot:
```

```
def __init__(self, axes):
31
32
          self.axes = axes
33
34
          self.lineplot, = axes.plot([], [], "ro-",label="BPM1")
35
          self.lineplot2, = axes.plot([], [], "go-",label="BPM2")
36
          self.lineplot3, = axes.plot([], [], color = "purple",label="
37
     BPM3")
          axes.legend([self.lineplot,self.lineplot2,self.lineplot3],['
38
     BPM1', 'BPM2', 'BPM3'])
39
      def plot(self, dataPlot):
40
          self.lineplot.set_data(dataPlot.axis_x, dataPlot.axis_y)
41
          self.lineplot2.set_data(dataPlot.axis_x, dataPlot.axis_y2)
42
          self.lineplot3.set_data(dataPlot.axis_x, dataPlot.axis_y3)
43
          self.axes.set_xlim(min(dataPlot.axis_x), max(dataPlot.axis_x))
44
          ymin = min([min(dataPlot.axis_y), min(dataPlot.axis_y2)])-10
45
          ymax = max([max(dataPlot.axis_y), max(dataPlot.axis_y2)])+10
46
          #self.axes.set_ylim(ymin,ymax)
47
          self.axes.set_ylim(0,200)
48
          self.axes.relim();
49
          self.axes.autoscale(False)
50
51
  def main():
52
53
      fig, axes = plt.subplots()
54
      plt.title('Plotting Data')
55
56
57
      data = DataPlot();
58
      dataPlotting= RealtimePlot(axes)
59
60
      try:
61
          count=0
62
          while True:
63
               count += 1
64
               data.add(count, 30 + 1/randint(1,5), 35 + randint(1,5),
65
     40 + 2*randint(1,5))
               dataPlotting.plot(data)
66
67
               plt.pause(0.001)
68
      except KeyboardInterrupt:
69
          print('nnKeyboard exception received. Exiting.')
70
          plt.close()
71
72
          ser.close()
          exit()
73
74
```

75 if __name__ == "__main__": main()

Listing A.2: LivePLots

A.2.2 Master Program

```
1 #Import MQTT packages
2 import paho.mqtt.client as mqttClient
3 import time
4 import datetime
5 import csv
6 #Import Plotting modules
7 import matplotlib.pyplot as plt
8 import matplotlib.animation as animation
9 from matplotlib import style
10 from LivePlots import DataPlot, RealtimePlot
11 #Audio
12 import numpy as np
13 from pygame import mixer
14 mixer.init()
15 mixer.music.load("Super Intelligence.mp3")
16 mixer.music.play()
17 mixer.music.set_volume(1)
18
19 v 1 = 0
20 v^2 = 0
21 v3 = 0
22 \text{ count} = 0
23 sync1 =100
24 sync2 =100
25 volume1 =0
26 \text{ voulme2} = 0
27 #Plotting
28 fig, axes = plt.subplots(figsize=(8,6))
29 plt.title('Synchronisation data',fontsize=25)
30 plt.xlabel("Time",fontsize=18)
31 plt.ylabel("BPM",fontsize=18)
32 data = DataPlot()
33 realdata = RealtimePlot(axes)
34
35 def on_connect(client, userdata, flags, rc):
36
      if rc == 0:
37
38
          print("Connected to broker")
39
```

```
40
                                              #Use global variable
           global Connected
41
           Connected = True
                                              #Signal connection
42
43
      else:
44
45
           print("Connection failed")
46
47
  def on_message(client, userdata, message):
48
      topic = message.topic
49
      global v1,v2,v3,ts1,ts2,ts3,volume1,sync1,sync2
50
      if(topic == "some/bpm1"): #Read BPM from client1
51
           #ts1 = int(time.time()*1000)
52
           #print("Current:", ct1)
53
           v1 = (int(message.payload.decode("utf-8")))
54
55
           #ct1 = datetime.datetime.now()
           #ts1 = int(ct1.timestamp()*1000)
56
57
      elif(topic == "some/bpm2"): #Read BPM from client1
58
           ct2 = datetime.datetime.now()
59
           ts2 = ct2.timestamp()
60
           v2 = (int(message.payload.decode("utf-8")))
61
62
      elif(topic == "some/bpm3"): #Read BPM from client1
63
           ct3 = datetime.datetime.now()
64
           ts3 = ct3.timestamp()
65
           v3 = (int(message.payload.decode("utf-8")))
66
           #print("BPM2:",v2)
67
      if (v1!=0 and v2!=0 and v3!=0):
68
69
           sync1 = v1 - v2
70
           sync2 = v1 - v3
71
72
73
      #Plotting the BPMs
74
      global count
75
      count += 1
76
      data.add(count,v1,v2,v3)
77
      realdata.plot(data)
78
      ctv = datetime.datetime.now()
79
      tsv = int(ctv.timestamp()*1000)
80
      plt.pause(0.001)
81
82
83
84
      #Audio Feedback
      abs_sync1 = abs(sync1)
85
      abs_sync2 = abs(sync2)
86
```

```
smoothness = 5
87
88
       if (abs_sync1>=0 and abs_sync1<=100):</pre>
89
           volper = 100-abs_sync1
90
           volume1 = smoothness * round(volper/smoothness)
91
92
93
       else:
           volume1,volume2 = 0
94
       mixer.music.set_volume(volume1/100)
95
96
       mylist = [tsv]
97
       with open ('data1_v4.csv', 'a') as f:
98
           writer = csv.writer(f)
99
           writer.writerow(mylist)
100
101
102
103
104 Connected = False #global variable for the state of the connection
105
106 broker_address= "192.168.0.104" #Broker address
107 \text{ port} = 1883
                                         #Broker port
108
109 client = mqttClient.Client("Python")
                                                         #create new
      instance
110 #client.username_pw_set(user, password=password)
                                                          #set username and
      password
111 client.on_connect= on_connect
                                                          #attach function to
       callback
112 client.on_message= on_message
                                                          #attach function to
       callback
113 client.connect(broker_address,port,60) #connect
114 client.subscribe("some/#") #subscribe
115 # ani = animation.FuncAnimation(fig,animate,interval=500)
116 # plt.show()
117 client.loop_forever() #then keep listening forever
```

Listing A.3: Subscsribe

Appendix B

Appendix B: Consent Form

Information Brochure

Faculty of EEMCS

Thesis Title: Interactive tool for Rhythmic Synchronisation Researcher: Carolynn Francis <u>c.f.francis@student.utwente.nl</u> Supervisor: Dr.ir. Dennis Reidsma d.reidsma@utwente.nl

Purpose

The purpose of the user study is to evaluate the audio-visual feedback feature of the interactive tool and its ability to help in rhythmic synchronisation with other individuals through an offline interview. The information gained from the interview will be used to evaluate the effectiveness audio-visual feedback feature of the tool.

Procedure

Your participation in the study will take 15-30 mins. In this period, you will be asked to play the integrated education tool along with two other individuals. After playing, a short offline interview will take place where the effectiveness of the audio-visual feedback will be evaluated. The answers will be recorded as written notes.

Participants:

It is a voluntary study, and the participants can choose to withdraw from the study at any time. Participants should be 18 years or older. The data will be stored anonymously and will be accessible only by the researcher and the supervisor.

Contact Person: Any questions or comments about the study may be directed to Carolynn Francis at <u>c.f.francis@student.utwente.nl</u>

51

UNIVERSITY OF TWENTE.

Informed Consent Form

YOU WILL	BE GIVEN A	COPY OF TH	IS INFORMED	CONSENT	FORM
100 11122	DE 017 E1171			CONCLUS	

Please tick the appropriate boxes	Yes	No
Taking part in the study		
I have read and understood the study information dated [DD/MM/YYYY], or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	0	0
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	0	0
I understand that taking part in the study involves taking part in an interview where my answers are recorded in written notes.	0	0
I understand that I have the right to withdraw from the experiment at any time and I can withdraw this consent without the need to give any reason	0	0
Use of the information in the study		
I understand that information I provide will be used for evaluating a section of the thesis project (Audio-visual feedback feature) and will be mentioned in the final report.	0	0
I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the study team.	0	0

Signature	s
-----------	---

Name of participant

Signature

Date

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

Researcher name

Signature

Date

Contact details for further information: Carolynn Francis, c.f.francis@student.utwente.nl

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee of the Faculty of Behavioural, Management and Social Sciences at the University of Twente by <u>ethicscommittee-cis@utwente.nl</u>

UNIVERSITY OF TWENTE.