Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technologies

Diploma Thesis

Rhythm-tracking for guitar players using smart watch device

Artemiy Mikheev, Bc.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Bc. Artemiy Mikheev

Systems Engineering and Informatics
Informatics

Thesis title

Rhythm-tracking for guitar players using smart watch device.

Objectives of thesis

Main objective of the thesis is to develop methods or algorithms to track rhythm for guitar players using a smart watch device.
The following list of partial objectives has been set:
- Design a proper set of experiments for gathering and processing the input data.
- Gather a representative set of input data.
- Analyze the gathered data using appropriate methods such as machine learning.

Methodology

The core of this work's methodology will be practical work on developing the suitable methods or algorithm for achieving the goal posed in the objective. Using the appropriate methods the experimental data set will be collected and processed. Then the series of algorithms will be developed and verified using the processed data. Afterwards, the results of the verification will be interpreted and explicitly represented.

The proposed extent of the thesis

70 pages

Keywords

Machine learning, Neural network, Music, Rhythm, Smartwatch, algorythm

---

Recommended information sources

BEALE, Russel and Terence JACKSON. Neural Computing: An Introduction. NY: Taylor & Francis Group, [1990]. ISBN 9780852742624.

GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad and Alex BUCKLEY. The Java® Language Specification. Oracle America, Inc., [2018]

LERDAHL, Fred and Ray JACKENDOFF. A Generative Theory of Tonal Music. 2 ed., illust., repr. MIT: MIT Press, [1983]. ISBN 9780262621076

LONG, Fred. Java coding guidelines: 75 recommendations for reliable and secure programs. Upper Saddle River, NJ: Addison-Wesley, [2014]. ISBN 9780321933157.

Expected date of thesis defence

2019/20 SS – FEM

The Diploma Thesis Supervisor

Ing. Jan Masner, Ph.D.

Supervising department

Department of Information Technologies

Electronic approval: 26. 8. 2019

Ing. Jiří Vaněk, Ph.D.

Head of department

Electronic approval: 14. 10. 2019

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 06. 04. 2020

Declaration

I declare that I have worked on my diploma thesis titled "Rhythm-tracking for guitar players using smart watch device" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 06.04.2020 ______________________________

Acknowledgement

I would like to thank my supervisor for his ideas, assistance and eager cooperation while supervising my work on this thesis. I also address my gratefulness to my family, friends and fiancée, who supported me throughout this whole period, my fellow musicians and programmers, who consulted me on their corresponding matters, and specifically the researcher friends from other facilities who generously shared their personal literature for my review and research despite the quarantine regime.

Rhythm-tracking for guitar players using smart watch device

Abstract

The thesis is dedicated to research that may enable development of a rhythm-tracker for guitar players. As the possible application is intended to be delivered onto wearable platforms, Android Watch device was used during the research. For this intent, sensor data were collected from the smart watch sensors (namely 6-axis MEMS, magnetometer and microphone) during the specifically designed set of experiments, targeted at covering as much of potential inputs variation as possible. The experimental data were collected with the help of physical equipment set, supported by auxiliary software, e.g. SensorCap application. Initial set of experiments proved to result in datasets which were insufficiently exact to be interpreted, therefore additional validation tool in form of synchronized slow-motion camera footage was introduced to align the data timestamps with actual physical events properly. Additionally, certain types of events have been discovered to produce unintelligible data due to high level of noise captured by smart watch sensors. Second set of experiments, supported by validation timestamps, was processed through the developed pipeline with minor variations, later the most suitable criterion was detected among the ones suggested by the sensor arsenal and the variety of derived values. The correlation has been discovered between validation set pulses and MEMS data from two specific axes, so this may be a foundation for the actual end-user-friendly solution.

Keywords: smartwatch, rhythm, IoT, sensor, data analysis, application, algorithm, music, tracker, tracking, guitar

Sledování rytmu za hry na kytaru s použitím chytrých hodinek

Abstrakt

Cílem teto práce je nalezení nástrojů, umožňujících vývoj aplikace pro sledování rytmu za hry na kytaru. Z toho důvodu že ten výzkum je zaměřen na vývoj pro nositelné chytré zařízení, byly v experimentech použité chytré hodinky se systémem Android Wear. K tomuto účelu byla posbíraná data ze senzorů chytrých hodinek (specificky 6-axis MEMS, magnetometru and mikrofonu) během sady experimentů, zaměřených na porovnání co nejvíce možných vstupů. Ke sbírání experimentálních dat byla použita soustava vybavení, podpořená kolekcí softwaru, např. aplikaci SensorCap. Výsledem první série experimentů byl soubor nedostatečně interpretovatelných dat v důsledku chybějící reference, což bylo vyřešeno ve druhé sérii experimentů, kde se využil dodatečný validační nástroj ve formě slow-motion záběrů z kamery. Navíc byli identifikované druhy vstupů, způsobující neinterpretovatelná data z důvodu vysoké hladiny šumu nalezené v údajích ze senzorů chytrých hodinek. Data ze druhé sérii experimentu byla zpracovaná podle výsledného algoritmu za minimálních uprav, pote z údajů senzorů a kolekce odvozených veličin bylo porovnáním zvolené nejspolehlivější kritérium. Korelace byla identifikována mezí daty z validačního souboru a výstupy ze dvou specifických os MEMS senzoru, což by mohlo sloužit teoretickým základem k vývoji funkčního řešení.

Klíčová slova: chytré hodinky, rytmus, IoT, internet věcí, senzor, analýza dat, aplikace, algorytmus, hudba, tracker, tracking, sledování, kytara

Table of content

List of figures

List of tables

List of abbreviations

2D	Two-dimensional
3D	Three-dimensional
A+G	Accelerometer and gyroscope
API	Application programming interface
BAN	Body-area network
BLE	Bluetooth low energy
BPM	Beats per minute
CPU	Central processing unit
CSV	Comma-separated value
DCNN	Deep convolution neural network
DSP	Digital signal processing
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of things
LTE	Long-term evolution (telemetry protocol)
MEMS	Microelectromechanical system
MIDI	Musical instrument digital interface
NTP	Network transfer protocol
PPG	Photoplethysmography
SI	*(fr.)* Système international
SIM	Subscriber identity module (telemetry protocol)
SVM	Support vector machine
Wi-Fi	Wire-free (telemetry protocol)

1. Introduction

Wearable smart devices and similar IoT-periphery have rapidly grown in popularity over the course of the last decade. Even though their hardware is keeping with up-to-date standards, and they often possess a variety of built-in additional functions, the open market of optional applications is still underdeveloped. This affects even the wearables subset which belong to the ecosystem of major platforms, like Android Wear or iWatch, not speaking of independent smart watch systems, like Samsung Tizen, for instance. And one of the most unexplored fields in this respect is applications which can utilize smart watch hardware, software and ergonomics to benefit their user's development, especially in developing potential business-critical skills.

And in this regard, musicians may possibly look forward to these solutions the most. Music is a very popular hobby and a significant share of entertainment business. Moreover, it requires immense effort for the participant to get their skills growing over time, and there are few smart solutions for musicians to use in their advantage. In the current era of open market and broad competition, even the most mainstream music tends to employ the growing complicacy to stand out of the competition, but the higher is the level of the complicacy, the more training it requires – especially for the rhythm section which is responsible for holding these complex pieces of music together. Drummers and keyboard players have their smart solutions pre-installed on their electronic instruments since long ago, whereas no such solution is available for guitarists.

In this sense, creation of a smart watch application beneficial for guitarists is a very appealing niche to conduct a research. The functionality of such application could rely on two core principles: one responsible for gathering and processing the data from the watch sensors of detection the pulses, and the other evaluating the accuracy of the pulses in real time. For the use of this solution performer would have to wear the watch on the wrist of the working hand, as it exercises the most intensive motion compliant with rhythm.

2. Objectives and Methodology

2.1 Objectives

Main objective of the thesis is to develop methods or algorithms to track rhythm for guitar players using a smart watch device.

The following list of partial objectives has been set:

- Design a proper set of experiments for gathering and processing the input data.
- Gather a representative set of input data.
- Analyze the gathered data using appropriate methods such as machine learning.

2.2 Methodology

The core of this work's methodology will be practical work on developing the suitable methods or algorithm for achieving the goal posed in the objective. Using the appropriate methods the experimental data set will be collected and processed. Then the series of algorithms will be developed and verified using the processed data. Afterwards, the results of the verification will be interpreted and explicitly represented.

3. Theoretical Part

3.1 Market research

Music has believed to be an ever-growing market since it has been introduced to humankind. The “rock star”/“rockstar” collocation has been relatively long ago reduced to a term with indirect meaning: according to Urban Dictionary (1), it may be used addressing a person who managed to make a success against the overwhelming opposing forces and despite being surrounded by controversies. In this meaning the term has been recognized by the official media, for instance, in the popular motto “Developers are the new rockstars” (2), or referring to the notorious former CEO of Microsoft, Steve Ballmer (3).

This definition precisely reflects the life of a contemporary musician: in order to achieve success in the field of converting talent into revenue, one must constantly struggle to stand out of enormous number of competitors. According to the year 2012 research of official data on number of music industry employees in US (4), the trend in number of musicians is overall declining, with a severe drop around year 2008 (see Figure 1), while in the same year there was a distinct rise in number of professional music directors and composers, also followed by the beginning of a slight decline, which, leaving out the methodological deviations, could be caused by (but not limited to) these three factors:

- Participants requalification – the situation looks like oversaturation of the labor market by musicians and deficient number of producers and composers created an economic bubble, which resolved in 2009 and is facing its consequences from that moment onwards
- Acquisition and promotion of YouTube platform by Google corporation, resulting in peak marginal popularity of the latest at around 2008 – notice also that the same source mentions the difference between self-proclaimed musicians number, revealed by a survey, against the number of employed and/or affiliated musicians

- Less successful employees driven out of the competition. All these processes being a result of escalated requirement for performing skills, promotion capability, originality or any other outstanding features.

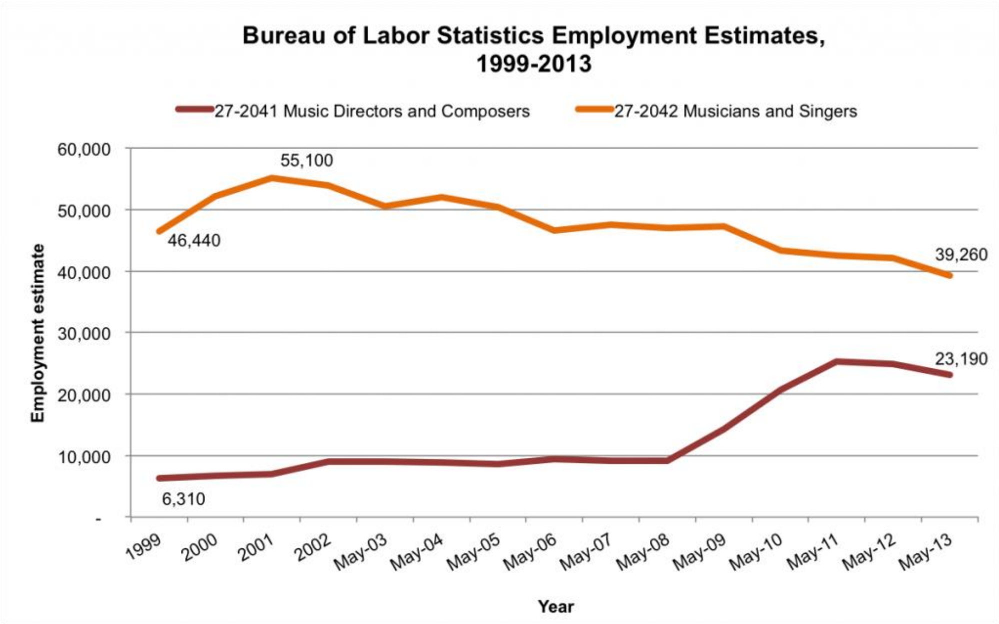

Figure 1 Musical industry employee number dynamics, taken from (4)

As mentioned above, availability of internet, as a ubiquitous information channel, contributed to popularization of certain concepts:

- Musicians are participants and audience of online marketing scene
- Previously unpopular bands from previously isolated regions become sources of national intangibles export (5)
- Popularity of YouTube as an information platform raised the overall degree of music education base of internet audience (6), contributing both to supply and demand for producing more complex musical pieces

- Introduction of crowd-funding platforms empowered the audience of potential musicians (7), allowing the business model which omits the label
- The bands targeting the mass market are more easily adopted by major labels (8), recording companies etc

But all of these concepts require investing enormous time, money and effort investments, none of the abovementioned popularity aspects can be left out while promoting a band or a performer. In order to compete on the musical market, one must possess a decent skillset, constantly augmenting it by all means necessary. In the era of smart devices prosperity, it would be definitely considered a missed opportunity to ignore the benefits which might be brought in by incorporating IoT into the rehearsal processes. Cultural background possessed by me and my supervisor allowed for the judgement that the support of guitar playing practice with modern IoT tools is lacking in a number of ways, therefore the decision has been made to opt into researching a new solution beneficial for guitarists. Guitar in most cases (9) (although arguably) belongs to rhythm-section, i.e. they perform instrumental parts of the composition which set up the overall rhythm of the song, rather than adopting the rhythm already set up by another instrument. This feature makes developing an internal sense of rhythm a crucial skill for a guitarist.

The possible platform for implementing this solution is a subject for a separate discussion, but this thesis will be focused on a solution for a smart watch device. First, exploring this platform poses a business value, as hardware market is saturated with smart watches, fitness bands and other wearables – at year 2013 wearables sales were predicted to hit 126.1 million units by the end of the decade (10), while application development was hindered throughout this whole period.

Second, as the platform-specific research shows, there has been a relatively constant demand for rhythm assistant apps for guitarists according to Google Trends data (11), but among all apps found on Play Market by the same query, none has provided actual real-time tracking – the best which was found was a series of interactive training tutorials; even less were compatible with wearables. The same is true for iOS apps compatible with iWatch. The variety of apps for other wearables is even lower. This makes the results of the current research a world novelty. The last and decisive argument for this choice was that smart watch is designed to wear on the wrist, therefore in the direct vicinity to the tracked process – i.e., playing the guitar.

3.2 Music theory

3.2.1 Rhythm

The first step to develop an algorithm for rhythm recognition is to define a rhythm. W. Sethares' (12) study defines rhythm on five main levels of understanding:

- Cognitive, as complex recurrent patterns happening in time
- Perceptive, as information ordered into certain chunks possessing distinct features
- Mathematical, as patterns and regularities occurring in data and mathematical tools developed for their description
- Processable, as series of signals produced and perceived by somehow complementary physical processes
- Musical, as organized patterns carrying unique cultural background

The statements above may be generalized as a one-sentence definition: rhythm is a series of physical processes each producing a unique feedback detectable by its complement, organized in chunks of regular longevity recurrent over time, each carrying the similar information, which can be described culturally and mathematically.

Rhythm is not a notion exclusive to music or performance arts in general. Etymologically, rhythm is a descendant of a Greek word meaning any recursion or symmetry (13). Currently the rhythmic notions are explored in linguistics (like rhythm of speech or poetic rhythm), medicine (like cardiac rhythm) and generally any other area of expertise dealing with repetitions and patterns, as long as they comply with the definition. Exhibitive arts and architecture are also considered to use this term in the similar meaning, if a certain recurrent pattern is observed in their corresponding compositions.

From the definition above, the main property of rhythm is obvious: it is periodicity. It means that rhythm is a pattern which is repeated from the beginning after every fixed interval, called period. Period may be defined in terms of time (e.g., every 10 seconds), discrete number of pulses (e.g., every 8 pulses), or some other feature defined by composer. Usually these criteria produce the same period, but musically it's more appropriate to hold on to discrete pulses criterion (so called mensural division (14), or beat) – i.e., the situation when the pattern is produced over different interval of time, but containing the same number of pulses holding constant longevity proportions between each other is considered a legitimate rhythmic pattern in music.

T. Garland (15), when addressing this paradox, suggests, that mathematics explains the patterns to observer, while musical theory assists in understanding for the performer. Usually to help performers correlate metrical and chronological definitions of a period, auxiliary notation is introduced: the beat frequency over time is usually denoted by BPM (beats per minute) in the modern scores or tempo marks in Italian (e.g., Moderato is moderate tempo, equal to approx. 108-120 BPM) in more traditional scores.

The second property of rhythm which is included into definition above, is the information carried in each period. It is questionable, whether pitch of notes in the pattern is a critical part of the rhythm: riffs (or rhythmic figures; although some dictionaries (16) consider this word originating from refrain) include the pitch and usually are specific to the song they are first performed in. Although, usually the rhythmic unit is a pattern of note duration (or simply length) relative to other notes in rhythm, and their relative intensity (also called dynamics). There is also a variety of bordering terms, like, for example, harmonic rhythm, which is periodic and include both time and pitch, but is not defined by pulses, rather by the frequency of chord progression in score's harmony. For the purposes of this thesis, rhythm will correspond purely to the relative duration of pulses within score's recurrent patterns.

In the Western tradition, the most common rhythmic unit size is four: this is defined by a so-called time signature (17). Time signature is a fraction, which has its numerator equal to number of steadily organized pulses (called beats) in a basic rhythmic unit (called measure), and its denominator equal to the length of these pulses relative to the universal. The simple time signatures are 2/n and 3/n, where n is a natural power of 2 (i.e., 2, 4, 8, 16…); ¾ in this notation will be pronounced as "three fourths" or "three above/against four". These time signatures have their first beat played loudly (also accented or strong beat), and the others considered unaccented or weak beats.

Complex time signatures, like 4/4 or 7/8, are constructed of simple ones (18) in a way which is made understandable by the composer: this way they have multiple strong beats in a measure, but the first one is always the strongest – e.g., for the 4/4 time signature the distribution of accent will be the following: $1^{st} > 3^{rd} >> 2^{nd} = 4^{th}$, as it is composed of two stacked 2/4 measures. In case of ambiguity, the composer may provide an additional notation hints on how to perform the rhythm dynamic-wise.

Rhythmic units may fall under one of the following categories (17):

- Metric (steady repetition of equal pulses played in accordance with beat)
- Intrametric (patterns of pulses played in accordance with beat and its division by powers of two, maintaining the intended dynamic)
- Contrametric (patterns of pulses played in accordance with beat and its division by powers of two, violating the intended dynamic, thus creating the syncopation effect – when the weak beat becomes accented and/or strong beat becomes unaccented)
- Extrametric (patterns of pulses played outside beat and its divisions by powers of two; this includes tuplets: division by numbers other than powers of two)

This discussion leads to defining a third feature of the rhythm, called phase. Phase is a pulse which starts the rhythmic pattern, and two neighboring phases delimiting exactly one period of rhythmic pattern. In musical theory phase corresponds to the first beat of the measure, although it doesn't necessarily mean that the songs always begin with the first beat: it's a common practice, especially in the songs, to put some pattern over the weak beat preceding the strong beat, which begins the rhythmic unit; this creates a "tail" of notes called an anacrusis (18). These may be later repeated, and in this way incorporated into the rhythmic unit, or appear exclusively at the beginning of the piece.

So far possible deviations of rhythmic pattern include:

- contraction/dilatation of a period, either chronological or metrical,
- deletion/addition of the elements (sometimes as a result of changes in metrical period),
- qualitative variation of the element(s) – including contraction/dilatation of exact pulses and changing the pulses' corresponding positions (12).

There is one more feature of rhythm, not explicitly emphasized by Sethares, but critical for the current thesis: it is a pair of physical processes, one producing a distinct signal at performer's will according to the cultural information stored in the musical score, and other translating it for the observer back into cultural information. Usually producing a single note involves numerous physical processes happening at the same time, which, due to the nature of human echoic memory, are processed into one sound (19).

The choice of physical processes (and a manner by which they are performed) initiating a pulse may define a shape of the waveform corresponding to this pulse. E.g., if a sound is initiated by a free damped oscillator (like a plucked guitar string), the waveform will approach the sinusoidal form. If the sound is initiated by a driven oscillator (like a violin string drawn over by a bow), the waveform is of different shape (12). Also force(s) being applied to an instrument by a performer, combined with a set of instrument and ambient damping factors define attack and release of the wave. To generalize, attack is a delay between the force being applied and a soundwave reaching its first peak, and release is a delay between the moment when driving force ceases and the moment when the sound vanishes completely. This general definition is specified more precisely for every separate instrument and playing style.

Each note, instead of being one vibration (called a pure tone), is a sum of harmonic vibrations of a pure tone, which defines the note we hear (20), and a series of vibrations that characterize the timbre (or a way the note is played: instrument, technique etc.). A pure tone defining the pitch is called a fundamental, while the added vibrations are called overtones. Overtones have the following properties:

- the same phase shift as a fundamental,
- frequencies equal to natural multiples of fundamental frequency – i.e., if fundamental frequency is F = 440 Hz, then frequencies F2, F3, F4 of 1st, 2nd and 3rd overtones respectively are 880 Hz, 1320 Hz and 1760 Hz,
- relative intensities of overtones vary from timbre to timbre but increase proportionally as the fundamental amplitude increases.

However, when designing an algorithm for a smart device, one addresses a computer as an observer, and computers have their own way of processing data. Every note and every chord is produced by a series of harmonic vibrations, which also have periodic properties, as the rhythm does, which makes the pitch an analogy of rhythm (21), especially for the computers, therefore a proper process should be selected to extract the data from, and a proper mathematical apparatus should be employed to correctly interpret these data.

The main property which allows to distinguish between rhythmic periodicity and tonal periodicity (12): on the lower side of frequency spectrum there is area of rhythm pulses (approximately between 0.1 and 3-5 Hz, these and following boundaries vary for different instruments), while high frequencies (above 10 Hz up to human perception limit) correspond to the tonal oscillations. The frequency band in between these two is questionable, and periodical sounds falling under this category (i.e. too frequent to be considered rhythmic, but too slow to be perceived as a pitch) are called fluttering – this category includes, for example, drum rolls.

Besides that, rhythm perception may differ for various people, and this difference does not always comply with the exact set of explicit factors: neither internal (like overall level of person's musical literacy), nor external (like quasi-rhythms of ambient noises), even if the pattern the observers are exposed to is the same. This effect of so-called subjective rhythm has been observed and described since last century (22), which increases the complicacy of rhythm perception by the machines, void of abstract thinking ability.

For the mapping of pitch, various spectral analysis tools, like Fourier transformation (19), are used producing a full picture of the sound spectrum at any given moment of time. However, by the definition, the rhythm is a notion which develops over time. This makes the "frozen" spectra, like output of the Fourier transformation, not suitable to represent the human perspective on rhythm in a way which could be perceivable by computers. In order to make a proper choice, all available variants of inputs should be discussed.

3.2.2 Guitar

Guitar is a string musical instrument played by exciting the strings and setting them to continuous oscillating motion. This motion establishes a steady wave, which as a reaction of atmosphere will produce sound (23). Guitar comprises neck (long and narrow part of the construction) and body (wide and massive part at the end, where guitar sound is usually amplified), for the reference see Figure 2. Neck is predominantly occupied by fretboard – a series of metal bars, which strings are pressed against to change the played note; the rest of the neck is called head, it contains nuts, regulating the string tension. Number of strings is usually six but depending on construction and genre this number may vary.

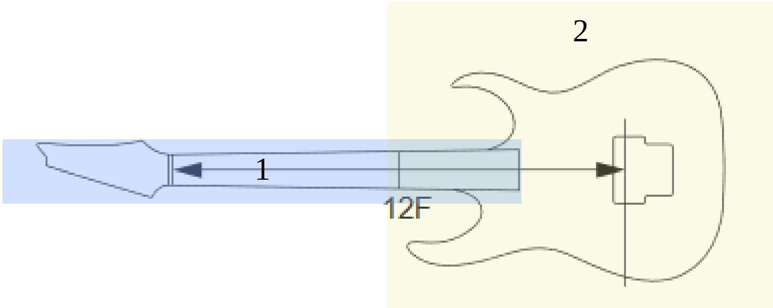

Figure 2 Guitar visualization, taken from (24). (1) – neck, (2) – body; 12F denotes a position of 12th fret on the neck

Depending on how sound is amplified, guitars are categorized as acoustic (amplification by resonance inside the hollow body) and electric (25) (amplification by translating into electric signal, amplified externally and played through the speaker).

Traditional guitar playing techniques can be divided into two categories: strumming and plucking (26). Strumming involves excitation of multiple strings at once by hitting them sideways on the outer surface, and plucking is usually done by picking up the single strings. An isolated motion of strumming is called stroke – it can be performed downwards or upwards, called downstroke or upstroke correspondingly, repeated pattern of upstrokes and downstrokes alike is called altering stroke. One of the most widely used plucking technique is called arpeggio, or arpeggiation, and is based on plucking separate strings within one chord to create rhythmic fills. Either technique may be played with fingers or with a pick (also called plectrum), a small rigid utensil held in hand and used to hit the strings.

Strokes are instantaneous for human perception, but in fact are continuous (27). Detailed view on the stroke extends the definition of attack and release discussed in chapter 3.2.1. Stroke attack is a moment of hitting the first string of the chord, while stroke release is a moment of letting ring the last string. Further on, the terms attack and release will refer solely to these definitions.

3.3 Mathematical apparatus

In order to correctly interpret the data, computer-based solutions must retrieve these data in a valid form, whereas in practice the experimental output of the sensors are mostly distorted, sometimes misleading and always contaminated by some level of background noise. Human eye may draw correct conclusions with some degree of reliability even with this kind of data, but machines require a decent pre-processing. For this purpose, a set of mathematical instruments is introduced.

3.3.1 Normalization

Gathering readings from multiple inequal sources results in heterogeneous datasets, each having different impact onto the overall process. In order to compare these magnitudes and enable to derive additional unitless properties from the existing ones by simple addition and subtraction, they should undergo a process called normalization (28). This process essentially transforms oblique scattered datasets $\{x_n\}$ with ranges $<x_{min,n}; x_{max,n}>$ and means equal to $\overline{x}$ into normally distributed sets $\{x'_n\}$ with similar range, usually $<-1; 1>$ and same mean, usually equal to 0. Target range and mean may vary, depending on the case. There is a variant on this procedure, called adaptive normalization. This happens in real time, as the new values are included into dataset: on each iteration the range and mean (also called normalization constants) are recalculated and retroactively applied to the entirety of a previous set.

3.3.2 Noise reduction

There are numerous solutions for elimination of noise from data, but the simplest, yet one of the most effective, solution is a noise gate (29). The principle of gate application is simple: a number from the data range is selected and labeled a threshold; then all the points of data (or rather their absolute values) get compared to the threshold, and in case threshold is greater, get changed to 0 (or any other selected value in the mid-range of the dataset). Being initially a sound producer term, noise gate in the context of guitar playing usually means post-processing the guitar sound with digital or analogous filters, and the affected value is sound amplitude. However, this instrument may successfully filter out the noise from any other properties with pulse-like behavior. The main point of caution while applying the noise gate is that if the gate gets open, it allows through not only the signal, but also the background noise, which is now added to the signal. This way, in the sets with high level of noise it becomes even more pronounced, because it is allowed through while surrounded by complete silence.

Like normalization, noise gate can also be adaptive (30), i.e., vary its threshold over the course of introducing new data to the set. However, this process is much more complex than the one for normalization, as setting the normalization constants follows the predefined algorithm, which in case of adaptive normalization is simply reiterated, whereas threshold for the noise gate is set up arbitrarily, usually based on the user's personal justification. The simplest way how this can be set up, is to dedicate a short period of silence before recording the data, to collect the static. As this is done, noise gate threshold calibrates automatically to the highest amplitude reached by the static, and when recording starts, threshold stops being adjusted.

The more advanced techniques of automatic noise reduction usually include spectral analysis tools, like Fourier transformation (12), but they are seldom performed in real time, due to the computational delay, and are usually reserved for post-processing.

3.3.3 Spectral analysis

In music-related software spectral analysis, and specifically Fourier transformation, have found a prominent use. As discovered and proven by numerous studies (20) (27), what we perceive as a single note, is always a combination of vibrations, and when several notes are played at once in a chord or multi-track polyphony, the waveforms may look totally unintelligible. In this case spectral analysis tools are used to visualize frequencies of partial vibrations as a spectrum plot. This spectrum is later processed in a manner suitable for each tool specifically.

Fourier transformation principle utilizes the Euler's identity equation, namely its property to relate exponential and goniometric functions (31). When Fourier transformation is applied to the function, an integrable function f(x) of an arbitrary property x is mapped onto function $\hat{f}(\nu)$ of frequency ν through integration through the whole function domain. If the output of Fourier transformed oscillation is to be represented graphically, usually it looks like a plot with series of peaks or bands, corresponding to each partial vibration in the sum. This plot's horizontal axis is usually the frequencies axis – the higher is the frequency, the farther it is to the right. The vertical axis contains some measure of intensity, and in some cases has the same measurement units as original function. In case of pure unobstructed vibrations, these bands are lines of zero width, and their height corresponds to their relative intensity, equal to a hypothetical amplitude of the isolated vibrations with the same characteristics.

3.3.4 General mechanics

Playing the guitar, especially strumming, can be approximated as harmonic oscillations of the picking hand with stability point near the strings (14). In case this is a starting point (i.e., its coordinates are [0;0]), the trajectory of a general point on a picking hand over time while playing may be imagined as a sinusoidal oscillator with zero phase shift, if talking about the vertical position. Acceleration component on a certain axis, which we are able to measure, is mathematically a second derivation of coordinate on the same axis, while the first derivation is velocity. When this is represented as a graph, it's obvious that every time the hand passes the initial point, its vertical velocity is maximal (it is an antinode for velocity oscillation) and its acceleration is zero (node for acceleration oscillation), which means that, theoretically, points in time when acceleration is equal to zero correspond to the points when the hand touches the strings. However, the experimental data are not perfectly harmonic, and seldom comply with some explicitly written formula, in this case numerical integration may be used.

Depending on the purpose, one may choose among the various ways of calculating the numerical integral. The most general and formal way to calculate numerical integral is Riemann sum (32). Riemann definition takes advantage of the fact that definite integral is equal to an area enclosed between x-axis and graph and approximates this area as consisting of rectangular slices. As the width of these slices approaches zero, their sum approximates the actual space under the graph with more precision.

However, Riemann integral is not not always suitable for solutions including, e.g., complex limits of integrated function. In this case, other numerical integrals, like Lebesgue integral, or other formulas derived from Riemann and Lebesgue definitions, are applied. In practice, one of the most recognized numerical integration formulas for oscillating time series is so-called trapezoid rule (33). It expands the idea of Riemann sums by approximating the area as consisting of trapezoidal slices.

Nodes and antinodes of near-harmonic oscillations can also be detected by their derivations. Series of displacement derivations and integrations with respect of time for an ideal or driven oscillator is infinite both ways, alternating between sine and cosine functions each iteration, and having the opposite signs per every group of two functions (34). In a way, this may also be considered a rhythm. But what is beneficial in this theory to be applied in the current project, is that nodes of velocity function can be found at the same timestamps, as they are for the first derivation of acceleration with respect to time, if this assumption holds for the practical data. This could be made use of, because integration is very sensitive to the quality of input data. Would an experimental error happen during measurement, due to which a random constant is added to the data, this constant is to be integrated, resulting in a linear function added to the output of integration. The derivative of constant, however, is zero, so this error will be mitigated.

3.3.5 Vector operations

As this thesis is predominantly dealing with vector properties, basic concepts of vectors and their processing also need to be introduced.

Vector in physics is a property which is defined both by direction and magnitude (35). Translated into mathematical apparatus, within n-dimensional Cartesian coordinates system a vector is denoted as an ordered set of n numbers, each corresponding to n^{th} coordinate of a vector, i.e., in 3-dimensional space a vector is triad of numbers [x, y, z], denoting projections of this vector onto x, y and z axes. A space, within which every point has a certain property vector is called a field. A thoroughly studied example of field is force field: a model of an enclosed spatial system with defined origin and coordinate axes, which has a vector assigned to every point, characterizing a total force affecting this point. Knowing the total force, affecting an object, and its scalar properties, one may derive the other vector (e.g., acceleration from the 2^{nd} Newton's law of motion) or scalar (e.g., kinetic energy) properties of this object.

If vector is represented as a directed straight line segment, beginning at the origin point of corresponding coordinates system and ending at the point with coordinates equal to those of the vector, the length of this segment is equal to the vector's magnitude, or norm, or modulus. Regardless of the term used, this property is calculated from the transformed Pythagoras theorem, by a formula called Euclidian distance (36). Physical meaning of modulus differs among the properties, but is usually understood intuitively: e.g., modulus of force vector represents, how much force is applied to the given point in space.

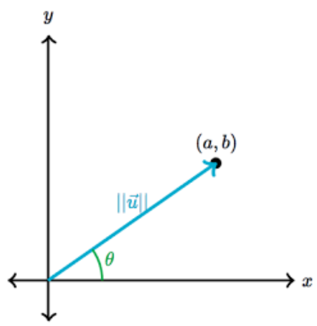

Figure 3 Vector u with coordinates (a;b), taken from (37)

3.4 Measurement principle

3.4.1 Accelerometer and gyroscope

Accelerometer is a device capable of measuring the proper acceleration of an observed object. Core principle of any accelerometer is gravitational forces affecting the sensitive element in a translational manner (38). Every accelerometer may have its own reference state, at which it displays zero acceleration, and operate in different units, but conventionally accelerometers are built with respect of Earth's gravity: accelerometer positioned still on the Earth's surface will read ≈ 9,81 m/s^2 straight upwards (as it will be exposed to the surface reaction equal to standard free fall acceleration, but directed oppositely), and readings of an accelerometer falling freely towards the Earth's gravity center will equal to zero.

The simplest accelerometer possible to build is a wooden plank with equidistant markings on it, an aligned metal spring and a weight attached to it. When dropped or moved forcefully in the direction of the spring, it will contract or elongate, and a weight will move a number of marks up or down, proportional to the acceleration it moves with. This principle of transforming mechanical forces onto a measurable unit space is mimicked by more technologically advanced accelerometers, but the initial terminology is retained. E.g., the constant, which is specific to an exact accelerometer, describing its susceptibility to the mechanical forces during measurement is called a spring constant, regardless of actual springs lacking from the majority of accelerometer builds. Modern commercial accelerometers are based on various effects, transforming mechanical external forces into electric signal (39), which is easily digitalized, stored and processed.

Gyroscopes are similar to the accelerometers in a way that its core functionality is based on projecting mechanical motion into measurable results, mainly electric signals. The main difference lies in the character of this motion: accelerometer measures the effect of translation-inducing force, and the gyroscopes deal with rotation. The main effects enabling gyroscopes to measure the rotation rate of the observed object are Coriolis force (40), created as a reaction of system to rotation-inducing forces, and the law of conservation of angular momentum.

Using an abovementioned hypothetical simple accelerometer enables measuring acceleration in one direction only. To determine the entire behavior of 3D acceleration vector, more complex technologies are to be involved (41). There are piezoelectric-based sensors, high-precision quantum accelerometers, relatively new thermal accelerometers and numerous other variants. Each separate solution has its pros and cons and works within specific range and with different precision, which are dictated by the requirements of the area they are applied in. The same variety holds for gyroscopes. It is also possible to have one sensor measuring both acceleration and rotation rate – such sensors are categorized as 6-axis sensors and find the most frequent application in radio electronics and portable devices engineering due to their area efficiency.

Accelerometers, gyroscopes and 6-axis motion sensors are used in vehicle engineering, in biology areas researching animal behavior, in navigation etc. Smartphones and wearable devices have microelectronic accelerometers built into their circuits alongside other sensors. Such sensor systems are often used in holistic tracking solutions (42).

3.4.2 Magnetometer

Unlike accelerometer and gyroscope, magnetometer is categorized as environment sensor, rather than being motion sensor (43). It means that instead of measuring the dynamic properties of an object, it evaluates properties of its surroundings. Magnetometers generally measure all magnetism-related properties: direction, intensity and/or relative change of magnetic field. Under this definition, compass may also be considered a simple magnetometer, capable of measuring solely the direction of magnetic field.

Most common property to measure by magnetometer is magnetic field intensity. This concept generalizes two closely related vector properties: magnetic flux density (measured in teslas according to SI) and magnetic field strength, measured in amperes per meter. Numerically these properties are different only when measured inside a magnetized material.There is a great variety of technologies magnetometer can be based on. The most notable are, again, MEMS magnetometers (38) (which are subject to further subcategorization) and inductive coil pickups, which are used the most prominently due to ease of their maintenance and manufacturing.

3.5 Equipment specification

3.5.1 Smart watch and other wearables

Smartwatch is a wearable appliance combining design and functionality of a regular wristwatch with more advanced flexible software platform powered by a CPU (44). Modern smartwatch devices usually come with the built-in sensor display and may occur to have to mechanical buttons whatsoever. The main selling point of smart watch is the same as it was for the smartphones during their most pronounced market growth: combination of multiple devices' functionality in one appliance. Majority of smartwatch-exclusive apps take advantage of either being positioned on the wrist (e.g., heart rate meter or hands-free talking and messaging tools for drivers) or being continuously attached to the body (e.g., fitness assistant or GPS tracker). Current generation of smartwatches is advertised as a part of a smart devices ecosystem, capable of communication with smartphones and other devices within BAN (body-area network) by telemetry, or even managing more complex solutions, like smart house, with a touch to one's wrist, further promoting the IoT (internet of things) concept.

Smart wearables' broad range of functions is supported by a variety of preinstalled microelectronic devices: sensors, peripheral inputs and outputs (speakers, microphone, sometimes even cameras), multiple protocols of telemetry (Wi-Fi, Bluetooth). Most of the smartwatches communicate with a device which is more advanced and power-consuming (e.g., smartphone or a PC), simply mediating the interaction between user and this device, serving as a remote access terminal. Although, the most advanced smartwatches have slots for a SIM-card and are able to advertise data under the LTE protocol, enabling them to function as an independent smart device (45).

3.5.2 Mounted sensors overview

According to the manufacturer data, provided on various wearables (44), and specifically Huawei Watch 2, (46), which is an experimental device for this thesis, smartwatch device in general has the following sensors mounted: 6-axis A+G sensor (accelerometer + gyroscope), 3-axis Compass (based on magnetometer), PPG (photoplethysmography) Heart Rate Sensor, Barometer, Capacitive Sensor, Ambient Light Sensor, as well as the Microphone. From all the sensors available, only accelerometer, gyroscope, magnetometer and microphone output data may prove useful for the scope of this thesis.

Luckily, Android Wear devices, regardless of the version, are provided with framework (43) to extract the readings via suitable applications and also exhaustive manual on the sensors' functionality.

Accelerometer and gyroscope readings are processed in compliance with Motion Sensors API (47). They are essentially produced by a single sensor (38), belonging to MEMS (microelectromechanical systems) sensor category, capable of tracking linear acceleration and rotation moment simultaneously. MEMS principles, application and discovered problems will be discussed later. The data exported from accelerometer are measured in m/s^2, while data exported from gyroscope are measured in rad/s.

Magnetometer belongs to the category of Position sensors, and its readings are accessed via similarly titled API (48). Magnetometer reads magnetic field measured in μT (microteslas), or, more precisely, magnetic flux density vector components in each of three cardinal axes. Principles of smartphone and smartwatch mounted magnetometers will be discussed later.

The device is powered by Android Wear 2.0 so there is also a step counter API built into the system and available to be used (47), although its readings are not self-sufficient, but derived from motion sensors.

Microphone is generally not considered to be a sensor, although it is also capable of exporting the readings of certain physical property: namely the amplitude of input sound wave (physical representation of sound volume). Microphone in apps is addressed initiating an instance of MediaRecorder class, and its readings may be accessed in accordance with MediaRecorder private methods (49).

3.5.2.1 MEMS motion sensor

MEMS motion sensor installed into smartwatch provides two sorts of data on three axes each: translational acceleration (accelerometer data) and rate of rotation (gyroscope/gyrometer data), which has this kind of devices to classify as 6-axis MEMS A+G (see Figure 4). This kind of device is characterized by balanced price related to other technologies, ease of installment into the circuit boards (creating an essential part of wearable hardware) and are capable of self-testing to adjust the measurements precision (50).

The principle on which MEMS functionality is based is semiconductor polysilicon-on-insulation wafer-like structures mimicking natural macro-sensors (38), like tuning forks and springs. Accelerometers are based on capacitive transduction properties, so that external disturbance evokes translational harmonic oscillations inside the device, changing the subsystem’s capacity, while gyroscopes do so on rotational axes, having Coriolis force create displacement between inequal plates of capacitor.

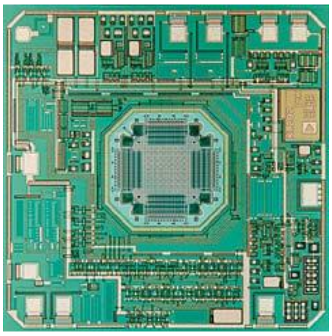

Figure 4 Circuit containing MEMS 6-axis (in the middle), taken from (38)

As IEEE standard for sensor performance states (39), among the specifications defining performance of an accelerometer are bandwidth, noise floor, cross-axis sensitivity, drift, linearity, dynamic range, shock survivability, and power consumption. For every device these characteristics are measured under normal conditions, as well as random selection of each sample undergoes a stress test under edge case states of temperature, voltage and other factors affecting their productivity. Unfortunately, exact data on Huawei Watch 2 sensors are confidential, so the parameters crucial for the current thesis will be identified experimentally.

Android developer documentation specifies (47) the measurement units on coordinate system used in the readings provided by the built-in accelerometer. Accelerometer and gyroscope both export their outputs as components of three-dimensional vectors in orthogonal Cartesian system, where x and y axes lie within a plane tangent to the surface of the screen, x being directed sideward on the screen, y directed upward and z containing a normal (perpendicular) to this plane. The measurement units of accelerometer and gyroscope comply with SI, i.e. m/s^2 and rad/s respectively.

MEMS technology sensors, due to their minuscule scale, are known to have their readings affected by Johnson noise (51), unlike their larger analogs. This noise is of Brownian nature and happens on the scale of agitated germs motion. This must be taken into consideration when evaluating the output of MEMS sensors.

3.5.2.2 Magnetometer

Due to the circuit scale and production affordability, magnetometers inside the wearable devices are also produced with use of MEMS architecture. Data provided by Huawei are again classified, so it is unknown, which specific technology was used to base the magnetometer on. The two known technologies of smart device compatible magnetometers are either based on building the sensor around a magnetoresistive element, which is considered an impractical solution from the perspective of power usage (52), or based on Hall voltage (potential difference created by Lorentz force affecting the electrons inside a conductor under direct current power placed into the homogenous magnetic field). The cited article discussing differences between these two approaches was released in 2012, while Huawei Watch 2 was launched in mid-2015, which makes Hall effect magnetometer a more likely option between the two. But use of the others, less conventional, methods to base the functionality of magnetometer on, is also possible.

IEEE standard for sensor performance (39) defines the same performance criteria for magnetometer, as for accelerometer (adjusted by the measured property and reference constants for zeroing), with addition of acquisition delay, alias “measurement time”, which suggests that the measurement of magnetic field is dynamic, rather than being instantaneous. This may interfere with attempts to collect high precision readings from the device, if this measurement time is too long.

Official manufacturer data refer to this magnetometer as 3-axis, which contributes to the fact that the output data are safe from possible aberrations caused by tilting the device, as expected from outdated 2-axis magnetometers (52), also z-axis data may be proven the most useful, as this axis is directed perpendicularly to a plane enclosed by guitar strings (i.e. both to the string itself and the direction of its vibration), if the hand is in default position. This way magnetometer z-axis is aligned with the component of a magnetic field vector which may happen as a reaction of the string to and induced field, if any occurs within measurable interval.

Android Developers online manual (48) specifies the same axes as for motion sensors, and the output data measured in μT (microtesla), therefore it's obvious, that this type of MEMS magnetometers measures magnetic flux density.

The initial idea is to detect the string vibration the same way as guitar pickups do on actual guitars (53): the pickup itself is an electromagnet, affecting the conductive metal string with the magnetic field it produces, then measuring the change in field created by reaction of the vibrating string and its induced magnetic field of opposite direction.

Application of this theory is complicated by the fact that guitar pickups rely on the technology (54) different from the one used in magnetometers. Albeit both are based on an electromagnet, they have different setup and positioning. Induction pickup might as well be more powerful, than the wearable device's magnetometer. According to the studies that have been conducted on this subject (55), magnetometer sensitivity is always a concern, when developing a solution for this kind of devices.

3.5.2.3 Barometer

There is also a piezoelectric element in the mounted barometer sensor (see Figure 5), which may approximate the application of piezoelectric component-based solutions of electroacoustic instruments. However, barometer readings will be discarded from the discussion for the following reasons:

- Readings resolution of 0.03 hPa at best doesn't enable this sensor to trigger from string vibrations (56)
- In case of sufficient sensitivity, application of a piezoelectric pickup requires it to have a fixed position with respect to the strings; the sensor mounted on a wrist, which is actively playing, will capture the waving motion prior to the string-emitted signal

- Being suited for long-term environmental measurements, smart watch barometers may have longer acquisition time, potentially being an obstacle on the way of capturing the string vibrations of instantaneous nature

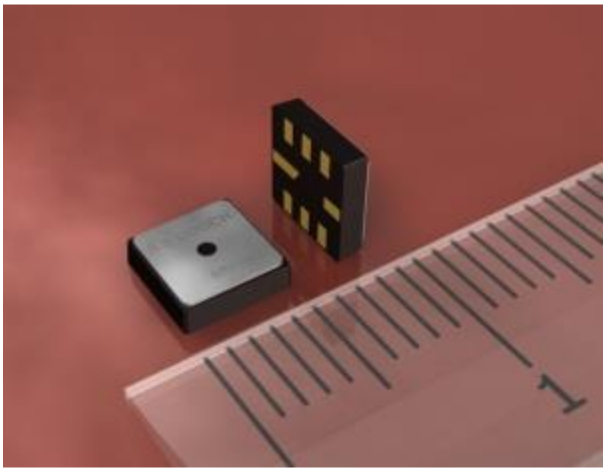

Figure 5 MEMS barometer visualization, taken from (56)

3.6 Previous solutions

The problematics of pattern detection using the environmental and motion sensors has been a hot topic for both academic and commercial studies for a long time. These solutions cover different aspects of human life, including music, and the principle of their functionality differs as well. This chapter will review the current state of academic field in regards of computer-assisted pattern detection, focusing mainly on the research targeting musical rhythm detection. Although, some literature and/or commercial solutions applied in different, even not so related, research spheres will also be mentioned briefly, as they present the concepts that are potentially beneficial for implementing an actual rhythm-tracking smartwatch application.

The most obvious and the most used solution is based on MIDI (Musical Instrument Digital Interface) protocol (57), finding its application in numerous musical instruments. The principle of this solution comprises transforming the mechanical force applied to the instrument into continuous sensor readings (or any other kind of a processable electronic signals), which are later translated into discrete computer-perceivable sound notation. Each MIDI note has a specific timestamp, pitch, dynamics, panning (shifting the symmetry of the stereo signal to left or right), but, what’s most important: attack and release timestamps.

One example of MIDI-based solution is a rhythm-training assistant software available both for PC (but require a MIDI keyboard connected to it) and for electronic keyboard-native platforms (58). The method for sound detection used in electronic keyboards is predominantly void of sensors but is rather based on the circuits similar to the ones used in the computer keyboards. This way, pressing the keys directly enables the current running through the distinct parts of this circuit, each corresponding to specific note. Then, whenever current flows to a processor of a keyboard, a specific MIDI signal is encoded and immediately sent to the sound module. The sound module than plays the note back to the performer through the audio output, i.e., speaker or headphones. If the current flow is altered, or if a specific algorithm is run on instrument’s firmware, the rhythm-training procedure is initiated. The MIDI signal from the performance is compared to a preselected pattern by the criteria of timestamps, duration and sometimes dynamics. The specifics of user-interpretable feedback are different on each specific device: some keyboards present it after the training ends (which suggests that the comparison is run post factum) or is displayed in real time. Sometimes both outputs are combined. The real time feedback on accuracy is affordable and supported even by instruments of the lower price segment, despite seeming complicacy compared to the post factum comparison.

The example which is closer to the object of current studies is MIDI drumset learning programs (see Figure 6). Electronic drumsets rely on sensors (59) (mainly piezoresistive) installed in every separate pad imitating a drum or cymbal. It means that unlike the keyboards, they rely on the processing of analogous signal, before encoding it into MIDI. Some drumsets have settings allowing to set the function determining effect of the pressure, which is applied to the sensor (and therefore the voltage allowed through the piezo element), on the MIDI note loudness. The function is later specified by the additional settings:

- Sensitivity (coefficient by which the voltage is multiplied when determining note loudness)
- Threshold (the lowest detectable voltage)
- Crosstalk (elimination of the interfering signals of neighboring drums, emitted by the vibration passed on through the rack)

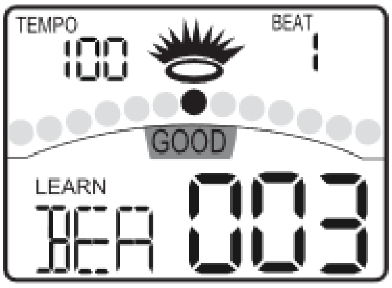

Figure 6 E-drums learning program feedback. Taken from (59)

MIDI tracks comparison for drumsets is simplified by the fact that only the attack timestamp is affected by the performer, release time is entirely encoded in the sound sample.

The only MIDI solution available for guitarists is a guitar synthesizer: a device which enables to process the signal of played strings into alterable electric signal. The later models of guitar synthesizers, however, are discarding the use of MIDI, giving their preference to DSP (Digital Signal Processing) technology. There are multiple options of guitar synthesizer: it may be a separate unit or built into the guitar body, bulk string detection or hexaphonic pickup (detecting signal of each string separately), protocol of sound processing etc. Although flexible, this instrument has its disadvantages:

- Audible processing latency and pitch recognition glitches, especially on the devices of the older generation
- High requirements for exact tuning of the guitar and setting up the synthesizer sensitivity, calibrating references and precise playing

- Even the instruments with the most pronounced issues belong to the upper price category, and the price increases with the quality of output
- These properties set a high entrance barrier for playing the guitar synthesizers, which makes the instrument unsuitable for beginners, therefore native training programs are rare
- Using additional software, other than pre-installed on the controller, requires connection to the PC or some other devices where this software can be run

The more universal solution, suitable for any instrument, is quantization filter (60). Quantization is a process of transforming rhythmically imprecise performance to fit the exact mensural grid, prescribed by notation. At the early days of sound production this was to be executed manually, but now this process is partially automated. Although, this process still requires a human supervision and interference. First, the grid must be pre-set manually either by musician or a sound engineer. Second, the performing instrument preferably undergoes sampling, which a reverse process of sound synthesis – more precisely, taking short sound samples of the instrument's timbre to use as a material for synthesis. The quantization usually has a predetermined efficacy percentage, determining how tightly will this tool snap the performance to the grid: finding the right percentage setting allows to improve the rhythmic structure, while retaining nuances of a humane playing style.

The other rapidly developing area to look at, while developing a rhythm-tracking solution, is rhythm-games. Rhythm-game is defined as a piece of software for portable devices or personal computers that mechanically relies on user's internal sense of rhythm, challenging and rewarding it in gamificational form. These games usually take minimalistic approach in graphics, focusing more on the details most critical for the gameplay: reliable feedback, lack of latency, synchronization of sound and visuals, clear and readable interface, distinct playable assets etc. The most renowned examples of such software are "osu!", rhythm-game for personal computers (although a functional port and analogs for portable platforms also exist), and Guitar

Hero, a series of mechanically similar games for gaming consoles. They share a list of common traits:

- The game consists of multiple independent levels, called “tracks” or “maps”
- Each level has exactly one musical composition played as the level progresses
- During the level progress, distinct elements appear on the screen (see Figure 7)
- These elements are to be interacted with (usually clicked on) at the exact moments in time, so that rhythm of activating these elements complies with the rhythm of selected track
- Each element has an unambiguous mean to mark the right moment of being interacted with, to a user
- User receives immediate feedback on their accuracy upon interacting with each element
- At the end of the level, user’s total accuracy is evaluated using predefined algorithms

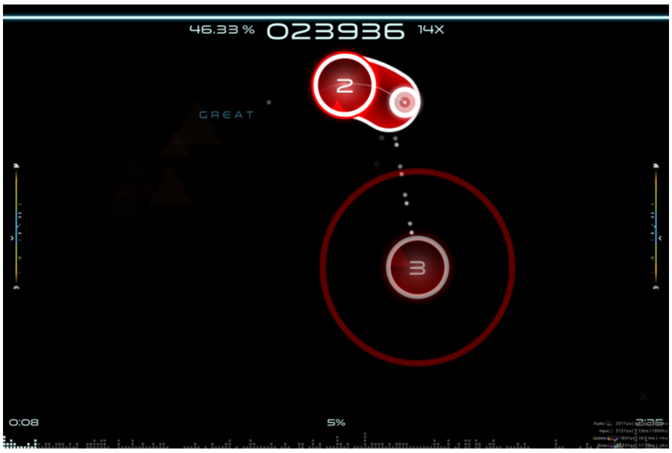

Figure 7 Rhythm-game screenshot, taken from (61)

Even though these games track the instantaneous actions of pressing the buttons on a peripherical device, window of opportunity for pressing each separate element is usually broader than the exact moment of the interaction. This evaluation is based on some sort of fuzzy logic, instead of the binary one (62). It means, pressing the element on time gives you the maximum possible points for it, slight inaccuracy lowers the score, medium inaccuracy gives the lowest non-zero score, and severe inaccuracy gets the element failed. This feature lowers the entrance threshold for the beginners, by not punishing them for pressing a little too fast or a little to slow, but rewards user proportionally to their accuracy, promoting their will to improve.

Abovementioned “osu!” and Guitar Hero games have dedicated level editors, enabling developers and players to build their own levels to play on. All the levels in these games (and majority of the others) are created manually, having the creator place the elements at certain timestamps guided solely by their internal sense of rhythm, therefore, in a way, evaluation of player’s accuracy is based mostly on human judgement. The other family of rhythm-games, like Audiosurf, is based on the procedurally generated levels to play on. Such games generate levels independently from human interference, and in order to do so, they also have internal automated rhythm-tracking algorithms. In most of the cases, the exact concepts this tracking is based on are classified as business-sensitive information.

Usage of rhythm games is being gradually incorporated into experimental facilities of musical education. A group of Chinese researches had applied the teaching methods based on the previous rhythm-games research in a group of elementary school students and presented the results of their experiment on Edutainment conference in 2011 (63). The effects of their experiment were evaluated as positive.

The most industrially recognized rhythm-tracking solutions belong to the Japanese research teams led by A. Nishikata. In 2003 his team published a paper (64) on an SVM (support vector machine) classifier, which was able to process the recording of a musical performance, evaluate the characteristic feature vector for it, describing the overall rhythm instability, and generate a feedback supported by a music knowledge base it was trained on. The system was tested on a focus group and evaluated as reliable, user-friendly and ready to use.

In 2006 another team with participation of Nishikata developed a learning support system for drummers (65), which evaluated multiple features of 24 tester musicians playstyles, and as a result of the use of this system their rhythmic skillset has improved, which was proven by validation experiments.

One of the most substantial researches in the sphere of rhythm analysis, modelling and computer-assisted tracking was conducted by W. Sethares (12). His book called Rhythm and Transforms, as well as numerous preceding publications, contained comprehensive theory of rhythm in various spheres of human activity. In the theoretical parts of these studies, he emphasized the paradoxes – both beneficial and aberrational – that may occur when human brain serves as an instrument engaged in rhythm analysis. Some of these paradoxes have specific notes on how they may be mimicked, avoided or otherwise taken into account while developing a software solution for this kind of problems. Sethares conducted a review of major preceding projects developed in this field, containing detailed description of the instruments used by their researching teams. This extensive theoretical background, combined with previous experience and results of his team's work, were summarized in form of "The foot-tapping machine" concept – an advanced computer-assisted tool capable of rhythm tracking. This tool was rather demanding in regards of computational power and was therefore more suitable as a post-analysis tool. Although, the powerful hardware with minimized latency could afford performing it in real time under the laboratory or studio conditions. This tool's efficiency is supported by numerous statistical models and experimental data.

Another very perspective tool explored in rhythmic studies is neural networks and machine learning. This is a very broad field, comprising various different technologies, which enable to develop software modules mimicking some aspects of human abstract thinking. The tools widely used in rhythm analysis are:

- SVM, mentioned above. SVM has found its application mostly in text analysis and speech recognition, but its property to generalize the properties critical for binary classification are beneficial for some musical decision support systems
- Monte Carlo experiments, which are not imperatively a machine learning, but serve as an effective tool to generate random imitations of certain events
- Cluster classifiers, which are prominently used to break seemingly chaotic data sets into initially unknown number of groups in which all members share some similarities
- Deep convolutional neural networks (DCNN), which are a popular tool for detecting anomalies and patterns on different levels of abstraction. DCNN algorithms have proven to be useful in macro-rhythmic analysis (e.g., biorhythmic analysis (66) conducted by a pet tracker) as reliably as in sub-rhythmic analysis (like detection of cardiac flattering (67) using MEMS sensor data)

To summarize, there are multiple module solutions that may possibly be combined to improve their overall effectiveness, but none of the solutions actually solves the problem raised by this thesis: being able to track rhythmic patterns in real time using a smartwatch and a guitar. More specifically, there is no reliable solution to identify the rhythm pulses, so they may be transferred as discreet signal chain to a module responsible for accuracy evaluation, which there are aplenty.

4. Practical Part

The elaboration of the work has been divided into four major parts (one part entirely theoretical and three experimental parts), containing three to four sub-processes each. This can be summarized as the workflow chart represented on Figure 8. The following discussion is dedicated to detailed description of the experiments and processing stages conducted while elaborating this work's goals.

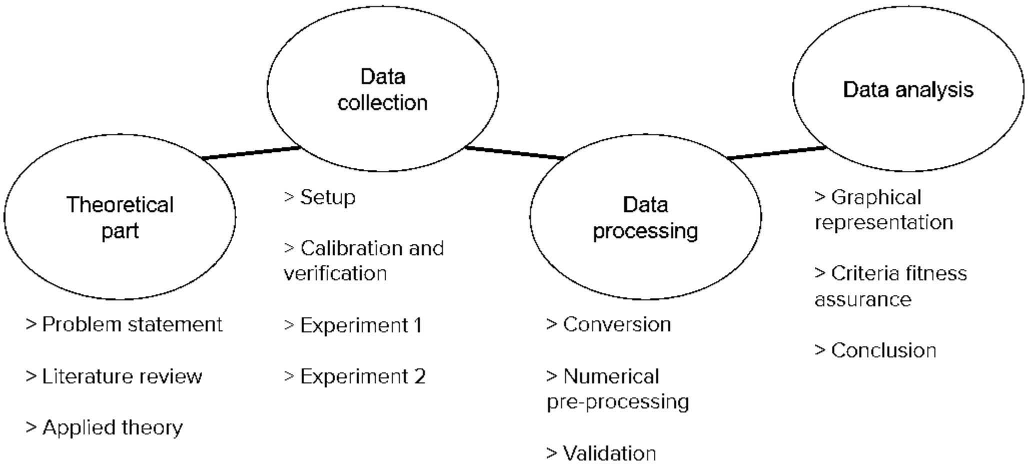

Figure 8 Flow chart of thesis elaboration

4.1 Data collection

4.1.1 Equipment and setup

In order to perform the experiments needed to collect the dataset for the future analysis, certain set of equipment was used: for the enumerative list see Table 1

Musical equipment	Hardware & Software
Generic acoustic guitar	Android smartphone (2x)
- .028 gauge nylon strings	SensorCap app for Android
Ibanez electric guitar	Camera app for Android
- .012 gauge steel strings	
.88 mm plectrum	
Digital metronome	
Intra-auricular monitors set	

Table 1 Equipment set for data collection experiments

The general idea of the experiment was to play the pulses of a steady metric rhythm on one of the guitars while wearing the smartwatch device on the wrist of the playing (right) arm and monitoring its sensor readings. In order to do so, the smartwatch device was paired to a smartphone using the Android Wear interface and Bluetooth debugging feature, then SensorCap application was installed on both devices. This application serves as the medium to establish the BLE (Bluetooth low-energy) connection between the phone and the watch, via which the watch sensors transmit their readings to the phone. The received data are then stored on a cloud storage, from where it can be exported in the form of the .CSV (comma-separated values) file. Additionally, the application allows to synchronize the phone and the watch with the NTP (network time protocol) server to increase the precision of the data timestamps.

For this experiment data was extracted from all sensors of the watch, but only the accelerometer data was sanity-checked initially. Further on the shift to linear accelerometer was suggested because of the reasons mentioned in chapter 4.1.2. In the later experiments directed magnetic induction vector and gyroscope data were exported as well.

In phase two of data collection the other Android smartphone was introduced. See chapter 4.1.4 for the details.

4.1.2 Calibration and verification

In order to reliably analyze the data yielded from the experiments one must make sure they are correctly interpreted. The theoretical insight into sensor data interpretation is provided within Android Developer Manual. Although it was decided to perform a set of calibration experiments in order to mitigate the risk of misinterpretation arising from the probability of SensorCap application or Huawei Watch firmware utilizing the different naming and/or directional conventions, as well as device-sensitive issues.

The first set of calibration experiments was designed in the following way: instead of playing with the watch on, watch was placed on a solid surface, then moved subsequently in all six relative cardinal directions of the watch itself along a straight trajectory. This set was to be adjusted in a way of increasing the initial acceleration and maintaining the constant velocity during the experiment, because in the test runs it had been discovered that in case of lower initial acceleration and variable velocity the exported data were getting lost in the signal noise. The data yielded from this set of experiments helped identifying the cardinal axes and their directions, and the corresponding names of exported values were assigned to them. For the ease of the further identification in the text of the current work, the following naming convention was used: the axis perpendicular to the watch display was dubbed as x-axis, output labeled as value1, the axis lying in the plane of the display surface and perpendicular to the arm of a potential wearer was dubbed as y-axis, output labeled as value2, and the axis directed alongside the arm of a potential wearer was dubbed as z-axis, output labeled as value3. The positive direction of the x-axis is directed underneath the screen, for the y-axis it is directed from bottom to top side of the screen, and for the z-axis it is directed from wrist to elbow, in case of right-handed wearing. The described convention had been found by locating the peaks on the accelerometer outputs and evaluating their directions and amplitude on each of the axes followed by assigning each peak to a distinct step of the calibration experiment.

The next set of calibration experiments was dedicated to evaluation of the external factors impact. It was designed in a similar way, but the experimental site was rotated halfway, and the linear accelerometer values were compared to the regular accelerometer values. These experiments lead to the conclusion that geographical cardinal directions have no impact over the relative cardinal directions of the device itself (i.e., external magnetic field does not affect the readings of the accelerometers), and the difference between linear and regular accelerometer was not very significant, which could be the result of one of the two circumstances: either regular accelerometer readings have built-in gravity correction (or the experiment conditions didn't involve the gravitational interference), or linear accelerometer's initial calibration is malfunctional, which can represent an issue of a great concern. As the conclusion, for the initial phase of the actual experiments regular accelerometer data would be evaluated and no geographical adjustments would be made.

The last set of experiments was dedicated to quantitative calibration. The same single straight trajectory experiment was performed, but this time the length of the trajectory was defined in order to be checked against the value calculated from the readings. The exported accelerometer readings were integrated twice over the timeline in order to calculate the covered path. Unfortunately, due to high level noise and confirmed malfunction of the gravity-induced adjustment, the readings could not be stabilized neither to a zero value over time, nor to any non-zero reference value to be subtracted afterwards, this way the output data was distorted, and the integration rendered impossible. However, the acceleration value approximated backwards from the known trajectory length corresponded to the unprocessed readings of the accelerometer, which meant that the accelerometer readings can be trusted, but in terms of this project evaluating the changes will be more reliable than evaluating the readings themselves or their integrated values, even though derivation of acceleration with respect to time has no distinct physical meaning.

4.1.3 Experiment one

For the statistical variety of input data, two different types of guitars with different string gauges and materials were used, different playing approaches were applied (with or without the plectrum, up/down- and altering stroke, arpeggio) and different tempos (60 and 120 BPM) were held. Pulse consistency was ensured by use of the metronome.

The watch was connected to the host phone as described in chapter 4.1.1. Data from the sensors were read every 15 ms at maximum (minimum frequency of approx. 66,6 Hz), with increasing intensity of the device's movement the frequency of readings would increase as well. The phone was connected to the PC via FTP (file transfer protocol), allowing to store the exported data directly onto the PC storage to be processed immediately.

Five sample songs with duration from 20 to 90 seconds were successfully recorded, data exported and pre-processed, as described in chapter 4.2.1. For one of the recordings acoustic guitar was used, with no significant differences observed while using it. Using plectrum also had no effect on the data quality. Aside the successful ones, there were three experiments resulted in smartwatch firmware failure and loss of the data. Multiple issues while connecting smartwatch to NTP had occurred even during the successful experiments.

These experiments revealed that in a long run usage of the data gathered from linear accelerometer is appropriate, as they contain less noise and over time the sensor can stabilize on a constant non-zero value. However, the noise cannot be omitted entirely, and gravity adjustment does not enable stabilizing on zero value, which is another obstacle to automation of this process.

Another calibration experiment was performed, designed to determine whether this malfunctional calibration was device-specific, app-specific or obstructed by some other external interference. The experiment consisted of predetermined set of waves and turns of both phone and watch in the similar way, then comparing the readings of their respective linear accelerometers exported with the same app. This results yielded from this experiment has revealed that this issue is actually device-specific, as the phone had shown the readings exactly corresponding to the expectations driven from the definition of a linear accelerometer in Android Developer Manual: linear accelerometer in the phone managed to stabilize all three axes' readings to near-zero values regardless of the position and orientation of the phone itself, while linear accelerometer in the watch had shown the similar behavior as in the major phase one tests.

The analysis of data gathered during these major tests has proven that certain limitations are to be held when applying this method of pulse detection: namely the minimal level of amplitude required for the pulses not to be overdriven by the signal noise and the maximal tempo of 90-100 BPM quarter notes for the peaks to be distinguishable in the current readings precision. Further details are described in chapter 4.2.3.

4.1.4 Experiment two

After the manual data analysis the decision was made to set up another series of experiments, which, first of all, are in compliance with the limitations discovered during phase one experiments, and what's more important, yields the precise timestamps of the strokes performed by the player to be checked against the readings in the way which will give an insight on what are the readings in the vicinity of the actual strokes and theoretically allow the resulting algorithm to detect the readings' anomalies corresponding to them.

In order to perform these experiments, another phone was introduced. This phone was set to stationary position with the slow-motion camera on. This camera would record the footage of actual playing, from which the timestamps of the hits would be derived. The frequency of the shots was set to 120 Hz, which would allow even higher precision than the lowest frequency of the data readings.

For this stage of experimental part only two recordings approximately 30 seconds each were made. The experiment consisted of playing upstroke, downstroke and altering stroke in 60 BPM quarter notes with a proper amplitude. One recording was made with establishing the NTP connection and one without. No significant inconsistencies in readings precision was encountered. Both recordings were played on the acoustic guitar without a plectrum.

As there was no possibility to synchronize the camera phone with the host phone and the watch via NTP, high amplitude distinct arm movement was performed as the reference point to synchronize the timeline of the video and the sensors output later during the post-production. In approximately 5 seconds after the reference the actual experimental data were recorded.

Initially only regular and linear accelerometers data was extracted from these experiments, similarly to the phase one experiments, but eventually the data from gyroscope, magnetometer and step counter were extracted and processed both to enhance the dataset for the neural network and, while processing, all of them but step counter were discovered to be suitable criteria on detecting the pulses. Step counter’s extracted .CSV-files contained no information, but nevertheless the step meter is usually implemented as a feedforward classifier with accelerometer and gyroscope data as the input, so it is basically a linear combination of the data successfully extracted.

At this phase an attempt was made to incorporate the microphone amplitude readings into the dataset, but test runs with the use of external noise meter app has shown that current smartwatch microphone is not suitable for this task, as it is too sensitive and is located right near the sound source, which made the data unreadable.

4.2 Pre-processing of sensor data

4.2.1 Conversion to table

The data measured over the course of experiment by smartwatch sensors are exported with SensorCap application in .CSV format. However, the dialect of these files differs from the dialect which can be correctly parsed by MS Excel software. In order to process the data and visually represent them as a scatter plot in this software they need to be either processed by an external converter or converted directly to a table using VBA macros. The second variant was chosen for this thesis to reduce the involvement of the 3rd party products, at least during the preparatory part of the work.

The macros written in built-in VBA developing environment utilized the regular expressions in its core. The dataset was selected as a range, then every cell in the range would be parsed under the following regular expression:

```
([^,]*)(?:[,])([^,]*)(?:[,])([^,]*)(?:[,])([^,]*)(?:[,])([^,]*)(?:[,])([^,]*)(?:[,])([^,]*)
```

For the readability reasons, the first line of the table was filled in by strings from header array. All the next lines contained the non-escaped groups from the regular expression as well as original unconverted line to retrace safely in case of the program failure, and custom timestamps starting from 0 and measured in seconds.

When the additional sensor readings were introduced to the experiments, they were processed by the similar macros, as their respective sensors' output strings were legitimately parceable by the same regular expression. The only adjustment to the original macros was headers array: for the easiest recognition, the notation described in Table 2. was used.

Abbreviation	Conventional name	Measurement unit
$a_{x,y,z}$	x, y, z components of acceleration vector	[m/s^2]
$g_{x,y,z}$	rate of rotation around axes x, y, z	[rad/s]
$m_{x,y,z}$	x, y, z components of magnetic flux density (m.f.d.)	[μT]

Table 2 Readings notation and used units

As mentioned in chapter 4.1.4, step counter data were omitted from the experiment, because current sensors provided no step data on the test runs. Proximity sensor was also omitted, because of its inconsistent and unreliable behavior across the devices. Magnetic field sensor was suggested analogously to sound pickups, which also rely on the magnetic field variation caused by oscillating metal or metal-wired strings.

According to the initial plan, described in chapter 4.1.4, microphone amplitude readings were initially considered to be processable, but, as experiment had shown, these values were claimed insignificant and excluded from the dataset. Which means they were not processed in a general way. All amplitude processing and visualization was made by the noise meter app *in situ*.

4.2.2 Numerical pre-processing

In order to fully explore the impact of certain features onto the overall evaluation of how rhythmically precise is played sequence, it's imperative to employ a series of numerical pre-processing methods presumably capable of eliminating the corrupt or noisy data,

4.2.2.1 Derived values

Various proposed criteria were derived from the measured data in order to have their time variation displayed on a scatter plot and checked for suitability as a pulse criterion. The first proposed criterion was velocity, calculated as a numerical integral of acceleration using the trapezoid rule. For the reference see (4-1), in which X(t) and X(t - dt) are current and previous integrated values respectively, x(t) and x(t - dt) are current and previous original values respectively, t is current timestamp and dt is difference between current and previous timestamps.

$$X(t) = 0{,}5 * dt * \big(x(t) + x(t - dt)\big) + X(t - dt) \qquad (4\text{-}1)$$

The second proposed criterion was the position vector, analogously calculated as a numerical integral of velocity. In case of the sensors other than accelerometer, the values are labeled as the first and the second primitive function to rotation rate and magnetic field strength respectively.

As discussed in chapter 4.1.2, using the integrated values of both regular and linear accelerometers as criteria is inacceptable because of the data distortion created by imprecise measurement tools and data noise, therefore first and second derivation with respect to time was introduced. The derivations were calculated numerically, according to (4-2, where x'(t) is current derivation value, x(t) and x(t - dt) are current and previous original values respectively, t is current timestamp and dt is difference between current and previous timestamps.

$$x'(t) = \frac{\left(x(t) - x(t - dt)\right)}{dt} \tag{4-2}$$

The abovementioned proposed criteria were calculated for every vector component, in case of accelerometer data: a_x, a_y and a_z.

In addition to the new qualities, vector modulus was proposed for each of them. For the modulus calculation the Euclidean distance formula for calculating vector length from its components was used, as (4-3)(4-3 suggests: there |l| is vector modulus and l_x, l_y, l_z are x, y, z components of vector l respectively.

$$|l| = \sqrt{l_x^2 + l_y^2 + l_z^2} \tag{4-3}$$

This way, from three components of the processed reading and its timestamp, 17 new values were calculated. For the sake of optimization, all these calculations were incorporated into the macros responsible for table conversion and therefore performed automatically.

The naming convention for the derived values and used measurement units are listed in Table 3.

Abbreviation	**Conventional name**	**Measurement unit**
$v_{x,y,z}$	x, y, z components of velocity vector	[m/s]
$r_{x,y,z}$	x, y, z components of position vector	[m]
$da_{x,y,z}$	x, y, z components of marginal acceleration	[m/s^3]
$G_{x,y,z}$	angle of rotation around axes x, y, z	[rad]
$dg_{x,y,z}$	marginal rate of rotation around axes x, y, z	[rad/s^2]
$M_{x,y,z}$	x, y, z components of total magnetic exposure	[µT*s]
$dm_{x,y,z}$	x, y, z components of marginal m.f.d.	[µT/s]

Table 3 Abbreviations, units and naming convention for derived values

4.2.2.2 Incorporation

After the measurement had been performed, it was found that different sensors readings are recorded at different timestamps. Plotting these readings onto the same timeline would produce a disjoint timeline, like the one represented on Figure 9. In order to be able to plot the different dependencies onto the same scatter plot and have their graphs uninterrupted, the incorporation procedure is used. First the added quality values are inserted into the new columns underneath the table where they are being added at, then their timestamps are pasted into preexistent timestamps column. After this the whole new dataset is sorted by the key of timestamps in increasing order, and the missing values are filled in with VBA macros.

Three possible ways of filling in the missing values were suggested: lingering last value (the missing values are considered unchanged at the time periods when given quality readings are missing), arithmetical progression (added values fill the gap between two known values in fixed increments) and linear trend. Due to the minuscule differences between the known values and greater effort required to implement and automatize the second and the third method, the lingering last value was implemented. Timeline adjusted this way is visualized on Figure 10.

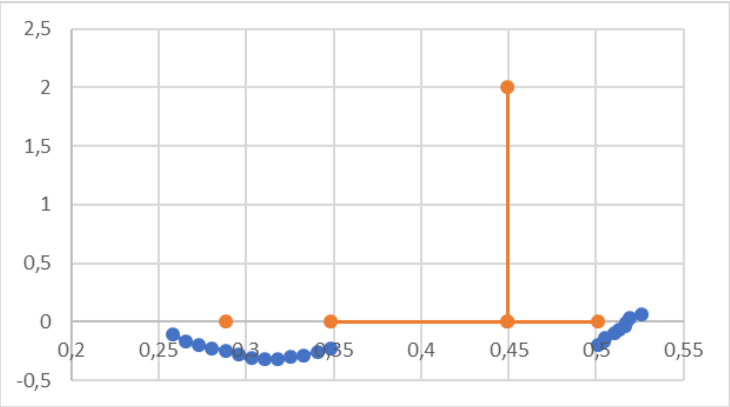

Figure 9 Disjoint timeline

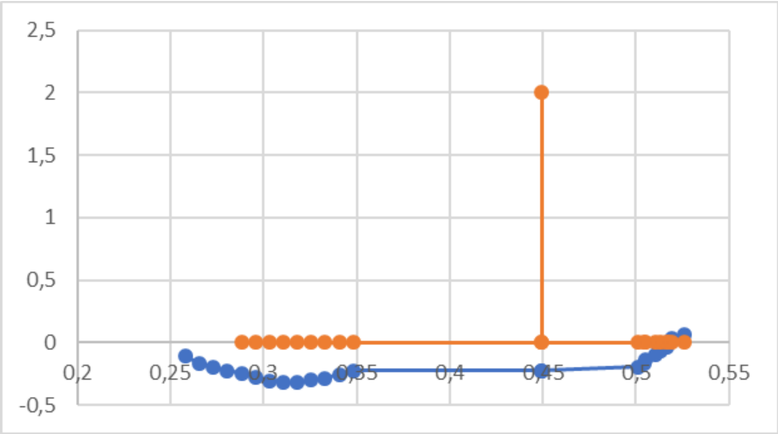

Figure 10 Timeline with incorporated values

4.2.2.3 Normalization

When evaluating the impact of separate vector components on the overall system state, it is crucial to use the original scale of the certain quality. However, in case of anomaly detection analysis, one may face the necessity to stack multiple qualities on one scatter plot. If more than two qualities are to be compared in such a way, Excel native graph constructor may not suffice, as it allows the usage of one primary and one secondary y-axis at a time. This problem is even more critical on the later stages of the data processing, as the secondary axis is always occupied by stroke indicators.

This may be mitigated by applying normalization to your dataset. In case of the qualities oscillating both upwards and downwards from the average, the normalization formula represented as Equation 3 is used. In this formula X_n is normalized value, X is non-normalized value, $\overline{X}$ is average value of the given quality and $X_{max} - X_{min}$ is value range of this quality. $\overline{X}$ and $X_{max} - X_{min}$ are sometimes also referred to as normalization constants.

$$Xn = \frac{(X - \bar{X})}{(X_{max} - X_{min})} \tag{4-4}$$

Depending on the situation, severely outlying values may be excluded from the selection on which average, maximum and minimum are calculated.

Data represented in the normalized way have the following properties:

- All values (except for possible outliers) lie within (-0.5; 0.5) interval
- Oscillating values gravitate towards 0

Normalized data can be safely stacked, as they have the same position of “gravitational zero” and the same scale, even if their non-normalized counterparts have their variation different by tens and hundreds of times. The differences between normalized (Figure 11) and non-normalized (Figure 12) datasets is visualized below.

If the magnitude of different vector components in relation to each other is to be sustained during the comparison with some other qualities, different vector components may share diapason (usually the greater of three) when evaluating the normalized form, but the average value should be selected separately for each component.

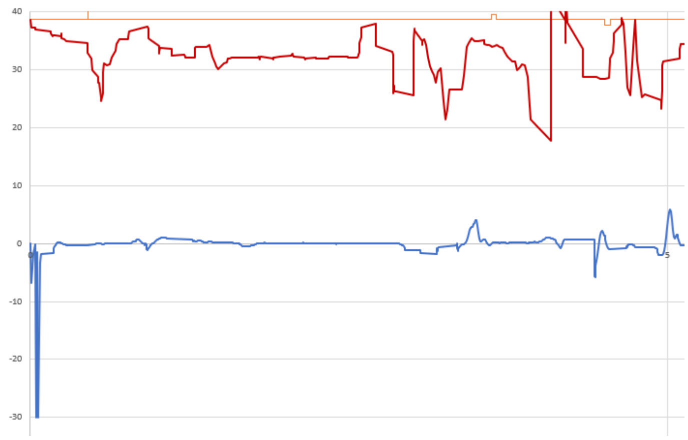

Figure 11 Non-normalized timelines

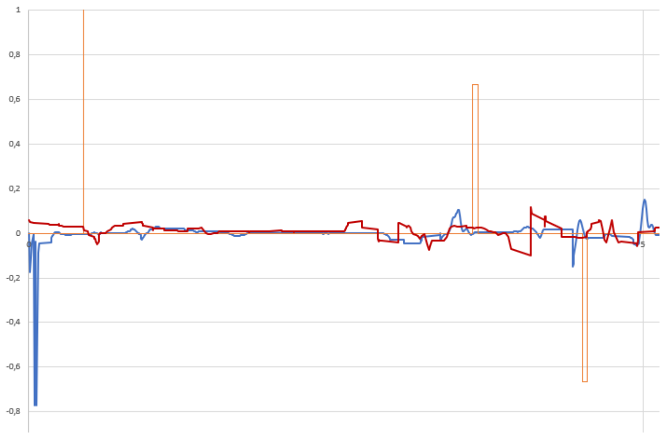

Figure 12 Normalized timelines

4.2.2.4 Noise gate

In order to reduce the noise contamination of sensor outputs, especially around extrema and zeroes of the examined function, the noise gate may be utilized. It is a simple, yet effective, method to clear the meaningful signal of randomly dispersed signal and accidental oscillations, which may create false local extrema. For the reference, see Figure 13.

Noise gate method consists of two steps. At first, some small positive number, called threshold within examined function's value domain is selected. Then all values of the function's domain (or rather their respective absolute values) get compared to the threshold: if the value is higher, it remains in the domain, else it gets replaced by 0. The result is visualized on Figure 14.

The shortcoming of this method is that it is suitable only for data with already high signal-to-noise ratio. If applied to data with low signal to noise ratio, the noise will be transmitted over gate with the signal itself, which will result in data still contaminated by noise and also very fragmented, as the parts consisting entirely of background noise will be cleared, but so will be some unaccented pulses, if gate threshold will be set too high. On contrary, if the threshold was assigned the value which is to low, this may result in increased false positive rate.

Applying noise gate to normalized data has a great advantage of using the same threshold for filtering different qualities, which would not be possible in case of two greatly varying non-normalized samples.

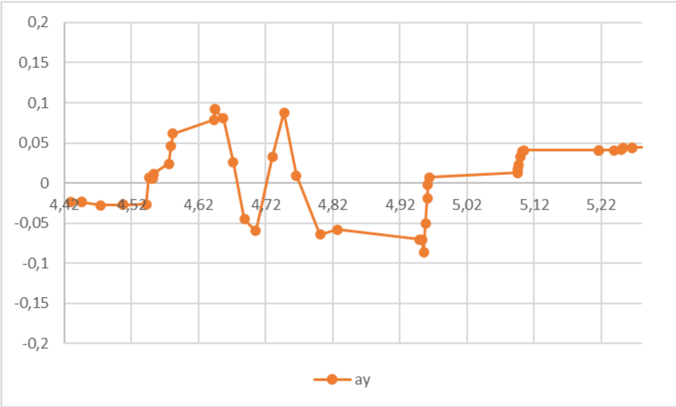

Figure 13 Data contaminated by noise

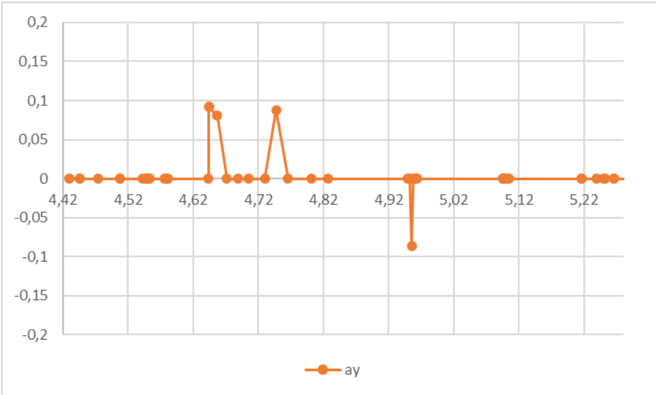

Figure 14 Data after applying noise gate

4.3 Data analysis

The remaining part of the work to be conducted was to analyze the gathered data using the available methods. Herein the form suggested in the Figure 8 will not be maintained hierarchically, as each process mentioned in this scheme is described from the methodology perspective, while logically most of these processes play unalienable roles in the processes not so representative from the methodology perspective, but rather are described in the logical way, where they are iterated over again – some conclusions have even already been drawn in the chapters above.

The preferred order for this chapter was selected as follows: first, the data gathered during the second experiment were validated against the video reference material; then various ways of representation were applied to all the suggested criteria; finally, the selected criteria were analyzed more thoroughly and undergone the whole developed pipeline.

4.3.1 Data validation

4.3.1.1 Discrete binary values assignment

For both recordings made at the experiment number two of data collection phase, processing of the data from the videos differed. The first variant required less effort and was more efficient from the time perspective. It consisted of simple manual check of the videos, frame by frame (see Figure 15), and detecting at first the reference point, mentioned in chapter 4.1.4, then subsequently listing all the frames where the stroking hand was in the middle of the guitar's fretboard. After that the frame numbers were converted to the timestamps in seconds using division by frame frequency. Formula for calculation of exact timestamp from the frame number is represented on the (4-5), where t is timestamp, F is frame number and ν is frequency, for the current experiment fixed at 120 Hz.

$$t = \frac{F}{\nu} \tag{4-5}$$

As the next step, the large peak corresponding to the reference point is located in the sensor data. The timeline of strokes calculated above was shifted so that reference point on the video data matches itself on the sensor data.

Finally, all the points on the video timeline are assigned a Stroke value of 1, the reference point assigned the value of 2, and incorporated into the sensor data table as described in chapter 4.2.1.3, having previously assigned the rest of the points Stroke value of 0. Cross-section of a sample timeline processed this way is visualized on Figure 16.

This algorithm of video data processing is easy to perform, and its output is sufficient for human eye to detect the anomalies once Stroke data are plotted over the criteria you want to analyze for fitness, but it suffers from numerous drawbacks. First of all, it doesn’t distinguish the strokes length and direction, making it useless for automated categorization. It also doesn’t distinguish between free waving and actual strokes, which damages the interpretability of this dataset, and it makes the Strokes plot look like there are some short periods of linearity, which there aren’t. This led to designing another method of processing the video data.

Figure 15 Cross-section of video analysis

4.3.1.2 Square pulse simulation

Processing of the second video and representing it in the form of a timeline of strokes has taken a moderately different approach, considering the shortcomings of the first video processing method. For this video it was decided to represent the pulses in a discrete square waveform. Cross-section of a sample timeline processed this way is visualized on Figure 17.

The main process of the new method was still a manual checking of the video frame by frame an calculating the relative timestamps from the frame number. But this time the following exhaustive list of changes was applied:

- Two timestamps were located per pulse: the moment when the player starts hitting the strings (called attack) and the moment of letting ring the last hit string (called release);
- Attack and release of the same pulse were assigned the value of 1, if the pulse is played as a downstroke, or the value of -1, if the pulse is played as an upstroke;
- Attack and release timestamps were also automatically assigned the value of 0, having these two zeroes to frame the pulse inside in order to imitate the square pulse and rid the data of false linearity intervals. For optimization of this process, insertion of the framing zeroes was automated using macros;
- Finally, the strokes timeline data were incorporated as described in chapter 4.2.1.3;

This way of representation allows to distinguish between up- and downstrokes, as well as to track the minute deviations of sensor data during the stroke, as this may give an insight on how to classify the actual strokes apart from waving.

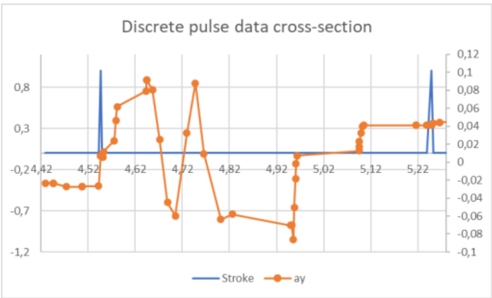

Figure 16 Cross-section of first video analysis method

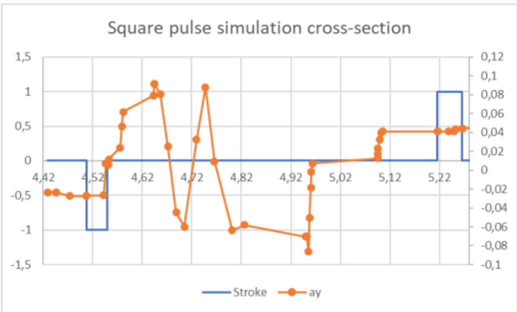

Figure 17 Cross-section of square pulse imitation method

4.3.2 Manual exploratory analysis

After the preprocessing has been accomplished, manual checking of potential criteria suitability has been performed. Normalized sensor data, as well as derived values and their combinations were mapped onto the scatter plot with stroke values, then empirical correlations between stroke and the analyzed values were searched. This analysis has confirmed the initial proposition that the values da_y and g_x have the closest correlation with the stroke due to the specifics of the hand movement.

The second step of manual analysis was applying different levels of noise gate to normalized data in order to eliminate the noise and, ideally, separate the peaks corresponding to the actual strokes, which would mean that the pulse could be located using trivial linear algorithm. Unfortunately, simple application of noise gate was generally insufficient. First reason of this being different offset between the actual pulse and different values extrema. Second reason being non-reliable elimination of backlash while playing in downstroke technique, however some distinction between backlash and actual upstroke was determined successfully. The third reason was unacceptably high rate of false positives and negatives on different sections of the same timeline with the same level of noise gate, which means that the gate should at least be adaptable.

Manual analysis has also discovered numerous other correlations between data and pulses: e.g., playing with altering stroke produces distinct peaks on g_x which can be filtered reliably using the noise gate, while da_y data are almost entirely overwhelmed by noise; changing the technique to straight up downstroke leads to the reverse result, namely more pronounced peaks on a_y and declining precision of g_x peaks. These properties are visualized on Figure 18. The following colors are used to distinct the plotted values: g_x – blue, da_y – red, stroke – orange. This graph cross-section represents the transition from altering strokes to downstroke of the same BPM. Magnetometer provided no significant readings for the duration of experiments.

As manual analysis shows, it is possible to some extent to suffice with linear algorithm of data processing, although the precision is not perfect, but may be enhanced introducing the additional operations. On the other hand, it should be kept in mind that the complicacy of input processing should remain as low as possible for the sake of the algorithm to be processed in real time. Applying this process would also require fine-tuning the constants, like the normalization constants and noise gate threshold.

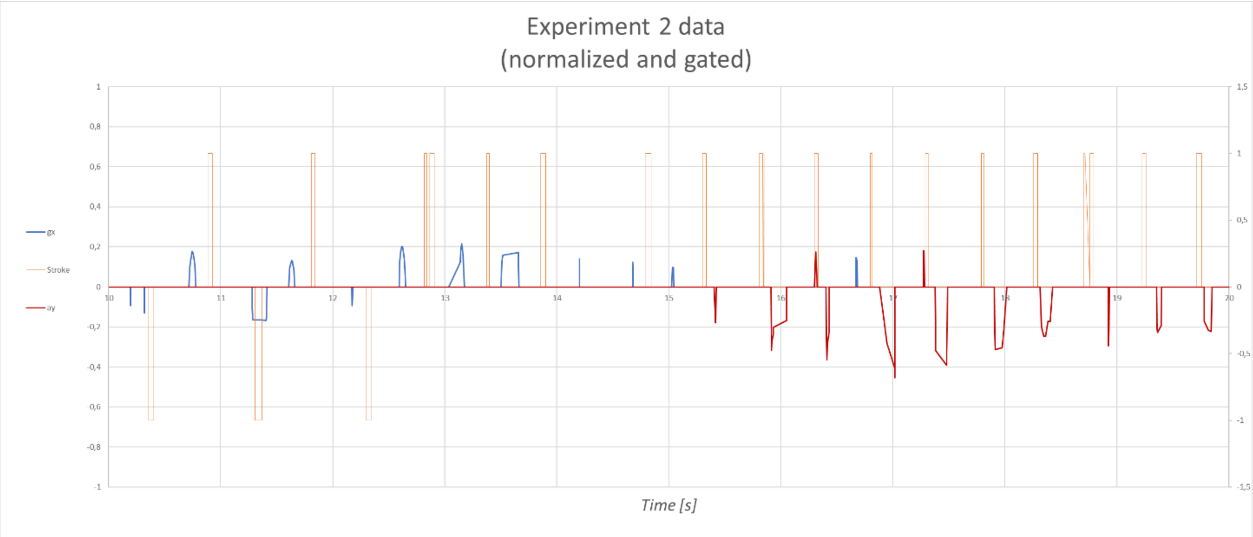

Figure 18 Cross-section of exploratory analysis of 2nd experiment data

The misalignment between the readings peaks and the bands corresponding to the stroke can be a result of the following:

- Initially incorrect assumption on how the reference peak on data should relate to the reference stroke on the video
- Imperfect playing style, resulting in smudged, unpronounced attacks on the stroke
- Programmed compression of the validation video, due to which the transition frames were inserted, obfuscating the correct attack timestamps

- Framerate optimization, violating the initial assumption about constant framerate; in-depth analysis of video specifications proven this to be true only at the very beginning of recording, even before the reference point

When implementing the actual app, retaining this work's pipeline, these details are to be explored more thoroughly.

5. Results and Discussion

5.1 Results

As all the possible criteria were summarized, visually represented and validated against the data from slow motion camera, the fitness of these criteria was discussed in order to identify the most suitable data vectors to base the tracking solution on, which is summarized on Table 4

Feature	Axis	Fitness	Commentary
a	x	Mostly positive	- Antinodes corresponded to the pulses, data may get lost after application of noise gate - Opposite sign of a criterion - Medium high impact on measurements
	y	Mostly positive	- Antinodes corresponded to the pulses, data may get lost after application of noise gate - High impact on measurements
	z	Questionable	- Data lost in the noise due to low amplitude - More quality sensors may provide the useful data for distinction of hand waving and actual strokes - Low impact
v	x	Questionable	- Due to calibration errors, data fluctuated dramatically, placing an obstacle to correct evaluation of impact on measurements - Nodes corresponded to the pulses
	y		
	z		

r	x	Negative	- Data went corrupt due to calibration errors
	y		
	z		
da	x	Mostly positive	- Nodes corresponded to the pulses - Opposite sign of a criterion - Medium high impact on measurements
	y	Positive	- Nodes corresponded to the pulses - High impact on measurements - Especially suitable for downstroke detection
	z	Negative	- Data not distinct
g	x	Positive	- Nodes corresponded to the pulses - Opposite sign of a criterion - High impact on measurements
	y	Negative	- Relative rotation insufficient
	z	Mostly positive	- Applicable on high amplitude strokes
G, dg	x	Not explored, as suitable partial criterion was already identified	
	y		
	z		
m	x	No readings acquired during measurements	
|a|, |v|, |da|, |g|	x	Negative	- No significant correlation identified

Table 4 Summary of criteria fitness evaluation

5.2 Discussion

The solution developed in this thesis is demonstrative and lacks in reliability and proper experimental testing. For the actual implementation of rhythm-tracking application it should be improved in numerous ways. Here is the list of possible improvements:

- Slow motion camera is synchronized with phone and smartwatch using NTP, to ensure the exact compliance of timestamps and excluding the reference point from the data
- Data set is enriched by playing more techniques and various playing patterns, as well as participation of several performers to increase variability and to consider individual playstyle nuances
- Devices from different manufacturers and different software participate in data collection; probably ensuring the constant readings rate, if possible
- More investigation on linear accelerometer calibration errors conducted
- Manual research data combined with machine learning: SVM or cluster categorizer run on the training data to verify the similarities criteria, then main solution based on DCNN
- Validation data from camera is used as mask for training data marking, instead of setting the outputs manually
- Final solution having both manual settings and automatic calibration to set the gravity correction, noise gate threshold, normalization constants and horizontal shim between actual pulses and signal peaks, if this issue reveals to be global

6. Conclusion

In theoretical part of the thesis major theoretical foundation was established. As this topic is widely cross-disciplinary, so the theory comprised musical definitions, mathematical apparatus and physical principles of sensors, as well as their engineering specification. Additionally, brief market and audience research has been conducted to ensure demand and novelty of the solution, proven by preceding studies of computer-assisted rhythm detection addressed at the end of theoretical part.

The first objective of this thesis was to design a set of data collection experiments. The method of gathering data, developed for this work, has meet the expectations regarding its precision and analytical value, but turned out to be exquisitely time-consuming due to the addition of the video analysis step and increasing complexity of data pre-processing step caused by the necessity to incorporate the video timestamps into the continuous data timeline. Although as analysis shown, these data are not excess in any way, and all the steps are necessary for producing the representative set of inputs data.

Used appliances and software created their own set of limitations, which can be overcome only customizing them more precisely for the means of this experiment. In the case of this thesis, however, the limitations were considered inalienable part of experiment, and, overall, these adjustments contributed to creation of an experimental pipeline which in 100% of cases yielded to interpretable dataset with no excess data

In two of the experiments conducted in chapter 4.1.3 assigning the timeline points to the actual pulses played was almost impossible; one of them being played by arpeggio technique – which served as a proof of this technology not being applicable for playing arpeggios, legato and basically any technique other than strokes. The other one, as described above, was partially played with weak amplitude, which affected the peaks strength and, given the same level of background noise, led to distorted data. Applying the noise gate on this data had no positive effect, as data extremes were comparable to noise extremes, and filtering precision was low.

The data from the second set of experiments were representative but suffered from low level of data variability.

Additional steps of pre-processing, namely deriving auxiliary values, normalization and noise gate were helpful while analyzing the data manually, but for actual implementation they can be omitted.

The gathered and processed data were manually analyzed to determine the most suitable criteria for detection of pulses. To perform this step, the outputs of each sensor and each auxiliary value was visualized as a 2D time series plot, the secondary y-axis of the plot contained strokes data. The analysis result was summarized in chapter 5.1

Research has shown that two manually recognized criteria for rhythm pulse identification among the sensor data were acceleration on the vertical axis, corresponding to forceful strokes, and rotation rate around the arm axis, corresponding to the rotational motion of the wrist which is a part of strumming technique. Other criteria are either insignificant or may prove useful as a part of machine learning training set.

Finally, the opportunities for improvement were briefly addressed in Chapter 0. The main suggested improvements were enrichment of the data set, increase of validation experiment reliability and adding a neural network module into the actual application.

7. References

1. **Anonymous.** Urban Dictionary: you're a rockstar. *Urban Dictionary.* [Online] April 2013. https://www.urbandictionary.com/define.php?term=you%27re%20a%20rockstar.

2. **SSI Author Group.** Developers Are The New Rockstars. *SSI People.* [Online] June 2015. https://www.ssipeople.com/2015/06/09/developers-are-the-new-rockstars/.

3. **Lillington, Karlin.** Has the rock star CEO finally left the tech business? *The Irish Times.* [Online] September 2013. https://www.irishtimes.com/business/technology/has-the-rock-star-ceo-finally-left-the-tech-business-1.1516026.

4. **Thomson, K.** How Many Musicians Are There? *Future Of Music.* [Online] June 2012. http://money.futureofmusic.org/how-many-musicians-are-there/.

5. **Mäkelä, J.** Alternations: The case of international success in Finnish popular music. *European Journal of Cultural Studies.* 2009, Vol. 12, 3.

6. **Marone, V. & Rodrigues, RC.** "What's So Awesome with YouTube": Learning Music with Social Media Celebrities. *ONLINE JOURNAL OF COMMUNICATION AND MEDIA TECHNOLOGIES.* 2019, Vol. 9, 4.

7. **Galuszka, P. & Brzozowska, B.** Crowdfunding: Towards a redefinition of the artist's role - the case of MegaTotal. *INTERNATIONAL JOURNAL OF CULTURAL STUDIES.* 2017, Vol. 20, 1.

8. **Abdiukhanov, E.: bass player for JINJER.** JINJER: дети олигархов на мировой сцене. [interv.] A. Glazastaya. *BRSCH.* March 2020.

9. **Camara, GS.** Effects of instructed timing on electric guitar and bass sound in groove performance. *JOURNAL OF THE ACOUSTICAL SOCIETY OF AMERICA.* 2020, Vol. 147, 2.

10. **Hajdarbegovic, N.** Smartwatch Development: Are Smartwatch Worth The Trouble? *Toptal.* [Online] 2013. https://www.toptal.com/apple/developing-for-smartwatches-is-it-worth-the-trouble.

11. **Google Analytics.** "guitar rhythm app" - Analysis - Google Trends. *Google Trends.* [Online] March 2020. https://trends.google.com/trends/explore?date=2009-01-01%202020-03-29&q=guitar%20rhythm%20app.

12. **Sethares, W.** *Rhythm and Transforms.* London : Springer-Verlag, 2007. ISBN 978-1-84628-639-1.

13. **Liddell, H. and Scott, R.** *A Greek-English Lexicon.* Oxford : Oxford University Press, 1996.

14. **Berry, W.** *Structural Functions In Music.* 2nd ed. New York : Dover Publications, 1987. ISBN 978-0-486-25384-8.

15. **Garland, T. and Kahn, C.** *Math and Music: Harmonious Connection.* s.l. : Dale Seymour Publications, 1995.

16. **Dictionary.com administration.** Riff. *Dictionary.com.* [Online] https://www.dictionary.com/browse/riff.

17. **Winold, A.** [book auth.] R. DeLone. *Rhythm in Twentieth-Century Music.* 1975.

18. **McPherson, S.** *Form in Music.* London : Williams, 1930.

19. **Bregman, A.** *Auditory Scene Analysis.* Cambridge : MIT Press, 1990.

20. **von Helmholtz, H. et al.** *On the sensations of tone as a physiological basis for the theory of music.* London : Longmans, Green and Co., 1875.

21. *Ableton Loop.* **Neely, A.** Berlin : s.n., 2018. Polyrhythms are Polypitch.

22. **Brelet, G.** *Le Temps Musical.* Paris : s.n., 1949.

23. **Anonymous.** GUITAR definition. *Merriam-Webster Dictionary.* [Online] https://www.merriam-webster.com/dictionary/guitar.

24. **Ibanez Inc.** *Ibanez electric guitars maintainance manual.* [Manual] 2013.

25. **Bacon, T.** *The Ultimate Guitar Book.* 1997. ISBN 0375700900.

26. **Snyder, J.** *Jerry Snyder's Guitar School.* 1999. ISBN 0-7390-0260-0.

27. **Snyder, B.** *Music and Memory.* Massachusets : MIT, 2001. ISBN 978-0262692373.

28. **Pavlík, J. et al.** *Aplikovaná Statistika.* Prague : ICT, 2005. ISBN 80-7080-569-2.

29. **Theriot, K.** What is a Noise Gate? *Homebrew Audio.* [Online] September 2015. https://www.homebrewaudio.com/what-is-a-noise-gate/.

30. **Hodgson, J.** *Understanding Records.* London : Bloomsbury, 2010. ISBN 978-1-4411-5607-5.

31. *Théorie analytique de la chaleur.* **Fourier, J.** Paris : s.n., 1822. OCLC 2688081.

32. *On the representability of a function by a trigonometric series.* **Riemann, B.** Göttingen : s.n., 1854.

33. **Bourne, M.** The Trapezoidal Rule. *InMath.* [Online] https://www.intmath.com/integration/5-trapezoidal-rule.php.

34. —. Derivatives of the Sine, Cosine and Tangent Functions. *InMath.* [Online] https://www.intmath.com/differentiation-transcendental/1-derivative-sine-cosine-tangent.php.

35. **The Editors of Encyclopaedia Britannica.** Vector. *Encyclopaedia Brittanica.* [Online] September 2006. https://www.britannica.com/science/vector-physics.

36. **Howard, A.** *Elementary Linear Algebra.* London : John Wiley and Sons, 1994. ISBN 978-0-471-58742-2.

37. **Anonymous.** Vector magnitude and direction review. *Khan Academy.* [Online] https://www.khanacademy.org/math/precalculus/x9e81a4f98389efdf:vectors/x9e81a4f98389efdf:component-form/a/vector-magnitude-and-direction-review.

38. **Bernstein, J.** An Overview of MEMS Inertial Sensing Technology. *Fierce Electronics.* [Online] February 2003. https://www.fierceelectronics.com/components/overview-mems-inertial-sensing-technology.

39. **IEEE.** *IEEE Standard for Sensor Performance Parameter Definitions.* [IEEE Standard] New York : IEEE Electron Devices Society, 2014. 2700-2014.

40. **Frautschi, S. et al.** *The Mechanical Universe: Mechanics and Heat, Advanced Edition.* Cambritge : Cambridge University Press, 2007. 978-0-521-71590-4.

41. **group, IEEE Std.** *Coriolis Vibratory Gyroscopes.* 2004. IEEE No. 1431-2004.

42. *IEEE International Conference.* **Nazare, AC. Jr, de Costa, FO. and Schwarz, W.** Auckland : IEEE, 2018. Content-Based Multi-Camera Video Alignment using Accelerometer Data. ISBN 978-1-5386-9294-3.

43. **Google.** Sensors Overview. *Android Developers.* [Online] December 2019. https://developer.android.com/guide/topics/sensors/sensors_overview.

44. **Kelion, Leo.** Smartwatches: Specs and reviews for the leading models. *BBC News.* [Online] September 2014. https://www.bbc.com/news/technology-29107354.

45. **Anzar, K.** Samsung Galaxy Watch LTE review: Apple Watch substitute for Android users . *Business Standard.* [Online] November 2019. https://www.business-

standard.com/article/technology/samsung-galaxy-watch-lte-review-apple-watch-substitute-for-android-users-119112900670_1.html.

46. **Huawei Technologies.** Huawei Watch 2 Specification. *Huawei Web.* [Online] June 2019. https://consumer.huawei.com/en/wearables/watch2/specs/.

47. **Google.** Motion Sensors. *Android Developers.* [Online] March 2020. https://developer.android.com/guide/topics/sensors/sensors_motion.

48. —. Position Sensors. *Android Developers.* [Online] March 2020. https://developer.android.com/guide/topics/sensors/sensors_position.

49. —. MediaRecorder. *Android Developers.* [Online] February 2020. https://developer.android.com/reference/android/media/MediaRecorder.

50. **Doscher, J.** Accelerometer design and applications. *Analog Devices.* [Online] December 2008. http://www.analog.com/en/technical-library/faqs/design-center/faqs/CU_faq_MEMs/resources/fca.html.

51. **Gabrielson, T.** Mechanical-thermal noise in micromachined acoustic and vibration sensors. *IEEE Transactions on Electron Devices.* 1993, Vol. 40, 5.

52. **Jones, W,.** A Compass In Every Smartphone. *IEEE Spectrum.* [Online] January 2010. https://spectrum.ieee.org/semiconductors/devices/a-compass-in-every-smartphone.

53. **Guadagnin, L., et al.** Analytical Modeling and Experimental Characterization of a Magnetic Pickup for Electric Guitar. *JAES.* 2017, Vol. 65, 9.

54. **Armstrong, D.** *Magnetic pickup for electric guitars. 4283982* California, January 26, 1979. 84/1.15.

55. *Magnetometer basics for mobile phone applications.* **Cai, Y., et al.** 2012, Vol. 54.

56. **BOSCH Engineering.** MEMS Barometric Pressure Sensor. *IEEE Sensors & Transducers e-Digest.* 2008, Vol. 92, 4.

57. **Swift, A.** A Brief Introduction to MIDI. *SURPRISE.* 1997, Vol. 1, 2.

58. **Yamaha Inc.** *Yamaha DGX-650 Owner's Manual.* [Manual] 16238.

59. **Medeli.** DD-512 E-Drum Set User Manual. *Kirstein.DE.* [Online] September 2012. https://www.kirstein.de/docs/manuals/xd-dd512-en-0912.pdf.

60. **Childs, G.W.** A Music Producer's Guide To Quantizing MIDI. *Ask.Audio.* [Online] March 2018. https://ask.audio/articles/a-music-producers-guide-to-quantizing-midi.

61. **osu! Team.** osu! Repository. *GitHub.* [Online] https://github.com/ppy/osu.

62. —. Score - Knowledge Base | osu! *Knowledge Base | osu!* [Online] January 2007. https://osu.ppy.sh/help/wiki/Score#osu.

63. *Edutainment.* **Wang, CY. & Lai, AF.: Development of a Mobile Rhythm Learning System Based on Digital Game-Based Learning Companion.** [ed.] Hwang W Chang M. Taipei : Springer-Verlag, 2011. 978-3-642-23455-2.

64. **Yonekawa, T. & Nishikata, A.** Rhythm pattern accuracy diagnosis system capable of objective evaluation and commentary feedback. *IEICE Transactions on Information and Systems.* 86, 2003, Vol. IV, 1.

65. **Tsuji, Y., & Nishikata, A.** Development and evaluation of drum learning support system based on rhythm and drumming form. *Electronics and Communication in Japan.* 2006, Vol. 89, 9.

66. **ActiJoy.** Health & Activity Tracker -- ActiJoy. [Online] Six Guys Labs, s.r.o., October 2016. https://www.actijoy.com/products/tracker.

67. **Fujita, H. & Cimr, D.** Computer Aided detection for fibrillations and flutters using deep convolutional neural network. *INFORMATION SCIENCES.* 2019, Vol. 486.