Fractal analysis of a human–played drum beat with principal component analysis

Saara Malila

May 2023

1 Abstract

The presence of $1/f$ type noise in a variety of natural processes and human cognition is a well–established fact, and methods of analysing it are many. Fractal analysis of time series data has long been subject to limitations due to the inaccuracy of results for small datasets and finite data. The development of artificial intelligence and machine learning algorithms over the recent years have opened the door to modeling and forecasting such phenomena as well which we do not yet have a complete understanding of. In this thesis principal component analysis is used to detect $1/f$ noise patterns in human–played drum beats typical to a style of playing. In the future, this type of analysis could be used to construct drum machines that mimic the fluctuations in timing associated with a certain characteristic in human–played music such as genre, era, or musician.

In this study the link between $1/f$–noisy patterns of fluctuations in timing and the technical skill level of the musician is researched. Samples of isolated drum tracks are collected and split into two groups representing either low or high level of technical skill. Time series vectors are then constructed by hand to depict the actual timing of the human–played beats. Difference vectors are then created for analysis by using the least–squares method to find the corresponding "perfect" beat and subtracting them from the collected data. These resulting data illustrate the deviation of the actual playing from the beat according to a metronome. A principal component analysis algorithm is then run on the power spectra of the difference vectors to detect points of correlation within different subsets of the data, with the focus being on the two groups mentioned earlier.

Finally, we attempt to fit a $1/f$ noise model to the principal component scores of the power spectra. The results of the study support our hypothesis but their interpretation on this scale appears subjective. We find that the principal component of the power spectra of the more skilled musicians' samples can be approximated by the function $S = 1/f^{\alpha}$ with $\alpha \in (0, 2)$, which is indicative of fractal noise. Although the less skilled group's samples do not appear to contain $1/f$–noisy fluctuations, its subsets do quite consistently. The opposite is true for the first–mentioned dataset. All in all, we find that a much larger dataset is required to construct a reliable model of human error in recorded music, but with the small amount of data in this study we show that we can indeed detect and isolate defining rhythmic characteristics to a certain style of playing drums.

Contents

2 Introduction

Drum machines are electronic musical instruments that produce a synthetic drum beat. The idea behind their invention and development is to eliminate the necessity of having a human drummer playing a drum set when playing in studio conditions or, for instance, performing electronic music. With today's accessible and easy-to-use applications it is possible for a solo artist to create and record drum tracks by themselves on a home computer. The first electronic drum machine was reportedly invented in the 1930s, and they saw rapid development from the 1980s on alongside other synthesizers used in music [37]. The instruments of that era were quite analogue and mechanical, producing the desired sound when played by hand or, alternatively, automatically to a set beat. The technology has come far since then, and today's synthesizers apply artificial intelligence and machine learning in their drum machines and other applications.

This thesis was inspired by the question of if, and how, it would be possible to build a drum machine that imitates the style of a certain genre, or a certain musician, for instance. This study concentrates on the theoretical basis of that idea: we will show that constructing and generating rhythmic patterns in such fashion is possible. At the heart of the problem is the idea of somehow being able to computerize the "human element" in live music. The key difference between a human and a drum machine playing is that a human makes mistakes in timing, whereas the simplest drum machines keep beat like a metronome. In an article by Hennig et al., in a chapter called "Humanizing music" [5] a study is introduced where 39 people listened to two different rhythmic samples of music, and a clear majority rated a beat containing "humanized" errors in timing over a more mechanical one. A great article by Räsänen et al., "Fluctuations of Hi-Hat Timing and Dynamics in a Virtuoso Drum Track of a Popular Music Recording" [6], concludes that a fluctuating beat results in a more enjoyable listening experience, and goes on to discuss how we could analyze "groove".

There are existing techniques of introducing to a drum beat a degree of recurrent delay. The most elementary method would be to mechanically set, for instance, every third beat in a bar to have a slight delay. It is easy to do and can be accomplished even with a vintage synthesizer. However, this doesn't do much to humanize the sound. A more sophisticated way used to naturalize the timing is to add a *white noise* signal to the track. White noise consists of samples that have zero correlation with each other but have equal variance. Therefore we end up with a computerized drumbeat containing a controllable level of delay occurring in uneven intervals. Comparing to a beat played with metronome level accuracy, or even that with a mechanical delay such as mentioned above, the signal is remarkably more "human" and results in a significantly more enjoyable listening experience.

A signal containing white noise is not, however, sufficiently similar to how a human would play. If we start rationalizing the definition, we might come to the conclusion that it is not at all how a human drummer would play, although in many cases the resulting beat may sound very convincing. The delays in a human–played beat more likely would correlate with each other, and conversely, their variance would likely not be equal. The argument made in the article written by Hennig et al. [5] is that a signal containing fractal or "pink" noise in lieu of white noise resembles a human–played drum beat considerably more closely. This was supported by a survey on which type of beat listeners enjoyed more. In this thesis we aim to prove that it is possible to dissect the elements in a piece of recorded music that contribute to the fractal pattern, and furthermore to create a machine learning model capable of reproducing similar patterns in new drum beats.

Fractal analysis on recorded music is a relatively new field of study, and its applications to music technology are still few and far between. The work we are pursuing here is therefore rudimentary, and its basis rests for a part in trial and error. The article by Räsänen et al. [6] is among the principal studies in the subject, and as such provides guidance for this thesis in a significant way. There is a simple reasoning behind the lack of research into pink, or $1/f$–noise in music: fluctuation created with white noise results in a sufficiently pleasant musical experience, and the mathematics behind it are that much simpler and easier to control. Keeping in mind that modern drum machines are automated and computerized, white noise fluctuation patterns are well suited for programming, and do not even require much mathematical background of the programmer. Most of all, white noise is well–known and well–understood, whereas despite its common recurrence in nature and science, very little is known about the technicalities and structure of pink noise. We will explore $1/f$–noise in greater detail in a subsequent chapter (3.3). Until then, the introduction from an article by Ward and Greenwood, "$1/f$ noise" [30], summarizes the nature of the phenomenon to the point:

"[...] $1/f$ noise can not be obtained by the simple procedure of integration or of differentiation of such convenient signals. Moreover, there are no simple, even linear stochastic differential equations generating signals with $1/f$ noise. The widespread occurrence of signals exhibiting such behavior suggests that a generic mathematical explanation might exist. Except for some formal mathematical descriptions like fractional Brownian motion (half–integral of a white noise signal), however, no generally recognized physical explanation of $1/f$ noise has been proposed. Consequently, the ubiquity of $1/f$ noise is one of the oldest puzzles of contemporary physics and science in general." ([30] chapter 1)

In the study by Räsänen et al. [6] detrended fluctuation analysis is used to analyse the hi-hat patterns in the song "I Keep Forgettin'" [34] by Michael McDonald played by the drummer Jeff Porcaro, and clear evidence of fractal–type fluctuation in the rhythm is found. Since such evidence already exists, the focus of this thesis is in finding whether samples of music sharing common characteristics also contain similar fluctuation patterns, and how the results could be applied to music production in practice. The characteristic subject to the study at hand is skill. The data consists of isolated drum tracks within sufficiently similar genres that are then divided into two categories: one where the drummer in question is famed for their technical skill, and one where the drummer is known for their lack thereof. Jeff Porcaro of Toto, Frank Beard of ZZ Top, Neil

Peart of Rush, Mike Portnoy of Dream Theater, and Jonathan Moffett, who worked with Michael Jackson and Madonna among others, represent the former category. Among the latter are unpolished studio recordings by acquintances, as well as samples played by Jerry Nolan of the New York Dolls, Paul Cook of the Sex Pistols, Tommy Ramone of the Ramones, and Scott Asheton of Iggy & The Stooges. The latter group is rather interesting. For example the Ramones were one of the most influential groups in the history of rock music, but the band was famous for how poorly they played. They were even notoriously bad at keeping a beat, which is naturally a point of fascination with regards to our subject of study. The same can be said about the other groups represented in the data. The musicians' popularity may even hint at a secondary conclusion, that at times a correlation can be found between the amount of human error in a recording, and the level of enjoyment attained from it by the listener.

Since the publication of Ward and Greenwood's article [30] in 2007 artificial intelligence and machine learning have become staples in science and industry. This is especially important to our cause since we can utilize machine learning techniques to find patterns that might, without them, require excessive computational effort. Such methods may lead to the scientific community overcoming the mathematical obscurity of $1/f$ noise. Another significant advantage of using machine learning in analysing time series data is that it does not require large datasets or very long data vectors like computational fractal analysis does.

The machine learning algorithm that was opted for use here is *principal component analysis*, often referred to as "PCA", which will be discussed in greater detail in chapter 4.5. Time difference vectors are created from our data that portray the fluctuations in delay in playing of a select drum, compared to a "perfect beat". By performing PCA on a set of vectors we aim to find recurring trends in the delay, and by doing so, to find delay patterns characteristic of a certain way of playing. The results obtained between the two groups of samples will be compared, but trends in the samples collected from the same musician will also be examined. We find that running PCA on Matlab is well suited for detecting categorical differences, as opposed to applying more rigorous and sophisticated machine learning methods. Such methods should, however, be utilized in further study of the subject.

3 Theory and Background

3.1 Time series difference vectors

In order to study musical fluctuations in an empirical way we must first translate the data to a form that enables it. Rhythm is, from that point of view, a rather approachable aspect to study, since we are examining fluctuations in time. Fractal analysis on time series has been established on numerous occasions and contexts, and $1/f$ noise has been proven to exist in other natural processes. For instance Kobayashi and Musha analysed fluctuations in human heart beat as a time series as described in their 1982 article "$1/f$ fluctuation of heartbeat period" [29]. Time series as a discrete, often single–variate function provides a well–manageable setting for noise analysis. Fluctuations in rhythmic delay may often be detectable to the naked eye in the graph of the vector. Conveniently, noisy fluctuations in timing occur naturally in various conditions, and time series analysis regarding such phenomena is common practice.

A time series is one where the data values of a variable are taken successively or periodically over time intervals and are often utilized to study fluctuations and trends with regards to timing or occurrence. Examples of this might be examining how frequently storms occur during a year within a certain region, or finding irregularities in a pulse. By finding and analyzing trends one can create forecasts from the information gained from time series data. If for five consecutive years a geographical area sees a surge of stormy weather during April, for instance, we can expect April to be the stormiest month of next year as well.

In this study our data consists of difference vectors. We will construct them from two "parallel" time series vectors by subtracting from one containing time values of the actual beats a vector containing the values for what would be the "perfect" beat. The latter is constructed from the collected data using the least squares method (see chapter 3.2), and depicts how the beats should be scattered over the time interval in question so they would be evenly spaced. The real data vectors display the beats on a select drum, for instance the bass drum or a snare drum. In our thesis the choice was made to examine a selection of different drum sounds in order to find out whether patterns occur in the drum track comprehensively. See chapter 4.3 for a detailed depiction of the process. The resulting difference vector consists of values for the deviation of each real drum hit from what would be the corresponding "perfect timing" for it. Naturally, negative values indicate a delay from the perfect beat, and positive values point to the hit occurring early. Patterns can then be studied among a sufficient amount of such vectors.

3.2 Least squares method

The necessity for finding the ideal beat that would approximately be equivalent to a metronome track was briefly explained in the previous chapter 3.1. Next we will demonstrate how this is achieved using the least squares method.

The simplest way to distribute n data values over a given time interval, t milliseconds, so that they are evenly spaced is to just divide the interval to n equal parts. In this case the vector would become

$$[t/n, 2t/n, 3t/n, \ldots, (n-1)t/n, t] \text{ or}$$
$$[0, t/n, 2t/n, 3t/n, \ldots, (n-1)t/n]$$

depending on our preferences. However, we run into a problem immediately if this method is used: the first drum hit in the real data cannot be assumed to happen at approximately t/n or 0 milliseconds. As it is in fact very unlikely that they do, subtracting a vector of this kind from the real data vector would display results of a significantly greater deviance in timing as there actually is, since the values don't align. The issue cannot be corrected by setting the first element to have the same timing as the first real drum hit either, since we cannot state that the timing of the first hit is exactly correct.

The method of least squares is defined in the Wolfram MathWorld entry "Least Squares Fitting" by Eric W. Weisstein [63] as "A mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets ("the residuals") of the points from the curve". In this study a linear fitting technique is used, in which a straight line is fitted in such a way that the maximum distance of any of the scattered data points is minimal. This type of least squares fitting is a form of linear regression, which is commonly used to model time series data. The desired time values are found by dividing the line to n parts and picking the values on the axis corresponding to the time interval.

We will discuss the practical application of the method in chapter 4.3, while now the idea is presented. Let the vector containing the beat timings from the human–played track be

$$t = [t_1, t_2, t_3, \dots, t_n]^\intercal.$$

The model for the perfect click beat is then of the form

$$p = [p_1, p_2, p_3, \dots, p_n]$$

where $p_j = a + jb$, $a \in \mathbb{R}$ and $b > 0$. In other words, we don't know the first value or the interval between consecutive values; only that they are evenly spaced and that there are n values.

Let's denote $x = [a, b]^\intercal$. This implies that

$$Ax = t$$

where A is an $n \times 2$-matrix such that

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \vdots & \vdots \\ 1 & n-1 \\ 1 & n \end{bmatrix}.$$

The values for a and b are found by minimizing $(A^\intercal A)x = A^\intercal t$. The minimization is performed on Matlab's "linsolve"–function, which solves a linear system of the form $Mx = B$ using QR factorization [31]. That means factorizing the $n \times 2$-matrix $A = QR$ where Q is an $n \times 2$ unitary matrix and R is a 2×2 upper triangular matrix [15]. As a result, the j:th component of the difference vector is found to be $a + jb - t_j$. See chapter 4.3 for further examples.

3.3 Noise

We have already touched upon the subject of noise on multiple occasions, and it is a paramount concept with regards to this study. A principal understanding of the phenomenon is therefore necessary in order to proceed. In mathematics noise appears and is defined widely. It is associated with for example signal processing, Fourier analysis and probability theory. A Wolfram MathWorld entry on noise by Eric Weisstein defines noise pithily as "An error which is superimposed on top of a true signal". [64] In other words, any type of random disturbance in a signal or function is considered noise. This is why noise is such a common occurrence in mathematical and physical applications; in real data we can assume signals to always contain imperfections.

Much of signal processing delves in denoising. In this thesis the character of the noise is examined. In terms of this study, the vectors constructed by the least squares method that contain the perfect timings for the beats can be considered as noiseless signal, the real data vectors as noisy signals, and the difference vectors ones that contain only the noise. As can be deduced from the frequency of noise in real world data, the presence of noise in signals is natural, whereas experiencing a completely noiseless signal can be considered unnatural. It seems all the more logical that musical signal containing noise - errors in timing - feels pleasant to listen to. In conclusion, what we aim to find is an opposite to a denoising method, and is instead a method of adding suitable noise to a signal.

We are focusing our attention to $1/f$, or *pink* noise. The article by Ward and Greenwood [30] was quoted in chapter 2 regarding the impossibility on defining $1/f$ noise mathematically. Since so little empirical information is known about this common phenomenon, it is convenient to familiarize oneself with it by examining *white noise* and *Brownian motion*; two other types of noise that have a similar appearance as pink noise in some of the same contexts. The article by Ward and Greenwood [30] describes pink noise as an "intermediary" between the two.

The types of noise under our scrutiny are characterized by their *spectral density*. This quantity is used in signal processing to measure or model the spectrum of the noise level over the whole interval (as stated for example by Michael Cerna and Audrey F. Harvey in their application note for National Instruments, "The Fundamentals of FFT–Based Signal Analysis and Measurement" [12]).

By its definition noise has no discernible structure or patterns, thus parameters need to be set for its other characteristics in order to scientifically describe it. Often spectral density is used to evaluate noise. White noise, pink noise, and Brownian motion all have spectral density of the form

$$S(f) = \frac{a}{f^\alpha} \tag{1}$$

where $a \in \mathbb{R}$ is a constant (in this case $a = 1$), f denotes frequency and α is a parameter depending on the signal [66]. The type of noise in question is characterised by the variable α having a value on the interval $[0, 2]$. When

$\alpha = 0$, the spectral density $S(f) = 1$ corresponds to white noise. Brownian motion appears when $\alpha = 2$, and the equation 1 becomes $S(f) = 1/f^2$. Thus, the spectral density of pink noise takes on the values of α on the open interval $(0, 2)$.

White noise is characterized by it having no correlation in time, containing independent samples from all frequencies, and, as defined in the previous chapter, having constant spectral density. The term is rather commonplace and widely used in everyday life, and most people are likely able to provide examples of white noise, such as the sound of rain, or static from an old television. A familiar application of white noise in sound signals may be its ability to soothe and help some people sleep, due to its smoothing effect on other background noise.

White noise can be obtained by deriving Brownian motion, or *Brown noise*, creating an obvious connection between the two. The phenomenon was discovered by a Scottish botanist Robert Brown in 1827 in the random movement of pollen particles in water [44]. Many occurrences of Brownian motion involve the movement of particles in a body of liquid, and it is strongly linked to diffusion. Therefore its applications in the fields of physics and chemistry are many [2]. In mathematics Brownian motion appears in signal processing and statistics and is linked to the phenomenon of random walks (mentioned by e.g. Ward and Greenwood [30]), and it is characterized by having stationary, independent increments (as opposed to white noise being independent of time). In Brownian motion the energy is concentrated in low frequencies. For instance vibrations from nearby running machinery can be thought of as an example of brown noise [7]. Figure 1 illustrates the fundamental differences between the three types of noise we are discussing. Brownian motion is occasionally referred to as "red noise", as in figure 1. [40]

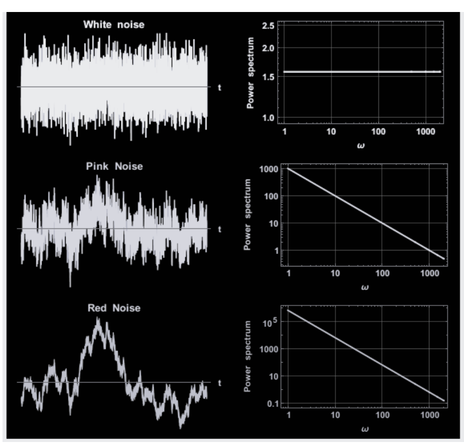


Figure 1: *Graphs of white, pink and brown noise. Source: WikiMedia commons, public domain [23]*

$1/f$ noise was first discovered in 1925 by the physicist Dr. John Bertrand

Johnson (1887 – 1970), who spent his career researching electronic circuits, oscillations, noise and various other problems [19] [30]. According to the article by Ward and Greenwood ([30] chapter "Early History") Johnson made his discovery as he was performing an experiment on shot noise in vacuum tubes as theorised by Schottky (1886 – 1976) [74] in 1918 [62]. The qualities of the noise observable in the data were not those of the familiar white noise or Brownian motion [9]. Johnson concluded he had found a new type of noise and it was initially classified as "flicker noise" by Schottky in his consequential 1926 study [61]. As catalogued in the article by Ward and Greenwood [30], the idea was further developed by Bernamont (1937), McWhorter (1957) and Bell (1960).

Another fascinating aspect of pink noise, in addition to the apparent impossibility of its exact definition, is its common occurrence in natural processes. Ward and Greenwood's article [30] lists among others electrical noise, geographical events such as tectonic movement, activity of neurons in the brain, stock market fluctuation, human reaction times, pulse, and of course, speech and music, as having attributes that contain $1/f$ noise. Some graphic examples can be seen in figure 2.

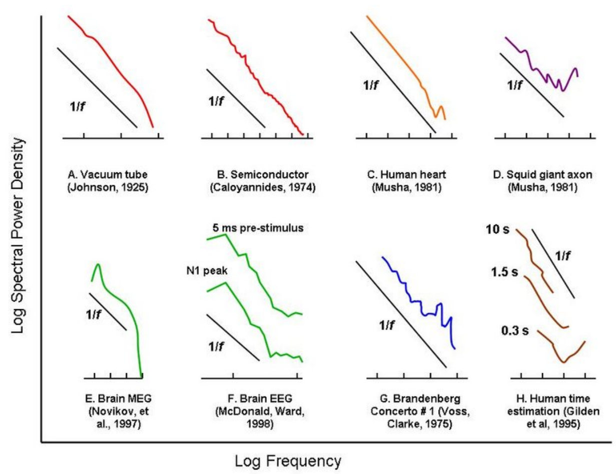


Figure 2: *Examples of* $1/f$ *noise signals. The same image can be found in the article by Ward and Greenwood [30]. Source: Scholarpedia, public domain [27]*

As stated in the article by Räsänen et al. [6], fractal analysis on musical rhythm is a new field of study as of now. There are, however, separate examples of $1/f$ noise found in rhythm and music. Voss and Clarke performed similar analysis on fluctuations in loudness and pitch in a piece of music in their 1975 article "$1/f$ noise in music and speech" [60]. Chen, Ding and Kelso found in their 1997 article "Long Memory Processes ($1/f^\alpha$ Type) in Human Coordination", when studying memory processes and reaction times, that when a subject attempts to coordinate finger–tapping to a metronome, the delay fluctuates along a $1/f$ noise pattern [14]. A claim can be made that such rhythmic fluctuations are a crucial characteristic of the human element in music.

We will go further into fractals and fractal analysis in the next chapter 3.4, but to conclude the present one, it is of interest to point out that because of its fractal pattern, pink noise is more structured than white or Brownian noise. To quote a post "Pink Noise in Neural Nets: A Brief Experiment" (2018) by Leo Korogodski on the blog "Chariot Solutions": "Taking a sample over a period of time, the distribution is Gaussian, but pink noise has a specific temporal dynamic." [26]. The entry in question is one of the few examples easily found at the time of the writing where pink noise is studied in machine learning. The subject of the post is neural networks. Korogodski goes on to compare the three different types of noise on a "spectrum of complete chaos to the more structured kinds of noise" ([26] "On Fractal Noise"), where pink noise falls the closest to "structured". Although the statement is not exactly scientific, it offers an apt analogy on the benefits of introducing pink noise to machine learning.

3.4 Fractal analysis and time series

Fractals are a mathematical concept related to the fields of chaos theory, topology and geometry, and is nowadays widely applied to computer technology and artificial intelligence. Fractal analysis is the mathematical study of fractals.

The discovery of fractals is often credited to the mathematician Benoît Mandelbrot (1924 – 2010), who suggested that the familiar geometrical metrics are not sufficiently sophisticated to model and examine nature in all its complexity (as summarized by Taylor and Pilgrim in their book "Fractal Analysis" [21], chapter 2). Mandelbrot wrote the pioneering paper on the study of fractals, "How Long Is the Coast of Britain? Statistical Self–Similarity and Fractional Dimension" [10] in 1967 upon facing a paradox in calculating how long the coastline of a given area is, and thus defining many of the basic ideas behind fractal analysis.

By his study on geographical shapes Mandelbrot was able to find and illustrate some of the fundamental properties of fractals. It is important to note that fractals are not exclusively geometrical shapes, and that any sequential forms can have fractal structure. For instance the chapter in Taylor and Pilgrim's book mentioned previously [21] describes fractal analysis on time series. The visual aid of geometrical examples is useful in understanding the elusive nature of fractals, since their infinitely complex structure is, by definition, challenging to grasp. Mandelbrot highlights in his paper that the metrics of these patterns are "undefinable", and that fractals are defined by their *self–similarity*; "each portion can be considered a reduced–scale image of the whole" ([10], abstract). He proposes that in lieu of the familiar geometrical quantities, such as length, we should measure fractals by their degree of complication, and on that basis define fractal dimensions.

The chapter "Fractal Analysis of Time–Series Data Sets: Methods and Challenges" in the book "Fractal Analysis" by Ian Pilgrim and Richard P. Taylor [21] is advantageous to this thesis, and we will discuss the methods introduced in the book futrher in chapter 4.6. It is specified in the chapter that while fractal patterns commonly appear in time series, they must be analysed with tools different from spatial data because of their two–dimensional structure. Spatial data is often assessed on the shape of its graph, which is not a convenient tool for investigating time series data, and their fractal dimension is often of interest. To quote section 3: "Time–series fractal structures" on the similarities between the two: "[...] a time-series structure may exhibit fractal scaling properties in either a statistical or an exact sense, which may be quantified using the formalism of fractal dimensions."

"Fractal Analysis" introduces different methods for analysing time series by estimating the value of the *Hurst exponent* [17], which can be treated as a measure of randomness. This is the core idea introduced in the chapter, but we will not deliberate on the methods further for the time being, since they are irrelevant in analysing the time series of this study. In practical sense, within the bounds of this thesis we are faced with a dilemma in terms of the quantity and scope of our data. Pilgrim and Taylor state as the foremost problem in fractal analysis of this kind is that inaccurate results are obtained when manipulating limited data, and when working on fine–scale detail ([21], introduction). The vectors constructed for this study amount to a few dozen and they contain only about 20 data points each on average, so it is a small data set. The book indicates that finiteness causes trouble in this approach to fractal analysis even for large data sets.

Using machine learning to detect fractal patterns is the very reason we are able to work with such small amounts of data. It is true that a machine learning algorithm is not as of yet able to explain the mathematical theory behind a reconstructed fractal noise vector, but there are some that are able to create them. Even though as scientists we are most interested in what causes the phenomena, in situations where we wish to find an elegant solution to a practical problem, machine learning may prove revolutionary. If we are constantly obstructed in fractal analysis of time series by data being finite, machine learning is likely the best means to obtain reliable results.

Alternate methods of studying shorter time series have not been able to produce results comparable to machine learning data in accuracy. Berndt Pilgram and Daniel T. Kaplan introduce in their 1998 article "A comparison of estimators for $1/f$ noise" [11] methods beyond the Hurst exponent for analyzing long–term correlation in fractal noise. MLE, or maximum likelihood estimation of α (in the function $1/f^{\alpha}$) is deemed as the best method of the five studied in the article, with FFT (fast Fourier transform) regression being a close second. Evidently, the results given by MLE are superior, but the method was computationally so heavy for 1990s technology that Pilgram and Kaplan deemed FFT, which was much faster to compute, as as good of a choice. Pilgram and Kaplan also examine detrended fluctuation analysis (DFA) in their article, which was the method used in the paper by Räsänen et al. [6]. The article "Fractal analysis for "short" time series: A re–assessment of classical methods" by Didier Delignieres et al. (2006) [28] re–evaluates various methods of fractal analysis for time series, including the ones mentioned in this chapter.

All three articles indicate that there are various advantageous methods that produce certified results of $1/f$ analysis of these series. However, the results appear rather dependent on the estimators and parameters used, thus implicating that their interpretation requires a level of subjectivity. Räsänen et al. mention an estimation error of up to 10% in DFA in their article (p. 6, [6]), and as stated by Delignieres et al.: "[...] each method seemed to present specific advantages and drawbacks, in terms of bias or variability." (chapter 4, [28]). In analyzing signals the methods in question have a lot of merit, and as was already mentioned, machine learning algorithms are incapable of producing analytical results of this sort. However, for generation of $1/f$ signals the "classical" methods catalogued in this chapter are too inaccurate, and that is exactly why machine learning processes are essential here.

4 Method

4.1 Choice of samples

In this section we will describe the course of the study, starting with a depiction of our data.

The aim of this study is to detect characteristics that are specific to a certain style of playing drums in popular music. We will call the defining quantity by which we have divided the data in two **skill**. It is however necessary to recognize that when discussing art, skill is not an objective or quantifiable phenomenon. In this context we are referring to technical proficiency in playing an instrument. The level of skill does not translate to a musician being "good" or "bad", as there are numerous, more or less subjective factors to what constitutes as someone being a good musician, or a piece of music being good. For example song–writing skills, appearance, originality, timbre, performance style and the audience's preferences effect the reception of music.

There were certain limitations to consider when selecting data. Firstly, time management has to be taken into consideration in a master's thesis study; the number of samples had to be such that processing them would take up a realistic amount of time. Another requirement for the data is that the samples should be comparable to one another at least to a degree. Different styles and genres in music are innumerable, and if we were to construct one of our two data sets solely of samples of, say, classical jazz, and the other one of pop, differences between the two sets would be more significantly due to vast differences in how jazz and pop are played, than in skill. As a third observation, we should note that the use of synthesizers, drum machines and click tracks in music production in recent years is the standard across all genres; even in "live" recordings computer software can be used in post–production to correct rhythmic delay. If we were to choose a modern piece of music at random, it would be quite unlikely that the beat can be treated as an example of natural human musical rhythm. Since we are not specifically interested in new music, it is notably easier to choose samples predating the generalization of accessible music software.

In conclusion, the samples used in this study were chosen to abide by the following parameters:

- They should be of succifiently similar genres.
- Their publication should predate preferably the 2000s.

The artists were chosen quite arbitrarily within these bounds; choises were made based on personal knowledge and music preference, and on suggestions by others. For instance, the writer's father was asked who he considers to be technically skilled drummers.

The real data vectors (see chapter 3.1) should contain approximately 20 data points each. The practical meaning of this is that we need to choose songs where the listener is able to detect by ear a sequence of an individual drum played roughly 20 in periodic rhythm. The interval must be contained in one segment of the song, so that for example one verse or chorus contains the interval in its entirety. The reasoning behind this is that it is fairly common in popular music that different sections of the song have a different drum pattern. For this study we have chosen two songs per each drummer, and three vectors are collected from each song. The songs selected have a 4/4 time signature and roughly follow the classical structure of a pop song, in which there is an intro, verse, possibly a bridge, chorus, verse, possibly another bridge, another chorus, solo, a final chorus, and possibly an outro.

To represent the group of the higher technical proficiency we selected mainly progressive rock drummers from the 1980s and 1990s. Jeff Porcaro's (1954 – 1992) playing was analyzed in the article by Räsänen et al. [6] as well. He was an iconic session musician who has recorded with countless artists, and is best known as the founding member and drummer of the American band Toto [69]. We collected samples of the snare drum track from two verses and the chorus of "Hold the Line" [56], and the first verse, chorus, and guitar solo of "Rosanna" [57] by Toto. Another renowned session musician chosen for our analysis is Jonathan "Sugarfoot" Moffett (b. 1954), whose collaboration with the Jackson family spanned over decades, and whose playing can be heard on recordings by for example Madonna, Quincy Jones and Elton John [71]. Here we are examining three snare drum tracks in "Fractal Zoom" by Brian Eno [18] and in the chorus, first verse and solo in "You Never Listen to Me" by Peter Cetera [13]. From the Canadian progressive rock band Rush we have chosen to study different cymbal tracks in "Subdivisions" [47], and the tom drum sound in "Tom Sawyer" [48]. The recordings were made by the drummer Neil Peart (1952 – 2020) who was famous for his disciplined performance [38]. We will also be analyzing the playing of Frank Beard (b. 1949), who is best known as a member of the American band ZZ Top [68]. Our vectors contain bass drum tracks from different segments of the song "Legs" [55], and snare drum tracks from the outro, chorus and verse in "La Grange" [54]. Finally, we chose to explore the playing of Mike Portnoy (b. 1967). He studied in a music program at college on a scholarship and was a founding member of the band Dream Theater. He has since participated in countless other musical projects, and has received multiple awards and denominations for his playing [35]. We have collected samples of a snare drum track in the second verse of "Another Day" [52] by Dream Theater, bass drum tracks in the solo and bridge of the same song, and snare drum sounds in two choruses and the first verse of "Wither" by Dream Theater [53].

The selection of the second group required slightly more reconsideration, since we wished to find musicians whose technical skills in their instrument are sub–par, but who have found success and admiration for their music in spite (or because) of it. Eventually, a handful of punk rock drummers from the 1970s were chosen to be studied. We also had at our disposal some live, unmixed studio recordings by the writer's father's, and supervisor's bands, which we felt belonged to this set of samples rather than the one introduced in the former chapter [36]. From those samples three tom vectors were isolated from the former recording, and six snare drum vectors from the latter. It should be noted that a click track was used when recording those last samples, although interestingly the resulting sample does not suggest that.

The first artist under scrutiny in this group is Jerry Nolan (1946 − 1992), who was notorious for his tendency to speed up the beat during the song. He started his musical career in the American band New York Dolls, whose members only began to learn to play their instruments upon forming the band. After the New York Dolls, Nolan went on to play drums for Johnny Thunders & The Heartbreakers [70]. We have samples of bass drum tracks in "Looking for a Kiss" by the New York Dolls [16], and bass drum tracks from the chorus and hi–hat tracks from the verse and break of "Pirate Love" by Johnny Thunders & The Heartbreakers [20]. The Ramones, perhaps the most influential punk rock band of all time, were an obvious choice for our study, since the inspiration for their music was drawn exclusively from its members' deficient skills in playing any instrument. We will be investigating the original dummer, Tommy Ramone's (Tamás Erdélyi, 1949 − 2014 [73]) playing on the band's admittedly under–produced first album. We collected bass drum hits from the verses and one chorus of "Havana Affair" [45], and from the first verse, solo, and chorus of "I Wanna Be Your Boyfriend" [46]. Scott Asheton (1949 − 2014) was a very similar artist to the previous two mentioned; he was best known as the drummer of Iggy & The Stooges, and was taught to play drums only upon his joining in the band by the singer Iggy Pop [72]. The drum tracks studied here are collected from the snare hits in the intro, chorus and bridge of the song "Dirt" [49], and first verse, chorus, and outro of "No Fun" [50]. Paul Cook (b. 1956) of the Sex Pistols fame completes the selection of musicians. Although Cook has then had a successful career as a skilled and sought–after drummer, his playing on the Sex Pistols' albums was still unpolished [41]. It can, however, be argued that what sets the Sex Pistols slightly apart from their American counterparts introduced here is that their playing "badly" was influenced by their desire to appear so, and as such the humanly errors in delay are not entirely due to internal groove. For our study we selected tom tracks from the intro, outro, and second verse of the song "Submission" [43], and snare drum tracks from two different verses and a bridge in the song "Holidays in the Sun" [42].

4.2 Software and Programs

We will briefly introduce the programs used in this study. The project was composed from real data, and as such was heavily reliant on computer software. All data was processed and all computations were performed using the applications mentioned in this chapter.

The principal tool at our disposal was Matlab's desktop application (version R2022a). Matlab is a programming language well–suited for mathematical imaging and analysis. All computations were performed and all the graphics were created in Matlab. Matlab has a built–in PCA (see chapter 4.5) function which made it both possible and convenient to run the machine learning algorithms on the same software, deeming it unnecessary to transfer data externally between different platforms.

Audacity was used for manual audio processing. It is a free, open source software designed for easy and accessible audio editing, that runs on all the major operating systems and supports most audio formats. More information can be found on their website at "audacityteam.org" [8]. A piece of music can be recorded directly in Audacity from an internal or external source, and converted into an MP3 file. Once the drum tracks were isolated on the Moises Application, which we will introduce shortly, they were reuploaded in Audacity and truncated to the desired length on the application. This could have been done with Matlab as well, but it was observed as convenient to collect all the samples as individual MP3 files, which was very easy to do in Audacity.

Having a set of isolated drum tracks was of crucial importance to this study, and their availability is not a trivial matter. Today's standard in recording music in studio conditions is to record each instrument in its own, separate track. These days the recordings are digital, and computers are capable of storing large amounts of data, so a recording containing nothing but drums would very likely exist for a recent piece of music, thus making our quest only a question of access. An immediate observation of this fact is that it limits greatly the number, style, and origin of recordings one might be able to use for an analysis of this sort. A major share of the samples chosen for this thesis were published before the 1990s, and most of the recording materials have likely been lost over these last decades on account of the inconvenience and unnecessity of storing them. The potential to study the playing styles in 1970s punk rock is particularly sensitive to what materials and software are available. The original recordings are analogue, and it was typical to the genre to produce albums on a budget. That means in practice that multiple instruments were often recorded on the same track, and the final result often contains errors, such as distorted or muffled sounds. For instance the group Iggy & The Stooges only used three tracks when recording their 1973 album "Raw Power": one for vocals, one for lead guitar, and one for everything else, thus making the album nearly impossible to remix later on. We can deduce the problem in a comprehensive research of different playing styles of a certain instrument to be the following: isolated tracks can not be mechanically recovered from many recordings.

In this thesis an AI–powered music software called the Moises App was used for separating the tracks, and it proved crucial to the process. The application is provided by Music AI, a company that creates and provides AI platforms for music and audio production [3]. In this study we used the online version of Moises App, which can be found at https://moises.ai [4]. A registered user can upload a piece of music as an MP3 or WAV. file, after which the application runs an AI algorithm that detects the sounds of different instruments played in the recording, and separates them into their own tracks. In this fashion we obtained complete drum tracks for each piece of music we chose for our analysis.

Further details on how the application operates are beyond the scope of this thesis, but the technology is revolutionary. The application of artificial intelligence in art and creative fields is a very recent development. A methodical, mathematical approach to a phenomenon with so many dimensions and such a complex, almost chaotic, structure as music seems an inconceivable idea.

4.3 Processing the sample

This section describes in detail how the difference vectors are constructed from the basis of our collected data. Two of the samples are chosen to illustrate the process throughout. One of these is Jerry Nolan's hi–hat track in the second verse of "Pirate Love" by Johnny Thunders & the Heartbreakers [20]. As a comparison, we will also examine the construction of the snare drum difference vector in the chorus of "Hold the Line" by Toto [56], and the playing of Jeff Porcaro.

The time stamps for the individual hits were hand–picked from the plotted signal. This, of course, adds to the element of error, but in an insignificant amount for our purposes. Ideally, the exact timings could be selected by applying a filter, but finding the desired beats by hand and eye allows the investigation of less dominant sounds just as conveniently as the most prominent ones. Our method is also advantageous in the case where the signal is damaged or muffled, or there are fluctuations in loudness, as there often is in live recordings. The desired sounds were identified by listening to the sample and comparing the audio to the graph of the signal.

The time stamps were then expressed as an $n \times 1$–vector, where n is the amount of hits. The "real data" vector for Jeff Porcaro's sample corresponding to the graph in figure 3 is

$$\begin{aligned} t_p =& [61698, 116198, 146075, 175009, 201996, 231664, 259468, 284662, 312714, \\ & 341566, 369893, 396385, 425762, 452515, 479218, 534131, 562193, 592057, \\ & 619249, 646869, 676778, 704580, 750285, 759095, 785932, 814251] \end{aligned}$$

where the $n = 26$ values represent timing in (tens of) milliseconds. On the graph they can be seen as represented by the most dominant peaks and the slightly smaller but easily detectable ones between them. The difference in their volume is caused by the bass drum being played on every other hit of the snare drum. There were samples where in similar situations the timings of two drums hit simultaneously were not synchronized, and its occurrence is clearly visible in

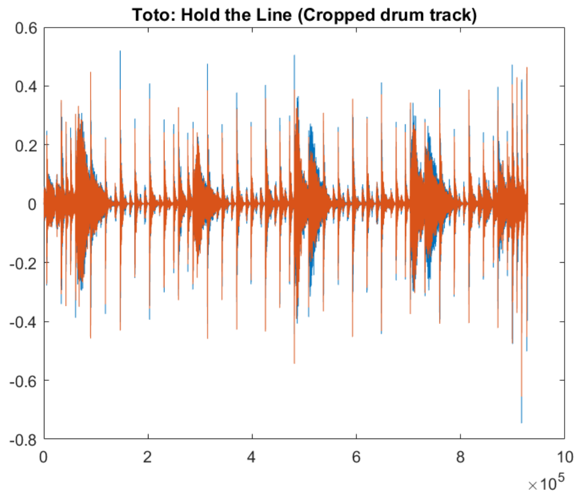


Figure 3: *The audio signal for the isolated drum track in the chorus of "Hold the Line". Drawn in Matlab.*

the plot. In figure 3 we can see that the two concurrent hits are so simultaneous that mistakenly selecting the bass drum hit on the graph would produce the same error as can already be expected from hand–picking the values.

The 35×1–vector depicting Jerry Nolan's sample (collected from the graph in figure 4) is

$$\begin{aligned} t_n =& [9320, 30370, 52714, 71790, 93156, 112993, 134168, 151909, 171938, 191468, \\ & 211999, 230311, 250159, 269731, 290303, 310142, 330172, 349975, 370719, \\ & 390643, 411054, 432036, 455732, 469762, 490790, 509866, 530278, 549289, \\ & 570338, 589568, 610397, 630427, 650648, 670220, 690256]. \end{aligned}$$

In the graph we can find the hi–hat hits a again at the points of the most prominent hits, and the less prominent ones centered between any two of those. This is again due to another drum being played simultaneously.

From these vectors we can construct vectors containing the corresponding "ideal" beat. The computations were done in Matlab (as demonstrated in figures) using the least squares method, which was described in chapter 3.2. To recapitulate, the goal is to minimize the equation

$$(A^\intercal A)x = A^\intercal t \tag{2}$$

where t is the real data vector, and A is an $n \times 2$–matrix, where the left–hand side column contains only row values of 1, and the right–hand side has the integer values between 1 and n in descending order.

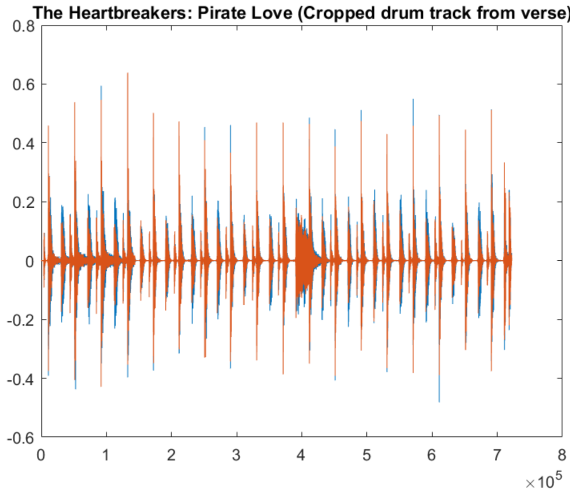


Figure 4: *The audio signal for the isolated drum track in the verse of "Pirate Love". Drawn in Matlab.*

```
n_jn5 = length(t_jn5);
A_jn5 = zeros(n_jn5,2);
[rows, cols] = size(A_jn5);

for iii=1:rows
    A_jn5(iii,:) = [1,iii];
end
```

Figure 5: *The creation of the matrix A with a for–loop in Matlab.*

Figure 5 shows how the matrix A is created. The script is very straightforward to follow for a reader with any experience in coding, so the explanation is kept brief here. This is a fragment of the code written to generate the difference vector for one of our examples, the chorus for "Pirate Love" [20]. The notation follows a personal system intended to keep track of all the data. Each sample was issued its own index, such as $jn5$ ("Jerry Nolan, sample no. 5") in the example, and the variables for each sample have the same name. As an example, the variables in the code behind our other example are indexed with $jp2$ ("Jeff Porcaro, sample no. 2"). In figure 5 we assign the length of the data vector as a value, and use that figure to call an empty matrix with the desired dimensions. Its rows and columns are then set as separate variables, and a for–loop is created that goes through all the rows of the matrix and replaces the values $[0, 0]$ with $[1, m]$, where m is the number of the row.

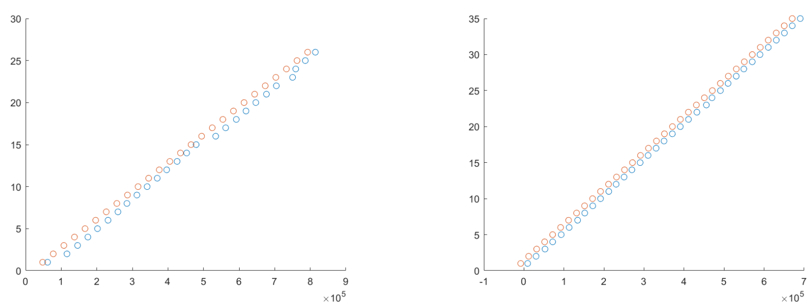
The script remains rather simple going forward, as can be seen in figure 6. The equation mentioned on the first line is the equation 2 with which the least–squares solution can be found. The "linsolve" function finds those solutions and stores them in a 2×1–vector x. We recall from chapter 3.2 that the j:th element of the "ideal" vector is of the form $a + jb - t_j$, where t_j is the j:th element of the real data vector, and a and b are unknown constants. Those constants can now be found as the values of x. The ones of this example are $a = -0.81 \cdot 10^4$ and $b = 1.9949 \cdot 10^4$ milliseconds. These values are stored in the variables $e5$

and $f5$ in the code in figure 6.

```
% Rename the right & left-hand sides of the equation for simplicity

A1_jn5 = (A_jn5.')*A_jn5;
B1_jn5 = (A_jn5.')*t_jn5;

% And now we can solve the equation for the ideal click beat:

x_jn5 = linsolve(A1_jn5,B1_jn5);

e5 = x_jn5(1,1);
f5 = x_jn5(2,1);
t_jn52 = zeros(n_jn5,1);
for i = 1:n_jn5
    t_jn52(i,:) = [e5+(i-1)*f5]
end


% We can now create the difference vector

t_diff_jn51 = t_jn52-t_jn5;
t_diff_jn5 = t_diff_jn51.';
```

Figure 6: *Matlab code used to find the "ideal beat" and construct the difference vector for "Pirate Love".*

The for–loop in the script is used to construct the time series vectors for a perfect beat element by element. In figures 7a and 7b those are plotted against the real data. Not much information is gained from these scatter plots, which is to be expected as the fluctuations from the click track are minute. One interesting observation we can visually detect is that the recorded drum hits appear to be consistently late from the perfect timing, as opposed to early. If we examine the plot 7b closely, it almost looks as if the two vectors are parallel and that the delay approaches a constant value.



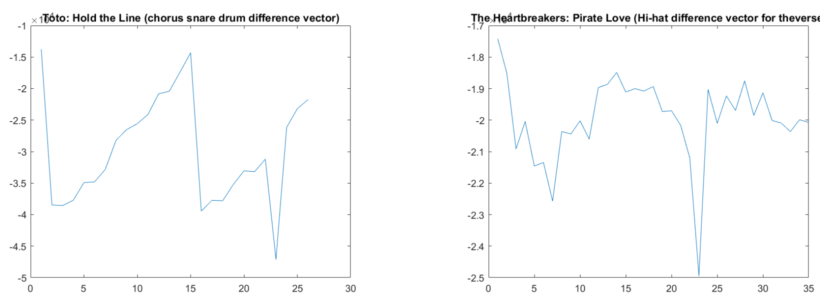
(a) The data from "Hold the Line" (b) The data from "Pirate Love"

Figure 7: *Scatter plots comparing a click track (red) to the timing of the actual drum beats (blue).*

In the final lines of script in figure 6 the difference vector is finally created by subtracting the actual hits from the ideal ones. At last, the vector is transposed to a $1 \times n$–vector. For our two highlighted examples they are the following:

$$
\begin{aligned}
t_{jp} = 10^4 \cdot [&-1.3751, -3.8469, -3.8564, -3.7715, -3.4920, -3.4805, -3.2827,\\
&- 2.8238, -2.6508, -2.5578, -2.4122, -2.0832, -2.0426, -1.7397,\\
&- 1.4317, -3.9448, -3.7728, -3.7809, -3.5219, -3.3056, -3.3183,\\
&- 3.1202, -4.7125, -2.6153]\\
t_{jn} = 10^4 \cdot [&-1.7420, -1.8521, -2.0915, -2.0042, -2.1458, -2.1346, -2.2572,\\
&- 2.0363, -2.0443, -2.0023, -2.0605, -1.8967, -1.8866, -1.8488,\\
&- 1.9111, -1.9001, -1.9081, -1.8935, -1.9729, -1.9704, -2.0165,\\
&- 2.1198, -2.4944, -1.9025].
\end{aligned}
$$

The machine learning model is then created with similar vectors, which will be described in an upcoming chapter (4.5). Before moving on to the subject of principal component analysis, we will visually assess the graphs of these vectors. Those corresponding to the two examples can be found in figure 8.



(a) The graph for "Hold the Line" (b) The graph for "Pirate Love"

Figure 8: *Graphs of the time series difference vectors of the two examples.*

Let us direct our attention to the graph in figure 8b first. This graph depicts the fluctuations in timing in Jerry Nolan's hi–hat playing in "Pirate Love". What we are looking for in this visual assessment is any apparent pattern or seasonality in the delay. In the one in question, there is none. One drum hit occurs with a significantly greater delay than the others (valued at $-2.4944 \cdot 10^3$ milliseconds), but besides that the delay seems to fluctuate randomly over an interval of approximately $0.4 \cdot 10^3$ milliseconds. However, the graph in figure 8a (visualizing Jeff Porcaro's playing of the snare drum in "Hold the Line") is very interesting in the sense that one can immediately see that there is a clear pattern in how the delay fluctuates. Furthermore, the shape of the graph resembles a fractal. A self-similar shape found in a singular, short time series sample is not evidence of a piece of music containing $1/f$ fluctuations more extensively, but it is very intriguing that indications of the occurrence can be found so easily.

Further investigation of the graphs of the vectors created for this study reveals that the samples collected from the same group representing either a high or low level of technical skill do not consistently share the same visual characteristics. The difference vectors created from the bass drum track in the chorus of ”Havana Affair” by the Ramones [45] and the third verse of ”Holidays in the Sun” both seem to mimic a pattern similar to that of ”Hold the Line” in figure 8a. We recall that the drummers behind these tracks, Tommy Ramone and Paul Cook, are both representatives of another group than Jeff Porcaro. The graphs are included here in figure 9.

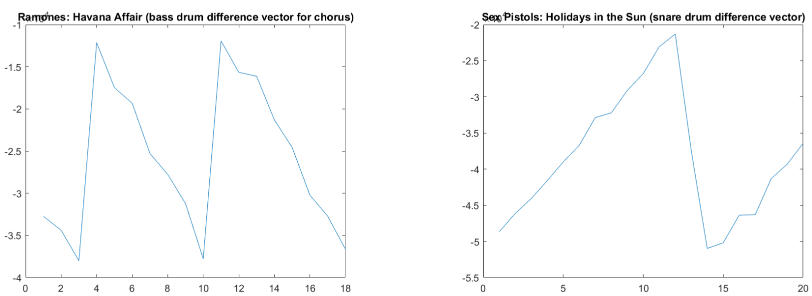


Figure 9: *Graphs of the time series difference vectors from ”Havana Affair” and ”Holidays in the Sun”.*

A recurring outline among the graphs of the difference vectors is one where a semi–evenly decreasing or increasing delay can be detected except for one hit that deviates greatly from the steady regression. Examples of this can be seen in figure 10. At least one such vector was constructed per almost every drummer studied in this thesis regardless of their skill level. The illustrated examples are the difference vectors for Tommy Ramone’s bass drum track in the chorus of ”I Wanna Be Your Boyfriend” by the Ramones [46], and the snare drum track in the verse of ”La Grange” by ZZ Top [54], played by Frank Beard. A major portion of all the vectors appear to consist of random fluctuations similar to those in figure 8b. As a final visual observation on the data we might suggest that some common characteristics can be seen in the graphs corresponding to the same drummer. Visual interpretation is subjective as is, and detecting any similarities with this method is ambiguous at best, but some leeway should be allowed in attempting to scientifically analyse any form of art. For example, when difference vectors based on Frank Beard’s playing are plotted, we can see smooth curves as opposed to jagged saw–tooth shapes (figure 11a presents a vector of the bass drum track in the verse of ”Legs” [55]). The vectors based on Rush’s Neil Peart’s playing, on the other hand, depict much gentler slopes than the other drummers studied here. A vector of a ride cymbal track from the intro of ”Subdivisions” [47] can be seen in figure 11b.

What we have in fact managed to compile from visually analysing the difference vectors is that we can’t see patterns in delay characteristic to a specific

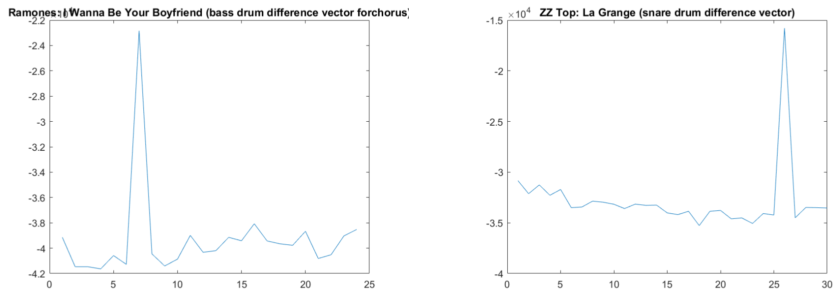
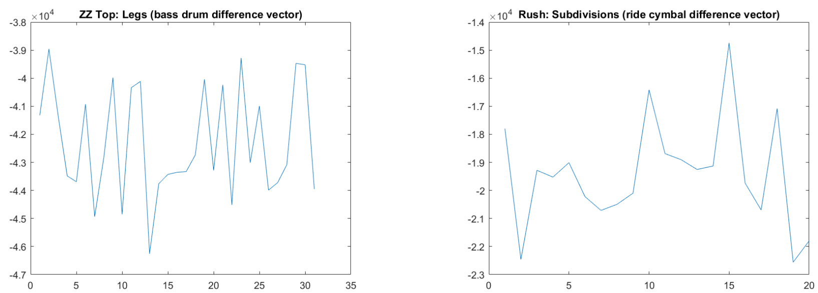


Figure 10: *Graphs of the time series difference vectors from "I Wanna Be Your Boyfriend" and "La Grange".*



(a) Curves in the graph appear smooth.

(b) Gentle slopes.

Figure 11: *Graphs of the time series difference vectors from "Legs" and "Subdivisions".*

data group. This is not surprising. The hand–picked data values fall within a margin of error, the fluctuations in rhythmic delay are very small, the Matlab plots are not an accurate depiction of the data, and, most importantly, the human element we intend to analyse is far too complex to be discovered with the naked eye. Machine learning algorithms are suited for this type of analysis exactly for their potential for detecting patterns a human cannot. We will move on to discussing time series modeling and principal component analysis in the following chapters.

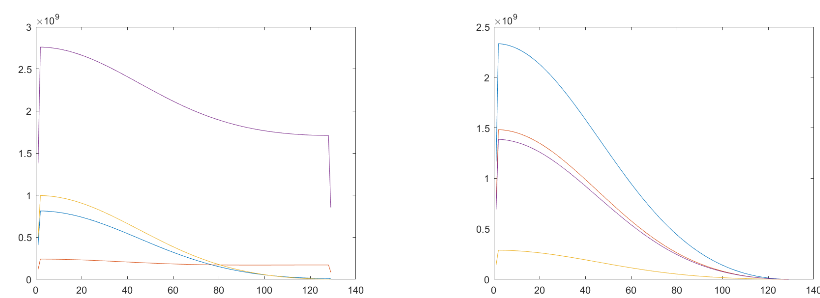
4.4 Power Spectral Analysis

In this chapter we will finally begin examining the quality of noise in our data. The article "$1/f$ noise" by Ward and Greenwood [30] ends in a chapter on identifying $1/f$ noise, in which some algorithms and methods are introduced, but the conclusion is stated that a canonical method of identification does not exist. Here, the power spectral density of our data is studied using Matlab. Power spectra are usually computed using fast Fourier transform, but it can also be calculated from the correlation function or the maximum entropy method [65]. Matlab has various built–in functions for determining the power spectrum of a vector. The command used here (*pwelch*) calculates Welch's power spectral density estimate (see [33] for further information).

We recall from chapter 3.3 that the power spectral density (often referred to as power spectrum) of pink noise is determined by the function $S(f) = a/f^{\alpha}$ (equation 1 in 3.3), where $a \in \mathbb{R}$, f is the frequency of the signal and $0 < \alpha < 2$. If $\alpha = 0$, we are working with white noise, and if $\alpha = 2$, brown noise.

Following the method in the article by Räsänen et al. ([6] figure 4 p. 7) we can analyse the results by detrending the power spectral data and finding the slope of the trend. The slope then corresponds to the exponent α. We gathered the difference vectors we previously created in two matrices, and ran the *pwelch*–function in Matlab on each vector, which performs the power spectral analysis. As a result we had two matrices containing the power spectra of all the vectors in each dataset.

One aspect worth noting is that the vectors weren't of uniform size, which they need to be for the construction of the matrices. For each group, two different–sized matrices were eventually constructed, one containing vectors of 20 samples and one of 15. The vectors that were shorter than the desired length were completed with zeros (so that the measurements were centered in the vectors), and the usage of $15 \times a$–matrices (where a is the number of vectors) is rationalised by that the amount of "missing" values in the analysis is thus smaller than in using $15 \times a$–matrices. Figure 12 shows the graphs of the power spectra of a few time difference vectors.



(a) Lower technical skill level. (b) Higher technical skill level.

Figure 12: *The principal component scores of the raw data.*

4.5 Principal component analysis and time series modeling

In this chapter we are going to continue the discussion on time series and forecasting that we began in chapter 3.1. Henceforth, we will place emphasis on the practicalities of this study, and the description of how principal component analysis was applied to the data at hand. We will delve into that after a brief discussion on time series modeling in general.

The scope of the subject is wide and its applications and benefits are many. Some examples of time series analysis in different scientific fields are mentioned in chapter 3.1. Forecasting is perhaps the most important and beneficial operation that a time series can be harnessed to, and modeling in time series is principally used for that purpose. Detecting trends and seasonality in standard time series is not a challenging task for a computer program, and a lot of the time it can be even computed by hand without much difficulty, although the large amounts of numerical data that are often required make it tedious and futile.

It is rather obvious that machine learning is extensively utilized in forecasting nowadays. The automated process, and potential for disregarding human error associated with such methods generate efficient time management and a relatively high level of reliability in results. That is when the algorithm is tested and proven to produce dependable data, naturally. The demand for automated forecasting tools in data analysis can be illustrated by considering the scope of the field of weather forecasting alone. The market for machine learning fit for similar purposes is clearly vast and diverse, which in turn means that the methods have been studied and developed from the moment their use in statistical forecasting became possible.

Various types of regression models are most regularly used in modeling the kind of time series data as we have collected. The book "Introduction to Time Series and Forecasting" by Brockwell and Davis (1996) [39] places emphasis on ARMA (autoregressive moving average) and ARIMA (autoregressive integrated moving average) models. Researching machine learning models used in time series analysis generally yields them as top results as well (for example blogs on the websites neptune.ai [24], Dataconomy [25], and Forbytes [59] have lists of efficient platforms). General linear models might be suitable for some datasets as well. For example the novel open source platform H2O does not provide ready–to–use autoregression algorithms, but the library contains linear regression models, as can be seen listed on H2O's Wiki page [67]. The versatility and adaptability of the algorithms available on such platforms might prove more advantageous in some cases than the availability of the most desirable frame for an algorithm.

The data of this thesis was investigated with **principal component analysis**, often abbreviated as PCA. The purpose of said method is to reduce a data set to its principal components, which could be described as the most prominent dependent variables [51]. In the context of this study, the idea is that running a PCA algorithm on our data recognizes and stores the most commonly occurring fluctuations in generated vectors. *Principal components* of a data set are defined in the article "Principal Component Analysis: A Review and Recent Developments" (2016) by Jolliffe and Cadima as linear combinations

$$Xa_k = \sum_{j=1}^{p} a_{jk} x_j \tag{3}$$

where the values a_k are the eigenvectors of a $p \times p$ covariance matrix S. These principal values correspond to an $n \times p$–matrix X, which contains the data in vectors, denoted in the article and equation 3 as x_j. The linear combinations Xa_k are uncorrelated and have maximum variance. ([51] chapter 2(a), p. 3)

Simplicity is the primary benefit in using PCA. The algorithm is suited for small and larger data sets alike, and distributional assumptions are not required [51]. Of course, the more widely used ARMA and ARIMA models are more sophisticated and yield more detailed results, and more accurate predictions, but application of autoregression algorithms requires knowledge about the parameters and structure of the data which PCA does not [58] [1]. There is a remark in the introductory chapter of the article by Jolliffe and Cadima ([51] p. 2) that PCA is a good exploratory tool, whereas one can deduct that autoregression models are superior for forecasting purposes. Said quality suits the purposes of this study well, since for the time being our interests lie mainly on finding patterns or shared characteristics among samples. The construction of actual drum machines that generate beats in a certain style will most likely benefit from the use of more predictive algorithms, but as we are still forming a theoretical basis for the process it suffices to use simpler tools. Applying a simple, flexible model is supported by the aforementioned argument as well. As was mentioned in chapter 2, time series analysis of this type is a new field of study. Obtaining information by the more elementary methods such as PCA adds to a yet small database on the subject while building a solid foundation for developing more complex options for practical applications in the future.

The objective in using principal component analysis in this thesis is to find similarities in the power spectra, and therefore in the behavior of noise, in the difference vectors. Matlab has a built–in PCA function which was applied to our collection of power spectrum vectors. Running the PCA algorithm in question does not require anything of the user apart from data entry, but we will briefly describe how principal component analysis is done by hand in order to illustrate what the Matlab function does. For detailed instructions, the technical report by Lindsay I. Smith, "A Tutorial on Principal Component Analysis" (2002) [22], is very helpful.

The data is first stored into a matrix, as described in the previous chapter on power spectral analysis 4.4. The difference vectors were stacked to form $20 \times m$– and $15 \times m$–matrices, where m is the number of samples. All the resulting power spectral vectors were of size 129×1. For the set representing a higher level of skill we have $m = 30$, and the lower $m = 33$. Following the tutorial by Smith ([22], chapter 3.1), the mean of the data needs to be subtracted from the values to produce a data set with mean 0. Let's denote our dependent variable as t_k for the time stamp in the kth sample vector. The subtraction procedure should be performed on the vectors independently, i. e. each mean $\bar{t_k}$ needs to be calculated separately and subtracted from the data values of t_k.

Principal values are derived from the eigenvalues of the covariance matrix of the data matrix, as defined by Jolliffe and Cadima [51] and demonstrated in equation 3. Thus, the covariance matrix for the aforementioned scaled data values and its eigenvalues must be determined before we can move on to PCA.

Covariance describes how much the vectors t_k fluctuate in mean with respect to one another ([22] chapter 2.1.3), as opposed to e.g. variance, which can only be applied to studying the data within one dimension. Let t_k and t_l denote any two of our vectors. The covariance between them is determined by the formula

$$cov(t_k, t_l) = \frac{\sum_{1=1}^{20}(t_{k_i} - \bar{t_k})(t_{l_i} - \bar{t_l})}{19}.$$

Positive covariance indicates positive correlation between the samples whereas negative covariance indicates negative correlation. We want to illustrate parallels between all the samples in the m–dimensional data set, which is achieved by constructing a covariance matrix

$$S = \begin{bmatrix} cov(y_1, y_1) & cov(y_1, y_2) & \cdots & cov(y_1, y_m) \\ cov(y_2, y_1) & cov(y_2, y_2) & & \vdots \\ \vdots & & \ddots & \vdots \\ cov(y_m, y_1) & \cdots & \cdots & cov(y_m, y_m) \end{bmatrix}.$$

The absolute values that deviate distinctly from 0 indicate strong correlations between select samples. Finding the eigenvalues of this matrix is a relatively trivial process. The eigenvectors a_k (see equation 3) are those that satisfy the equation

$$Sa_k = \lambda a_k,$$

where $\lambda \in \mathbb{C}$ is an eigenvalue. These can be found by calculating a matrix's determinant and characteristic function, but built–in functions in Matlab and other programming software produce the same result faster and with no need for performing basic calculus.

The definition of principal components as stated by Jolliffe [51] was presented on page 27 as a sum of the eigenvectors of a covariance matrix. Now that we have found the principal components they need to be rearranged for easier interpretation of the data. Chapter 5 of Smith's tutorial [22] states that the eigenvector containing the highest eigenvalue is the **principal component** of the data set. Accordingly, the eigenvector that contains the largest eigenvalue of those remaining is the second most significant one, and so on. The vectors are then rearranged from the highest to the lowest value column by column. Smith denotes this matrix as a *feature vector* ([22] chapter 5). The Matlab function we have been referring to sorts the principal components by using a singular value decomposition (SVD) algorithm by default, which finds and ranks the singular values of a matrix from the largest to the smallest.

The tasks we have listed have just been preparatory steps for the actual PCA, which in itself is a simple procedure. As Smith presents it, the desired data is obtained by transposing the feature vector and the original data matrix X, and multiplying them in that order ([22] chapter 5).

Matlab offers multiple output options for its PCA function, as can be seen on the Mathworks documentation for PCA [32], starting from the rudimentary option of only printing out the principle component coefficients. In our analysis we opted to use the largest possible number of output matrices. By doing so, for each matrix we perform PCA on we gain the following numerical data: the coefficients we just mentioned, the principal component scores (the linear combinations X_{ak} in equation 3), the principal components i. e. the eigenvectors of the covariance matrix, and the percentage of the total variance explained by each principal component.

Matlab runs the PCA algorithm autonomously, as well as optimizes the alignment of the data and arranges the principal components, as has already been stated. Only the data matrix is needed as input. The single line of code in figure 13 is where the algorithm is run on the matrix stacked above it. The additional parameters "Rows" and "pairwise" trade the SVD algorithm in for an eigenvalue decomposition algorithm, which improves the results when the data contains missing values. Eigenvalue decomposition dismisses those when calculating covariances, whereas SVD does not.

Before analysing the power spectra of the difference vectors, PCA was run on the difference vectors as well to find points of correlation in delay patterns. We looked for similar fluctuations in timing occurring in multiple samples at a corresponding or comparable time. In other words, we are looking for similar cyclical alteration in delay between samples.

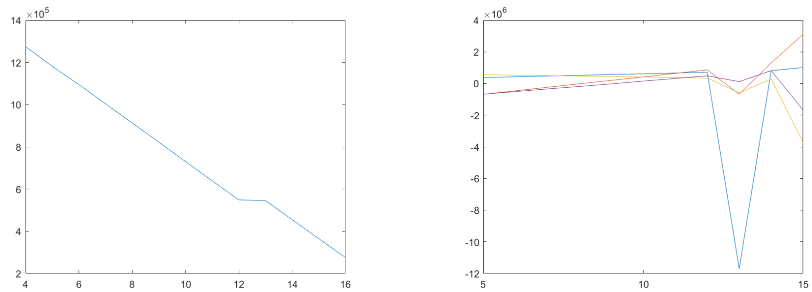
```
% Arrange into a matrix (15x33)
Bad = [am1; am2; am3; tr1; tr2; tr3; tr4; tr5; tr6; jn1; jn2; jn3; jn4; jn5;
    jn6; sa1; sa2; sa3; sa4; sa5; sa6; t1; t2; t3; s1; s2; s3; pc1; pc2;
    pc3; pc4; pc5; pc6].';

% Perform PCA with as many response matrices as possible
[coeff,score,latent,tsquared,explained,mu] = pca(Bad, 'Rows','pairwise');
```

Figure 13: *How the Matlab function for principal component analysis was used.*

The explanatory vector for principal component analysis on the data set representing the lower level of technical skill yields that 99.92 percent of the fluctuation in variance is explained by the principal component, which along with its coefficients is stored in the first column of the PCA matrix. The graph of the vector can be seen in figure 14a.

The explanatory vector for the set of technically skilled drummers tells us that while 64.52 percent of the variation is explained by the primary principal component, the next three account for 21.96, 8.28, and 5.22 percent of the fluctuation, clearly depicting greater deviance between the components than in the other data set. Figure 14b displays the graphs of the four vectors, where the blue graph with the most prominent range corresponds to the principal component, the red to the second most significant vector, the yellow to the third, and the purple to the fourth.

What these graphs display are points of correlation between the different data vectors. It is instantly visible that no fractal pattern can be detected in



(a) Lower technical skill level.

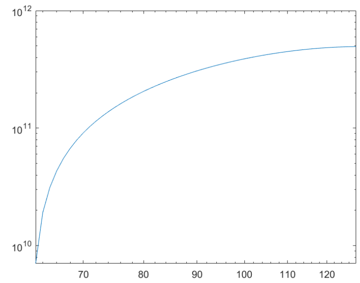
(b) Higher technical skill level.

Figure 14: *The principal component scores of the raw data.*

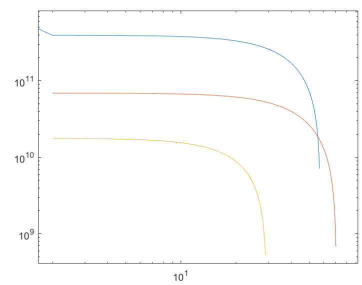
either of the sets with the data as is, but we can gain some knowledge from these results. Figure 14a depicts an obvious linear regression in the similarity of rhythmic fluctuations between the samples. The beginning suggests that the first hits are very much ahead of time, and the consequent ones decay towards the ”perfect” timing. Figure 14b does not indicate a pattern either, but all four vectors presented here depict an anomaly at the same point in time. According to this data, it appears the more talented drummers tend to deliver the 13th hit too late. We will shortly discuss PCA of the power spectrum data, which was performed on various subsets of the two datasets. The same procedure as described above was repeated for each of the corresponding sets of difference vectors to detect seasonality in fluctuations. We will return to discuss them as needed.

Principal component analysis was then run on the matrices comprised of the power spectra of the difference vectors. The results are consistent with the PCA of the difference vectors. The analysis on the set of musicians of the lower skill level depicts that $99.99\ldots\%$ of the variation is explained by the principal component alone regardless of whether 20×1– or 15×1–vectors were used. Accordingly, analyzing the set of the higher–skilled musicians' samples implies more variance between the components; when selecting to use 20×1–vectors the first three components explain 84.8, 15.1 and 0.2 per cent of the variation in that order. For 15×1–vectors the corresponding numbers are 74.8, 25.2 and 0.2 per cent. Figure 15 shows the principal component scores of the 20×1–vectors plotted on a semi–logarithmic scale. In both pictures, the blue graph is the vector that explains the majority of variance.

Before moving on to analysis of the quality of noise, we will perform PCA on some smaller subsets of our data and see if $1/f$ noise characteristics can be detected in the playing of a select piece of a drum kit, or in the playing of a select drummer. The following chapter will entail more detailed discussion on the noise models, but we recall that a power spectrum approximating the function $1/f^{\alpha}$, $0 < \alpha < 2$, is indicative of pink noise. That means we know that if the graph we get as a result presents a descending slope, we can look for



(a) Lower technical skill level.



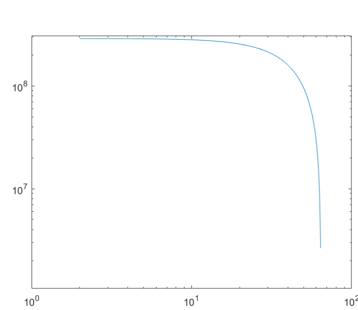
(b) Higher technical skill level.

Figure 15: *The principal component scores of the power spectral data.*

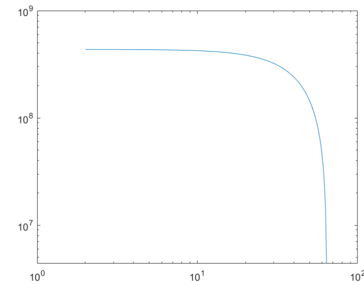
fractal noise, and if the slope is ascending, it definitely does not contain fractal noise. Figure 16 displays some examples of the principal component scores of smaller subsets.

Using the aforementioned quality as a quick visual test to assess the principal components of the power spectrum data leads us to a different direction than the initial analysis of the data groups as whole indicate. The graphs of the subsets of the less–skilled group are very consistently all descending in a seemingly desired way, whereas the opposite is true for our other group. Therefore it seems that fractal noise characteristics are more commonly found in less rhythmically accurate drumming than more, but their discovery and investigation is very dependent on how the data is selected and arranged.

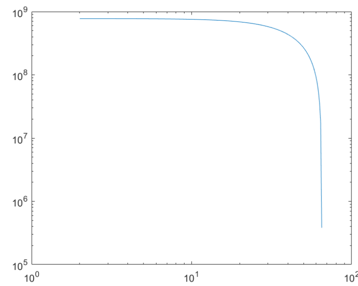
We noticed the same phenomenon in analyzing the principal components of the difference vectors. It appears as well that the algorithm is quite suitable for small data sets, perhaps even better for this study. When analysing the group of less skilled drummers, visible results are only obtained when we are more selective of the samples chosen. In the case of the more skilled drummers, limiting the sample size appears to have a noise–removing effect. For instance the principal component score in Frank Beard's samples (figure 17a) looks very much like a denoised graph of the component vectors of the whole group in figure 14b. We can see that the principal component score of the corresponding power spectra (figure 17b) resembles almost a constant value. The uniformity in the principal components of the power spectra of the smaller subsets of the group representing lower level of skill suggests that we get better results for smaller sets. From this elementary visual analysis we can perhaps conclude that the successful use of PCA in this manner calls for a notable level of similarity in the data.



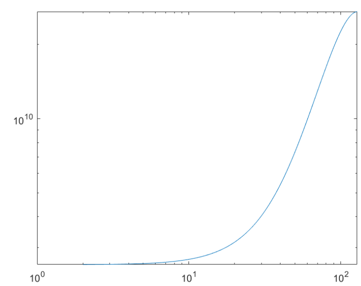
(a) Samples played by Jerry Nolan.



(b) Bass drum samples from the less skilled group.



(c) Samples played by Jeff Porcaro.



(d) Samples played by Neil Peart.

Figure 16: *The principal component scores of some subsets of power spectral densities.*

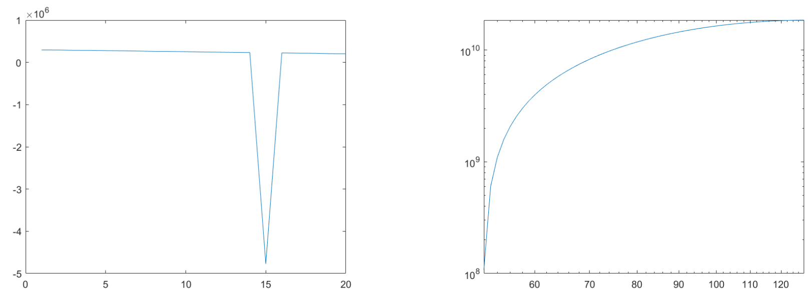
4.6 Fitting a Noise Model

We will now return to the function 1

$$S(f) = \frac{1}{f^{\alpha}}.$$

As we have mentioned multiple times before, a power spectral density determined by such a function indicates fractal noise. In the previous chapter we found the ”collective” power spectra for selected sets of samples that share at least one common characteristics. In this chapter we will examine whether the trends of those vectors follow the function 1 with $0 < \alpha < 2$. The principle for doing this is quite simple; it suffices to find out if the slope of the graph is between -2 and 0.

Matlab's *detrend*–command was used to remove the trend from the function, after which we could compute the slope, or the gradient, of the trend. Following the example set by Räsänen et al. [6] we focused our evaluation on the interval where the L–curve appears on the logarithmic scale. The article makes a good observation on the interval α should be found on as well; since there is a margin



(a) PCA of time difference vectors. (b) PCA of power spectral densities.

Figure 17: *The principal component scores of Frank Beard's samples.*

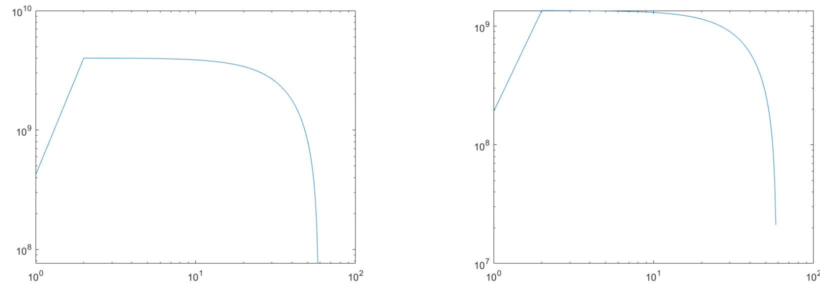
of error in our data collection and previous computations, we shouldn't treat values outside the interval $[0.5, 1.5]$ as definite proof of $1/f$ noise. It is evident from the scale in the graphs of figure 4 in the article by Räsänen et al. that the value found for α is a decimal coefficient, which is how the values in this study were treated as well.

As we already deduced visually in the previous chapter, the principal component corresponding to the power spectra of the lower skill level group does not approximate the function 1 where the exponent would indicate fractal noise. In fact, the value we get for α in this case is approximately -0.995. However, for the higher skill level group we find the value to be $\alpha \approx 0.717$, which is evidence of the fluctuations approximating pink noise (see figure 15.)

We computed values α for smaller subsets as well, like was mentioned once before in chapter 4.5. The results for these small groups were almost contrary to the previous findings. The following subsets were analyzed: only the punk rock recordings (the set of lower skill level with the unmixed live recordings excluded), only the bass drum tracks from the lower skill level group, only the snare drum tracks from the higher skill level group, and a set containing each individual drummer's samples (5 sets of 6 samples from both the lower and higher skill level group).

Limiting the lower skill level set to contain only samples of old punk rock drumming did change the value of α drastically (to approximately 2.2), but it still doesn't indicate pink noise. However, it is quite close to the value associated with Brownian motion, which is $\alpha = 2$. The same type of noise can be detected in Tommy Ramone's playing with $\alpha \approx 2.18$ (figure 18b). The value of α for the live recordings is in the neighborhood of -1, but for Jerry Nolan (figure 16a), Paul Cook and Scott Asheton (figure 18a) the values fall within the range of pink noise (approximately 0.54, 0.46 and 0.65, respectively). The set of just the bass drum tracks yielded the value of $\alpha \approx 0.81$, which is again indicative of pink noise (figure 16b).

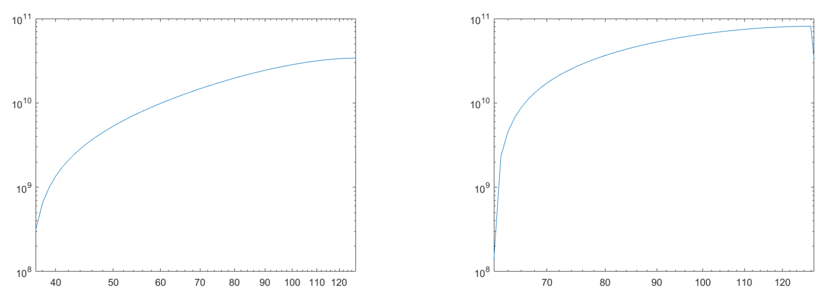
However, among the subsets of the higher skill level only the sets of Jeff Porcaro's samples ($\alpha \approx 1.44$) and Jonathan Moffett's samples ($\alpha \approx 0.43$) were



(a) Samples of Scott Asheton's playing. (b) Samples of Tommy Ramone's playing.

Figure 18: *The principal component scores of two select drummers.*

found to likely contain pink noise. Of these two, only Jeff Porcaro's set falls within the margin of error. All the other subsets yielded negative values for α between approximately -0.2 and -1.8. We can draw a comparison here with the other group where indications of Brownian motion were found; within these subsets there are signs of white noise fluctuations (with the value of α approaching 0). It's also worth noting that the principal component of the power spectra that was investigated from the set of higher skill level samples explained only approximately 85% of the variation, which might suggest that the value we found for α may not be a suitable representation of all the data. Figure 19 contains two examples of cases where $1/f$ noise is not detected.



(a) 15×1–vectors. (b) Samples of Mike Portnoy's playing.

Figure 19: *Principal component scores of some power spectral densities.*

In conclusion, we have found several characteristics in rhythmic fluctuations according to which we could construct a $1/f$ noise model, most notably in the set of technically skilled drummers. The model might be built in such a way that the fluctuations follow along the lines of the principal components we computed for the difference vectors (see figure 14b). So, the fluctuations should perhaps be concentrated in the last quarter of the sample (or a $15 - 20$ beat cycle), and their power spectral density should be approximately

$$S_1(f) = \frac{1}{f^{0.72}}. \tag{4}$$

5 Results

A dataset of isolated drum tracks was split into two groups based on the musicians' technical skill level with the objective to find a $1/f$–noise pattern typical to each, and thus demonstrate that constructing a synthesized drum machine that mimics the human element in music is feasible. Time series difference vectors were created to illustrate the rhythmic fluctuations in each sample from a metronome beat. Principal component analysis was then performed on both these vectors and their power spectra in order to find common characteristics in delay within the two groups. Lastly, the principal component scores were analyzed to determine if a $1/f$–noise model can be fitted.

An important goal of the study was to identify the noise detected in the rhythmic fluctuations as pink or fractal noise. This quality of noise in comparison to for example white noise or Brownian motion was discussed in further detail in chapter 3.3. It is commonly found in many natural phenomena, and its occurrence in human–played music and other human processes, such as speech and reaction time, is well established. Defined patterns or formulas for pink noise have not, however, been defined due to its structure being infinitely complex. Some techniques of identifying $1/f$–noise are introduced in for example by Pilgrim and Taylor in their book "Fractal Analysis" [21] and the articles "$1/f$ noise" by Ward and Greenwood [30] and "Fractal analyses for "short" time series: A re–assessment of classical methods" by Delignieres et al. [28] (see chapters 3.4 and 4.6). We have concluded that there is no standard method of identification but various options for it. The method selected for this thesis was examining whether the power spectral density of a signal is determined by the function $1/f^{\alpha}$, $0 < \alpha < 2$ (equation 1 as introduced by Ward and Greenwood [30]).

We were able to construct some models based on our data to mimic the style of either bass drum playing in punk rock, or the polished progressive rock playing spanning an approximately 20–beat interval. Given a metronome track, this is accomplished by adding noise to the timing so that its power spectral density is approximately determined by the equation

$$S_2(f) = \frac{1}{f^{0.81}}$$

in the former example, and by the equation 4 in the latter. Such results from the power spectral analysis indicate that the noise is indeed pink noise. The fluctuations typical to a lower skill level in playing occur randomly over the time interval with no significant peaks. For the higher skill level, the fluctuations should be concentrated approximately over the last quarter of the interval, with little change from the ideal beat before that. These evaluations are based on principal component analysis of the difference vectors (see figure 14).

Principal component analysis and power spectral analysis were also performed on smaller subsets of the data to examine whether similar noise characteristics can be detected as those mentioned in the previous chapter. We were unable to produce consistent results for subsets of the higher skill level group so that the value of the constant α in the power spectral density model was lower than 2, although we did find clear signs of pink noise fluctuations in Jeff Porcaro's playing, and indications of that in Jonathan Moffett's as well. Their power spectral densities follow approximately the respective functions $1/f^{1.44}$ and $1/f^{0.43}$. As for the drummers depicting a lower level of technical skill, we found implications of pink noise in the playing of Jerry Nolan, Paul Cook and Scott Asheton. Analyzing only the recorded punk rock tracks yielded the power spectral density function to be approximately $1/f^2$, which is indicative of Brownian motion. The same result was found when analyzing the playing of Tommy Ramone. Accordingly, implications of white noise was found in some of the subsets of the higher skilled group; analysis of 15×1– instead of 20×1–vectors, Frank Beard's, and Neil Peart's playing yielded power spectral densities that approach a constant function.

Wide variance in results may be indicative of power spectral analysis not being the ideal tool for examining small sets of data. It is also evident that PCA and therefore all analysis performed on the principal component scores is very sensitive to how the data is arranged and presented; for example the figures in chapter 4.5 depict how the graphs of the components change depending on how the difference vectors of the lower skill group are processed. As for the other dataset, we were able to duplicate the results with smaller subsets, and can thus conclude that we can construct a simple model of noise typical to 1970s punk rock drumming over a short interval. The fluctuations should be randomly placed, and have the power spectral density of approximately

$$S_3(f) = \frac{1}{f^\beta} \tag{5}$$

where the values of β fall on the interval $(0.52, 0.81)$.

6 Conclusion

The research we have performed is rather rudimentary and exploratory, and the results obtained are more directional than anything. The data we have constructed does, however, indicate that a method such as the one at hand generates auspicious results. We were able to repeat the analysis with similar results within the subsets of the lower skill level group, which is of special interest to us. Noise models typical of playing at a certain skill level were also found, which was the goal of this project.

In the introduction of this thesis we mentioned an idea of a possible connection between how flawed the timing of a drummer is and how enjoyable their playing is to listen, and wondered whether that could be part of the reason behind the popularity of such drummers as Paul Cook or Tommy Ramone. The study by Hennig et al. [5] concluded that music containing $1/f$ fluctuations in its elements leads to a more enjoyable listening experience than those containing other noise patterns. The discovery of signs of $1/f$ noise in the playing of successful drummers who are known as "bad" in the sense of technical skill suggests that there might be some truth to the idea; pink noise in rhythmic fluctuation is a crucial factor in what makes music pleasant to listen to. Räsänen et al. [6] discuss the meaning of the term "groove" in their article, and associate groove in playing music with pink noise fluctuation. In less scientific terms $1/f$ noise could be thought of as a measure of grooviness in that sense. It should also be pointed out that the analysis of Jeff Porcaro's playing in this thesis agrees with the findings of Räsänen et al. [6].

A probable reason why $1/f$ noise couldn't be identified in the whole set of the less skilled musicians is that because the drummers make mistakes in their playing in a distinctive way, as a whole the group is too diverse. Our research and results indicate consistently that in order for PCA to work reliably for this type of purpose the data needs to be fairly homogeneous. That is a possibility for why we found a decent model for the whole group of technically skilled musicians, but when we analyzed smaller subsets we failed to repeat the result. The samples of a very talented drummer might contain so little error that the power spectral density function approaches a constant value, as was the case in for example Frank Beard and Neal Peart's playing. However, the most important result of this study is the discovery that it is possible to detect $1/f$–noise patterns within it without a larger amount of data or more complicated tools than PCA.

A relevant observation on the analysis of this study is that its results were strongly subject to fluctuation on the basis of how they were arranged and chosen especially in the set of the lower skill level samples. The initial time–series data was hand–picked using only a graph, which in part makes the results subjective as well. It is highly probable that some of the fluctuation apparent in the difference vectors is caused by errors in determining the time stamps by hand. The margin of error should be similar in every sample, but the decreased level of objectivity must still be recognized. In further studies this problem should be addressed for instance by collecting larger samples or using a filter to construct the time series data. We will discuss in chapter 7 how fractal analysis on human–played music should be improved from here.

The objectivity issue hinders treating the results obtained here as empirically factual without further research. Nonetheless, the aim of this study has been reached, which was to find a way to program synthetic drum tracks whose playing resembles human in terms of rhythm. It could be stated that the subjectivity - the human element - in the models constructed here is an improvement over a mechanically computed model. After all, the drum machine towards whose construction we are working is a tool for creating art. For that reason some minor inaccuracies and unpredictability should be allowed. On the other hand those inaccuracies may and likely will cause the imitation of a style of playing to be imperfect. Although the research clearly implies that drum machines such as described in chapter 2 can be created, these results on their own are not sufficient for building them.

7 Discussion

It is an established fact that pink noise is commonly found in music patterns, nature and human cognition in general, as the various studies we have discussed in this thesis demonstrate. The motive behind this research was to discover ways to potentially develop drum machines that can mimic the playing style characteristic to a specific musician or genre by introducing $1/f$–type fluctuations to the timing of the rhythm. We have showed that it is possible and that we can detect $1/f$ fluctuations even in relatively small datasets, but for creating an accurate noise model more sophisticated methods are required.

Principal component analysis could yield more reliable results when applied to larger and/or more homogeneous datasets. Performing the analysis on for example a hundred samples of the same drummer's playing is likely to result in a very good model. Constructing the real data vectors by hand is slow, so realistically a filtering algorithm that would automate and thus accelerate the process is a requirement. The research of this thesis indicated that using PCA does not limit the choice of the characteristic in playing which one wants to simulate, in the sense that indications of fractal noise were found in the subsets of both our two main groups. It appears, however, that the optimal sample size varies between them.

There is much more to be discovered within this subject in machine learning. Models created with predictive ARIMA, ARMA or GLM algorithms are more detailed and able to find and mimic more intricate patterns in rhythmic fluctuations. The advantage of PCA over the methods mentioned previously is that it requires no additional parameters or data processing, but the increase in input information means the resulting model is more accurately adjusted to the desired purpose. For example the open source platform H2O [67], which was briefly mentioned in chapter 4.5, provides predictive general lineal model algorithms that can be used on fairly small datasets. The models are versatile and customizable and should absolutely be used in further study on fractal analysis on recorded music. All in all, different algorithms should be run on a large variety of data to find general results regarding the field of study, and possibly to attempt to define the nature of $1/f$–noise in humanized musical rhythm.

Models containing $1/f$–noise rhythmic fluctuations may be constructed using the methods introduced in this thesis and mentioned in this chapter according to a multitude of different characteristics. Now that it is established that pink noise in the rhythm can be found and isolated, a fruitful and interesting sequential step would be to construct noise models on a variety of datasets. Returning once again to the original incentive, drum machines in the style of a specific genre or drummer certainly have a demand, and are evidently feasible. There are virtually no limitations to how the data could be collected as long as the samples are sufficiently similar. We could choose for the samples to share one common characteristic, or multiple ones. We can select samples of for example a specific drummer, but we can also limit the samples to a specific year of recording. Upon further and wider research, and complex machine learning algorithms, we can aim to construct drum machines with incredibly detailed parameters. They may mimic genres, eras, musicians, albums, recording studios and personnel, or anything we might think of.

We have treated the presence of $1/f$–noise in musical rhythm as an established fact, and we have mentioned various sources which support the claim. We have not, however, concentrated much on the claim of Ward and Greenwood [30], Räsänen et al. [6] and Clarke and Voss [60] (among others) that such fluctuations are in fact what causes the "human feel" in music. The article by Räsänen et al. [6] suggested that pink noise presents as what we call groove in drumming. It is not necessary to prove the aforementioned results again, but as our last topic of discussion we will select the quantifiability of groove. Already by researching the fairly small amount of data at our disposal, we detected very similar patterns within one of the two groups. After some additional research we might possibly come to the conclusion that the power spectral density is somewhat constant for musicians of a sufficient skill level, thus correlating with how enjoyable their music is to listen to. We noticed that the rhythmic patterns of drummers who are known for their lower level of skill deviate greatly from our other group. As a conclusion of this last subject we might imply that various differently parameterized noise models can be suggestive of a positive listening experience, but we can also find a function for playing music in an enjoyable way (equation 5).

References

[1] Hayes A. *Autoregressive Integrated Moving Average (ARIMA) Prediction Model*. An entry on the subject of ARIMA models on investopedia.com. Reviewed by Stapleton C and fact checked by Velasquez V. Feb. 23, 2024. URL: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp (visited on 03/18/2024).

[2] Tokmakoff A. *11: Brownian Motion*. From Chemistry LibreTexts. URL: https://chem.libretexts.org/Bookshelves/Biological_Chemistry/Concepts_in_Biophysical_Chemistry_(Tokmakoff)/03%3A_Diffusion/11%3A_Brownian_Motion (visited on 03/06/2024).

[3] Music AI. *About*. The info page on Music AI on their official website. URL: https://music.ai/about/?utm_source=google.com&click_section=footer_about (visited on 03/12/2024).

[4] Music AI. *Discover Moises Web App: The Essential Tool for Music Practice*. The introductory page on the Moises App. URL: https://moises.ai/products/moises-web-app/ (visited on 03/12/2024).

[5] Hennig H Fleischmann R Fredebohm A Hagmayer Y Nagler J et al. "The Nature and Perception of Fluctuations in Human Musical Rhythms". In: *PLoS ONE* (2011). DOI: https://doi.org/10.1371/journal.pone.0026457.

[6] Räsänen E Pulkkinen O Virtanen T Zollner M Hennig H et al. "Fluctuations of Hi-Hat Timing and Dynamics in a Virtuoso Drum Track of a Popular Music Recording". In: *PLoS ONE* (2015). DOI: 10.1371/journal.pone.0127902.

[7] Woolf P et al. *2.5: Noise Modeling - White, Pink and Brown Noise, Pops and Crackles*. From Engineering LibreTexts. URL: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/02%3A_Modeling_Basics/2.05%3A_Noise_modeling-_more_detailed_information_on_noise_modeling-_white%2C_pink%2C_and_brown_noise%2C_pops_and_crackles (visited on 03/06/2024).

[8] Audacity. *Audacity: Free audio editor, recorder, music making and more!* The official Audacity home page. 2024. URL: https://www.audacityteam.org/ (visited on 03/11/2024).

[9] Johnson J B. "The Schottky effect in low frequency circuits". In: *Phys. Rev.* 26 (1925), pp. 71–85.

[10] Mandelbrot B. "How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension". In: *Science* 156.3775 (1967), pp. 636–638. DOI: 10.1126/science.156.3775.636. eprint: https://www.science.org/doi/pdf/10.1126/science.156.3775.636. URL: https://www.science.org/doi/abs/10.1126/science.156.3775.636.

[11] Pilgram B and Kaplan D T. "A comparison of estimators for $1/f$ noise". In: *Physica D: Nonlinear Phenomena* 114.1 (1998), pp. 108–122. ISSN: 0167-2789. DOI: https://doi.org/10.1016/S0167-2789(97)00188-7. URL: https://www.sciencedirect.com/science/article/pii/S0167278997001887.

[12] Harvey A F Cerna M. *The Fundamentals of FFT-Based Signal Analysis and Measurement*. Tech. rep. Application note 041. National Instruments, 2000. URL: https://www.sjsu.edu/people/burford.furman/docs/me120/FFT_tutorial_NI.pdf.

[13] Peter Cetera. *You Never Listen to Me*. One More Story [CD]. Written by Cetera P, Leonard P. 1988.

[14] Ding M Chen Y and Kelso J. "Long Memory Processes ($1/f^\alpha$ Type) in Human Coordination". In: *Phys. Rev. Lett.* 79 (22 Dec. 1997), pp. 4501–4504. DOI: 10.1103/PhysRevLett.79.4501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.79.4501.

[15] Waldron A Cherney D Denton T. *14.5: QR Decomposition*. Contributors: David Cherney, Tom Denton, and Andrew Waldron (UC Davis). URL: https://math.libretexts.org/Bookshelves/Linear_Algebra/Map%3A_Linear_Algebra_(Waldron_Cherney_and_Denton)/14%3A_Orthonormal_Bases_and_Complements/14.05%3A_QR_Decomposition (visited on 03/04/2024).

[16] New York Dolls. *Looking for a Kiss*. New York Dolls [CD]. Written by Johansen D. 1973.

[17] Hurst H E. "The Problem of Long–Term Storage in Reservoirs". In: *Hydrological Sciences Journal* 1.3 (1956), pp. 13–27. DOI: 10.1080/02626665609493644. URL: https://doi.org/10.1080/02626665609493644.

[18] Brian Eno. *Fractal Zoom*. Nerve Net [CD]. Written by Eno B. 1992.

[19] ETHW. *John B Johnson - Biography*. From Engineering and Technology History Wiki. 2019. URL: https://ethw.org/John_B._Johnson (visited on 03/07/2024).

[20] Johnny Thunders & The Heartbreakers. *Pirate Love*. L.A.M.F. [CD]. 1977.

[21] Pilgrim I and Taylor R P. "Fractal Analysis of Time-Series Data Sets: Methods and Challenges". In: *Fractal Analysis*. Ed. by Ouadfeul S. Rijeka: IntechOpen, 2018. Chap. 2. DOI: 10.5772/intechopen.81958. URL: https://doi.org/10.5772/intechopen.81958.

[22] Smith L I. *A Tutorial on Principal Component Analysis*. Technical Report OUCS-2002-12. Computer Science Technical Report. Dec. 2002. URL: http://hdl.handle.net/10523/7534.

[23] Bertolotti J. *Colored noise.png*. From WikiMedia Commons. 2019. URL: https://commons.wikimedia.org/wiki/File:Colored_noise.png (visited on 03/08/2024).

[24] Korstanje J. *How to Select a Model for Your Time Series Prediction Task [Guide]*. A blog post on MLOps Blog on the site neptune.ai. July 31, 2023. URL: https://neptune.ai/blog/select-model-for-time-series-prediction-task (visited on 03/17/2024).

[25] Gülen K. *Time Series Forecasting and Machine Learning*. A blog post on the subject of machine learning on dataconomy.com. Nov. 25, 2022. URL: https://dataconomy.com/2022/11/25/time-series-forecasting-machine-learning/ (visited on 03/17/2024).

[26] Korogodski L. *Pink Noise in Neural Nets: A Brief Experiment*. From the technology blog of Chariot Solutions. 2018. URL: https://chariotsolutions.com/blog/post/pink-noise-in-neural-nets-a-brief-experiment/ (visited on 03/07/2024).

[27] Ward L. *Scholarpedia fig2.jpg*. From Scholarpedia Commons. 2007. URL: http://www.scholarpedia.org/article/File:Scholarpedia_fig2.jpg (visited on 03/08/2024).

[28] Delignieres D Ramdani S Lemoine L Torre K Fortes M and Ninot G. “Fractal analyses for ”short” time series: A re–assessment of classical methods”. In: *Journal of Mathematical Psychology* 50.6 (2006), pp. 525–544. ISSN: 0022-2496. DOI: https://doi.org/10.1016/j.jmp.2006.07.004. URL: https://www.sciencedirect.com/science/article/pii/S0022249606000939.

[29] Kobayashi M and Musha T. “1/f fluctuation of heartbeat period”. In: *IEEE Trans Biomed Eng.* 6.29 (1982), pp. 456–7. DOI: 10.1109/TBME.1982.324972.

[30] Ward L. M. and Greenwood P. E. “1/f noise”. In: *Scholarpedia* 2.12 (2007). revision #137265, p. 1537. DOI: 10.4249/scholarpedia.1537.

[31] MathWorks. *Matlab Documentation - MathWorks Nordic.* MathWorks help center for Matlab documentation. URL: https://se.mathworks.com/help/matlab/index.html?s_tid=hc_panel (visited on 03/04/2024).

[32] MathWorks. *PCA - Principal component analysis of raw data.* MathWorks help center for Matlab documentation. URL: https://se.mathworks.com/help/stats/pca.html (visited on 03/20/2024).

[33] MathWorks. *Welch's power spectral density estimate.* MathWorks help center for Matlab documentation. URL: https://se.mathworks.com/help/signal/ref/pwelch.html (visited on 04/21/2024).

[34] Michael McDonald. *I Keep Forgettin'.* If That's What It Takes [CD]. Written by McDonald M, Sanford E, Leiber J, Stoller M. 1982.

[35] *Mike Portnoy.* From the official website of Mike Portnoy. URL: http://www.mikeportnoy.com (visited on 03/11/2024).

[36] Angles Mortis. *Promise.* Unpublished. 2023.

[37] Computer Music. *A short history of the drum machine.* URL: https://www.musicradar.com/news/a-short-history-of-the-drum-machine.

[38] *Neil Peart – THE OFFICIAL NEIL PEART WEBSITE.* From the official website of Neil Peart. URL: https://neilpeart.net/ (visited on 03/11/2024).

[39] Brockwell P and Davis R. *Introduction to Time Series and Forecasting.* 3rd edition. Switzerland: Springer, 1996.

[40] Sutter P. *What is brown noise?* Updated in 2022. 2013. URL: https://www.livescience.com/38547-what-is-brown-noise.html (visited on 03/07/2024).

[41] *Paul Cook.* Bio from the Sex Pistols official website. URL: https://www.sexpistolsofficial.com/bio/paul-cook/ (visited on 03/11/2024).

[42] Sex Pistols. *Holidays in the Sun.* Never Mind the Bollocks, Here's the Sex Pistols [CD]. Written by Cook P, Jones S, Rotten J and Vicious S. 1977.

[43] Sex Pistols. *Submission.* Never Mind the Bollocks, Here's the Sex Pistols [CD]. Written by Cook P, Jones S, Rotten J and Vicious S. 1977.

[44] Smith R. "The Miscellanous Botanical Works of Robert Smith". In: London: R. Hardwicke, 1966. Chap. On the Particles Contained in the Pollen of Plants; and On the General Existence of Active Molecules in Organic and Inorganic Bodies, pp. 463–486.

[45] Ramones. *Havana Affair*. Blitzkrieg Bop/Havana Affair [Single]. Written by Ramone D and Ramone J. 1976.

[46] Ramones. *I Wanna Be Your Boyfriend*. Ramones [CD]. Written by Ramone T. 1976.

[47] Rush. *Subdivisions*. Signals [CD]. Written by Lee G, Lifeson A and Peart N. 1982.

[48] Rush. *Tom Sawyer*. Moving Pictures [CD]. Written by Dubois P, Lee G, Lifeson A and Peart N. 1981.

[49] Iggy & The Stooges. *Dirt*. Fun House [CD]. Written by Alexander D, Asheton R, Asheton S and Pop I. 1970.

[50] The Stooges. *No Fun*. The Stooges [CD]. Written by Alexander D, Asheton R, Asheton S, Pop I. 1969.

[51] Jolliffe I T and Cadima J. "Principal Component Analysis: A Review and Recent Developments". In: *Phil. Trans. R. Soc. A.* 347 (2016). URL: `http://dx.doi.org/10.1098/rsta.2015.0202`.

[52] Dream Theater. *Another Day*. Single. Written by LaBrie J, Moore K, Myung J, Petrucci J, Portnoy M. 1993.

[53] Dream Theater. *Wither*. Black Clouds & Silver Linings [CD]. Written by Myung J, Petrucci J, Portnoy M, Rudess J. 2009.

[54] ZZ Top. *La Grange*. Tres Hombres [CD]. Written by Beard F, Gibbons B and Hill D. 1974.

[55] ZZ Top. *Legs*. Eliminator [CD]. Written by Beard F, Gibbons B and Hill D. 1983.

[56] Toto. *Hold the Line*. Single. Written by Paich D. 1978.

[57] Toto. *Rosanna*. Single. Written by Paich D. 1982.

[58] Rastogi V. *ARMA (Autoregressive Moving Average) model*. A blog post on the subject of ARMA models on medium.com. Aug. 13, 2023. URL: `https://medium.com/@vaibhav1403/the-arma-autoregressive-moving-average-model-is-a-popular-statistical-model-used-in-time-series-24ce43049366` (visited on 03/18/2024).

[59] Yakymiv V. *Machine Learning Time Series Techniques for Accurate Predictions and Forecasting*. A blog post on the subject of machine learning on forbytes.com. July 19, 2023. URL: `https://forbytes.com/blog/machine-learning-time-series-techniques/` (visited on 03/17/2024).

[60] Clarke J Voss R. "1/f noise in music and speech". In: *Nature* 258 (1975), pp. 317–318.

[61] Schottky W. "Small-shot effect and flicker effect". In: *Phys. Rev.* 28 (1926), pp. 74–103.

[62] Schottky W. "Über spontane Stromschwankungen in verschiedenen Elektrizitätsleitern". In: *Ann. der Phys.* 57 (1918), pp. 541–567.

[63] Weisstein E W. *Least Squares Fitting*. From MathWorld–A Wolfram Web Resource. URL: `https://mathworld.wolfram.com/LeastSquaresFitting.html` (visited on 03/04/2024).

[64] Weisstein E W. *Noise*. From MathWorld–A Wolfram Web Resource. URL: `https://mathworld.wolfram.com/Noise.html` (visited on 03/05/2024).

[65] Weisstein E W. *Power spectrum*. From MathWorld–A Wolfram Web Resource. URL: `https://mathworld.wolfram.com/PowerSpectrum.html` (visited on 03/24/2024).

[66] Greenwood P Ward L. *The mathematical genesis of the phenomenon called "1/f noise" (10frg132)*. Tech. rep. Banff International Research Station, 2010-2012. URL: `https://www.birs.ca/workshops/2010/10frg132/report10frg132.pdf`.

[67] H2O Wiki. *Machine Learning Models*. A description in H2O documentation of ML models available on H2O. URL: `https://h2o.ai/wiki/machine-learning-models/` (visited on 03/17/2024).

[68] Wikipedia. *Frank Beard*. URL: `https://en.wikipedia.org/wiki/Frank_Beard_(musician)` (visited on 03/11/2024).

[69] Wikipedia. *Jeff Porcaro*. URL: `https://en.wikipedia.org/wiki/Jeff_Porcaro` (visited on 03/10/2024).

[70] Wikipedia. *Jerry Nolan*. URL: `https://en.wikipedia.org/wiki/Jerry_Nolan` (visited on 03/11/2024).

[71] Wikipedia. *Jonathan Moffett*. URL: `https://en.wikipedia.org/wiki/Jonathan_Moffett` (visited on 03/11/2024).

[72] Wikipedia. *Scott Asheton*. URL: `https://fi.wikipedia.org/wiki/Scott_Asheton` (visited on 03/11/2024).

[73] Wikipedia. *Tommy Ramone*. URL: `https://fi.wikipedia.org/wiki/Tommy_Ramone` (visited on 03/11/2024).

[74] Wikipedia. *Walter H. Schottky*. From Wikipedia. 2023. URL: `https://en.wikipedia.org/wiki/Walter_H._Schottky` (visited on 03/07/2024).