The Development of a Drum Machine Using the Steinberg VST-Specification

Implementation of Extracted Timing Behaviors from Human Drum Performances

KAHL HELLMER
Abstract

Five different drummers were asked to perform two short drumming performances to a reference tempo click delivered through headphones– one in 110bpm and one in 160bpm. These performances were analyzed for obvious and, both between the subjects common and individual, timing patterns and –behaviors. Amongst the most important discoveries unveiled were the phenomena which are referred to as drift, flutter and swing. Drift is the drummer’s own timekeeping variations to the reference click, flutter is every strikes individual deviation from the drift itself and swing is a tendency to displace unemphasised hi-hat strikes earlier in time. These behaviors were isolated from each other using various statistical means and described as performance rules, which in turn were translated into C++ code and implemented into a drum machine using the Steinberg VST specification.

Parallel to the analysis of the drum performances a listening test was conducted where the participants were instructed to rank the drum performances in order of their own preference (tightness, credibility and musical feel). The results of the test showed that drummers with a higher amount of swing were more preferred.

The implementation and design is described in general so that readers with little or no programming experiences can understand the main design and functionality of the drum machine. Further details in forms of code examples with explanations are also included which may require some acquaintance with programming in general.

The drum machine has 10 different parameters which can be adjusted by the user; master gain, kick drum flutter, snare drum and tom’s flutter, kick drum’s offset to the hi-hat, snare and tom’s offset to the hi-hat, hi-hat and cymbals’ swing, maximum drift and drift speed. The sound samples that are used include kick drum, snare drum, two different toms, various hi-hat types, ride cymbal, two crash cymbals and a splash cymbal. The drum machine’s design and the sound samples’ style suggests that plug-in be used in the genres of rock, punk rock and various sub-groupings of metal.

Finally an evaluative listening test was conducted to put the performance of the drum machine into context. The results show that the drum machine is significantly more preferred than the drummers’ original performances in 110bpm at 99% confidence level– and at 95% confidence level at 160bpm.
## Table of Contents

1. **Introduction**  
   1.1 Background  
   1.2 Previous Research  
   1.3 Theories  
   1.4 Aim  
   1.5 Limitations  

2. **Extraction of Timing Behaviors and a Listening Test**  
   2.1 Method  
      2.1.1 Recording  
      2.1.2 Data Analysis  
   2.2 Analysis Results  
   2.3 Listening Test  
   2.4 Listening Test Results  
   2.5 Discussion of Results  
   2.6 Limitations of the Conclusions  

3. **Implementation of Extracted Timing Behaviors**  
   3.1 Introduction  
   3.2 The Steinberg VST-specification and the VST SDK  
   3.3 Sound Samples  
   3.4 The Design of the Plug-in  
      3.4.1 Polyphony  
      3.4.2 Drift  
      3.4.3 Flutter  
      3.4.4 Offsets and Swing  
      3.4.5 Fill-in Haste  
   3.5 GUI  
   3.6 Results  
      3.6.1 Discussion of Limitations and Known Bugs  
      3.6.2 Usage  
   3.7 Drum Machine Specifications  

4. **Listening Test – Evaluating the Drum Machine**  
   4.1 Aim  
   4.2 Listening Test Design
4.2.1 Selection of Stimuli 38
4.2.2 Drum Replacement 38
4.2.3 Drum Machine Stimuli Settings 39
4.3 Listening Test Results 40
4.4 Discussion of Results 42

5 Conclusions 45
5.1 Final Remarks 45
5.2 Further Research and Development of the Drum Machine 45

6 References 47

7 Appendices 48
1 Notation for T1 48
2 Notation for T2 48
3 Listening Test Form for both Listening Tests 49
4 Screenshot of the Drum Machine 50
5 CD-ROM 51
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>IOI</td>
<td>Inter-Onset-Interval</td>
</tr>
<tr>
<td>ISIP</td>
<td>Isochronous Serial Interval Production</td>
</tr>
<tr>
<td>JND</td>
<td>Just Noticeable Difference</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>s</td>
<td>Standard Deviation</td>
</tr>
</tbody>
</table>
Acknowledgements

Sofia Dahl, Håkan Ekman, Stefan Hellmer, Anders Häggbloem, Guy Madison, Tomas Nilsén, Robert Stjärnström, Stefan Sundström, Erica Öberg, the listening test participants and the drummers.
1. Introduction

1.1 Background
A drum machine was developed by the author in the spring of 2004. This drum machine was programmed to only circumvent the dullness of computer performed drum tracks by adding intentional timing errors on every note’s onset. These timing errors, or note onset deviations, were rendered using a normal distribution algorithm at a note-to-note level, which means that the algorithm did not take the onset deviation of a prior note in consideration when rendering the deviation of the current note. The size of the error’s normal distribution was controlled by the user. But as we all know, the human timing function is far more advanced than random timing errors on a note-to-note level, but still the aim of the drum machine was not to imitate human drumming but only to circumvent the dullness of computer performed drum tracks [1].

1.2 Previous Research
The research that has been made in the field of human timing is rather limited in terms of how drummers unintentionally add timing errors when playing more than one part of the instrument simultaneously, i.e. research has mainly been focused on finger tapping and isochronous striking of only a single drum. Nonetheless this research is very valuable as the conclusions of the author’s own research have shown similar patterns and behaviors as that of finger tapping and less complex drumming. A great part of the useful research is in the field of Isochronous Serial Interval Production (ISIP). In this field it is most commonly finger tapping; subjects that tap their fingers on a sensor plate to a given beat. The data is then analyzed for the main purpose of trying to understand the human timing mechanism. Finger tapping under these circumstances is much like playing a drum track to a given beat, in most cases a static click delivered to the drummer through headphones during the recording session. What differs is the complexity; playing an entire set of drums in sometimes very complicated patterns does add more variation to the performers’ timekeeping and more data to be analyzed. The standard deviation of ISIP series is found to be about 3-6% of the Inter-Onset-Interval (IOI), which is the time between two, in this case finger taps. What this means is that when tapping a finger in - to make it easy for us - 1000ms intervals, the standard deviation for each tap is 30-60ms. This is true for within the approximate range of two or three hundred milliseconds to two seconds [2] (intervals of 200ms equal to 300bpm and intervals of 2000ms equal 30bpm).
As tempo increases – generally the standard deviation decreases, up to a critical point of every individual performer where the tempo is too high to be maintained. In a test performed by Guy Madison, PhD. at Uppsala University, three professional drummers were asked to play a series of 128 events with an IOI of 300ms. This produced a standard deviation of 8.3ms which is 2.8% of the IOI [2].

Sofia Dahl, PhD. student at the Royal Institute of Technology in Stockholm, Sweden, has conducted research with the focus at drumming; the movement of the performers arm during different striking velocities and also the timing aspects of “free” playing without a click. Her conclusions are similar to those of finger tapping studies – there are two kinds of variations in the timekeeping of the performer; short-term variations between the IOI’s referred to as flutter and long-term variations, or temporary changes in tempo, referred to as drift [3]. Naturally, drift behaves differently in the absence of the click as a continuous reference to the performer. When there is a reference, the performer adjusts and compensates the timekeeping as soon as it starts to drift and when there is no reference – the drift is “free” and there is no way to control to where it strives at.

According to Dahl, the average flutter ranges between 10-40ms, or 2-8% of the IOI. This flutter was influenced by neither tempo nor dynamic rate. The Just Noticeable Difference (JND) for a displacement of a beat in an otherwise isochronous sequence is around 2-3% of the IOI, which means that for a sequence of beats with a consistent IOI, a single beat can be displaced about 10-16ms when IOI is 517ms or 6-9ms when IOI is 300ms without a noticeable difference. For a beat with an inconsistent IOI (e.g. a human performance) the JND is 2-5% of the IOI obviously depending on how prior beats were manipulated, which is lower than that of the performances in the test (2-8%) [3]. Another interesting thing that Sofia Dahl mentions is that emphasized strikes are prolonged. In addition to raised dynamic, the emphasized strike also is given a longer duration. This means that the following note is, unless if it is also emphasized delayed slightly in time [3].

Another interesting viewpoint of timing analysis was taken by Anders Friberg at the Royal Institute of Technology [5]; he measured the swing ratio of professional jazz drummers. The definitions and results of his research are presented parallel to the results presented in this essay in chapter 2.2.

1.3 Theories
The fundament of which the author’s prior drum machine is based upon is that the human ear can be tricked into believing that a computer performed drum track is performed by a human – more easily than what is expected.
Since every drummer has his/her own very unique way of performing a drum track and since every time a specific drummer plays a specific drum track it will be performed timing-wise dissimilar. If this can be taken as a fact, then in a controlled way adding random timing errors to every note onset can to some extent (satisfying or not) trick listeners into just taking for granted that the drum performance is human when it indeed is not. If this can also - at least for the sake of this argument – be accepted by the reader, then if certain fundamental timing behaviors in a group of drummers can be extracted and defined as performance rules, by implementing these rules to the drum machine a more credible result should be obtained.

1.4 Aim
The intention is to improve the existing drum machine from a “dullness circumventing drum machine” into a “human performance emulating drum machine” with as few limitations as of which in the given time frame and programming skills of the author is possible. The algorithms will be programmed to be as correct to the performance rule as possible and its size or greatness of which it affects the drum track with shall be controlled by the user.

1.5 Limitations
Given that a limited amount of subjects performed two relatively short drumming performances it is likely that certain timing behaviors were overlooked or not even present in the data. There is a practically infinite amount of styles, drumming patterns and different drummers, and what has been measured in this work is in that context very limited.
2. Extraction of Timing Behaviors and a Listening Test

2.1 Method
Two different but similar drum tracks were created in the tracker program Jeskola Buzz. These two will henceforth be referred to as $T_1$ and $T_2$. Both $T_1$ and $T_2$ were 8 bars long and included both two different fill-ins; one located in bar number 4 and one in the last bar; number 8. The tempo of $T_1$ was 110bpm and $T_2$ 160bpm. The test subjects were supposed to perform both $T_1$ and $T_2$ in a studio on a drum kit and the performances were to be analyzed timing-wise parallel to conducting a listening test to see if any of the performances of the test subjects were preferred or disliked in general of the listening test participants.

2.1.2 Recording
A drum kit consisting of a kick drum, snare drum, hi-hat, tom and a crash cymbal were set up in a studio. All parts of the drum kit were recorded with their microphones placed as closely as possible, to in an extent as great as possible avoid the microphone from picking up the other parts of the drum kit. An aesthetic recording was not desired, the only aim was to as effectively as possible isolate the different instruments on their respective channel.

The participants were familiarized with the drum tracks in the control room in order to minimize the amount of time spent in the recording room. As the drummers felt that they could perform the track, with all its details they one by one moved into the recording room and were seated at the drum stool. The click was delivered through headphones during the recording session and was set at a comfortable and clearly audible level. The drummers all requested some “boost” of the kick- and snare drum which was also delivered through the headphones.

The participants were instructed to play the track to the click and that there was no rush; the task can require any amount of takes – the participant himself decides when a take is “a keeper”. Typically that was around 10-15 takes. The reason for this was to obtain a drum track that reflects what the drummer would be satisfied with under “real” circumstances.

The drum tracks were recorded into a standard hard drive recording interface using Steinberg Cubase 2.0. All wave-files were recorded as 44.1 kHz linear PCM Wave-files. Worth to mention is that the click tracks were pre-recorded and were added as tracks in the sequencer thus ensuring that
the clicks were exactly timed with the drums. There were two click-tracks for both tempi, one with quarter notes and one with 32\textsuperscript{nd} notes. The latter one was muted; it was to be used in the following analysis.

2.1.3 Data Analysis
The data was manually analysed using the software sequencer Cool Edit 2.0 by Syntrillium Software. By comparing every drum track individually with the 32\textsuperscript{nd} note click track, the task of checking every onsets deviation from the click with great preciseness was easy but time consuming. Every onsets deviation (measured in milliseconds) was notated using Microsoft Excel.

2.2 Analysis Results
By first plotting a graph showing how the note onsets deviation from the click varies over time the phenomenon of drift is revealed. To analyze the drift properly it must be discriminated from the flutter. This was done by first calculating the average deviation from every reference click, i.e. for every click where a part or two of the drum kit was struck, the deviation was calculated with the formula:

\[
d_i = \frac{h_i + s_i + k_i}{n}
\]  

\(d = \text{average deviation}\)
\(h = \text{hi-hat deviation}\)
\(s = \text{snare drum deviation}\)
\(k = \text{kick drum deviation}\)
\(i = \text{index; indicates the location in the performance measured in 16\textsuperscript{th} notes. (i.e. } i=17 \text{ is the location of the first beat in the second bar.)}\)
\(n = \text{number of parts in the drum kit used for that specific click (index). E.g. if only the hi-hat was struck; } n = 1, \text{ if nothing was struck; } n = 0 \text{ and } d \text{ is ignored.}\)

A 16-point moving average was calculated upon this data in order to clarify the drift and disregard the flutter.

\[
MA_i = \frac{d_i + d_{i+1} + d_{i+2} \ldots d_{i+15}}{16}
\]  

\(MA = \text{moving average}\)
\(d = \text{deviation}\)
\(i = \text{index}\)
The moving average in equation 2 takes the first 16 data points and calculates the average, then it takes one step ahead to the 17th and removes the first etc. In this data every point is a 16th note thus the moving average is in a sense the average of the index data point and the following bar. Since the hi-hat is only struck at 8th note intervals, the only 16th notes are found in the fill-ins.

As can be seen in figure 1 there was a tendency amongst all but one (subject 5) of the participants to in the second half of T1 rush ahead of the click. This was generally a 20 to 50 ms leap following or during the short fill-in and lasted the remaining of the take.

Figure 1: All subjects deviation (ms) from the click T1 shown as 16 period moving averages over 32nd note data. A negative value on the Y-axis indicates that the drift is ahead of the reference click. Subject 5 is marked grey.

This behaviour is also shown in T2 with one difference; the fast recovery time. As figure 2 shows, all subjects tend to rush – however with different amounts – the first fill-in just before halfway through the take.
Minor drifts, i.e. visible over one and sometimes two bars in length, are found in every take.

To analyse the flutter - the drift must be taken as a factor. When a timing error occurs, the size of its deviation is dependent on where the drift is placed. If a drummers drift at a given point in a break is -30ms as an example, if a snare drum is hit with the deviation -35ms, its actual deviation is only -5ms. So to collect the flutter data, every deviation is compared to the moving average of that given point:

\[
dMA_i = d_i - \left( \frac{d_i + d_{i+1} + d_{i+2}...d_{i+15}}{16} \right)
\]

(3)

\(dMA = \) deviation from the moving average  
\(d = \) deviation  
\(i = \) index

The distributions of flutter were nowhere near a tidy normal distribution curve. Instead they were multi-modal distributions and every subject differed from the other.
The distribution curve does of course harbour certain behaviours or patterns when it is bi- or multimodal. In the case of the hi-hat, usually the drummers would place de-emphasised strikes further ahead in time, i.e. strikes that occurs between kick drum or snare drum strikes. These strikes are found as
the mode on the right hand side in these figures. This behaviour is also shown clearly in T2 for subject 5 as can be seen in figure 5:

![Figure 5: Subject 5’s flutter distribution (ms) from the 16 period moving average (T2); black is hi-hat, broken line is snare drum and grey is kick drum.](image)

Another approach of flutter analysis is to look at the internal time deviation which in essence is the mistiming between two different parts of the drum kit which are supposed to be hit simultaneously. In these specific takes the hi-hat is struck simultaneously with the snare – or kick drum a number of times. What was discovered is that the snare drum is more timed with the hi-hat than the kick drum is, i.e. when the snare drum and the hi-hat are struck simultaneously their respective onset times differs less than those of the kick drum and the hi-hat. The internal deviation was calculated as follows:

\[ d_i = sd_i - hd_i \]  

Where:
- \( d \) = internal deviation
- \( sd \) = snare drum deviation
- \( hd \) = hi-hat deviation
- \( i \) = index

To make some sense of this data the mean and standard deviation was calculated for all subjects, referred to as S1, S2, S3, S4 and S5 in table 1-4.
This data alone does not enlighten the complexity of the timing patterns appreciably but some findings can still be mentioned. To reiterate - the $s$ for all subject’s kick drum vs. hi-hat is larger than their snare drum vs. hi-hat. What also was found is that every subject’s snare drum vs. hi-hat $s$ was improved from T1 to T2 except for Subject 4 (though worth to mention is that Subject 4’s mean went from -14,21ms to 1ms). This is not true in the case of the kick drum where Subject 3, 4 and 5 increased their $s$.

To further see how the internal time deviation works, graphs over the distribution were rendered. Since the data is not sufficient in amount (only 12-18 simultaneous hits per take and deviation source) the distribution graphs are not reliable enough to draw any solid conclusions. Still it seems as if the distribution does contain more than one mode, but it requires more data to safely reject the possibility that if more data is added the distribution strives to a single mode.

A tendency of an early displacement of unemphasised hi-hat strikes was also discovered, especially for Subject 5 in T2. What this means is that hi-hat strikes that are placed on beat two, four, six and eight in an 8/8-beat were more often struck earlier in time than those of beat one, three, five and seven.
This tendency is in contradiction to what Sofia Dahl discovered which was mentioned in the last section of chapter 1.2.

The early displacement of the unemphasised hi-hat strikes is a form of “shuffle” most commonly used by jazz drummers to achieve a “swing-feel”. To compare the drummers the approach was to compare their hi-hat onset times with their respective moving average. So explained in words the equation extracts the average difference between the emphasised hi-hat’s deviation from its moving average and the following unemphasised hi-hat’s deviation from its moving average.

\[
S_A = \frac{\sum_{i=0}^{32} (Eh_{4i+1} - dMA_{4i+1}) - (Eh_{4i+3} - dMA_{4i+3})}{n}
\]

(5)

\[S_A = \text{swing average}
\]

\[Eh = \text{Emphasised hi-hat strike’s deviation}
\]

\[dMA = \text{The moving average’s deviation}
\]

\[n = \text{number of emphasised hi-hat strikes}
\]

\[i = \text{index}
\]

Also to confirm that the swing average has any clear significance a standard deviation of the difference between the deviations of emphasised hi-hats and their respective following unemphasised hi-hats was calculated.

\[
s = \sqrt{\frac{n \sum_{i=0}^{32} H^2 - \left( \sum_{i=0}^{32} H \right)^2}{n(n-1)}}
\]

(6)

\[s = \text{Standard deviation}
\]

\[H = (Eh_{4i+1} - dMA_{4i+1}) - (Eh_{4i+3} - dMA_{4i+3})
\]

\[Eh = \text{Emphasised hi-hat strike’s deviation}
\]

\[dMA = \text{The moving average’s deviation}
\]

\[n = \text{number of emphasised hi-hat strikes}
\]

This was calculated for all drummers and shown in table 5 and 6. The takes in which this striking behaviour was most obvious the values have been notated in italic type.
<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_A$ (ms)</td>
<td>1.46</td>
<td>3.05</td>
<td>7.27</td>
<td>-1.55</td>
<td>14.69</td>
</tr>
<tr>
<td>$s$</td>
<td>11.16</td>
<td>9.75</td>
<td>8.35</td>
<td>11.85</td>
<td>6.97</td>
</tr>
</tbody>
</table>

Table 5: The average of how much earlier in timing an unemphasised hi-hat strike is compared to the previous emphasised strike in T1. In the bottom row; the standard deviation of these differences are shown.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_A$ (ms)</td>
<td>1.126</td>
<td>-</td>
<td>-0.96</td>
<td>9.00</td>
<td>18.92</td>
</tr>
<tr>
<td>$s$</td>
<td>9.00</td>
<td>-</td>
<td>4.89</td>
<td>6.27</td>
<td>5.74</td>
</tr>
</tbody>
</table>

Table 6: The average of how much earlier in timing an unemphasised hi-hat strike is compared to the previous emphasised strike in T2. In the bottom row; the standard deviation of these differences are shown.

The reason why Subject 2’s values are missing in table 6 is due to the fact that it proved to be too strenuous to perform at that high tempo thus the task itself added an overall decrease in tempo. To avoid this and yet still being able to obtain relevant timing data from the subject a performance exception was made, enabling the subject to still remain in the group by skipping the unemphasised hi-hat strikes.

The tables show that Subject 5 in both tempi and Subject 4 in T2 have the clearest tendency to place the unemphasised hi-hat earlier. This phenomenon is to the author audible - even if scarcely, when compared to the other subjects and the knowledge of what to listen for is granted.

To put this “swing-thing” into context; similar research has been made by Anders Friberg [4] at the Royal Institute of Technology in Stockholm, Sweden, as mentioned in chapter 1.2. Here what is referred to as swing ratio is measured on jazz drummers. The swing ratio is a measure of how much shorter the shorter note is than the longer note (IOI’s are used instead of deviation from a reference click). The shorter note in this case is a note that is a hi-hat strike which subsequent strike is placed earlier in time. This forces the first hi-hat strike’s duration to be shortened. To illustrate this very simply; the bold numbers (1, 3, 5 and 7) symbolize the emphasised strikes and the regular numbers (2, 4, 6 and 8) symbolize the unemphasised strikes over the time period of one bar.

```
1------2-------3------4------5------6------7------8------
1------2-------3------4------5------6------7------8------
```
In the first line there is no “swing” since all strikes are equal in duration but in the second line the unemphasised strikes are placed earlier in time – forcing a shortening of the emphasised and an elongation of the unemphasised strikes. The ratio between these durations is thus the swing factor.

To compare the swing of Subject 5’s performance of T2 (where the most distinct swing was found), the hi-hat’s deviation from the click was recalculated as IOI’s. The average of the longer hi-hat strikes was calculated to 197,7ms and the shorter strikes to 179,9ms (index 187,5ms). This gave the swing ratio the value: 1,1. To relate this to Friberg’s research the hi-hat IOI (187,5ms) gives a tempo for the hi-hat of 320bpm. In Friberg’s research the swing ratio is approximately 1,1 to 1,3 at 300 – 320bpm for the famous jazz drummers Tony Williams with the Miles Davies Quintet and Jeff Watts with the Wynton Marsalis Quartet. Two things that should be mentioned before any parallels are drawn between subject 5 and the jazz drummers of Friberg’s research which has not been clearly explained is that the jazz drummers play eighth notes and Subject 5 plays quarter notes, and also, the long and short hi-hat strikes are in reversed order here. Friberg refers to swing in a jazz context to be “long-short-long-short” etc. whereas to in the performance of subject 5 it is “short-long-short-long” etc [5]. So in a way the swing found in subject 5’s performance should more properly be referred to as an “inversed swing” with quarter notes instead of eighth notes. Still, the amount of “swing” is interesting; it is just that the two types of swing give different kinds of “feel”.

2.3 Listening Test
Parallel to the analysis of the drum performances the listening test of these was also conducted. The main purpose of this test was to see if there were any common preferences or dislikes of any of the drum performances between the participants. By making ten different packages (.zip-archives) where the stimuli were arranged in different orders a controlled randomization of the stimuli’s order was created. The five stimuli in each package were named 1-, 2-, 3-, 4- and 5.wav and contained the subjects’ drum performances in different orders and could be transcribed to the subjects’ numbers using a package number and a cross reference chart. In a .pdf-file that was enclosed in every package the following instructions were given to the participant:

$\frac{197.7}{179.9} = 1.1$
“Rank the sound files in order of your personal preference, taking only the drummer’s performance as a factor. Do NOT judge by the sound- or the timbre quality of the drum kit, the only relevant factor is the performance (‘musical feel’, ‘tightness’ and ‘reliability’) of the individual drummers. Place the sample you think has the best performance as number one, second best as number two etc.”

Subjects were also instructed to add brief information about themselves in a text-file and return to the author by e-mail together with the package number. The distribution of the packages was handled over FTP and participants were located over e-mail listings and the author’s personal contacts to reach musicians, drummers, song writers and students.

2.4 Listening Test Results
The data from the listening test was analyzed using Statistical Graphics Corp. Statgraphics Plus 4.0⁴. The results were not as reliable as expected due to the fact that the number of participants of 12 was lower than anticipated. Given a higher number of participants perhaps more could be revealed and what is assumed for now could perhaps be assumed with greater certainty.

Nonetheless, what is shown in Table 7, Figure 6 and Figure 7 is a common low ranking of S1’s performance in both tempi, a noticeable higher ranking of S5 in T1 and a higher ranking of S4 together with S5 in T2.

<table>
<thead>
<tr>
<th>Subject</th>
<th>N</th>
<th>Average rank</th>
<th>Subject</th>
<th>N</th>
<th>Average rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>12</td>
<td>48,1667</td>
<td>S1</td>
<td>12</td>
<td>47,5833</td>
</tr>
<tr>
<td>S2</td>
<td>12</td>
<td>30,5</td>
<td>S2</td>
<td>12</td>
<td>28,8333</td>
</tr>
<tr>
<td>S3</td>
<td>12</td>
<td>24,5417</td>
<td>S3</td>
<td>12</td>
<td>30,8333</td>
</tr>
<tr>
<td>S4</td>
<td>12</td>
<td>33,7917</td>
<td>S4</td>
<td>12</td>
<td>22,7083</td>
</tr>
<tr>
<td>S5</td>
<td>12</td>
<td>15,5</td>
<td>S5</td>
<td>12</td>
<td>22,5417</td>
</tr>
</tbody>
</table>

Test statistic = 23,9655  
P-Value = 0,0000811572

Test statistic = 17,2389  
P-Value = 0,00173687

Table 7: Kruskall-Wallis test of the ranking from 1-5 (1 being preferred the most and 5 the least) of the subjects’ performances in the two tempi.

⁴ http://www.statgraphics.com
An explanation of the higher rankings of S4 and S5 in T1 and T2 is not found amongst the analyses of drift behaviors nor flutter distributions – but fits the swing behavior in table 5 and table 6, i.e. the three stimuli that was ranked higher than the other stimuli had a significantly greater amount of inversed swing.

2.5 Discussion of Results
The results from this research can be grouped into four categories:

- Drift behaviour
- Flutter amount
- Internal deviations
- Hi-hat swing

---

3 A Box-and-Whisker chart shows the rankings for each subject’s performance as a graphical summary. The central boxes shows fifty percent of the ranking data – the sides of the box are the upper and lower quartiles and the line between them is the median. The whiskers extend to the lowest and top rankings and the mean is plotted as a plus-sign (+). Single smaller boxes to the sides of the whiskers are outliers in the data.
To discuss these and draw any conclusions which are relevant to the drum machine we begin with the most basic category; the drifting behaviour. What was called minor drifts are for now be presumed to be random, i.e. over the parts where fill-ins or other complex passages were absent there were smaller drifts (+/- 10-15ms over approximately a bar in length) which are presumed to be of a random nature. The part of the drift of which is presumed to not be random are during the fill-ins since every subject played through the fill-in with haste in 160bpm and compensated for that directly after it by de-accelerating, and that in 110bpm two out of five subjects accelerated through the fill-in and stayed ahead of the click, two out of the remaining three accelerated directly following the fill-in and the last one was unaffected by it.

The flutter distribution is considered for now to be a single mode distribution curve for the hi-hat. The fact that the distributions by the subjects were bi- or multimodal was to some, but not quite satisfying extent explained by the swing phenomenon. But still, a ratio of swing should be added to the hi-hat by the user using a fader in the drum machine’s GUI to make sure that a somewhat satisfying result is obtained. The flutter of the snare- and kick drum is for now also presumed to have a somewhat normally distributed error that the user should be able to manipulate in shape and wideness.

The internal deviations are a bit trickier. They are highly dependent on the flutter and the best way for the user to control these is probably a control of the average of how much it deviates from the hi-hat. The distribution curves of these deviations were too insufficient to draw any conclusions and they are on the other hand dependent on the flutter distribution curve.

To ensure that the swing can be used with as few limitations as possible it must be written in such a way that both the common jazz-swing and inverted swing can be emulated. An easy approach to this is letting the parameter that controls the swing being able to have both negative and positive values which states how the unemphasised strikes should be displaced.

A proposed list for the drum machine’s controls:

- **Drift**
  Controlled by the three faders; *Drift Amount, Drift Frequency and Fill-in haste*.

- **Snare drum flutter**
  Controlled by; *Distribution curve’s size (standard deviation) and Average Offset to Hi-hat*.
• Kick drum flutter
Controlled by; Distribution curve’s size (standard deviation) and Average Offset to Hi-hat.

• Hi-hat flutter
Controlled by; Distribution curve’s size (standard deviation) and Swing factor (linear from \(-x\) - \(+x\) in order to achieve both jazz-accentuated swing and its inverse equivalent).

2.6 Limitations of the Conclusions
To reiterate, the purpose of this chapter is to find and isolate - individual or common between the subjects – obvious timing behaviours which are to be used as a foundation in the development of the drum machine. These conclusions are not claimed to in any way reflect the timing behaviour of drum performers in general. It is also most likely that if this test was to be conducted again – even with the same drummers – the results would not be identical to that of which is presented here. Every drummer has his/her very own and unique way of playing the drums due to a spectrum of factors such as motoric skills, music preference, current mood etc. Even when playing a short musical piece twice subsequently, an exact reproduction of the previous performance will never be performed if factors other than timing also are taken into consideration. These factors are timbre\(^4\) and striking velocity. To further entangle this matter – these factors are also connected in various ways which in detail will not be discussed here. So, because of this phenomenon it is not possible to by such a small amount of data as found in this research draw any more precise conclusions of the participants or drummers in general. This leaves the conclusions to be estimates or guidelines.

In the case of the possible connection between inversed swing and a higher ranking in the listening test there is obviously not enough data to state that there is a connection. There are several other factors that could have affected the listening test participants to rank those specific performances higher and those factors may perhaps be unseen or even unthought-of. It may also be the case that the swing ratio is a bi-product of a behaviour that has been over-looked?

---

\(^4\) If the drum stick’s point of impact on the snare drum head is moved a fraction of millimeter, if its striking angle is changed the slightest or if struck with a different velocity – the timbre when struck will be dissimilar.
3. Implementation of Timing Behaviors

3.1 Introduction
To put the extracted timing behaviors into practice they were translated into C++ code and implemented into a drum machine. The reason for programming the entire drum machine “from scratch” is to have complete control of all its variables thus being able to insert the different timing errors at any point in the note triggering chain (from MIDI-signal to sound output).

The drum machine from the author’s prior C-extended essay was used as a starting point [1]. As the task of eliminating the glitch that was present in that version of the drum machine; most of it was rewritten and improved.

3.2 The Steinberg VST Specification and VST SDK
A VST-Plug-In is a software application, it is not a stand-alone program; it is an application that is used “inside” a host application. It can vaguely be described as the digital equivalence of the analogue domain’s hardware signal processing or analyzing devices. VST-Plug-Ins are developed by both companies with commercial interests and by amateur programmers, who mostly do it for the fun of sharing their produce with fellow audio engineers. On the Internet there are great amounts of various plug-ins - free for download by anyone. A Plug-In can be made to accomplish almost any task at all; except for making coffee - the sky is the limit.

The VST specification has been developed by Steinberg Media Technologies GmbH, and following quotes are taken from the VST 2.0 specification:

“In the widest possible sense a VST Plug-In is an audio process. A VST Plug-In is not an application. It needs a host application that handles the audio streams and makes use of the process the VST Plug-In supplies […]

Generally speaking, it can take a stream of audio data, apply a process to the audio and send the result back the host application […]

From the host application’s point of view, a VST Plug-In is a black box with an arbitrary number of inputs, outputs (Midi or Audio), and associated parameters. The Host needs no knowledge of the Plug-In process to be able to use it […]” [6].

This specification is supported by the vast majority of audio sequencers and editors which are focused on the aesthetic approach to audio mixing and editing. This makes VST plug-ins very versatile since if an audio effect plug-
in or a synthesizer plug-in is developed, they are not limited to just one host application, or for that fact one single Operating System. This means that if someone develops a reverberation plug-in, as an example, it is compatible with all host applications that support the VST-specification, no matter what platform it uses.

The Virtual Studio Technology Plug-in Specification 2.0 Software Development Kit [6] (VSTSDK) can be downloaded free of charge from the Steinberg website. It consists of sample C++ code, a few simple example projects that can be compiled into ready to use VST plug-ins and an empty starter project that compiles into a “do-nothing plug-in”. With the help of the SDK it is much easier to get started with the programming since you can peek at the example code and see how different basic tasks (e.g. how a variable can be controlled by a fader in the GUI) are handled efficiently.

3.3 Sound Samples
To record the different drum samples a drum set was set up in a recording studio using conventional means and techniques. The drummer was asked to strike the different parts of the drum set with a few second intervals – at first with great force and then descending to using lesser force. This way a decent amount of samples from each part of the drum kit with various strike forces could be obtained. The drummer was also asked to strike the hi-hat twice, letting the second strike ring out, this way the first strike of the two could be deleted in the wave editor afterwards giving the remaining second strike the special timbre that is given when striking an already ringing hi-hat.

3.4 The Design of the Plug-in
First of all it needs to be explained that - and why – all sound samples when loaded from the HDD into RAM when the drum machine is initiated are added 8820 sample frames (200ms) of introductory silence: Since the onsets of the strikes must be able to be triggered with a certain delay, and also in advance – some measure must be taken since triggering sounds before they have occurred belongs in another field of research. So, by adding 8820 sample frames of silence to every sound buffer’s beginning, and from there telling the drum machine’s pointers that the default offset in this silence is at sample frame 4410, we have delayed every sound sample with 4410 sample frames, or 100ms.
Figure 8: Illustrates graphically how sound samples are stored in RAM with a 200ms introductory silence. The red cursor line shows where the default offset is located. By moving the cursor to the right the time until the first actual sample frame of the sound sample is reached is shortened, and vice versa.

So, by setting the default offset to 4410 sample frames into the sound buffer it is very easy to manipulate when the actual sound sample is reached. By decreasing the default offset by 441 to 3969 sample frames the sound is delayed with 10ms, as adding 882 sample frames forcing the offset to 5292 places the sound 20ms ahead. This is the basics of how the timing deviations are achieved in the drum machine.

To explain in overall how the drum machine works an explanation of how it handles incoming note events will be described. When a MIDI-signal is sent from the host program to the drum machine the object `processEvents()` is called. This object sorts out the specifications of the MIDI-event, i.e. it checks what note was sent, its velocity and what number of sample frames has passed. The note is value which tells which key that was struck, where the value 0 is the note C0 and 78 is F#4. The velocity is a value between 0-127 which tells us how hard the note was struck where a low value gives a low velocity or strike force and vice versa. The number of sample frames is explained in chapter 3.4.1.

For every individual MIDI-signal that is received, the object `processEvents()` calls the object `noteOn()`. This object assigns the values of the MIDI-signal to the variables `currentNote`, `currentVelocity` and `currentDelta` and calls the next object which is named `sampleRand()`.
Figure 9: An overview as to how the major objects in the drum machine are connected to each other and to the host program. The bold arrows symbolize the output sample flow (output sound).

`sampleRand()` first checks what value the `currentNote` variable has to see which part of the drum set that is to be triggered. Then the timing error (flutter) is generated by `timeDev()` and added to the offset position along with the current drift disposition which is continuously changing and generated independently from the other timing errors. How the flutter and drift are generated will be explained further in the following sections.

```
1 kpk1 = &k[4410 - delta1 + currentDrift + DevKick + offsetKick];
```

Code example 1 – shows which variables affects the offset in the introductory silence when the sound `k1` (Kick01.wav) is triggered using the pointer `kpk1` (first out of 4 pointers). Note that all variable values are specified as number of ± sample frames.

The first value in Code example 1 within the brackets is 4410 which is the default offset as mentioned earlier. The variable `delta1` is the number of delta frames which will be discussed in the next chapter – polyphony. `currentDrift`
is a rather self-explanatory variable with a value between -4410 and 4410. 

DevKick is a variable which maximum deviance from 0 is specified by the user through the GUI and offsetKick is a constant value – also controlled through the GUI. These variables, and how they are assigned their values, will be explained in this chapter.

3.4.1 Polyphony

Delta samples, or deltaFrames as the variable is called is related to time – the number of sample frames that has passed in the block of sample frames of the input/output buffers, i.e. if a note is received at 512 sample frames into the input buffer it must not be triggered to play at the offset 0 sample frames in the output buffer. This also enables the polyphonical feature since if two or more MIDI-signals are received simultaneously by the drum machine, it can process them one-by-one and still trigger them with the same sample frame value in the output buffer thus forcing them to be played as simultaneously as they were received.

```cpp
long kDrummer::processEvents (VstEvents* ev)
{
    for (long i = 0; i < ev->numEvents; i++)
    {
        if ((ev->events[i])->type != kVstMidiType)
            continue;
        VstMidiEvent* event = (VstMidiEvent*)ev->events[i];
        char* midiData = event->midiData;
        long status = midiData[0] & 0xf0;
        if (status == 0x90 || status == 0x80)
        {
            long note = midiData[1] & 0x7f;
            long velocity = midiData[2] & 0x7f;
            if (status == 0x80)
                velocity = 0;
            if (!velocity&&(note == currentNote)||velocity == 0)
                noteOff ();
            else
                noteOn (note, velocity, event->deltaFrames);
        }
        else if (status == 0xb0)
        {
            if (midiData[1] == 0x7e || midiData[1] == 0x7b)
                noteOff ();
            else if (status == 0xb0)
            {
                if (midiData[1] == 0x7e || midiData[1] == 0x7b)
                    noteOff ();
            }
            event++;
        }
    return 1;
}
```

Code example 2 - the object processEvents().

The object `processEvents()` is called when one or more MIDI-signals (events) have been received. The objects’ statements are all inside of a `for-loop` which is
declared on line 3 which is based upon how many events \((\text{numEvents})\) that has been received (if two events are received – it will loop twice). To facilitate everything in the object; if the if-statements at lines 14 and 16 are not true, that is, if the event received is not the termination of a prior note (velocity 0) – \text{noteOn()} \text{will be called from line number 19 with the arguments note number, its velocity and delta sample frames. After that, the following event \((\text{event++})\) will be selected (given that there is one) and the for-loop starts over.

3.4.2 Drift

The drift is generated continuously in the “background” at all times – even when there are no incoming MIDI-signals – the drift is being generated. The drift is as previously mentioned, variations in the tempo that were presumed to be random.

In the object \text{processReplacing()}, which is called when the input/output buffers are empty, the object \text{driftGen()} is called accordingly with the variable \text{drift frequency from the GUI}.

\text{driftGen()} basically generates values that forces the default offset (4410 sample frames) to vary over time (see Figure 6). The drift affects all parts of the drum set causing all sounds that are triggered at a specific point to have the same drift. The drift generated in the drum machine continuously adds generated drift values to the previous value. This is to ensure that great differences between preceding values are avoided, i.e. instead of generating drift values between maximum and minimum tolerable deviances (-4410 – 4410 sample frames) with the mean 0 – a new drift value is generated with a smaller deviance (-2205 - 2205) and added to the previous value. Also, to ensure that the drift does not exceed the -4410 – 4410 barrier which would occur if a series of positive or negative drift values are generated subsequently it must be taken in consideration where the current drift is positioned.
1 void kDrummer::driftGen ()
2 {
3   currentDrift = futureDrift;
4   kDriftSC = 0;
5   timeDev (0, .3);
6   kDriftNum = kTimeDevNum;
7   if (kDriftNum > 1) kDriftNum = 1;
8   if (kDriftNum < -1) kDriftNum = -1;
9   if (kDriftNum > 0 && currentDrift > 0) {
10      kDriftNum = kDriftNum * fDriftAm * ((-1 * (currentDrift - 2205)) / 2205);
11   }
12   if (kDriftNum > 0 && currentDrift < 0) {
13      kDriftNum = kDriftNum * fDriftAm;
14   }
15   if (kDriftNum < 0 && currentDrift > 0) {
16      kDriftNum = kDriftNum * fDriftAm;
17   }
18   if (kDriftNum < 0 && currentDrift < 0) {
19      kDriftNum = kDriftNum * fDriftAm * ((-1 * (currentDrift + 2205)) / 2205);
20   }
21   futureDrift = currentDrift + (2205 * kDriftNum);
22   if (futureDrift < -2205) futureDrift = -2205;
23   if (futureDrift > 2205) futureDrift = 2205;
24   driftDiff = currentDrift - futureDrift;
25 }
Code example 3 - the object driftGen().

At line 7 in Code example 3 the object timeDev(), which generates a random normally distributed value with the mean 0 and the standard deviation 0.3, is called. Translated into an equation from C++ code it looks like this:

\[ k\text{TimeDevNum} = \text{std} \times \sqrt{-2 \log(rand() + 1) \times \sin \frac{2\pi \times \text{rand}()}{r_{\text{max}}} + \text{mean}} \]

(7)

\[ k\text{TimeDevNum} = \text{The global variable in which the generated number is stored.} \]
\[ \text{std} = \text{The standard deviation which is retrieved as an argument when the object is called (0,3).} \]
\[ \text{rand()} = \text{a random number between 0 and r_max.} \]
\[ r_{\text{max}} = \text{a variable associated with the functions srand()} \text{and rand().} \]
\[ \text{mean} = \text{the mean of the normal distribution, also retrieved in the argument (0).} \]

The value is stored in the variable kDriftNum at line 8 of Code example 3. At lines 10 and 11 the value is limited to -1 – 1. Now to the interesting part, as
can be seen at lines 13 and 22: If \( k\text{DriftNum} \) is positive when \( \text{currentDrift} \) also is positive – or vice versa: the new drift value in \( k\text{DriftNum} \) is scaled down in proportion to how far from 0 \( \text{currentDrift} \) is:

\[
k\text{DriftNum} = k\text{DriftNum} \times f\text{DriftAm} \times \frac{-1 \times (\text{currentDrift} - 2205)}{2205}
\]

(8)

\( k\text{DriftNum} \) = New generated drift value (ranges from 0 – 1 in this specific scenario).

\( f\text{DriftAm} \) = The variable drift amount – controlled by the user through the GUI.

\( \text{currentDrift} \) = the position in sample frames from 0 (4410 sample frames) that the current drift is positioned.

By subtracting 2205 from \( \text{currentDrift} \) and multiplying by -1, \( \text{currentDrift} \) which ranges from 0-2205 is inversed (the greater \( \text{currentDrift} \) is – the smaller it will be). By then dividing it by 2205 a value between 0-1 is obtained – inversely proportioned to the size of \( \text{currentDrift} \). What this returns in the equation is that the greater the drift – the greater the down-scaling of \( k\text{DriftNum} \).

Figure 10: Illustrates how new positive drift values are scaled down in proportion to the current positive drift.

Another precaution that is made to see that there will not be any sudden and great drift differences is slicing every step to the new drift position into 4. As
the future drift is already known when a current drift position takes place it is possible to smoothen out the drift as seen in figure 8:

![Figure 11: The grey line illustrates how the drift differences between current and future drift is split up into 4 steps instead of 1. The broken grey line shows how the drift would be without this precaution.](image)

The function in the drum machine that does this adds a quarter of the difference between currentDrift and futureDrift to currentDrift every quarter of fDriftFreq, which is the variable that the user controls through the GUI.

Metaphorically; if the randomness of the generated drift values over time can be compared with the randomness of white noise – then by eliminating great differences between subsequent drift values can be compared with a low-pass filter, thus the rendering of drift in this object is much like generating a sort of LP-filtered noise.

### 3.4.3 Flutter

Flutter is every onsets individual deviation and is generated “on top” of the drift. The flutter is just like the drift also generated through the object timeDev(). The returned value is stored in a variable which is multiplied with the standard deviation of the specific part of the drum kit that is being used in accordance to the user’s setting in the GUI and then translated to sample frames:

```
DevKick = (2205 * DevNumKick * fStandDevK);
```

Code example 4 – how the returned flutter value to a kick sound is treated.

The DevKick-variable stores what in this case is the flutter value of a kick. DevNumKick is the value returned from timeDev() and fStandDevK (ranges from 0-
1) is controlled by the user through the GUI. By multiplying \texttt{DevNumKick} with \texttt{fStandDevK}, \texttt{DevNumKick} is scaled down, then by multiplying by 2205 it is translated into sample frames and is then a value which ranges from -2205 to 2205.

### 3.4.4 Offset and Swing

With the offset, the effect of an overall slightly delayed or early onset time of the part of the drum kit in question. By adding a negative or positive number to the offset position in the sound sample’s introductory silence this is rather effortlessly achieved (see Code example 1). By moving the fader in the GUI the variable in question (in the case Code Example 1 it is \texttt{offsetKick}) is given a value translated into sample frames.

The swing is a bit trickier than the offset. Since the aim of the drum machine is to make it with as few limitations as possible and the author’s GUI-programming skills are nonexistent – the swing was given an extra MIDI-note at C3 to mark that any notes struck simultaneously are unemphasised. This way the user can decide if the swing is to occur on half notes, quarter notes, eighth notes or sixteenth notes and also specify specific parts of a track where there should not be “swing”.

![Figure 12: Part of a screenshot from the built-in MIDI-editor in Steinberg Cubase SX which shows how the note C3 is used to mark specific hi-hat strikes at the note E4 as unemphasised which enables the drum machine to render swing.](image)

### 3.4.5 Fill-in Haste

The Fill-in Haste function statically increases or decreases (according to the parameter in the GUI) the tempo. The alteration is done parallel to the drift generation and the two objects do not communicate with each other. Just like the swing function – the fill-in haste function has a MIDI-note for controllability. The note number in use is 62, or D3, and is used similarly as C3 for swing: add the note at the desired start position of the tempo alteration.
and the length of the note controls its duration, i.e. throughout the duration of a note at D3 the drum machine will with a continuous linearity increase or decrease the drift by adding a value to the variable currentDrift ten times every second. The drift generation is functioning as usual while this occurs and strives to decrease the drift both during and afterwards as if it was produced by itself.

3.5 GUI
Through the GUI the user can control the drum machine easily. Table 8 shows, in the same order as shown in the drum machine itself, the parameters, the ranges in within they can be set and a brief explanation of each parameter.

<table>
<thead>
<tr>
<th>Parameter’s name</th>
<th>Range</th>
<th>Variable type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>gain</td>
<td>-∞ - 0dB</td>
<td>static</td>
<td>Master gain</td>
</tr>
<tr>
<td>kick dev</td>
<td>-50ms – (+)50ms</td>
<td>random</td>
<td>Kick drum s</td>
</tr>
<tr>
<td>snare/tom dev</td>
<td>-50ms – (+)50ms</td>
<td>random</td>
<td>Snare drum and tom’s s</td>
</tr>
<tr>
<td>oh dev</td>
<td>-50ms – (+)50ms</td>
<td>random</td>
<td>Hi-hat and cymbals’ s</td>
</tr>
<tr>
<td>kick offset</td>
<td>-50ms – (+)50ms</td>
<td>static</td>
<td>Kick drum’s offset to hi-hat</td>
</tr>
<tr>
<td>snare offset</td>
<td>-50ms – (+)50ms</td>
<td>static</td>
<td>Snare drum’s offset to Hi-hat</td>
</tr>
<tr>
<td>oh swing</td>
<td>-50ms – (+)50ms</td>
<td>static</td>
<td>Hi-hat and cymbals’ swing</td>
</tr>
<tr>
<td>drift amount</td>
<td>-50ms – (+)50ms</td>
<td>random</td>
<td>Maximum drift5</td>
</tr>
<tr>
<td>drift freq</td>
<td>0-1Hz</td>
<td>static</td>
<td>Drift speed6</td>
</tr>
<tr>
<td>fill-in haste</td>
<td>-100ms – (+)100ms</td>
<td>static</td>
<td>temporary drift control</td>
</tr>
</tbody>
</table>

Table 8: Listing of the drum machine’s parameters.

3.6 Results
The drum machine has been tested and is working properly on an AMD 2800+ fitted computer with 512MB dual DDR Ram on Microsoft Windows 2000 Professional Service pack 4. The host for the plug-in has been Audio Mulch 0.9b18 for debugging and Steinberg Cubase SX 1.0.6 for testing.

3.6.1 Discussion of Limitations and Known Bugs
The most obvious and least annoying and harmless bug is located in the GUI; a number of unnamed faders appear along with the ordinary faders. The

5 Drift amount is the maximum tolerable difference between two subsequent drift values (i.e. current and future drift).

6 Drift frequency is the time between two subsequent drift values (i.e. how long until the future drift becomes the current drift).
origins of these unnamed faders are to the author unknown as this is written and due to their harmless nature the effort that has been made to remove them has been small – instead the work done at the eleventh hour has been focused on removing audible bugs that has a higher degree of interference to the actual output and end-result.

Due to using only four pointers for every “sound bank” (i.e. kick samples or snare drum samples) and the fact that there is an introductory silence in each sound sample another issue evolves: If 5 or more sound samples are triggered for playback and the introductory silence, depending on the generated deviation and the current drift, it can take up to 200ms (8820 sample frames) before the sounds are triggered. So, in the cycle of the four pointers, if the time between one specific pointer is used and reused does not exceed 200ms the first sound may not have even started playing before the pointer is reassigned to another sample buffer. This must be known by the user when making excessive snare drum rolls. The limit in theory in a worst case scenario (very unlikely) is when a specific sound sample is delayed maximally to 200ms: then the highest tempo allowed is an IOI of 
\[
\frac{200\text{ms}}{4} = 50\text{ms}
\]
. This is an equivalent of eighth notes at 150bpm. Though what is worth to be mentioned as a corollary is that if such great deviations were allowed at that tempo the individual snare sound samples in the snare roll would be too mistimed to be musically appreciated.

When in playback – recording or not – the first 4 to 8 bars can sometimes contain one or more sudden leaps in the drift. This can be sidestepped by playing the entire performance in loop-mode and then rejecting the first performance.

3.6.2 Usage
How get started and loading the plug-in into a host depends on which host is used and therefore no information other than what concerns the drum machine in itself is discussed.

First of all – the minimum and maximum levels of all parameters in the GUI are extremes, except for the maximum level of drift frequency (1Hz) which is a useable value. Secondly, the fill-in haste must be greater than the drift amount or else \texttt{driftGen()} will immediately neutralize the fill-in haste. It can roughly be described as the actual fill-in haste being equal to the drift amount subtracted from the fill-in haste (given that fill-in haste is positive). Excessive use of fill-in haste will cause improper play-back of the sound samples (too “fast” drift) or program error (too “slow” drift).
### 3.7 Drum Machine Specifications

<table>
<thead>
<tr>
<th>Name</th>
<th>MIDI note</th>
<th>Vel. Groups</th>
<th>Samples</th>
<th>Memory Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swing</td>
<td>C3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Haste</td>
<td>D3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kick</td>
<td>C4</td>
<td>2</td>
<td>4</td>
<td>983kB</td>
</tr>
<tr>
<td>Snare</td>
<td>D4</td>
<td>2</td>
<td>12</td>
<td>2,88MB</td>
</tr>
<tr>
<td>Hi-hat</td>
<td>E4</td>
<td>4</td>
<td>28</td>
<td>7,61MB</td>
</tr>
<tr>
<td>Tom1</td>
<td>F4</td>
<td>2</td>
<td>8</td>
<td>2,59MB</td>
</tr>
<tr>
<td>Tom2</td>
<td>G4</td>
<td>2</td>
<td>8</td>
<td>2,59MB</td>
</tr>
<tr>
<td>Splash</td>
<td>A4</td>
<td>2</td>
<td>8</td>
<td>3,27MB</td>
</tr>
<tr>
<td>Crash1</td>
<td>B5</td>
<td>1</td>
<td>4</td>
<td>2,98MB</td>
</tr>
<tr>
<td>Crash2</td>
<td>C5</td>
<td>2</td>
<td>8</td>
<td>5,96MB</td>
</tr>
<tr>
<td>Ride</td>
<td>D5</td>
<td>4</td>
<td>18</td>
<td>8,36MB</td>
</tr>
</tbody>
</table>

Total: 37,22MB

Table 9: Chart over how sound samples, velocity groups and allocated memory is distributed in the plug-in.

<table>
<thead>
<tr>
<th>Name</th>
<th>Vel. Gr. 1</th>
<th>Vel. Gr. 2</th>
<th>Vel. Gr. 3</th>
<th>Vel. Gr. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kick</td>
<td>Hard kick</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Snare</td>
<td>Hard snare</td>
<td>Soft snare</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hi-hat</td>
<td>Hard closed</td>
<td>Soft closed</td>
<td>Hard open</td>
<td>Hard closed</td>
</tr>
<tr>
<td>Tom1</td>
<td>Hard tom</td>
<td>Soft tom</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tom2</td>
<td>Hard tom</td>
<td>Soft tom</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Splash</td>
<td>Hard Splash</td>
<td>Soft Splash</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Crash1</td>
<td>Hard Crash</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Crash2</td>
<td>Hard Crash</td>
<td>Soft Crash</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ride</td>
<td>Hard Ride</td>
<td>Soft Ride</td>
<td>Hard bell</td>
<td>Soft bell</td>
</tr>
</tbody>
</table>

Table 10: Chart over how sound samples are distributed inside the velocity groups.

---

Memory Allocation refers to the actual amount of memory being allocated. The facts that every sound samples frame is stored in RAM as 32-bit floating point variables and the introductory silence has been considered.
<table>
<thead>
<tr>
<th>Name</th>
<th>Vel. Gr. 1</th>
<th>Vel. Gr. 2</th>
<th>Vel. Gr. 3</th>
<th>Vel. Gr. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kick</td>
<td>101-127</td>
<td>0-100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Snare</td>
<td>101-127</td>
<td>0-100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hi-hat</td>
<td>101-127</td>
<td>91-100</td>
<td>81-90</td>
<td>71-80</td>
</tr>
<tr>
<td>Tom1</td>
<td>101-127</td>
<td>0-100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tom2</td>
<td>101-127</td>
<td>0-100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Splash</td>
<td>101-127</td>
<td>0-100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Crash1</td>
<td>0-127</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Crash2</td>
<td>101-127</td>
<td>0-100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ride</td>
<td>101-127</td>
<td>91-100</td>
<td>81-90</td>
<td>71-80</td>
</tr>
</tbody>
</table>

Table 11: Chart over which velocity to use to reach a specific velocity group.
4. Listening Test – Evaluating the Drum Machine

4.1 Method and Aim
The purpose of this listening test is to evaluate the performance of the drum machine and put the performance into some kind of context. This is done by executing a second listening test that was, apart from the stimuli, identical with the prior listening test; letting the participants rank the stimuli in order of musical preference in the very same way as the prior test - the difference being two performances of the drum machine hidden with the real drummer's performances. This way an evaluation of the drum machine can be made.

4.2 Listening Test Design
This listening test is designed to be comparable with the previous listening test; to enable comparability and thus verify that the rankings of the drum performances used in the previous test are ranked somewhat similar to each other in this test as well. What this means is that the possibility of looking at a specific stimulus used in the prior test and verifying that it is ranked in the same proportion to another stimulus from the prior listening test is granted. If, as an example; one stimulus is ranked much higher than another – those two should have a proportional distance in ranking in this test as well.

4.2.1 Selection of Stimuli
To see how the drum machine’s performance was ranked amongst the drummers’ performances two of the stimuli from each category from the previous listening test were replaced to avoid an excessive amount of stimuli. The two stimuli that were extracted were the two in each test who ranking-wise did not distinguish themselves from the group or from another stimulus. Considering the average rank and the spread of the individual rankings for each subject; S2 and S3 were not used in T1, S2 and S4 were not used in T2. This left S1, S4 and S5 in T1 along with S1, S3 and S5 in T2 which also were the highest, lowest and centermost ranked in respective tempi.

4.2.2 Drum Replacement
Since the performances of the drummers were played on another drum kit using another microphone technique than the recorded sound samples that are used in the drum machine – the audio output of the drum machine is vastly different from the sound of the drummer’s performances. To avoid
disclosing that two out of the five performances in each category are different from the other three some measure had to be taken.

Using the VSTi plug-in Drumagog 3.0 by Wavemachine Labs⁸, every strike in the performances of the drummers were replaced with the sound samples used by the drum machine. What Drumagog basically does is analyze a given track in real-time and when the amplitude of the signal on that track is greater than the user-specified threshold it triggers one of the replacing sounds to be played. The sound of the track itself is muted resulting in only the sound of Drumagog’s replacement sounds are heard.

The kick drum, snare drum and the toms are quite commonly replaced in contemporary music productions; mainly to attain “clean” tracks (i.e. tracks which are free of sound leakage from the rest of the drum kit) and thus leaving the mixing engineer with greater possibilities of altering the tracks by means of equalization, dynamic reduction and such. The hi-hat and the other cymbals are to the author’s knowledge rarely replaced in the post-production. This is probably because these instruments are too complex due to the large varieties of sound each can generate depending on how - and where – they are struck, but mainly because there is not the same need to have these tracks “cleaned” from the other parts of the drum kit.

The hi-hat track had received a substantial amount of both snare- and kick drum which made the work of the plug-in Drumagog impossible since it can not tell the hi-hat strikes from the other strikes. The solution to this problem was easily sidestepped by filtering the hi-hat track with a steep high-pass-filter at approximately 5 kHz. In this part of the spectrum the transient onsets of the hi-hat have enough amplitude compared to the snare- and kick drum that Drumagog can be set to disregard them.

Also worth mentioning is that Drumagog was set to disregard the dynamics of individual strikes on all tracks and to randomly pick sound samples from a sound bank. This means that it did not reduce or increase the gain of a replacement sound sample if it was triggered by a weaker or stronger strike and every sound sample was randomly picked for each individual sound replacement. This was to ensure that the output of Drumagog had the same characteristics as the drum machine.

4.2.3 Drum Machine Stimuli Settings
In order to create a realistic and musically preferable stimulus with the drum machine a variety of different settings along with numerous trials with each

---

⁸ http://www.drumagog.com/default.asp
setting were tried. Finally, the drum machine’s settings were set to reflect the
subjects’ own performances were set and the four stimuli were rendered.

<table>
<thead>
<tr>
<th></th>
<th>Dm01 T1</th>
<th>Dm02 T1</th>
<th>Dm01 T2</th>
<th>Dm02 T2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kick Dev.</td>
<td>10ms</td>
<td>12,5ms</td>
<td>5ms</td>
<td>12,5ms</td>
</tr>
<tr>
<td>Snare Dev.</td>
<td>5ms</td>
<td>7,5ms</td>
<td>5ms</td>
<td>7,5ms</td>
</tr>
<tr>
<td>OH Dev.</td>
<td>5ms</td>
<td>10ms</td>
<td>5ms</td>
<td>10ms</td>
</tr>
<tr>
<td>Kick Offset</td>
<td>0ms</td>
<td>-2,5ms</td>
<td>-1ms</td>
<td>-2,5ms</td>
</tr>
<tr>
<td>Snare Offset</td>
<td>0ms</td>
<td>-5ms</td>
<td>-2,5ms</td>
<td>-5ms</td>
</tr>
<tr>
<td>Swing</td>
<td>0ms</td>
<td>5ms</td>
<td>0ms</td>
<td>5ms</td>
</tr>
<tr>
<td>Drift amount</td>
<td>10ms</td>
<td>10ms</td>
<td>10ms</td>
<td>15ms</td>
</tr>
<tr>
<td>Drift Frequency</td>
<td>1Hz</td>
<td>0,5Hz</td>
<td>1Hz</td>
<td>0,5Hz</td>
</tr>
<tr>
<td>Fill-in Haste</td>
<td>0ms</td>
<td>0ms</td>
<td>0ms</td>
<td>0ms</td>
</tr>
</tbody>
</table>

Table 12: Settings used in the drum machine when rendering the stimuli used in the evaluative listening test.

4.3 Listening Test Results

The data in the evaluative listening test was also analyzed using Statistical Graphics Corp. Statgraphics Plus 4.0. As can be seen in table 13; the rankings of the two stimuli rendered by the drum machine are ranked as more preferred than the stimuli of the drummers.

<table>
<thead>
<tr>
<th>Subject</th>
<th>T1</th>
<th>Average rank</th>
<th>Subject</th>
<th>T2</th>
<th>Average rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dm01</td>
<td>14</td>
<td>18,2143</td>
<td>Dm01</td>
<td>14</td>
<td>23,8929</td>
</tr>
<tr>
<td>Dm02</td>
<td>14</td>
<td>27,8214</td>
<td>Dm02</td>
<td>14</td>
<td>34,3929</td>
</tr>
<tr>
<td>S1</td>
<td>14</td>
<td>53,2143</td>
<td>S1</td>
<td>14</td>
<td>45,5714</td>
</tr>
<tr>
<td>S4</td>
<td>14</td>
<td>41,5714</td>
<td>S3</td>
<td>14</td>
<td>41,0714</td>
</tr>
<tr>
<td>S5</td>
<td>14</td>
<td>36,6786</td>
<td>S5</td>
<td>14</td>
<td>42,5714</td>
</tr>
</tbody>
</table>

Test statistic = 24,5778
P-Value = 0,000 061

Test statistic = 15,3653
P-Value = 0,004

Table 13: Kruskall-Wallis test of the ranking from 1-5 (1 being preferred the most and 5 the least) of the subjects’ performances in the two tempi.

The p-values indicate that there are statistically significant differences in ranking between at least two of the stimuli in both T1 and T2. By looking at the Box-and-Whisker charts in Figure 13 and Figure 14 the diversion of the rankings for each stimulus along with the average and median rankings can be seen.
The Box-and-Whisker charts indicate that there are evident differences between the rankings of each stimulus. In table 14 and table 15 are the p-values obtained when each stimulus was individually tested with each other using the rank-test Mann-Whitney (Wilcoxon) test.

“The Rank Test combines the two samples, sorts the data from smallest to largest, and compares the average ranks for the two samples of combined data. If the p-value is small (less than a desired alpha level, such as .05), there is a statistically significant difference between the means at the 95 percent confidence level.”

In other words; when testing the rankings of two stimuli in the Mann-Whitney test and the obtained p-value is 0.05 or less - the possibility that the difference in rankings has not occurred by chance is 95% or more. But since the total amount of tests is 20, 10 for T1 and T2 each, the probability of a Type

---

9 Statistical Graphics Corp.
I error in one of the significance tests increases and therefore the confidence level is increased to 99%. These rank-tests enable the possibility of seeing between which specific stimuli there are, and where there are not, significant differences (if the p-value is lower than or equals to 0.01 there is a significant difference).

<table>
<thead>
<tr>
<th></th>
<th>Dm01</th>
<th>Dm02</th>
<th>S1</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dm01</td>
<td>-</td>
<td>0.253</td>
<td>0.0001</td>
<td>0.0009</td>
<td>0.003</td>
</tr>
<tr>
<td>Dm02</td>
<td>-</td>
<td>-</td>
<td>0.003</td>
<td>0.077</td>
<td>0.168</td>
</tr>
<tr>
<td>S1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.053</td>
<td>0.005</td>
</tr>
<tr>
<td>S4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.464</td>
</tr>
<tr>
<td>S5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 14: The p-values of each individual stimulus in T1 tested with each other using the Mann-Whitney (Wilcoxon) test.

In T1 - the rankings of stimulus Dm01 has a statistically significant difference compared to all the drummers; S1, S4 and S5. Dm02 only has a statistically significant difference compared to S1. There is also a significant difference between S1 and S5.

<table>
<thead>
<tr>
<th></th>
<th>Dm01</th>
<th>Dm02</th>
<th>S1</th>
<th>S3</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dm01</td>
<td>-</td>
<td>0.867</td>
<td>0.005</td>
<td>0.037</td>
<td>0.015</td>
</tr>
<tr>
<td>Dm02</td>
<td>-</td>
<td>-</td>
<td>0.003</td>
<td>0.035</td>
<td>0.018</td>
</tr>
<tr>
<td>S1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.671</td>
<td>0.742</td>
</tr>
<tr>
<td>S3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.907</td>
</tr>
<tr>
<td>S5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 15: The p-values of each individual stimulus in T2 tested with each other using the Mann-Whitney (Wilcoxon) test.

In T2 - the rankings of both Dm01 and Dm02 are statistically significant compared to the rankings of S1, but not to S3 and S5 at a 99% confidence level.

4.4 Discussion of Results

By just glancing at the results of this evaluative listening test it seems quite obvious that the drum machine is not only accepted – but also generally preferred over the drummers with a convincing statistical significance. The conundrum is how a drum machine which is based on human timing patterns can with a very limited amount of timing data render similar timing patterns which are rated as musically superior over the original timing
patterns. First of all to the rendering of the drum machine stimuli; the drum machine has – even with humanizing timing settings – a natural “inclination” to avoid larger timing errors. Human drummers tend to timing-wise misplace strikes since drumming is a very complex task; a momentary loss of concentration or when in doubt as to which striking combination is to be used next can distract the drummer and can cause a small, but musically audible hesitation. The second issue is the fact that the stimuli provided by the drummers have been drum replaced. This was a necessary task to perform due to the fact that the output of the drum machine has a different sound compared to the real drummers’ and thus the listening test participants would immediately recognize which stimuli were “real” and which were not. Still the sound replacement must have affected the musical feel of these specific stimuli since it deprived them of their original timbre variations and dynamics. When comparing the rankings of the drummers’ stimuli in T2 of the second listening test (figure 14) with T2 in the first listening test (figure 7) it is clear that their joint rankings are somewhat dissimilar. Where S1 in T2 of the first listening test was ranked less preferable than S3 and S5 and when drum replaced in the second listening test it is ranked as being comparably preferable. This matter is not within the boundaries of this essay to analyze but it is still an interesting fact that when drum replacing drum tracks it may very well be so that the perception of their musical timing is altered.

Although Dm01’s ranking was not significantly different from Dm02’s ranking in either T1 or T2 the Box-and-Whisker charts of figure 13 and 14 implies that Dm01 is slightly more preferable. This can be due to a various number of factors; every parameter in the drum machine was set to a more tolerant, or greater, value in Dm02 in both tempi (except for drift amount in T1 which was identical, see table 12). Any of these parameters – or any combination of these parameters – could have given the Dm02 stimulus a less preferable characteristic. And to make it more complex; the drum machine is random in nature and will not generate a drum track exactly according to its parameters; increasing parameter values does not guarantee an increase in these actual parameters in short-time drum tracks. Thus Dm02 does not have to differ as much from Dm01 as the parameters suggest.

In the individual rank-tests, which results are presented in tables 14 and 15; the confidence interval was as said set to 99% which obviously increases the possibility of a Type II error (failing to reject a false null hypothesis – or stating that the result occurred by chance when it has statistical significance). The comparisons of the rankings of both Dm01 and Dm02 would have a statistical difference compared to the rankings of all the subjects’ performances at a 95% confidence level in T2 and which is very important to
mention. The reason why the possible occurrence of a Type II error is preferred to a possible occurrence of a Type I error in T2 is because I feel that it is more appropriate to state that the results imply that the rankings of the drum machine’s stimuli were slightly different than the rankings of the subjects’ drum performances than it is stating that the drum machine’s musical timing is significantly more preferred over human drum performances - when it indeed might not be.
5. Conclusions

The aim of this essay was to analyze real human drummers for timing patterns, extract these patterns and isolate them from each other, describe how and when they operate and implement them as C++ code in a drum machine. To tie the ends together and put the drum machine’s performance into context a listening test was conducted and the results show that the drum machine performs well above expectations.

The original aims of improving the existing drum machine from a “dullness circumventing drum machine” into a “human performance emulating drum machine” with as few limitations as possible has been achieved. Based on human drum performances of five amateur drummers a set of parameters were described and implemented into a drum machine. All parameters can be set with high precision between 0 and well above what can possibly be musically desired, thus giving the user virtually endless combinations of parameter settings.

5.1 Final Remarks

Unlike the original drum machine developed for the C-extended essay, the drum machine is in its current shape and form fully useable. It has already been tested by a few amateur songwriters who have said that it was both easy to get started with and easy to configure, and also that the output was above their expectations. My educated guess is though that most songwriters, in whatever profession or level of experience, will not be satisfied with the fact that no other than the original complementary sound samples can be used in the drum machine.

During the process of developing the drum machine, conducting listening tests and analyzing drum performances, there were no greater unanticipated problems or set-backs that were encountered.

5.2 Further Research and Future Development of the Drum Machine

In research there are matters which need to be further investigated to enable a more precise and accurate description of the timing patterns. Given enough data it would be of interest to see how different conditions such as changes in tempo, variations in complexity and extreme tempi effect drift, flutter and internal deviations. Another aspect of research is that with more data from individual drummers (old recordings and such) – authentic timing profiles of these drummers could be created. This would enable the creation of timing presets based on, as an example, celebrity drummers.
Of course there is another view-point which must be considered in order to be able to strive for perfection; the striking dynamics. For now these are rendered randomly although they also are a part of the musical performance.
6. References


   URL: http://www.acoustics.org/press/137th/friberg.html

[6] Steinberg VST 2.0 Software Development Kit

   URL: http://www.steinberg.de/steinberg/ygrabit/index.html
7. Appendices

Appendix 1 – Notation for T1 (110bpm)

Appendix 2 – Notation for T2 (160bpm)
Appendix 3 – Listening Test Form for both Listening Tests

Please fill this out first:

Name:

Occupation:

I play the following instrument(s):

I generally listen to this genre of music:

I generally play/mix/master/perform this genre of music:

Category 1
1:
2:
3:
4:
5:

Comments (optional):

Category 2
1:
2:
3:
4:
5:

Comments (optional):

Thank you very much for your participation.
Kahl Hellmer
Appendix 4 – Screen shot of the Drum Machine
Appendix 5 – CD-ROM
- Drum Machine Plug-in
- Drum Machine’s sound files
- Sound examples
- Source code