Music Synthesis - Automated Rhythm Generation

by

Chintapalli Tarun (Roll No. 150102014)

Sai Charan Bodda (Roll No. 150102013)

SUBMITTED IN THE PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

Under the guidance of

Dr. K. Karthik

DEPARTMENT OF ELECTRONICS & ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

APRIL 2019

CERTIFICATE

This is to certify that the work contained in this thesis entitled

Music Synthesis: Automated Rhythm Generation

is the work of

Tarun Chintapalli

(150102014)

Sai Charan Bodda

(150102013)

for the award of the degree of Bachelor of Technology, carried out in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.

____

Guide

Date:____________

Place:___________

DECLARATION

The work contained in this thesis is our own work under the supervision of the guides. We have read and understood the "B.Tech./B.Des. Ordinances and Regulations" of IIT Guwahati and the "FAQ Document on Academic Malpractice and Plagiarism" of EEE Department of IIT Guwahati. To the Best of our knowledge, this thesis is an honest representation of our work.

_______________	_______________
Author	Author
Date:______________	Date:___________
Place:_____________	Place:___________

Acknowledgments

First and foremost, We want to thank our supervisor Dr. Kannan Karthik for his warm encouragement and thoughtful guidance. It has been an honor to work under his distinguished guidance. He has taught us, how good experimental research is done. I appreciate all his contributions of time, and ideas to make this Bachelor's Thesis Project experience productive and stimulating.

I would like to thank my family members for holding my hand during tough times and for motivating me for the studies.

1. Abstract

Music is one of the most complex yet beautiful forms of art where different sounds can invoke different feelings within self and each feeling can be accurately represented and conveyed in terms of music. Rhythm, a pattern of sound in time, is a core element of music which when tangled with harmony and melody invoke different feelings. Not every pattern of sound makes a powerful rhythm.

This thesis describes the process of generation of several families of rhythm and their analysis for finding out what properties make a rhythm sound good. Many good rhythms satisfy properties such as *Maximally even property* and *Syncopation* are presented and explained with examples. These properties help us in better understanding of music and a model to generate good Rhythms. It is also described how morse code and modified morse code can be used to produce quite interesting rhythms including folk drumming

In the later section of this thesis, different methods of generation of rhythmic sequences like 'Rhythm generation using probability distributions' and state transition methods morse words are explored and finally, various modifications are made to Morse code model to generate appealing rhythms.

Table Of Contents

2. List of Figures

3. Nomenclature

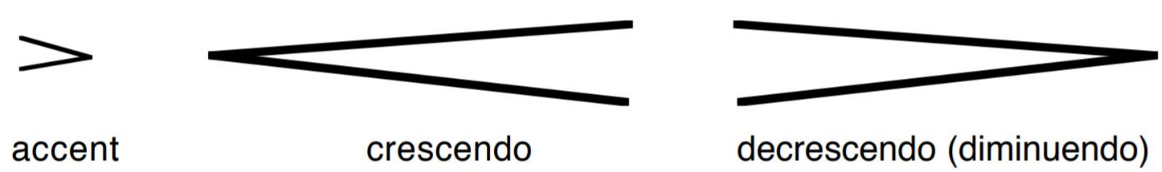
Rhythm: We usually think of rhythm as the element of music, but it's actually found everywhere in the world around us, from ocean tides to our very own heartbeats, a rhythm is essentially an event repeating regularly over time.

Pitch: Pitch is how high or low a sound is, and it corresponds to the frequency of a wave. Women, generally tend to have high pitch than men.

Harmony: when we combine notes or pitches into chords, we come up with something called Harmony. Harmony is the verticalization of the pitch. Harmony is described in terms of its relative *harshness*. *Dissonance* is a harsh sounding harmonic combination, *Consonance* is the smooth sounding harmonic combination.

Pulse: This is typically what listeners sweep along with the flow as they tap their foot or dance along with a piece of music. It consists of beats, in a repeating of identical yet distinct periodic short duration impulses, perceived as points in time.

Dynamics: This is a general element involving all musical aspects relating to the relative loudness or quietness of music. When music gradually gets louder, it is *Crescendo*. When it gradually gets quieter, it is *Decrescendo*. *Accent* is punching a note harder to emphasize it.



Timbre: (pronounced as Tam-Ber) When "C" is played in a piano and "C" is sung by a chorus, or when "C" is played in Guitar, even though the underlying pitch is same, all of them sound different. We can safely say that each musical instrument produces its own characteristic pattern of "overtones", which gives it a unique TIMBRE.

Texture: The number of individual musical lines (melodies) and its combination of these lines with one another are called texture. These are divided into *monophonic, homophonic* and *polyphonic* textures.

Duple: A meter with only two mail pulses or beats, one weak and one strong is called duple. Eg: a rhythm involving only THOCK and a THACK.

Onset: It is the beginning of a musical note or a sound.

Syncopation: It is an interruption of the regular flow of rhythm, which makes part of the rhythm, off-beat. This element in rhythm makes it unexpectable.

Staff: The parallel lines in the music bar notation which represents a time signature.

Timeline: That part of the rhythm, which repeats itself over time again and again.

4. Introduction

Music is an art form whose medium is sound organized in time. It is the art medium that communicates interiority, being perceived by the ears, and received by the mind. Music is composed of pitch, rhythm, dynamics, timbre, and texture. Rhythm is the timing of musical sounds and silence over time. It is Rhythm that moves our hands, feet, and bodies to the pulses of the universe[Jason martineau].it is the backbone of music. Rhythm can be found everywhere around us, It is the voice of ocean tides. It is the music of our heartbeats, But every periodic repetition of a single beat does not make it a beautiful rhythm. There has to be at least one opposing beat which can be the unstressed offbeat or the accented backbeat to make it a good rhythm.[John Varney]. These beats can be made distinct in several ways such as high and low drums or long and short beats or a beat and silence.It's always been a question for ancient scholars, philosophers and even modern scientists why some rhythms sound so sweet while others are downright dreadful and wondered if there is any relation between Rhythm and Mathematics.

In this Thesis, different approaches such as constant meter, Euclidean and maximally even rhythms and Morse code and Morse words and modified morse words and its variations along with ‘Rhythm generation using probability distributions’ and state transition methods explored to generate different classes of appealing rhythms.

As per future work, These above-mentioned properties along with the analysis of an above-generated class of rhythms to find a common property that can differentiate good rhythm from a bad rhythm and develop a metric for quality assessment of music to generate rhythms automatically.

In this research context, A rhythm is considered as the recurrence of beat or pattern over time. Duples are also considered for this purpose.

5. Literature Review

5.1. Representation of Rhythms

Musical Bar Lines Notation:

It is a standard notation for music, Rhythm is indicated on musical bar lines where each beat is represented by stoke and staff represents time signature.

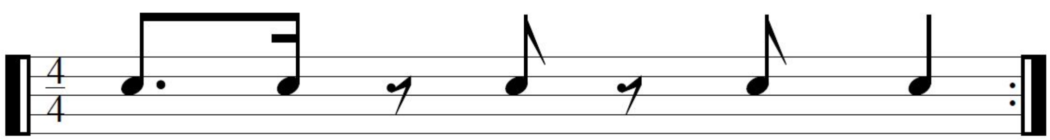


Fig 1: Musical Bar lines notation of *Clave-Son*

Image source : [Toussaint]

Wheel Notation:

Wheel Notation is one of the most intuitive notations for tracing a rhythm described by John Varney.

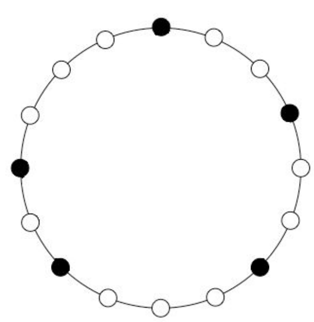


Fig 2: John Varney's Wheel representation of *Clave-Son*.

In this notation, the circular clock represents the linear passage of time. Its circumference is divided into equispaced points representing pulses.

While black dots represent onsets, white dots represent silence. A circular wheel notation can be a better way to visualize a rhythm, than a normal linear scale for rhythmic timelines which needs to be looked back and forth.

Box Notation:

It is one of the famous representation of rhythm and is also called as *Time Unit Box System* (TUBS).In this notation, a pulse is represented as box and the dot inside the box represents onset at that particular pulse.

●			●			●				●		●			

Fig. 3: Box notation of the *Clave-Son*

Interval Notation:

In *interval Notation,* the duration(in terms of pulses) between successive on sets is given in row matrix. Interval notation for Clave Son is shown below.

[**2-2-3-2-3**]

Here first integer ‘’2’’ in the above notation represents an onset followed by two pulses of silence followed by an onset.

In this thesis, we mostly explore duple rhythms composed of either a long note and a short note or onset and silence. We also make use of Box Notation and Interval Notation.

5.2. Constant Time Rhythm Generation

The basic underlying mathematical rule in this method is sum to a number N. Here we take the value of N as 8. There can be only two integers 1 and 2 in the sum expression. Using these two integers, with a replacement we need to find the sequence that sums up to 8. Where 1 represents a short beat and 2 represents a long beat. There can be as many as 34 ways of representing 8 as sum if 1's and 2's.

1+1+1+1+1+1+1+1 = 8
1+1+1+1+1+1+2 = 8
1+1+1+2+2+1 = 8
.
.
.
2+2+2+2 = 8

From such family of sequences take a sequence, for example, 1,1,2,1,2,1. This sequence when played, where 1's replaced by a short beat, 2's represented by a long beat, sounds pleasant to hear.

5.3 Euclidean rhythms

Euclidean Algorithm for finding the greatest common divisor of two numbers can be used to generate most of the famous Music Rhythms[Godfried]. These Rhythms generated using Euclidean algorithm are called Euclidean Rhythms.

Euclidean Algorithm for Finding Greatest Common Divisor

Euclidean Algorithm is a recursion based algorithm where the smallest number is recursively subtracted from a larger number until both are equal or the smallest number is 1

Explanation: Consider two numbers 11 and 5 represented as (11,5)

ie.,(larger number, smaller Number)

(11,5) ->(6,5)->(5,1)

11-5=6

Larger 6, smaller number 5

6-5=1

Larger Number:5 , smaller Number: 1

GCD of(11,5) is 1;

Generation of Rhythmic Pattern using the Euclidean Algorithm

Euclidean Algorithm of Euclid has the same structure as Bjorklund Algorithm for SNS timing systems[Godfried]. The aim is to distribute k onsets over n time intervals. It can be viewed as a binary sequence of k ones and n-k zeros spread in such a way that distance between any two consecutive 1's is maximum. Where '1' represents a pulse of onset and '0' represents a pulse of silence.

By representing in this manner, the problem statement reduces as follows: to construct a binary sequence of length n with k ones distributed within the sequence as evenly as possible. At the beginning of the algorithm, ones and zeros are divided into 2 groups. In each iteration elements from larger group equal to that of the smaller group are removed and merged with a smaller group until the size of the smaller group is '1'.

Explanation:

Let us take an example of a sequence of 13-time intervals and 5 onset pulses, i.e 5 ones and 13-5 = 8 zeros, which can be represented in binary as follows:

[1 1 1 1 1 0 0 0 0 0 0 0 0]   (8,5)

After first iteration, 5 elements from larger group(0) is merged with (1's).

[10 10 10 10 10 0 0 0]   (5,8-5) = (5,3)

After 1st iteration larger group will have 5 elements and smaller group have 3 zeros.

Which is similar to the Euclidean Algorithm.

After the 2nd iteration, we get

[100 100 100 10 10]   (3,2)

After 3rd iteration, we get the following sequence with the smaller group has 1 element

[10010 10010 100]

This is where we terminate the algorithm.

Thus the sequence [1 0 0 1 0 1 0 0 1 0 1 0 0] is the resultant Euclidean Rhythm.

Similarly for the example of 16-time intervals and 5 pulses , which can be represented in binary as

(5,11)   [1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]

(5,6)   [10 10 10 10 10 0 0 0 0 0 0]

(5,1)   [100 100 100 100 100 0]

Which gives rhythm [1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0]

5.4. Maximally Even Rhythms

we can observe that the onsets in Euclidean rhythms are evenly spaced out which paves for speculation of, "does Rhythmic patterns satisfying Maximally even property are pleasant to hear?".

For Maximally even duple made of Onset and silence, the duration between adjacent Onset should be maximum.

For a rhythm made up of 16 pulses (11,5) i.e. (11 pulses of silence and 5 pulses of onset).

Maximum Onset distance will be 16/5=3.2, which when used will result in periodic onsets which makes rhythm boring. *Syncopation* in rhythm makes it more interesting.

So the onset is played at nearest integer either less than or greater than the decimal number [Godfried].

For a 16 pulse rhythm with 5 beats, there are 16 maximally even patterns containing famous rhythms such as Rap, Shiko, Clave Son.[Godfried] and Euclidean Rhythm generated will be one of the 16 patterns generated by Maximally even Property.

5.5. Morse Code in Rhythm Generation

In Morse code, Each alphabet or symbol are represented as unique sequences of "Dits" and "Dahs" separated by silence and is mostly used in the transmission of text information.
For a given Sentence, Morse code representation of the sentence/ word gives a binary sequence. if we consider "Dit" as a short node and "Dah" as a longer node. Each unique word will give rise to a unique Duple.

6. Methodology

6.1. Project Aim

The aim of the project is to generate rhythmic patterns that are pleasing to hear. In this work, we show the different ways to generate good rhythms employing various algorithms.

6.2. Approach

The problem statement can be broken down into two parts.

1) Synthesis of Rhythmic Patterns: generating of rhythmic patterns
2) Analysis of Rhythms: quality assessment of rhythms.

A rhythm can be represented mathematically as

$$Y(t) = \sum_{i=1}^{n} A_i h(t - t_i)$$

where,

A_i is the amplitude of the beat which is a random variable in [0,1].

h(t) is an impulse function representing a beat.

t_i is the time instant at which onset is present.

Understanding the variations in amplitude and the time intervals between the onsets is essential for the quality of rhythm.

we can observe that the synthesis of Rhythmic patterns can be divided into 2 parts.

1) Rhythmic patterns with constant Amplitude across all beats.
2) Rhythmic patterns with different amplitude across beats.

In this phase of the project. rhythmic patterns with constant amplitude (part 1) and rhythm with long and short beats are considered.

6.3. Generation of 8 phase(constant Time) Rhythmic patterns

This method constrains the total time taken by the rhythm to be constant. Then the time is filled with long beats and short beats to exactly fit the total time with short beats spanning one pulse and a long beat spanning 2 pulses.

For the generation of these patterns, the stochastic approach is implemented, where the long beat or short beat are added to the pattern randomly until total time equals is 7 or 8 pulses. If the sum is 8, iteration stops and the pattern generated is played. If the sum is 7, a short beat is added at the end the pattern generated is played accordingly.
As explain in Literature review, most of the good rhythms are from sequences with the sum of long beats almost equal to that of short beats but not vice-versa.

6.4. Generation of Euclidean and Maximally even Rhythms

As Explained in Literature Review, Maximally Even rhythms can be generated by a syncopation of rhythmic onset to its nearest Integer.
For a (16,5) rhythm i.e, for 16 pulses and 5 onsets. For Maximally Even the positions of onsets should be at 1, 4.2, 7.4, 10.6 and 13.8.
Here beat at 4.2 can be played at either 4 or 5 similarly for 7.4, 10.6 and 13.8 giving rise to a total of 16 patterns which can be generated using Backtracking method.

16 almost maximally even shown in Fig. 4 where empty box represents silence and the dot represents onset. We can observe that the Euclidean rhythm generated in Literature review is $16^{th.}$ Rhythm in the below sequence.

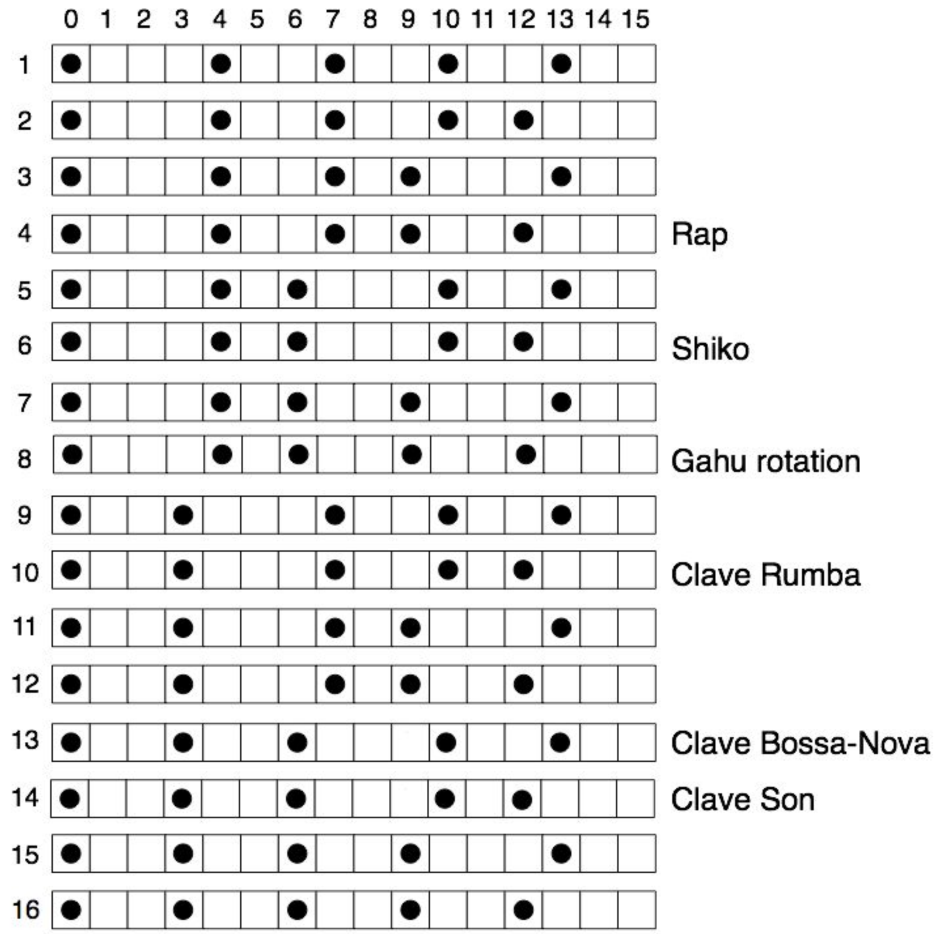


Fig.4: 16 Box Notation of 16 maximally even rhythms.

Image source:[Toussaint]

We can observe that not all maximally even rhythms all good to hear such as pattern 1, 9, 16 compared to those of 10,13.

We can observe that even though 1 is just a rotation of 13. Rhythm 1 is not good whereas rhythm 13 is one of the popular world rhythm called Clave Bossa-Nova. Names of other famous rhythms are mentioned next to rhythm sequence

6.5. Generation of Rhythmic patterns using Morse Code

As Explained in Literature Review, Each alphabet of the considered sentence is replaced with the corresponding Morse Code which gives rise to rhythmic pattern with “dit” being a short beat and “Dah” being a long beat.

For Example, consider the word “DELTA DAWN”

Morse Code for D=” - . .”

Morse Code for E=" . ”

Morse Code for L=”. - . .”

Morse Code for T=” - ”

Morse Code for A= ” . - ”

Morse Code for W= ” . - - . ”

Morse Code for N= ” - . ”

Total Rhythm=” - . . . . - . . - . - - . . . - . - - - .”

Which can be represented as [4-2-1-3-1-3] in interval notation when viewed as offset and silences. We can observe that some sentence such as “JUMPING JACK” is good to listen whereas sentence such as “LIVE LOVE LEARN” is not that good.

Rhythms from many Sentences are generated and classified into good or bad rhythms Subjectively.

Future Work: Finding a common pattern in these good and bad rhythms which can be useful for quality analysis of rhythms.

6.6. Stem Chart Extraction from Rhythmic patterns

Stem chart of music represents the beat instances of a rhythm over time. From the stem chart, we can extract time intervals between the beats, which can be used for the next phase, i.e. for quality assessment. These time intervals and the subjective score, based on the quality of rhythm, can be used to train an algorithm to assess the further quality assessment or for automatic rhythm generation.

To extract the stem chart from audio (music):

1. The audio clip is divided into windows of size 30 ms with an overlap of 10 ms between consecutive windows
2. The short-term energy of each window is calculated.
3. The onset is said to be at the particular window if the short term energy of the particular window is greater than the average of the short term energies across all windows.
4. These onsets found in above step plotted and the time intervals between the onsets are calculated.

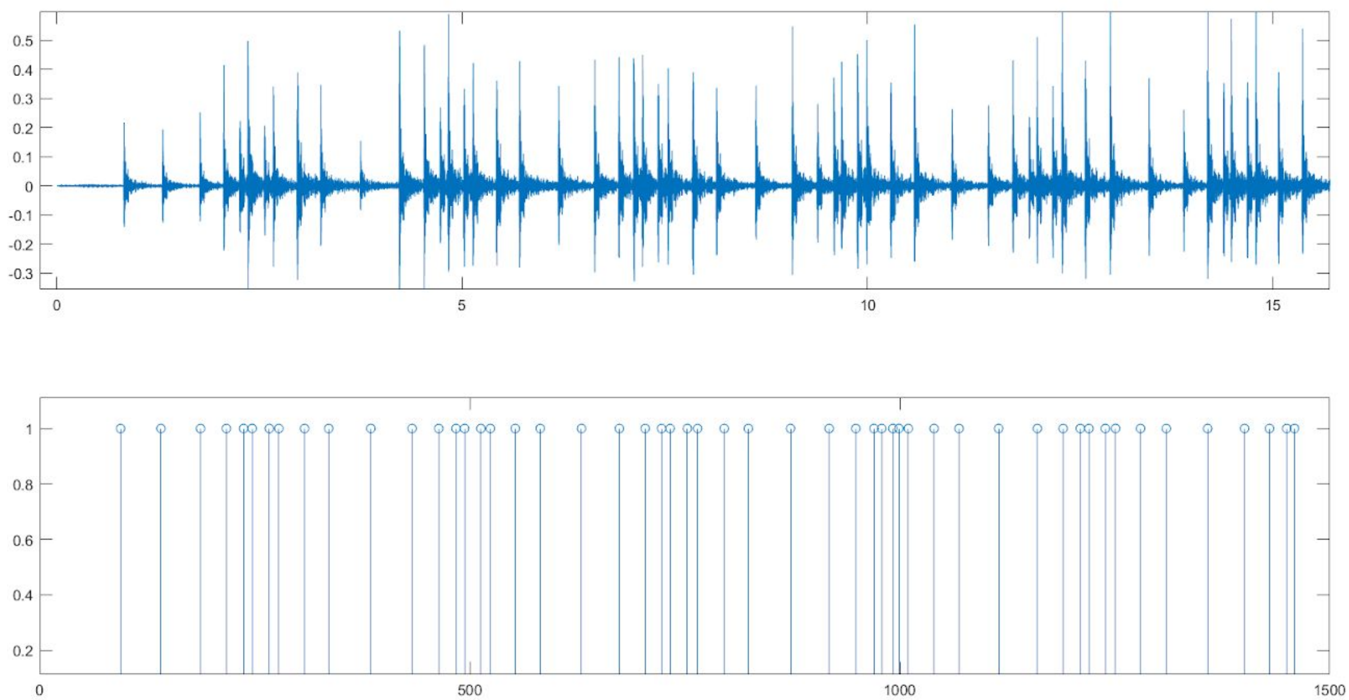


Fig.4: Audio file and Corresponding Stem chart

6.7. Generation of rhythms using Probability Distribution

Introduction to model:

Every rhythm can be represented in discrete time in its stem chart representation with each impulse in stem chart corresponding to beat**.** We assume that the occurrence of the beat at a particular discrete instant 'n' depends on the previous beats based on some known probability distribution. In other words, the probability of occurrence of the beat at discrete instant 'n' is a function previously occurred beats.

Model Description:

We assume that probability function to be discrete Gaussian distribution $f = D(N(\mu,\sigma^2))$ with μ is mean and σ^2 is variance.
Where $D(n,q) = 1\text{-integral}(N(n,\sigma^2),q\text{-}p/2,q+p/2)$ where p is the time period between 2 successive discrete time instants.
In other words, if a beat occurs at an instant 'n'

The probability of occurrence of beat at 'n+t' is

$f(n+t) = 1\text{-integral}(N(n+1,\sigma^2),n+t\text{-}p/2,n+t+p/2)=D(n+1,n+t)$..

For variance = 1,
The probability of occurrence of beat at next 5 instants

0.6171   0.7583   0.9394   0.9940   0.9998

For variance = 2,
The probability of occurrence of beat at next 5 instants

0.8026   0.8253   0.8790   0.9344   0.9722

At every instant, according to the probability of occurrence of a beat at that particular instant decided by previous beats. i.e., whether a beat will occur at that particular instant is decided by previously occurred beats.
If a beat has occurred, the probability of occurrence of the beat at next instants is decided by this new beat else it is decided by the previous beat only.

A random number generator function is used to duplicate probability.
I.e., for example, consider that a beat occurs at an instant 'n+k' with a probability of 0.6. A random number is generated between 0 to 1 and if the generated number is less than 0.6 then the beat occurs at that instant.

Algorithm:

#initialization(consider beat occurred at n=0)

- An array A of length equal to the length of rhythm which each element corresponding to time instant.in each element, probabilities of occurrence of the beat at that particular instant are stored. (initially, all are set to 0.5).

Iterate at each instant(i) until termination condition(length of rhythm):

- A random number (r)is generated between 0 to 1
- If r<A(i)
  - Sound beat at that instant
  - Update probabilities of occurrence of a beat in all successive instants according to function f as explained above.

Results:

This model is capable of producing infinite time sequence of rhythms.
Since it is dependent on probability, each execution of the algorithm with result in a new sequence with high probability.
Some of the sequences generated from this algorithm are good to hear.
As the variance increases, the density of beats occurred increased resulting in large subsequence with all instants having beats.

For variance = 1,
The probability of occurrence of beat at next 5 instants
0.6171 0.7583 0.9394 0.9940 0.9998

For variance = 2,
The probability of occurrence of beat at next 5 instants
0.8026 0.8253 0.8790 0.9344 0.9722

For variance = 3,
The probability of occurrence of the beat at next 5 instants
0.8676 0.8747 0.8938 0.9193 0.9451

We can observe that as time difference from the latest beat occurred increases probability tends to unity.
This implies that in most of the practical scenarios a beat will occur within 4 instants from the previous beat.

Version 2:

We can observe that as variance increase in above method probability of occurrence of beat after a beta had occurred tends towards 1.this results in subsequences with a large number of beats in it which are in general not very amusing to listen. To tackle this problem, the distribution is multiplied with a constant so that the probability of occurrence of the beat at next instant will be 0.5.

Define k = D(0,0)

$F_2(n+t) = 1\text{-integral}(k*N(n+1,\sigma^2),n+t-p/2,n+t+p/2)$.

For variance = 1,
The probability of occurrence of beat at next 5 instants

0.5000   0.6844   0.9209   0.9922   0.9997

For variance = 2,
The probability of occurrence of beat at next 5 instants

0.5000   0.5576   0.6936   0.8339   0.9295

For variance = 3,
The probability of occurrence of the beat at next 5 instants

0.5000   0.5268   0.5988   0.6953   0.7928

As the variance increase, the probability function tends to uniform distribution with probability at each instant is equal to 0.5.

Version 3:

Rather than multiplying with a scalar constant. The function is multiplied with another Gaussian with lower variance.

$F_3(n+t) = D(n+1,n+t)*(1-g(n))$; where $g(n)= (N(n,\sigma_{,}^{2}))$

For variance $\sigma^2= 1$,$\sigma_{,}^{2}=2$,
The probability of occurrence of a beat at next 5 instants

0.5093   0.6086   0.7753   0.8738   0.9342

For variance $\sigma^2= 3$, $\sigma_{,}^{2}=2$,
The probability of occurrence of a beat at next 5 instants

0.6624   0.6624   0.7255   0.8214   0.9084

For variance $\sigma^2 = 3$,$\sigma_{,}^{2}=2$,

The probability of occurrence of the beat at next 5 instants

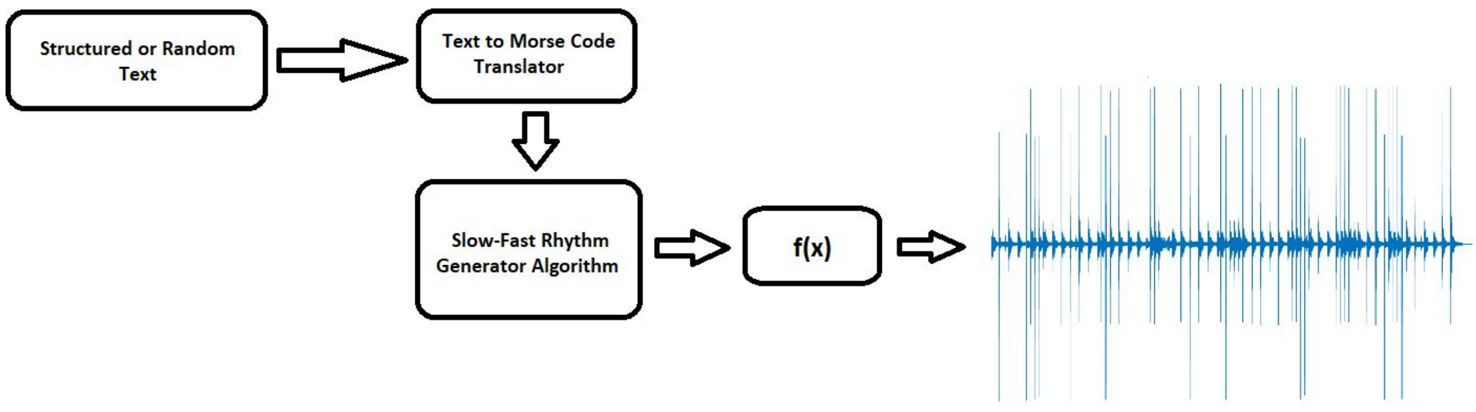
0.7161 0.7020 0.7377 0.8081 0.8831

We can observe that, by multiplying the probability for a higher variance does not converge fast to 1.the rhythms sequences generated by this version shows some significant (author feeling) better results than that of other 2 versions.

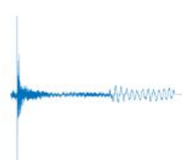
6.8. Morse Code rhythm generation with slow and fast words

This is similar to the morse code rhythm generating algorithm in section 6.5 but involves variation in tempo of different segments .i.e., there will be alternate fast and slow words. The algorithm ensures that different segments in given rhythm sequence speed up or slow down naturally and alternate in this fashion to generate good rhythms.

Flow Chart of the algorithm



Where f(x) is as follows

f(.) = THACK 

f(-) = THOCK 

Assumptions

- The words contain only 2 alphabets dit(**.**) and dah(**_**).
- The time gap between the successive beats in a slow word is 0.37 seconds and the time gap between the successive beats in a fast word is 0.17 seconds.
- In a slow word or a fast word, the minimum number of beats inside a word is assumed to be 3 and the maximum number of beats is 6.
- The reason for choosing the number is as follows:
  - The maximum time taken for slow word is considered as 2 seconds, hence the maximum number of beats is 2/0.37 ~ 6 beats.
  - The minimum time taken for fast word is considered as 0.5 seconds, hence the minimum number of beats is 0.5/0.17 ~ 3 beats.
- As said earlier, *dit* in morse code corresponds to THACK and *dah* corresponds to THOCK.

Algorithm

1. Initially, the Structured or random text is converted into morse code using morse code translator. Any space in between is neglected
2. The morse code sequence is taken as a whole (At this point of time, the morse code cannot be converted back to text)
3. Firstly, a random number is generated between the minimum and the maximum number of alphabets allowed in a word, in our case, it is 3 and 6 respectively. A random number is generated between 3 and 6, both 3 and 6 inclusive.
4. Let say, the number is **X**, now, the first **X** alphabets are taken as a word. The word is considered as a fast word, so the time gap between the alphabets are taken as 0.17 seconds
5. Again, a random number is generated between 3 and 6, both inclusive. Let the number be **Y**.
6. Next **Y** alphabets in the morse sentence are taken and considered as a slow word. The time gap between the alphabets is taken as 0.37 seconds.
7. The algorithm continues from step 3 until the whole sentence is completed.
8. The dits and dots are replaced by THACKs and THOCKs respectively.

Example:

Let the text be

“RAPID FIRE WITH THE HIGH HILLS”

The translated Morse code is as follows(neglecting all the spaces in between)

“...--.--.-.-.-.---.-..-..-..-.-.--.-.-”

Let the random number between 3 and 6 is 4,

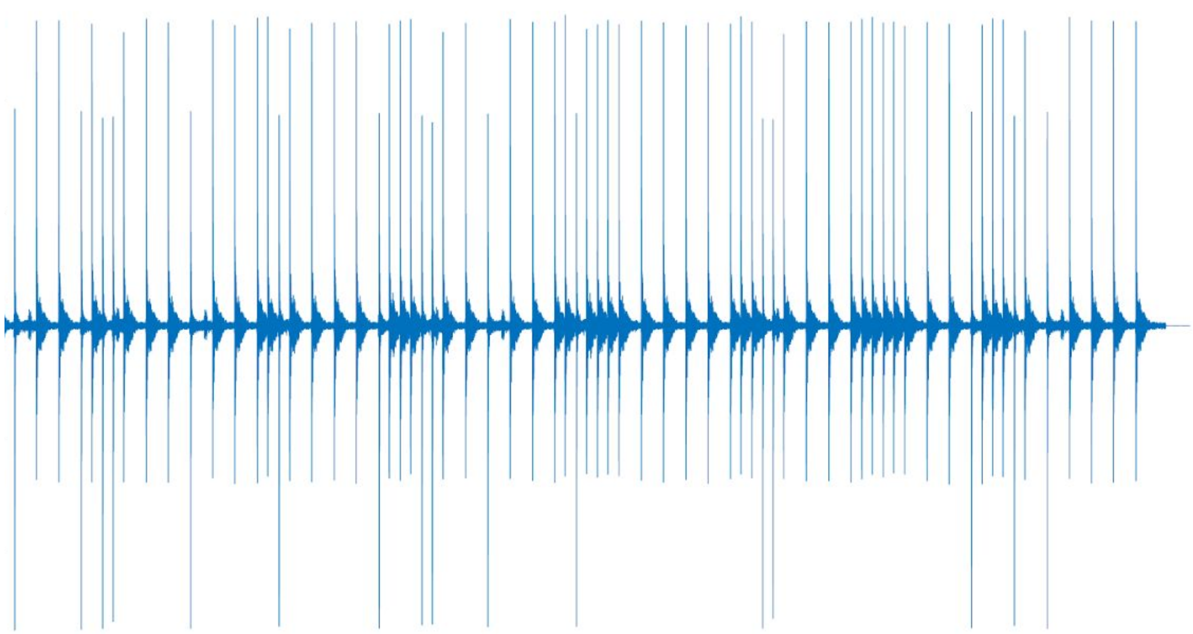
The first four alphabets, “ **...-**” are taken into consideration as a fast word, it, the time gap between the words is taken as 0.17 seconds.

. 0.17s **.** 0.17s **.** 0.17s **-** 0.17s

Now, let the next randomly generated number be 5. The next 5 alphabets, “**-.--.**” , are taken as the slow word. The time gap is 0.37 seconds

- 0.37s **.** 0.37s **-** 0.37s **-** 0.37s **.** 0.37s

Now. the final step, replacement of dits and dahs,
The final result is a piece of pleasant music as follows



Observations

The algorithm is implemented in Matlab. various audio files are generated, taking different types of the input text.
Three types of text are taken into consideration, structured English sentence, random text, and text with repeating alphabets.

Structured text: "**RAPID FIRE WITH THE HIGH HILLS**"

The audio file obtained from the algorithm with this text is quite good. There is no predictability in the rhythms which makes it even more interesting(syncopation).

Random text: "**IWACKJLNLKWEK**"

This text also gives out a good rhythm

Repeating alphabets: "**OOOOOOOSSSS**"

The morse code of this sequence is a structure with repeating words. This makes the rhythm output predictable and makes it a bit boring.

Conclusion:

The rhythms produced by the tempo variation of Morse code are better than that of rhythms generated by morse code alone. If the input text is any structured English text or any random text, the algorithm gives out good rhythms. But the algorithm produces a bit boring rhythms if the input text contains repeated alphabets.

6.9. Rhythm generation using Morse words transition.

Introduction:
Musical word: In the context of this thesis, a musical word is a unique sequence of beats of finite length. Morse words are a class of musical words with each word sequence being morse code of the English letter.

I.e., B = ' - . . . ‘is a word.
G = ' - -. ' is another word …

In this approach for rhythm generation, we assume that rhythm is composed of musical words(analogous to English sentences) and translation from one word to another is controlled by a state transition chart.

A state transition chart is a matrix with each element A(i,j) corresponding to the state transition probability(probability of transition from i to j)

$$\rightarrow \Sigma_{j} A(i,j)=1.$$

State transition chart for one word to another word is assigned as the normalized distance between 2 beats.
A new measure Distance between two beats is defined as the Euclidean distance between the last 3 symbols of previous words to the first 3 symbols of the next words. For the computation of distance the morse symbol‘- ‘is assigned 1 and ‘.’ is assigned as ‘1’
Ex: the distance between words L (. -..) and P(. - - .) is computed as follows

Last 3 symbols of L (-1,1,-1,-1)~(1,-1,-1)
first 3 symbols of P (-1,1,1,-1),~(-1,1,1)
Distance between L and P is Euclidean distance between (1,-1,-1) and (-1,1,1) =12;
Distance between P and L is Euclidean distance between (1,1,-1) and (-1,1,-1) =8;
Observe that LP is not equal to PL.

State transition probability for one word to another word is assigned as the normalized distance between 2 beats.

Probability of transition from word A to word B is calculated as

$$\frac{\textit{distance between a and b}}{\textit{sum of the distance between a to all possible words}}$$

These probabilities are used for the transition of one word to another.

The beat generation algorithm will assign the next word randomly based on probabilities from STC.

This whole rhythm generation model can be viewed as a **Markov chain** with each word corresponding to states and state transition chart defines the probability of transition from one state to another.

Results:

- This algorithm is capable of producing infinite sequences.
- Although the rhythm sequences produce are not appealing to hear, **An effective definition of state transition measure can result in good rhythmic sequences.**
- Since the generation of rhythm used probability for finding the next word in sequence. The same starting word can result in a different outcome and also it will not produce a cyclic sequence.

6.10. Morse Code music generation with modified Morse words

In this version of rhythm generation algorithm, a sequence is generated from the class of morse words with the time gaps between the beats varying randomly. Morse words in this version are different from the typical morse code used in communication. (basic *dit*s and *dah*s remain same).

The alphabets and the respective morse codes used in this algorithm are as follows:

'a' → ..-.
'b' → .---
'c' → ..-
'd' → -.--
'e' → .--
'f' → .-.-.-..-

In this version, the alphabets are limited to 6 to reduce the complexity.

Algorithm

1. Initially, a blank array **T** of **n** elements is considered.
2. The array **T** is randomly filled with -1 and 1.
   T(n) = -1 or 1.
3. An array **G** is built in the same fashion as the Fibonacci series but in modular form.
   G(n) = (G(n-1) + G(n-2)) % 6
   G(1) = 1, G(2) = 0;
4. At any time instant **n**, an alphabet (a,b,c,d,e,f,g) with index G(n) is played with time gap 0.1 seconds or 0.2 seconds, depending on whether T(n) is 1 or -1.

Conclusions

The above algorithm can generate rhythmic sequence based on morse words where tempo of the word is decided randomly.

6.11. Folk Drumming using modified words in Morse Code

This version of Morse words music generator aims at generating folk drummings using modified morse words. This algorithm divides morse words into two categories, *Opening words* and *Closing words*. There is a word onset or *Opening word*, which is immediately followed by a *Closing word*. The *Opening* and *Closing words* come from disjoint subsets. The *opening words* may vary, but the *closing word* remains same and repeats itself after every *opening word*

The *Opening words* are as follows
a → ..P--
b → .P.P--
c → .P.P--
d → .P---
e → .-.P--
Here, P represents a pause,

Similarly, the *Closing word* is
f → --P..

Algorithm

1. Initially, a blank array **G** is taken with n elements
2. An array **G** is built in the same fashion as the Fibonacci series but in modular form.
   G(n) = (G(n-1) + G(n-2)) % 5
   G(1) = 1, G(2) = 0;
3. At any time instant **n**, an *Opening word* (a,b,c,d,e) with index G(n) is played with time gap of 0.2 seconds. When it is a pause **P**, then it is 0.3 seconds silence.
4. After G(n) is played, which is an *Opening word*, *Closing word* f is played immediately.

Conclusion

The above algorithm generates a good rhythm of folk drumming pattern, where closing word is same and opening words differ.

6.12. Morse Code Music Generator with rhythm Alternation

This version of Music generation using morse words is similar to the previous section 6.11 (*Opening words* and *Closing Words*). The previous version contains only one *Closing word*. This version contains 3 *Closing words*. There is a special property for *Opening* and Closing words. *Opening words* are simple and slow. *Closing words* are fast and complex.

Opening words are as follows:

a → **..P-**
b → **.P.P.P--**
c → **-.P.P.P--**
P represents a pause

Closing words are as follows:
f → **.-.--.--..**
g → **..--.-.-**
h → **--..**

Algorithm

1. Initially, a blank arrays **G** and **H** is taken with n elements
2. Arrays **G** and **H** are built in the same fashion as the Fibonacci series but in modular form.
   G(n) = (G(n-1) + G(n-2)) % 3
   G(1) = 1, G(2) = 0;
   H(n) = (H(n-1) + H(n-2)) % 3
   H(1) = 1, H(2) = 0;
3. At any time instant **n**, an *Opening word* (a,b,c) with index G(n) is played with time gap of 0.1 seconds. When it is a pause **P**, then it is 0.3 seconds silence.
4. After G(n) is played, which is an *Opening word*, a fast and complex *Closing word* (f,g,h) with index H(n) is played immediately with time gap 0.2 seconds.

Conclusion

The above algorithm generates a good rhythm where closing words and opening words differ. A simple word followed by a fast and complex word sounds quite interesting.

7. Results and Conclusions

Different techniques such as Constant time technique, Euclidean Algorithm, Maximally even pattern, Morse encoding, Morse words and modified morse words and probability distribution and state transition methods are used to generate different families of rhythms and each of these class of Rhythms are determined whether they are interesting or not.We found that some of these generated rhythms sound good while some are not pleasant to hear.

In Constant time (8 phase) rhythm, Most of the good rhythms are from sequences with the sum of long beats almost equal to that of short beats and long and short beats evenly spread.we have also found that a few of maximum even rhythms generated are good to hear and these are the sequences having syncopation in their pattern.

Most of the rhythms synthesised from Morse code representation of structured text are not that appealing whereas rhythms generated from the unstructured text are interesting.irrespective of the structure of text changing the tempo of subsections of above rhythmic sequences improves the quality of rhythm.

Although the rhythms generated from ‘probability distribution’ and ‘state transition models’ are not that amusing a perfect distribution and transition measure can result in better rhythms.

8. Future Work

8.1. Synthesis of Rhythmic pattern for beats with different amplitudes:

As of now, we have generated rhythmic patterns with a single amplitude or two amplitude (short beat and a long beat). Our future work includes consideration of multiple levels of amplitude and finds the relation between amplitude variations and quality of rhythm.

8.2. Quality assessment of Rhythm:

We have generated a family of rhythms which includes all the rhythms ranging from very good rhythms to downright dreadful rhythms. Our next step is to give subjective scores to each of the rhythms and find the correlation between pattern and its quality and design a model to find the quality of rhythms and to generate good rhythms automatically.
Most of the Rhythms explored in this phase are based on percussion instruments (drum etc). Extending the above work to generate rhythms from different instruments and genres (vocals, Instrumental solo) and their quality assessment.

8.3. Automated Rhythm Generation:

Using the above-mentioned models, and by finding the features that make rhythm good, we can synthesize rhythms that are pleasant to hear.

8.4. Rhythm Generation using probability distribution:

In this thesis, Gaussian distribution is used for rhythm generation whose variations produced some interesting rhythms. Better rhythms may be synthesized using different probability density function like Poisson or mixture of Gaussians.

8.5. Rhythm generation using morse word transition:

As explained in section 6.9 of methodology, An effective definition of state transition measure can result in good rhythmic sequences. The measure can be based on quantities defined in research by [**Toussaint G**] to generate pleasing rhythms.

9. References

[Jason Martineau]

The Elements of Music: Melody, Rhythm and Harmony.

[Godfried]

Toussaint, Godfried T. "The Euclidean algorithm generates traditional musical rhythms." *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*. 2005.

[SNS]

Bjorklund, Eric. "The theory of rep-rate pattern generation in the SNS timing system." *SNS ASD Tech Note, SNS-NOTE-CNTRL-99* (2003).

[Toussaint]

Toussaint, Godfried. "The geometry of musical rhythm." *Japanese Conference on Discrete and Computational Geometry*. Springer, Berlin, Heidelberg, 2004.

[Toussaint G]

Toussaint, Godfried. Generating “Good” Musical Rhythms Algorithmically” .8th International Conference on Arts and Humanities,Honolulu, Hawaii, January 13-16, 2010, pp. 774-791.

[John Varney]

https://ed.ted.com/lessons/a-different-way-to-visualize-rhythm-john-varney