ALGORITHM FOR THE GENERATION OF ALAMKAARA IN CARNATIC CLASSICAL

[1]Mahesha Padyana, [2]Bindu A Thomas
[1]Research Scholar, [2]Professor and Head
Department of Electronics and Communication
Vidya Vikas Institute of Engineering and Technology
Mysuru - 570028, India

***Abstract*—Carnatic classical music is specialized for improvisations and creativity. Kalpana swara is one kind of improvisational music which requires many years of practice and cannot be learned easily. Synthesizing kalpana swara is an important area that requires some attention and gaining momentum in recent years. Alamkaara or decoration of musical notes is one of the core parts of carnatic classical music that is widely used during improvisations. Alamkaara is taught during basic lessons, but complex forms of alamkaara has to be institutionalized own their own. This paper presents an algorithm used for generating various types of alamkaara in carnatic classical, which are ultimately used in synthesizing kalpana swara.**

***Index Terms*—Carnatic classical music, creative music, kalpana swara, musical pattern generation, alamkaara, alamkaara, music synthesis**

I. INTRODUCTION

Carnatic classical music is one kind of music having its own specialty and found commonly in southern parts of India [1]. "Creative Music" (also known as Manodharma sangeetha or improvisational music) is the specialty of Carnatic classical. Creative music comes out spontaneously without memorizing, but within the boundary of raaga and also adhering to the rules of improvisational music. Kalpana swara (or swara kalpana) is one of the complex forms of creative music. Synthesizing kalpana swara using software is not an easy task and this paper basically lays foundation for the kalpana swara synthesis.

The basic elements of Carnatic music are "Swara", "Raaga" and "Thaala". There are 12 swaras positions (swara sthaana) in Carnatic music and represented using seven swaras(equivalent to sol-fa syllable) S, R, G, M, P, D, N and some variants of R, G, M, D and N. Thus 12 swaras are notated as S1, R1, R2, G2, G3, M1, M2, P1, D1, D2, N2 and N3. Though S and P do not have any variants, suffix 1 is used for the sake of consistency. These swaras are associated with specific frequency and swaras are arranged in ascending order of frequency like S1 R1 R2 G2 G3 M1 P1 D1 D2 N2, N3 in an aarohana pool and also arranged in descending order of frequency like N3 N2 D2 D1 P1 M2 M1 G3 G2 R2 R1 S1 in an avarohana pool. After N3 in aarohana pool, swaras repeat again, but in higher octave notated using ' (quote) symbol such as S1' R1' R2' G2' and so on'. Similarly swaras below S repeat in lower octave with dotted notation S1. R1. And so on. So both aarohana pool and avarohana pool consists of twelve swaras in each octave and there are typically 3 octaves, but sometimes it could be even 5 octaves in case of instruments.

Raaga is defined using 5 or more swaras [5] p.1 and it is characterized by embellishments or intonations [1][2]. Structurally, raaga has two components termed as aarohana and avarohana. Aarohana means the pool of swaras in ascending order and hence comes from aarohana pool. Avarohana means pool of swaras in descending sequence and hence comes from avarohana pool. Swaras in aarohana and avarohana need not be same and also their order need not be same. Though certain order is maintained for defining raagastructurally using aarohana or avarohana, it need not be maintained while weaving the swaras during musical rendering. But if the flow is reversed, swaras from respective pools have to be considered.

The most important specialty of Carnatic classical is that raaga is not just framed by combination of swaras. Raaga is framed by certain phrases (group of swaras arranged in a specific way) that identify the raaga uniquely [2].These phrases are referred as raaga characteristic phrases and played frequently to show the characteristics of a raaga. Various characteristic phrases are grouped together to form sanchaara for the given raaga. There can be many sanchaara patterns associated with given raaga.

Thaala (or tala) is nothing but the rhythmic aspect of the Carnatic classical and it is constructed from various fundamental parts named anga. Thaala is basically a meter for measuring musical time. In their paper titled "Modular Representation of Thaala in Indian Classical System" [4], authors explained the structure of thaala and its representation in modern day.

While weaving swaras for rendering compositions in a given raaga, alamkaara (or alamkaara) also contributes a lot in beautifying Carnatic classical. In Sanskrit, alamkaara means "decorate", that means beautiful grouping of swaras that adorn the classical music [6] p.140. In Carnatic classical, alamkaara is the embellishment techniques used for generating pleasant patterns of swaras. Unlike sanchaara, same alamkaara may appear in other ragas having same set of swaras [7]. However sanchaaras cannot be omitted altogether. They are inserted frequently to retain the raaga flavor.

There are various types of alamkaaras in use that add beauty to the music. We have fourbasic types of alamkaarasnamed as aarohi (group of swaras arranged in ascending order frequency), avarohi (group of swaras arranged in descending order frequency), sthaayi (end swara of the group of swaras is same as that of start swara) and sanchaari (swara arrangement in any order) [6] p.140. Yathi is another type of alamkaara wherein various smaller patterns called inner patterns are generated to form beautiful combination that gives particular shape to music.

There are 6 types in yathi type of alamkaara.

- Sama Yathi: In this case the length of each inner pattern (pattern group) remains same. Example: MGR|PMG|DPM|NDP
- Srothovaaha Yathi: In Srothovaaha yathi, length of each generated inner pattern increases gradually. Example: S|SR| SRG|SRGM| or S'|S'N|S'ND|S'NDP
- Gopuccha Yathi: In this case, length of each inner pattern starts decreasing in subsequent cases. examples of gopuccha yathi are: SRGMP|SRGM|SRG|SR

- Mridanga Yathi: In case of mridanga yathi, inner pattern length initially increases and then decreases in subsequent patterns. It is the combination of Srothovaaha and Gopuccha yathi type of alamkaaras. Example: SR|SRG|SRGM|SRGMP|SRGM|SRG|SR
- Damaru Yathi: In this type of yathi, inner pattern length initially decreases and then increases. It is the combination of Gopuccha and Srothovaaha yathi type of alamkaaras. Example: SRGMP|RGMP|GMP|MP|GMP|RGMP|SRGMP
- Vishama Yathi: All other patterns which are not covered by other 5 types of yathi fall in this category. In case of vishama yathi, there is no definite relation to previous inner patterns of yathi. For example: SRS|SRGMP|DPMGMGR|SRGR|SMGR|SN.D.N.SR

Kalpana swara is the method of weaving swaras in various fashions around the chosen line of lyric called kalpana swara lyric [6] p.66. In kalpana swara, musician presents the chosen line of lyric first and then weaves set of swaras involving sanchaara and alamkaaras and again presents the kalpana swara lyric again. Sambamoorthy outlined several rules for the kalpana swara [6] p.67 and these stringent rules make the kalpana swara rendition little complex. Learners find kalpana swara very difficult and kalpana swara synthesis thus can generate a list of patterns in any given raaga and thereby simplifying the task of framing common patterns during learning process.

This paper basically briefs about the algorithm used for generating various types of alamkaara used for synthesizing kalpana swara.

We use certain terminology in this paper. This paper considers only 3 octaves, consisting of aarohana pool swaras S R G M P D and N and their variants in base octave, S' R' G' M' P' D' and N' and their variants in higher octave, S. R. G. M. P. D. and N. and their variants in lower octave. In case of avarohana pool, the order of swaras is reversed. The terms before and after is used with respect to the order of swaras in either aarohana or avarohana pool. In case of aarohi, swara 'before' means the adjacent swara has frequency lower than the current swara and 'after' means the adjacent swara having frequency higher than the current swara. In case of avarohi, swara 'before' means the adjacent swara having frequency higher than the current swara and 'after' means adjacent swara has frequency lower than current swara.

Unless and otherwise mentioned specifically, this paper assumes parent raaga (raaga consisting of all seven swaras) for quoting examples, even though alamkaara can be generated for any derived raaga (consisting of 5 or more swaras). Swara variants may or may not be specified in examples and any variant as allowed by the raaga can be used. Pipe (|) symbol is used to separate patterns into various groups. Unit time is the absolute duration of the swara in the selected speed of rendering. Stretch is indicated by the symbol comma (,) and silence is indicated using hyphen (-) and both of them are of unit time. For example; in the pattern G,M-P,,M the swara G is stretched for one unit duration and P is stretched for two unit time and silence of one unit time is introduced after first occurrence of M. Working range indicates the highest and lowest swaras considered in a given raaga from three octaves and comes from raaga database consisting of raaga and their characteristics [1]. Langhana swaras are the set of swaras from where starting of new pattern is not permitted. Also ending on langhana swara or stretching langhana swara is not permitted. Similarly restricted end swara is the swara where ending is not permitted and restricted start swara is the swara from where starting new pattern is not permitted. Daatu means skipping one swara in a given swara sequence and jumping directly to the subsequent swara. For example, if swaras in aarohana pool of raagaare SRGMPDN, jumping from G to P makes daatu. Similarly samvaadi swara means consonant swara. Graha swara is the swara where kalpana swara generation can start. Moorchana is the word used to indicate either aarohana (or even aarohi) or avarohana (or even avarohi) type of patterns.

IMPLEMENTATION

Generation of alamkaara is possible only once the fundamental swaras are generated. Generation of swara, raaga and basic patterns are discussed in detail in the papers published by Mahesha Padyana and Bindu A Thomas [1][3] and this paper does not discuss this topic. Swaras from aarohana and avarohana pools are represented graphically as nodes and alamkaara generation utility ultimately makes use of this graphical structure to know the movement of swaras in aarohana or avarohana pool and composition of each swara.

A. Aarohi and Avarohi alamkaara

The first step in generation of alamkaara is to generate aarohi and avarohi type of alamkaara. These patterns are ultimately used for constructing other alamkaara patterns. These simple aarohi and avarohi type of alamkaaras are categorized as basic patterns. The algorithm specified in the paper on basic pattern generation [3] is utilized and revised/rewordedhere to generate basic patterns. The flow outline for generating basic patterns is shown in Figure1.

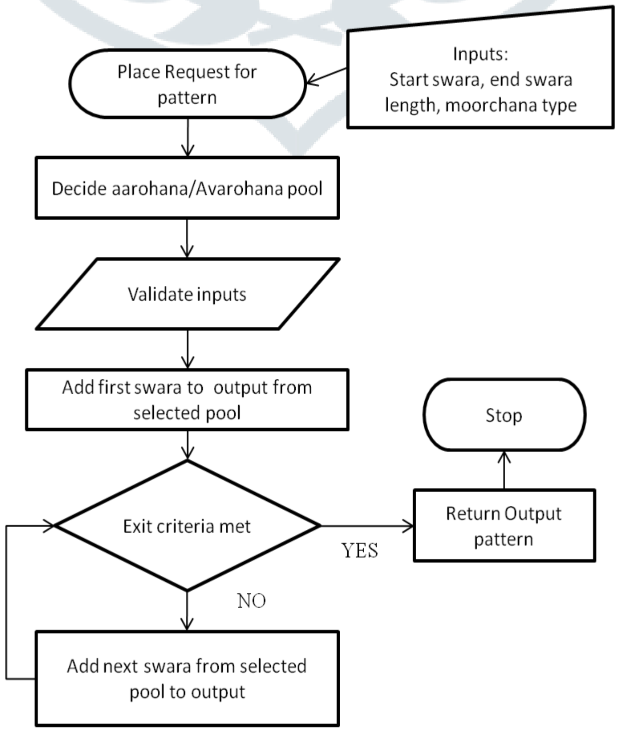

Fig. 1. Generation of Aarohi and Avarohi Varnas

Any service requiring aarohi or avarohi patterns must place the request for the generation of aarohi/avarohi with following input parameters.

- Moorchana: Aarohi, avarohi or random
- Start Swara: First swara of the pattern to be generated
- End swara: Valid end swara or empty value
- Length: Total length of the pattern to be generated in terms of swara count

Depending on moorchana (indicates wither aarohana or avarohana), either aarohana pool or avarohana pool is used to generate aarohi/avarohi patterns. In case random moorchana type is requested, either of the types is considered by the algorithm randomly. Algorithm will first validate the input parameters by looking at the raaga graph [1][3] and see if they belong to raaga and also parameters are within limits. Algorithm will also check to see if either length or end swara is specified as exit condition (the condition used to terminate the pattern generation) and in case both are given, algorithm will consider either of them, whichever occurs first. Once inputs are specified and validated, algorithm will go through the raaga graph [1][3] and find out the next swara to be generated and add the swaras until required exit criteria is met.

B. *Connected Patterns*

Connected pattern is a powerful algorithm used to generate various types of alamkaaras. In case of connected patterns, several types of smaller patterns especially aarohi or avarohi patterns are connected one after the other to form one long pattern. We call these smaller patterns as inner patterns and final resulting pattern as connected output pattern. These inner patterns are generated based on several parameters that are supplied to the connected pattern generation algorithm. These parameters are listed and explained below and any service requesting this connected pattern service must supply these input parameters. Short name inside the bracket is used in flowchart.

- Start Swara: First swara of the output pattern generated by this connected pattern service. It is also the start swara of the first inner pattern.
- Reference Length (Ref_Len): This is the length of the first inner pattern or reference pattern. This pattern is called as reference pattern because this is taken as reference for generating subsequent inner patterns.
- End Swara: Last swara of the last inner pattern (or last swara of the overall output pattern) where connected output pattern must end.
- Number of patterns (numPatterns): Number of inner patterns to be generated and connected one after the other.
- Swara count: Total number of swaras to be generated in the output pattern.
- Moorchana :Decides whether subsequent inner patterns should be of type aarohi or avarohi. Random means it can be either aarohi or avarohi and decided randomly by the connected pattern algorithm.
- Connection Point Reference (Conn_Ref) indicates one of the two swaras to be taken as reference swara from the previous inner pattern for deciding the start swara of the subsequent inner pattern. Connection point reference swara could be the start swara or end swara of the previous inner pattern. Random means it can be either start swara or end swara of the previous inner pattern and decided randomly by the connected pattern algorithm.
- Connection Point (Conn_Point) indicates from which swara next inner pattern should start with respect to connection point reference swara of previous inner pattern. Subsequent inner pattern's first swara could be connection point reference swara itself (Conn_Point=same), one swara before the reference swara (Conn_Point=Prev), one swara after the reference swara (Conn_Point=next), one of the daatu swaras of the reference swara (Conn_Point=daatu) or one of the samvaadi swaras of the reference swara (Conn_Point=samvaadi).
- Progress type: Indicates the progress of next inner pattern such as gradual increment, gradual decrement, fixed or random. In case the progress type is gradual increment, subsequent inner pattern's length will increase by one, in case the progress type is gradual decrement, subsequent pattern's length will be reduced by one, in case it is fixed, no change will be there in the length and if the progress type is random, the length of subsequent parameter could be selected randomly between 2 and 7 swaras with typical value being 3, 4 or 5.

Various patterns can be generated randomly by using connected pattern services with various combinations of inputs.

High level implementation outline of connected patterns is shown in Figure2. R(x,y) in this figure indicates the random value between x and y. 'Inputs' in the figure indicates the list of input parameters we discussed at the beginning of this section.

Algorithm will generate the first inner patter based on start swara, moorchana and reference length parameters using aarohi/avarohi alamkaara generation algorithm. Algorithm will call aarohi or avarohi depending on the moorchana parameter. Once the first pattern is generated, next pattern can be generated and connected to the previous pattern based on other parameters.

1) Input Parameter Validation

Certain validation is done on input parameters so that pattern generation algorithm executes smoothly without going into endless loop. In certain cases, pattern generation will continue by assuming default values. Main validation is done to ensure that start swara and end swara are valid, reference length is not empty and within limits, connection point reference and progress type are not missing.

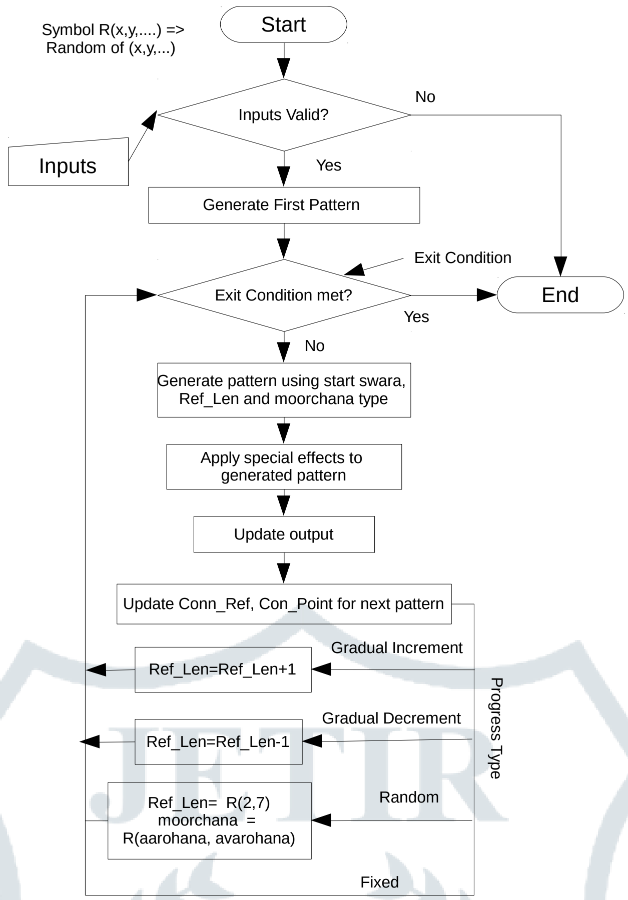

Fig. 2. Connected Patterns

Fig. 3.

In addition to the above validation, certain other validation is also performed on certain combination of parameters. If the progress type = FIXED, Conn_Ref=START, Conn_Point=SAME, end swara of generated inner pattern will remain same and there is no progress in either direction. In such scenario, end swara cannot be the exit condition, otherwise endless loop is entered. Similarly in case end swara is given as exit condition and if progress type=FIXED, connection point reference=START, Conn_Point=NEXT, there is a possibility of end swara happens to be in opposite direction compared to the progress of first pattern. In such case exit condition may be met only because of working range limits. In case of gradual decrease, number of patterns cannot be more than reference length. For Ex: Reference length=3 and number of patterns =4. Some of the generated inner patterns could be SRG|SR|S and in such case pattern count 4 cannot be achieved. So in case of gradual decrease, pattern must always end when number of patterns is achieved or swara count of one is reached (whichever occurs first).

When swara count is given for generating lengthy patterns, parameters such as moorchana, progress type, Conn_Ref and Conn_Point must be random. In case the progress type is other than random (that means gradual increment, gradual decrement or fixed) and if valid end swara is given as exit condition, there is a possibility that end swara encountered at the middle of the pattern and there is a possibility of not generating even one pattern and hence this condition will be avoided. In some cases, if end swara is given and progress type is either gradual decrease or gradual increase, there is a possibility of end swara being outside the limit and hence if progress type is other than random, end swara cannot be the exit criteria.

Taking end swara as Conn_Ref and 'next' as Conn_Point will give rise to either aarohana or avarohana type and this may not be desired always though it is valid. Calling service must take care of this.

If restricted start swara happens to be the start swara of the entire pattern, pattern generation will not start.

2) Exit Condition

Connected pattern is capable of generating many types of patterns connected one to another to form lengthy patterns. There must be an exit condition to stop the pattern generation. Whenever exit condition is met, pattern generation process will be stopped and whatever patterns already generated will be returned.

Following are the exit conditions depending on the input parameters given to the connected pattern algorithm:

- Generated pattern reached end swara and total number of swara generated is equal to swara count in case end swara and swara count together forms the exit condition.
- Total number of swaras generated equals to swara count if swara count is the exit condition.
- Total number of patterns generated equals to number of patterns if number of patterns is the exit condition.

In addition to this, pattern generation may be stopped automatically in certain other cases such as:

- Any swara goes beyond the working range.
- Langhana happens to be end swara of any pattern
- Other than the alamkaara, if restricted end swara happens to be the end swara. However even in alamkaara, restricted swara cannot be the last swara of the final output. It can be the end swara of an inner pattern.

3) Runtime parameter correction

During runtime, algorithm may encounter certain invalid conditions and in such situations, algorithm may change input parameters dynamically. This dynamic change in parameter is possible only in case the progress type is random. This will ensure that the pattern generation is not stopped abruptly.

C. *Generation of Alamkaara from connected patterns*

Alamkaara generation makes use of connected pattern, aarohi and avarohi services with appropriate input parameters. Not all combinations are possible. Movements that results in only aarohana or avarohana combinations are avoided. That means combination of connection point reference=end swara and Conn_Point=next, moorchana = aaroha OR connection point reference=end swara, Conn_Point=next and moorchana=avarohi are avoided. Start swara can be any valid swara taken with respect to the end swara of previous pattern. Or it could be even one of the graha swaras in case previous pattern ends with a long stretch or silence.

This section shows how various types of alamkaaraare generated using connected patterns algorithm.

1) Sama Yathi

In sama yathi, total number of swaras in output pattern can be calculated using the formula (1).

$swaraCount=Ref\_Len*numPatterns$ (1)

- Num Patterns : randomly selected between 3 and 5
- Ref_Len: Selected randomly between 2 and 5 with 3 and 5 being most common.
- Progress type: Fixed
- Moorchana: Aarohi, avarohi or random
- Conn_Ref: Start, end or random.
- Conn_Point: Same, next, prev, daatu, samvaadi or random

2) Srothovaaha Yathi

In case Srothovaaha yathi starts with an input pattern of reference length (r) and generates n patterns, the formula for calculating total swara count in the output pattern would be derived using the formula (2).

$Swara\ Count = (n^2-n+2nr)/2$ (2)

- Number of patterns : randomly selected, but typically between 3 and 5
- Start swara: Any valid swara taken with respect to the end swara of just concluded pattern
- Ref_Len: Selected randomly, but typically selected between 1 and 3.
- Progress type: Gradual Increment
- moorchana: Aarohi, Avarohi or random
- Conn_Ref: Start, end or Random
- Conn_Point: Same, next, prev, daatu, samvaadi or random

3) Gopuccha Yathi

In case Gopuccha yathi starts with an input pattern of reference length (r) and generates n patterns, the formula for calculating total swara count in the output pattern would be derived using the formula (3).

$Swara\ Count = (n-n^2+2nr)/2$ (3)

- Number of patterns : randomly selected, but typically between 3 and 5
- Start swara: Any valid swara taken with respect to the end swara of just concluded pattern
- Ref_Len: Selected randomly, but typically selected between 3 and 5.
- Progress type: Gradual Decrement
- moorchana: Aarohi, Avarohi or random
- Conn_Ref: Start, end or Random
- Conn_Point: Same, next, prev, daatu, samvaadi or random

4) Mridanga Yathi:

In case of mridanga yathi, rules specified for gopuccha and Srothovaaha yathi are applicable. But Srothovaaha starts with Ref_Len=1 and gopuccha starts with Ref_Len=(length of last Srothovaaha pattern-1).

5) Damaru yathi

In this type of yathi, rules specified for gopuccha and Srothovaaha yathi are applicable. But Gopuccha must end on one swara and then Srothovaaha yathi starts with two swaras and retains the same number of patterns of gopuccha yathi.

6) Vishama Yathi:

There is no specific formula for vishama yathi since it does not follow any rule.

- Number of patterns: randomly selected between 2 and 5 with 3 and 5 being most common.
- Start swara: Any valid swara taken with respect to the end swara of just concluded pattern. It could be the end swara, one swara before the end swara or one swara after the end swara.
- Ref_Len: Selected randomly between 3 and 5.
- Progress type: Random
- Moorchana type: Random
- Conn_Ref: Start, end or Random with the condition that if start swara is taken as connection point reference, anti divergence rule is taken into account.
- Conn_Point: Same, next, prev, daatu, samvaadi or random

II. Results

Results of few types of alamkaaras generated by this algorithm are given below. Though S and P does not have any variants, S1 and P1 is used for the purpose of consistency. Tests were performed with several raagas, however results with only one raaga having swaras S1, R1, G3, M1, P1, D1, N3 in both aarohana and avarohana are given below.

7) Sama yathi

Inputs: Start Swara=G3 Ref_Len=3 Pattern Count=4 Conn_Ref=end Conn_Point=random Progress Type=fixed

Output: G3 R1 S1| M1 G3 R1| S1 R1 G3| R1 S1 N3.

8) Srothovaaha yathi

Inputs: Start Swara=P1, Ref_Len=3, Pattern Count=4, Conn_Ref=start,Conn_Point=previous, Progress Type=gradualIncrement

Output: P1 D1 N3| M1 P1 D1 N3| G3 M1 P1 D1 N3| R1 G3 M1 P1 D1 N3

9) Gopuccha yathi

Inputs: Start swara=G3, Ref_Len=4, Pattern Count=3, Conn_Ref=random, Conn_Point=random,Progress Type=gradualDecrement

Output: G3 M1 P1 D1| M1 P1 D1| M1 P1 (Generated Swara count = 9)

10) Mridanga yathi

Inputs: Start swara=R1, Pattern Count=4, Conn_Ref=random, Conn_Point=random, Progress Type: gradualIncrement

Output: R1| G3 M1| M1 P1 D1| P1 D1 N3 S1'| P1 D1 N3| M1 P1| G3

III. Conclussion

Alamkaara generation based on the algorithms specified has given very good results. The algorithm was developed using java programming language with the help of JFugue music library. These patterns could be utilized further in synthesizing kalpana swara or even for generating pre-composed music. Small graphical user interface could be designed to select various types and their parameters to generate required alamkaara so that learners could benefit from this.

REFERENCES

[1] M. Padyana and B. A. Thomas, "Modeling Carnatic classical raaga using packetization technique," 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2015, pp. 612-617

[2] M.Subramanian, “Synthesizing Carnatic Music with a Computer”, Journal of Sangeet Natak Akademi, , No 133,134, pp 16-24, 1999

[3] Padyana, Mahesha, and Bindu A. Thomas. "Basic Pattern Generation for Kalpana Swara Synthesis Using Graphical Approach.", International Journal of Computer Science Trends and Technology, Volume 4 Issue 3, May - Jun 2016, ISSN: 2347-8578

[4] Mahesha Padyana and Bindu A Thomas "Modular Representation of Thaala in Indian Classical System". International Journal of Computer Science Trends and Technology (IJCST) V5 (1): Page (1-6) Jan-Feb 2017. ISSN: 2347-8578

[5] Prof P Sambamoorthy “South Indian Music” Book III, The Indian music publishing house

[6] Prof P Sambamoorthy “South Indian Music” Book IV, The Indian music publishing house

[7] P Malathi "Alamkaaras in Alapana" , Dissertation of master of philosophy, University of Madras, 1992