CHAPTER 4

SEGMENTATION AND FEATURE EXTRACTION

According to the overall architecture of the system discussed in Chapter 3, we need to carry out pre-processing, segmentation and feature extraction. This chapter discusses the contributions that have been made in the Segmentation and Feature extraction stages of Carnatic Music processing. A good set of features is essential for extracting meaningful information available in a given music signal. For this purpose, a good segmentation of the signal is essential. In the context of music signals, the characteristics of the signal play an important role in the segmentation and feature selection, resulting in an efficient identification of the content. In this thesis, contributions have been made in the segmentation and feature extraction modules by exploiting the characteristics of Carnatic music.

4.1 OVERVIEW OF SEGMENTATION

Research issues in audio content analysis can be categorized along four directions: audio segmentation and classification, content-based audio retrieval, audio analysis for video indexing, and integration of the audio and video (Zhang and Kuo 2001). In this thesis, the issues considered are segmentation, classification, indexing and retrieval of music. In this chapter, the focus is on the first issue of audio content analysis: segmentation. For performing audio segmentation, the signal needs to be pre-processed. As discussed in Chapter 2, in the context of music signal pre-processing, the main modules to be considered are noise removal and signal separation. Noise
removal algorithms that are available for speech can be applied to music signals. However, noise removal from music results in removing the information content of the music signal along with the noise. Instead, signal separation that isolates the voice and non-voice parts of the noisy music signal, is carried out in order to individually process these signals later. Hence, before discussing the algorithm proposed for segmentation, we shall discuss the modifications that have been performed in an existing algorithm for signal separation.

4.2 SIGNAL SEPARATION

Signal separation can be defined as the process of separating the vocal and non-vocal sub-signals from a given music signal. In this work, a comparative study of two existing signals separation algorithms (Zhang and Zhang 2005, Every and Szymanski 2004), originally proposed for Western music but now applied to Carnatic music, is performed. We have found that the one proposed by Zhang and Zhang (2005) is better suited for Carnatic music.

4.2.1 Spectral Filtering Approach

As discussed in Chapter 2, in the approach to signal separation proposed by Every and Szymanski (2004) the separation is performed using a bank of filters. The spectral filtering approach is based on examining the spectral characteristics and designing a filter for the same.

This method to separate the voice from the non-voice signal was explored for Carnatic music. The drawback of this approach is its use of MIDI for the representation of music. Since Carnatic music is rich in harmonics and Gamakas, the conversion to MIDI results in a loss of content of the signal.
Hence, we tried an alternative approach for signal separation which is discussed in the next section.

### 4.2.2 Harmonic Structure Modeling Approach

As discussed in Chapter 2 the algorithm proposed by Zhang and Zhang (2005) is based on Harmonic Structure modelling, where the harmonic of the signal is considered to be more stable compared to its monophonic representation.

In the first step of the three step Harmonic Structure model algorithm, the input signal is converted into frames of fixed duration. Then in each frame, all the spectral peaks exceeding a certain threshold are determined. Let the frequencies of these peaks be \( f_1, f_2, f_3, \ldots, f_k \) where ‘\( k \)’ is the number of peaks in each frame. Then for a fundamental frequency \( f \), the number of \( f_i \) that satisfy the following condition is calculated, using the equation given by:

\[
\text{floor } \left[ \frac{(1+d) f_i}{f} \right] \geq (1-d) \frac{f_i}{f}.
\] (4.1)

where \( \text{floor}[x] \) denotes the greatest integer less than or equal to ‘\( x \)’, and ‘\( d \)’ is an arbitrary integer constant satisfying equation 4.1. In this algorithm, all the frequency components including the harmonic frequencies are extracted to calculate the harmonic structure coefficient \( B \).

The harmonic structure coefficient \( B^l \) is given by the following equation:

\[
B^l = [B_1^l, \ldots, B_R^l], \quad B_i^l = \log \left( \frac{\rho^l A_i^l}{\rho^l A_1^l} \right) / \log \left( \frac{\rho^l A_i^l}{\rho^l A_1^l} \right). \quad i = 1, 2, 3, \ldots, R
\] (4.2)

where \( l = 1, 2, \ldots, L \) is the frame index, \( A_i^l \) describes the amplitude at the \( l^{th} \) frame and \( \rho^l = C / A_1^l \) is a multiplying factor and \( C \) is any arbitrary constant.
In the second stage of the algorithm, a data set of harmonic structures is estimated from the harmonic structure coefficients. In the data set, all the music harmonic structures are clustered into a set of high-density structures, where each cluster would correspond to one Instrument. Voice harmonic structures are scattered around like background noise, since the harmonic structure of the voice signal is not stable (Pinquier J et al, 2002). Therefore, the calculation of the harmonic structure is essential, which is done in the third stage of the algorithm.

In the third stage of the separation algorithm, the NK clustering algorithm (Zhang et al 2003) is used to determine the music Average Harmonic Structures (AHS). The AHS’s are obtained by calculating the mean of each cluster, using the equation:

\[
\text{Average Harmonic structure (AHS)} = \text{Average of } B = \left( \frac{1}{L} \right) \sum_{i=1}^{L} B^i. \tag{4.3}
\]

In the separation stage, all Harmonic Structures of an Instrument in all the frames are extracted, to reconstruct the corresponding music signals, and then removed from the mixture. After removing all the music signals the rest of the mixture is the separated voice signal.

4.2.3 Signal Separation Algorithms and Carnatic Music

The algorithms proposed by Zhang and Zhang (2005) and Every and Szymanski (2004) were applied to Carnatic music. On comparing the performances of the algorithms the following observations were made.

a. Carnatic music uses a just-tempered scale with 22 to 24 music intervals for an octave and hence, the frequency range between two swaras is very narrow, when compared to the successive notes in Western music that uses 12 intervals per octave. In addition, the special Gamaka characteristic of this music, the
conversion into the MIDI representation is not well suited for Carnatic music, as it results in loss of data. As already discussed, the spectral filtering approach is based on the conversion to MIDI, and therefore, is not considered for Carnatic music processing.

b. On the other hand, for Western music signals, the algorithm proposed by Zhang and Zhang (2005) provides a significantly high SNR, and has separated the music signals into voice and Instrument components. Since Carnatic music is rich in harmonics, this algorithm was considered for Carnatic music. In addition the accompanying instruments are a typical scenario of Carnatic music and hence this algorithm is adopted.

4.2.4 Modification to the Harmonic structure modelling algorithm to suit Carnatic music

As already discussed, the Harmonic structure modelling algorithm is suitable for harmonic-rich Carnatic music. The absence of conversion to the intermediate MIDI representation was also an advantage of this algorithm. One important part of the algorithm was the determination of the spectral peaks \( f_i \) using the constant value \( 'd' \). In the initial algorithm proposed for Western music, this integer constant was arbitrarily chosen to satisfy equation (4.1). However, since Carnatic music is heavily affected by the Gamakas, the value of \( 'd' \) as specified in Equation (4.1) was chosen using an adaptive algorithm, where the value of \( 'd' \) changes on subsequent iterations, to re-calculate and separate the voice and non-voice signals. We iteratively modified the constant integer \( 'd' \) to a real value between 0 and 1 in steps of 0.1, to compute the spectral peaks until the voice and non-voice components are separated. The computed spectral peaks are later used for computing the Harmonic structure coefficient. In addition to identifying just one pitch for
each frame, this algorithm also identifies all the values of the pitches for which the corresponding ‘$d$’ is below a threshold.

At the end of the signal separation phase, the input signal is separated into two signals, one non-voice signal consisting of the Instrument, and the other voice signal, both of which are used for subsequent processing. In addition to this, the input signal as a whole consisting of both voice and Instrument is also considered for identifying the Emotion content of the signal. After separating the signal, the signal needs to be segmented for further processing.

4.3 SEGMENTATION

In order to extract meaningful information for audio signal processing applications, there is a need to segment the signal. In this thesis, contributions have been made in the Segmentation stage of music signal processing by exploiting Carnatic music characteristics, to determine the swara components.

4.3.1 Classification of the Segmentation Algorithms

As discussed in Chapter 2, audio segmentation algorithms are divided into two categories: Model based algorithms and Novelty based algorithms.

Herrera et al (2000) proposed various strategies for the analysis of music content, which examined different model-based methods based on supervised learning, like the SVM, Neural network, and Bayesian Classifiers. These systems for music segmentation were based on identifying the musical Instruments. Gao et al (2003) used Hidden Markov Models to segment musical signals into a continuous sequence, based on the presence or absence of notes.
As discussed in Chapter 2, the typical music characteristics that have been used for segmentation are the pitch, beat, loudness, and rhythm which are derived from signal features, like the periodicity pitch, spectral flux, spectral centroid, Short-term energy etc. This indicates that both temporal and spectral features can be used for segmentation, either by using them directly or mapping them to musical characteristics.

In this thesis, the focus is on the segmentation of traditional South Indian Classical Carnatic music and Tamil movie songs, based on the regularity defining characteristic of Carnatic music – the Tala, which is a periodic pattern that is associated with a given musical piece. We use a novelty-based algorithm, which utilizes this regularity defining unique feature of Carnatic music for segmentation.

4.3.2 Segmentation Algorithms for Carnatic Music

In our work, the aim of segmentation is the determination of the points of swara beginning or end, and therefore we initially considered a simple segmentation algorithm, which traces the pitch envelope of the signal to yield the rise and fall of the pitch as points of segmentation. However pitch contour did not yield accurate results, due to a typical characteristic of Carnatic music called the Gamaka, which refers to pitch inflexions as already discussed in Chapter 1. Tracing the pitch contour and identifying the rise and fall of pitch as points of segmentation, typically lead to over-segmentation.

The work of Jian et al (2003), which was based on human perceptual properties, motivated us to develop a new approach for segmentation, which is based on Carnatic music characteristics. In Carnatic music a musician corrects a mistake committed in singing or playing an Instrument, with reference to the beginning and end of the Tala. This motivated us to use this concept for segmentation since each Tala could
correspond to a swara, and essentially our aim in performing segmentation is to identify the swara or the notes from a given music signal. Therefore, after studying the two algorithms for segmentation, it was concluded that the Tala based algorithm yielded segments associated with swara components, which are required for subsequent processing, to identify the content of Carnatic music. Both algorithms required the identification of the Onset and Offset, which is discussed in the following section.

4.3.2.1 Onset and Offset detection

Onset refers to the point at which the information content of a music signal commences. Typically, onset can be classified into hard and soft onsets (Zhou and Reiss 2007) (Pradeep et al 2007). Hard onset refers to the sudden change in energy, while soft onset refers to the gradual change in energy. Onset can also refer to the beginning of a note, and offset the end of a note (Duxbury et al 2003); hence, the duration between the onset and offset can be considered for segmentation.

As indicated in Chapter 2, a Carnatic song is associated with a Raga and a Tala. Also, as indicated in Chapter 1, a Tala is identified by a Thattu, Veechu, and Count, which is indicated using the Laghu, Anudhrutham, and Dhrutham. Each of these Thattu, Veechu or Count can accommodate 1, 2, 4 or 8 swaras. Each song is associated with one of the 175 Talas.

The Tala’s first count starts at the beginning of the song and ends with the song. In fact, the accompanying Instruments in a concert play till the completion of a Tala. Therefore, it is mandatory that any song is an integral multiple of a pre-specified Tala. This fact is the regularity that could be used to segment the song. In Carnatic music, a song is divided into three sections, namely, the Pallavi, the AnuPallavi and the Charanam. Typically, the music characteristics of the song are conveyed in the Pallavi. Therefore, the Pallavi
of the song can be used as a logical component sufficient for segmentation. This requires us to identify the beginning and ending of a Pallavi, which would correspond to an integral multiple of the Tala. The beginning and ending of a song is specified as the Onset and Offset values of a given musical piece. Onset refers to the beginning of a musical note, in which the amplitude rises from zero to an initial peak. The process of detecting the Onset can be performed in the time domain, frequency domain, phase domain, or complex domain. The onset can be detected by looking for the following changes:

- Increase in the spectral energy – a sudden rise in the amplitude corresponding to a given frequency which normally happens at the beginning of a song

- Changes in spectral energy distribution – Accompanying Instruments add up to the musical piece, which result in a change in the spectral flux or phase

- Changes in the detected pitch – Abrupt changes in frequency

- Spectral patterns

This was used as the basis for identifying onset using the characteristics of time-based energy and phase (Duxbury et al 2003), peak changes in the spectral energy (Gainza 2004), combining the pitch and energy (Zhou and Reiss 2007).

Simple techniques, such as identifying the increase in the time-domain amplitude to determine the onset can typically lead to an unsatisfactorily high amount of false positives or false negatives (Bello et al 2005). The technique that is being adopted in this work on Carnatic music is based on identifying the change in the spectral energy (Gainza 2004, Zhou
and Reiss 2007). This is due to the fact that in Carnatic music a sudden rise in the spectral energy indicates the beginning of a Pallavi, where the energy of the Singer or Instrumentalist along with their accompanying Instruments reaches a maximum, and at the completion of the Pallavi drops to a minimum.

In a typical Carnatic music concert the beginning of a song could be initiated by different Instruments or an “Aalapana”. Due to this, the determination of a change in pitch or spectral distribution could be difficult for Onset detection, as it could correspond to the different Instruments that are played before the beginning of a song. Therefore, for detecting the Onset, the input voice-only signal which could either belong to the “aalapana” or the voice separated signal from the input is considered. This signal is converted into the frequency domain, and the change in the spectral energy is observed in a frame of 2 msec using an overlap of 1 msec. The value of 2 msec is chosen as this duration is closer to one Tala count. After observing for successive frames of 2 msec, the point at which the spectral energy starts stabilizing to a value greater than the previous frame by 80%, is identified as the Onset. The threshold is determined by observing the typical value from a training set. The Offset is detected in a similar manner but here the decrease in the spectral energy is observed and the point at which the spectral energy starts drastically dying is identified as the Offset. In case, the offset is not identified within the 2 minute duration, the duration from the onset till the end of the 2 minutes is considered for segmentation.

4.3.2.2 Segmentation based on Pitch Contour

The first algorithm that we proposed is based on using the pitch contour which serves as the feature for the novelty based segmentation algorithm. This involves the determination of Onset and Offset points which
are computed as discussed in the previous section. The segment extracted between the Onset and Offset is used for determining the Pitch contour. The variations in the pitch contour are used as points of segmentation. This algorithm did not yield good results due to the presence of the Gamakas and hence resulted in over segmentation. Hence we proposed another algorithm for segmentation, which is based on the Tala characteristic of Carnatic music.

### 4.3.2.3 Tala Based Segmentation algorithm

The novelty-based algorithm that is being proposed in this thesis involves the determination of Onset and Offset followed by two-level segmentation, and then a re-combination of the segments. The identification of Onset and Offset has already been discussed, while the remaining modules are discussed below. The overall flow of the segmentation algorithm is shown in Figure 4.1.

![Figure 4.1 Tala Based Segmentation Algorithm](image-url)
4.3.2.3.1 First Level Segmentation

As already discussed, since the main purpose of segmentation is to extract the swaras, the segmentation algorithm has been designed with this in mind. After determining the Onset and Offset, the signal component between these two points is considered for the first level of segmentation, which is based on separating it with respect to an integral multiple of Tala and segmenting each Tala’s duration into its individual components as discussed in Table 1.3. We initially considered a total of 350 Talas with either 1 or 2 strokes for each of the Tala components - Count, Dhrutham, and Anudhrutham. The results that we obtained with 1 or two strokes were similar, and hence, we reduced the database of Talas to 175 with only 1 stroke for each of the Tala components, which are maintained in a Tala Database, with the associated time duration of every component.

This database is sorted based on the decreasing frequency of the usage of the Tala. A mapping is performed to map the Tala’s time duration with that of the input music signal, for which a fitness function based on the Tala’s duration has been designed. This fitness function determines from the list of available Talas, the Tala that closely matches the segment of the input song. In case two Talas match closely, the longer pattern Tala is chosen, so as to derive a longer stream of swaras.

After obtaining the Tala associated with a segment, the segment is divided into the individual Tala segments, by observing the time duration of the associated Tala from the Tala database. Any Tala has a fixed predetermined component pattern. Each Tala component depending on whether it is a Laghu, Dhrutham, Anudhrutham, has a varying duration. After identifying the Tala, using its beat pattern, the signal is further segmented based on the Laghu, Dhrutham, and Anudhrutham. After this segmentation, each segment corresponds to one of the Tala components, such as the Laghu,
Dhrutham or Anudhrutham. The result of this stage of the segmentation is shown in Figure 4.2.

**Figure 4.2 Output of First level segmentation indicating segmentation points**

### 4.3.2.3.2 Second Level Segmentation

In Carnatic music, a Laghu, Dhrutham or Anudhrutham can accommodate one, two, or four swaras depending on the tempo of the song. However, as per our requirement, when each of the segments finally obtained needs to correspond to a swara, a second level of segmentation becomes necessary. For this purpose, the segment obtained after the first level of segmentation is split into four equal parts, initially assuming that the segment consists of four swaras. However, there is a possibility that depending on the tempo of the songs, each Tala component can have 8 swaras. To tackle this situation, linear warping by 100% in order to accommodate these 8 swaras in a Tala is carried out before the second level of segmentation. However, this can create a problem if the swaras in a Tala component is 4 or 2 since there is a possibility of over segmentation which needs to be tackled.
4.3.2.3.3 Reducing over segmentation

The segments that have been obtained from a single Laghu, Dhrutham or Anudhrutham could also belong to a fraction of a swara, when the individual Tala components had corresponded to one or two or four swaras. In order to remove this over segmentation, we use the autocorrelation which is a measure that tells us where the signal is most similar to itself (Foote et al 2001). Autocorrelation is given by:

\[
y[k] = \frac{1}{N} \sum_{n=1}^{N} x[n] \times x[n-k].
\]

where \(N\) is the number of samples in a given segment, \(x[n]\) is the amplitude at \(n\), \(k\) is the lag at which \(y[k]\) is the autocorrelation value at lag \(k\). If the autocorrelation measure between adjacent segments is above a threshold, then these segments are combined. The threshold of 70% is chosen by observing a training data set of 200 songs. In the case of ambiguity the individual segments are not combined which could result in additional number of segments. If the adjacent segments correspond to the same swaras and the auto-correlation function is not able to merge these two segments to one segment, then this could result in the adjacent segments being identified as the same swara that occur contiguously during swara identification.

For segmentation, we consider the beginning 2 minutes of a song which may comprise of only Pallavi or Aalaap and Pallavi. In case where the Pallavi is present the duration between onset and offset is used for Tala based segmentation. In case of the input containing the Aalaap and Pallavi, the determination of onset will separate the Pallavi, where Tala based segmentation is used as before. Table 4.1 gives the performance of our segmentation algorithm, when tested in a dataset containing 1200 songs from
Vani Compact disc, Kosmic music, Inreco, Amudham music, Laya music, Songs from Ilayaraja composition. The table gives the accuracy in determining one dominant frequency in each segment which may correspond to a swara.

**Table 4.1 Segmentation accuracy**

<table>
<thead>
<tr>
<th>Segmentation accuracy when the input contains aalap + pallavi</th>
<th>Segmentation accuracy when the input contains only the pallavi</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>88%</td>
</tr>
</tbody>
</table>

The difference in segmentation accuracy is due to the performance of the determination of onset and offset. It was easier to determine the onset when the input contains the aalaap followed by Pallavi. In some situations where we find more than one dominant frequency in each segment we record and associate these frequencies with the segment. Presence of Gamakas and incorrect Tala association are the reasons for the error in determining a dominant frequency in each segment.

**4.4 FEATURES FOR MUSIC PROCESSING**

The music segments obtained after segmentation are to be assigned labels, by extracting the characteristic features of the music signal (Brossier et al 2004, Jian et al 2003). Selecting and extracting a good set of feature vector helps to efficiently determine the characteristic content of a given music signal. As discussed in Chapter 1, features can be classified into Temporal, Spectral and Cepstral features, which are typically used for analysing and identifying the content of a given musical piece. These classes of features are used independently or as a combination for the process of musical content identification.
4.4.1 Role of Temporal, Spectral and Cepstral Features in Music Processing

Typically, a combination of features is used by researchers to analyse and determine the content of a music signal. In the work done by McKinney and Breebaart (2003), four sets of features were used for Music classification. McKinney and Breebaart (2003) also used spectral features such as the spectral tilt and spectral flux to classify music. Schubert et al, (2004) have used spectral centroid and timbre characteristics as features for analyzing the adjacency of two notes in Western music. The other spectral features that are normally considered for analysis are amplitude of sinusoids, amplitude of residual, spectral envelope, spectral shape of residual, vibrato, etc.

Temporal features and Cepstral coefficients are used for Instrument recognition (Eronen and Klapuri 2000). Eronen compared the performance of the LP coefficients, the MFCC and WLPCC for Instrument identification and confirmed the robustness of WLPCC (2001). Agostini et al have used spectral features like centroid mean, in-harmonicity mean, bandwidth standard deviation and harmonic energy percentage for Musical timbre identification yielding to Instrument identification (2003). Cepstral features are used not only for Instrument identification, but also considered for identifying the characteristics of a music signal. The Cepstral features that are used for Music signal processing are the MFCC, and LPC (Mckinney and Breebart 2003, Mandel and Ellis 2005), which typically convey the timbral characteristics. Mandel and Ellis (2005) have used the short-time spectral characteristics of the MFCC for the process of identification of Timbre characteristics.

Maddage et al (2004) have designed and used a new set of Cepstral Coefficients called Octave Space Cepstral coefficients (OFCC/OSCC) that convey the timbre characteristics for the process of Singer identification based
on the Octave interval of Western music. The frequency corresponding to the successive notes of an Octave is chosen for designing the filter banks. Therefore, it is evident that since the Cepstral features convey timbre characteristics these features are useful for Singer and Instrument identification.

### 4.4.2 Features for Carnatic Music Signal Processing

The features that have been discussed so far have been basically used for processing Western music or speech. We considered the possibility of using these feature extraction algorithms for Carnatic music processing. Some of the existing spectral and Cepstral features could be used for processing while certain Carnatic music relevant features like the Tonic indicating the frequency of the Shadja ‘S’, required the design of new algorithms for their extraction. On the other hand the spectral features like the spectral density, spectral centroid, spectral flux, and spectral energy can be used for processing. In this work, the use of temporal features is very minimal, as it conveys very little information in the identification of the swaras, since the swara refers to the frequency component of the signal. Cepstral features as used by Western music, namely, the OSCC, and MFCC were designed for Western music and speech, and hence, we wanted to design a new set of Cepstral features that caters to Carnatic music.

Hence, in this thesis, the focus is on designing new algorithms for Tonic estimation, and incorporating this estimated frequency to design a new set of Cepstral coefficients for Carnatic music.

### 4.4.3 Need for Tonic Estimation in Carnatic Music

Due to the importance in understanding the characteristics of Carnatic music (Raga, and Tala), a new algorithm to determine the tonic of
the input song is designed and implemented. In order to determine the Raga of a particular Carnatic music song it is mandatory to know the swaras available in the song, this in turn, is highly dependent on the tonic. In the speech analysis and Western music scenario, typically, the fundamental frequency is estimated as the lowest frequency of a periodic wave, of a given song. However, in Carnatic music, this lowest frequency component is not the frequency of ‘S’, since a song spans two octaves. The range of two octaves starts from the upper half of the lower octave covers the entire middle octave and the lower half of the higher octave. The frequency of the middle octave ‘S’ which corresponds to the C of Western music, is typically referred to the tonic. In Carnatic music, the Singer normally starts at a frequency higher than the absolute frequency of ‘C’, and refers to this starting frequency as the swara ‘S’. Hence, in order to span a frequency range of two octaves it is very important that the Singer chooses this tonic with necessary caution. This choice of the tonic ‘f’ ranges from ‘f/2’ to ‘3f’ and spans two octaves. Therefore, depending on the starting frequency which is referred to as the frequency of the middle octave ‘S’ the other frequencies would slide, depending on a ratio given in Table 1.4 of Chapter 1. The determination of the tonic, and mapping the other associated frequency components of a given Carnatic musical piece, are essential in the determination of the various swara components.

As tonic refers to the relative fundamental frequency, in order to identify the tonic, the various algorithms available for fundamental frequency estimation of Western music were analysed for Carnatic music (Hara et al 2009, Chevigne and Kawahara 2002). The algorithm YIN was proposed by Cheveigne and Kawahara (2002), which is a generalized algorithm for speech and music based on the auto-correlation method. In the Sawtooth Wave Inspired Peak Estimator (SWIPE) algorithm developed by Camachho and Harris (2008) the fundamental frequency of speech and music signal were
estimated based on spectral comparisons. The average peak-to-valley distance of the frequency representation of the signal, is estimated at harmonic locations.

The algorithm developed by Klapuri (2003), is also based on harmonicity, spectral smoothness and synchronous amplitude evolution of the input signal for determining the fundamental frequency.

We considered the three algorithms, namely, the YIN (Cheveigne and Kawahara 2002), the SWIPE algorithm (Camachho and Harris 2008) and the algorithm proposed by Klapuri (2003) for Carnatic music, and since the estimated fundamental frequency did not correspond to the frequency of ‘S’, we designed a new algorithm that exploits Carnatic music characteristics for its estimation.

The algorithm proposed by Klapuri (2003), motivated us to use a spectral comparison based approach. Klapuri (2003) used spectral smoothness and harmonicity as features and performed spectral comparisons within the segments of the same file. In our algorithm this comparison between the spectral features is made between the original and modified original signal. The modified signal is chosen based on the biological theory of mutation where we have used the principle behind the Biological Neutral mutation theory. For performing this comparison we determine features like spectral flux, centroid and the MFCC and compare the original signal’s features with the same features of the mutated signal. For mutating the original signal, the octave interval characteristics of Carnatic music are used.

4.4.4 Tonic Estimation based on Mutation Theory

The concept of mutation is a well known methodology in biological science. It is used in many computer applications, and in particular, in signal
processing (Munteanu and Lazarescu 1999, Lu 2006, Reis et al 2008). Mutation is normally identified in a DNA molecule as a change in the DNA’s sequence which is due to radiation, viruses or exposing a body to a different environment or surrounding (Ochman 2003). The process of mutation, which can influence the change of the DNA sequence, could result in an abnormality in the exposed cell. Some mutations are harmful and others are beneficial. In this context, we have a concept called neutral mutation, which does not have any effect, be it beneficial or harmful, but just changes the DNA’s sequence without affecting the overall structure of the DNA. In computer applications the exposure of the features to another environment is treated as mutation, and it has found applications in various fields, including signal processing.

Munteanu and Lazarescu (1999), have utilized the concept of mutation to perform genetic algorithm coding, to design IIR filters. They have utilized mutation operators like uniform mutation and non-uniform mutation that would select a gene from the available gene pool. After creating a gene pool, a Principal Component Analysis was performed on the created pool set which is also based on the concept of mutation: “mutation tends to homogenize the components to avoid having few principal components and neglecting the others”. Using the determined code values, IIR filters were designed, where the coefficients of the IIR filters are determined, using the proposed mutation technique. The authors claimed that the results of the IIR filters were better than those of the Newton-based strategy.

Lu (2006) and Reis et al (2008) have used a mutation strategy to decide the notes to be used for transcribing a piece of music. Here, the authors create a gene pool of possible transcriptions for a particular piece of music, and then, use the mutation theory that would assign a fitness value to determine the exact transcription against the possibilities of all the available transcriptions. The authors have used the mutation theory of irradiate, nudge,
lengthen, split, reclassify and assimilate, to determine the transcription sequences.

These algorithms for the IIR filter design and music transcription motivated us to use the biological based mutation theory, where we exploit the feature of neutral mutation to determine the tonic of the signal. The signal’s frequency components are similar to the DNA’s sequence. In the event of neutral mutation, there is no impact and the structure of the DNA sequence is retained. Similarly, in our algorithm if the mutating signal is made to imbibe with the input signal, the mutated signal’s frequency characteristics will be the same as those of the original input signal and therefore, the mutated signal and the input signal would have the same set of frequency components. After mutating the signal, if the signal characteristics are identical to those of the original signal, then the tonic of the original signal is the same as that of the mutating signal. The tonic of a given song does not vary and hence, the “aalapana” is considered for identifying the tonic. The minimum duration of aalapana required is 20 seconds.

4.4.4.1 Mutation Algorithm

The mutation based algorithm requires a database of mutating signals to be created to be used as the basis for tonic identification. This database consists of pre-recorded signals of ‘S’‘P’‘S’ of all the 22 intervals of Carnatic music created using string instrument of the keyboard. The proposed system for determining tonic is shown in Figure 4.3.
The mutating signal was imbibed into the original signal at three positions: the beginning, the middle and the end to obtain three mutated signals for computing the features. The necessity of three positions arises, due to the fact that the characteristics of ‘S’’P’’S’ can occur anywhere in the signal and here we consider three positions, beginning, end and middle. The three mutated signals are given individually to the feature extraction module to compute the MFCC, Spectral flux and Centroid features. The 3 sets of features are compared with the features of the original signal. The Euclidean distance value between the original signal and that of the mutated signal is determined first with the MFCC feature. If the algorithm is not able to unambiguously determine the tonic, the spectral flux and centroid are additionally used as features for its determination.

The proposed algorithm has a fixed number of iterations corresponding to number of intervals in the octave of Carnatic music. The pseudo code of the basic algorithm is given below:
Algorithm_Mutate (Input Signal, Mutating signal)
{
    While (mutating signal)
    {
        Mutatedsignal[3] = Original signal is mutated at three positions
        beginning, middle and end
    }
    Return Mutatedsignal
}

Determine_RelativeFundamentalFrequency(Original signal, Mutated Signal)
{
    Extract MFCC, Spectralflux, Spectral centroid from input signal
    For all mutated signals
        Extract MFCC, Spectralflux, Spectral centroid
    Q1 = Q2 = Q3 = \infty
    i = 1
    While (Mutatedsignal)
    {
        Determinedvalue[3] = Compare features of Mutated
        signal at three positions with original signal’s features
        If (Determinedvalue1 < Q1 & Determinedvalue2 < Q2
        & Determinedvalue3 < Q3)
            Then
                Q1 = Determinedvalue1
                Q2 = Determinedvalue2
                Q3 = Determinedvalue3
                Tonic = Frequency of the ith
                Mutating signal’s S
        i = i+1
    }
}

4.4.4.2 Evaluation of the Mutation Algorithm

The Tonic is the basis for deriving the swara components, and
hence, it is vital for this algorithm to be error free and hence need to be
evaluated (Bay et al 2009). Therefore, we analyse the performance of our
Mutation based algorithm with the YIN, by determining the tonic of various
songs of four Singers. The algorithm was evaluated for the following parameters based on the evaluation suggested by Kotnik et al (2006). The authors have used the parameters already proposed by Martino and Laprie (1999) and Ying et al (1996) for parameters like gross error high, gross error low, voiced errors, unvoiced errors, average mean difference in pitch and average difference in standard deviation. All these parameters estimated the percentage of difference between the actual frequency and the computed frequency, by considering the speech signal as voiced and unvoiced signals. Ying et al (1996) also estimated the precision, recall and F-measure for evaluating the fundamental frequency.

All the measures discussed above gave an estimate of identifying the wrong frequency as the fundamental frequency. This motivated us to introduce a new evaluation parameter, based on the analysis of observed results, from our tonic determination algorithm. A typical problem in identifying the fundamental frequency is that harmonic frequencies are sometimes identified as the fundamental. The harmonic could be the next lower or higher multiple of the tonic – the frequency of the ‘S’. If the harmonic is identified as the tonic, the algorithm would yield the wrong identification of swaras. Hence, to determine this type of error, we introduce a new evaluation parameter called the harmonic frequency estimation error.

Some of the algorithms that are already available for speech and Western music determine mostly the harmonic of the lowest frequency as relative fundamental, and hence, we used this as one more parameter for evaluation. In addition, we have also used the existing parameters like absolute mean difference in pitch and absolute difference in standard deviation, to analyse our algorithm. The parameters are discussed below.
1. **Harmonic frequency Estimation Error (HE)**

In Western music or Speech, in general, the lowest frequency component is termed as the fundamental frequency. However, in Carnatic music, the lowest frequency component is not the tonic as already explained. When a comparison of the algorithms of YIN and our mutation based algorithm was made, it was observed that YIN determined the Harmonic of the lowest frequency in more number of situations, than the mutation based algorithm. This is because of the voiced and unvoiced components present in the input musical piece. When the tonic is available in the unvoiced component segment, this frequency is skipped, and the algorithm identified the harmonic of the tonic (Kotnik et al 2006). The Harmonic frequency estimation error is defined as the ratio of the harmonic of the relative fundamental estimated against the determination of the tonic, and is given by:

\[
\text{Harmonic frequency estimation error} = \frac{\text{Harmonic of Relative Fundamental identified}}{\text{Relative Fundamental is identified}}
\]  

(4.5)

The determination of the harmonic error is important for Carnatic music signal processing, since the determination of the harmonic instead of the actual tonic would result in the determination of the wrong swara pattern, and hence the wrong Raga in later stages of processing. In addition, the tonic indicates the singing range of the Singer. For example, if the harmonic frequency is 500 Hz as against the correct fundamental of 250 Hz, then it wrongly indicates the singing range as 250 Hz to 1500 Hz, instead of the correct range of 125 to 750 Hz. Therefore the determination of this error is important in the correct Raga identification and correct singing range of the Singer.
The tonic values of four Singers are listed in Figures 4.4 to 4.7. We have however determined the tonic of nearly 10 Singers covering a total of 1500 songs for the process of Raga identification which is discussed in Chapter 5. We compare the performance of our tonic estimation algorithm against the performance of the YIN (Cheveigne and Kawahara 2002) for nearly 20 songs each for four singers and the results are shown in Figures 4.4 to 4.7. A survey is also conducted with musicologists to know the tonic of these songs of the four Singers and have used for comparison. In Figures 4.4 to 4.7, the reference line marked is the frequency as estimated by musicologists. The YIN in most cases either estimated the lowest frequency component or its higher harmonic as the fundamental frequency but this did not correspond to the expected tonic. This is because the YIN estimated the formant frequency as the fundamental frequency, and hence, the tonic of different Singers for different songs typical of Carnatic music, could not be accommodated. The performance of the YIN was closest to the musicologist’s estimate only when the singer used mostly notes near the middle octave. However, our mutation based algorithm was very close to the musicologist’s estimate, and in no case, estimated the harmonic of the tonic for any of the four Singers, even when they rendered songs using different relative fundamental frequencies.
Figure 4.4 Tonic of Singer Nithyasree

Figure 4.5 Tonic of Singer M.S. Subbulakshmi
Figure 4.6 Tonic of Singer M. Balamuralikrishna

Figure 4.7 Tonic of Singer Ilayaraja
A reference line was marked for the Mutation based algorithm, YIN algorithm, and the survey obtained from musicologists for all the Singers. As can be seen from Figures 4.4 to 4.7 the reference line of a musicologist either overlapped or was very close to that of our mutation algorithm, while the reference line of YIN deviated much from that of the musicologists.

These results are validated by estimating the harmonic error for the four Singers based on the tonic identified. The results are tabulated and the performance chart is shown in Figure 4.8. As can be seen from the figure, the mutation based algorithm and the one determined by musicologists have a low error rate, when compared to the YIN.

![Harmonic Error Chart](image)

**Figure 4.8** Harmonic Error of four Singers using YIN, Mutation based (our) and Musicologist
As can be observed from YIN, where the lowest frequency is termed as the fundamental frequency, the probability of the harmonic frequency being identified as the fundamental frequency as against the actual fundamental frequency is high. As can be seen from Figure 4.8 the performance of the mutation algorithm was low for Singer Ilayaraja when compared with the other Singers. This is due to the fact that the typical singing range of Singer Ilayaraja is generally low, with the tonic varying between 200 Hz to 240 Hz. This resulted in the mutation algorithm also identifying the harmonic of the tonic.

2. Absolute difference between the mean values (ABDM) and Absolute difference between the standard deviations (AbsStdDiff)

We also evaluated the performance of our algorithm by using two standard measures, the absolute difference (in Hz) between the mean values (ABDM) and the absolute difference (in Hz) between the standard deviations. These measures are computed using tonic, which is the normal singing range of the Singers, and the actual estimated tonic with the help of the following equations.

\[
\text{ABDM}[\text{Hz}] = |\text{MeanRefPitch}[\text{Hz}] - \text{MeanEstPitch}[\text{Hz}]| \quad (4.6)
\]

\[
\text{AbsStdDiff}[\text{Hz}] = |\text{StdRef}[\text{Hz}] - \text{StdEst}[\text{Hz}]| \quad (4.7)
\]

The average tonic as estimated by the mutation algorithm, the YIN and the musicologists was determined and the reference pitch was chosen as 400Hz, 320 Hz, 300Hz, 250 Hz for Nithyasree, M.S. Subbulakshmi, Balamuralikrishna and Ilayaraja respectively. The reference pitch is chosen by observing their normal range of singing.
The above mentioned mean values and standard deviations are computed with the reference and estimated tonic respectively, where the mean conveys the average values of all the fundamental frequencies used by the Singer, while the standard deviation shows the range of all the fundamental frequencies used by the Singer. The differences between the computed values are shown in the graphs shown in Figures 4.9 and 4.10. It is observed that the YIN deviated to a greater extent from the other two algorithms, since it computed the harmonic of the lowest frequency for most of the songs. The mutation algorithm as well as the one determined by the musicologists was almost the same for all the singers, since in both the cases the estimated tonic is the same.

As already explained, the absolute difference between the mean values is estimated and is shown in Figure 4.9. It was observed that for Singers Nithyasree, and M.S. Subbulakshmi, the estimations made by YIN and mutation algorithm were comparable, while for Singer Balamuralikrishna, the YIN gave a higher difference between the computed mean value and the reference value. The YIN did not consider the range of fundamental
frequencies; hence, its estimate in general was not comparable with the musicologists' estimate as shown in Figure 4.10. However, the tonic estimated by our algorithm was comparable with the musicologists’ estimate across the range of fundamental frequencies used by the Singers.

![Figure 4.10 Absolute differences between standard deviations](image)

**Figure 4.10 Absolute differences between standard deviations**

### 4.4.5 Carnatic Interval Cepstral Coefficients

MFCC’s and its extensions like HFCC, OFCC are the Cepstral features mostly used for processing speech and music. MFCC’s are extracted by designing a bank of filters based on mocking the human auditory system. These coefficients were initially designed for Speech Processing, and have now found applications in understanding the characteristics of music, like the Singer, Instrument, and Genre, as they represent the timbral characteristics.

While the MFCC and HFCC are specially designed by considering the auditory properties, the OFCC considered the 12 octave interval characteristics of Western music for filter bank design, for coefficient determination. These Cepstral features however did not consider the source
properties of the input signal and were basically used for Singer, Instrument, and Genre identification. Therefore, we considered including the tonic of the input and the 22 octave interval of Carnatic music for designing the new set of Cepstral coefficients which we call as the *Carnatic Interval Cepstral Coefficients* (CICC). The inclusion of the tonic essentially makes the Cepstral coefficients vary from one singer to the other and even within songs of the same singer.

### 4.4.5.1 CICC computation

The input to the feature extraction module to determine the CICC are the segments that have been identified by the Tala based segmentation algorithm. We assume that the segments obtained of the output of the Tala based algorithm would correspond to one swara. These segments are converted into frames; go through a process of windowing, and conversion to the frequency domain, similar to the pre-processing performed for the MFCC (Yoshimura et al 1999).

Figure 4.11 shows the computation of the CICC which is part of the Feature extraction module of our proposed music signal processing system.

![Figure 4.11 Computation of CICC](image-url)
After converting the signal into the frequency domain the frequencies can be used directly as features for processing. Here, at this stage, we have made use of Carnatic music’s 22 interval octave system and have designed and proposed the conversion to compute the Carnatic frequency. The relationship between the Carnatic frequency and the input frequency is given by:

\[
f_{\text{CARNATIC}} = \left(\frac{22}{7}\right) \log_{10}(1+(f/f_0)).
\]  

(4.8)

where, the component 22/7 refers to the assignment of seven swaras for an octave that consists of 22 intervals; this assumption is valid since we are extracting the coefficients from a segment that would probably correspond to a swara; ‘f’ is the frequency that is estimated in the current frame, \(f_0\) is the tonic (the frequency of the ‘S’) of the signal. The frequency \(f\) typically varies from \(f_0/2\) to \(3f_0\). The computation of the Carnatic frequency involved computing a ratio, adding unity to the ratio, taking logarithm and multiplying it by a constant. The value of ‘1’ is added to the computation of logarithm to avoid negative coefficients due to the frequency range as indicated. The multiplication by ‘22’ is to consider the coefficients for one octave, with the assumption that we are considering the 22 intervals per octave scheme. This coefficient is further divided by ‘7’ so as to map this value to an interval per swara. We define the filter banks gain for estimating the CICC using the following equation:

\[
CI_j = \sum_{k=0}^{N-1} |X(k)|^2 H_j(k).
\]  

(4.9)

where \(0 \leq j \leq p\), where \(p\) is the number of filters required, \(X(k)\) is the value of the FFT, \(H_j(k)\) is the gain of the \(j\)th filter, and \(CI_j\) is the energy at the \(j\)th filter bank. The computation of the filter banks energy is similar to the MFCC but
the gain $H(k)$ of the filter is based on Carnatic Interval’s frequency value as the centre frequency, as defined in Equation (4.8).

After estimating the Carnatic interval filter bank values we perform Discrete cosine transform to determine the coefficients. The coefficients are given by:

$$C_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{p} CI_j \cos \left( \frac{\pi i}{N} \left( j - 0.5 \right) \right).$$  \hspace{1cm} (4.10)

where $CI_j$ refers to the filter outputs energy estimated in Equation (4.9), $N$ is the number of samples in a window, and $p$ is the number of filters. The $CI_j$’s are called the Carnatic Interval Cepstral coefficients (CICC), and the first ten are found to be the most useful. They are used for music signal processing as they produced reasonable results, compared to the 13 to 15 coefficients of the MFCC.

These CICC coefficients have a dynamic varying filter bank due to the incorporation of the tonic and are show in Figures 4.12(a) and 4.12(b).

Figure 4.12 (a) CICC filter bank – singer I
Other features are extracted from every segment, namely, the Spectral flux, Spectral Centroid, MFCC and our newly designed Carnatic music specific tonic, and the CICC. These coefficients are used by the Raga identification module. The use of the CICC is also tried for the process of Singer, Instrument, Genre and Emotion identifications. The performance of the CICC as a Cepstral feature for Carnatic music processing, can be estimated only by using this feature in music component identification, which will be discussed in chapters 5 and 6.