Using Fast Fourier Transform, the digital signals of a musical sound can be effectively broken into its constituent frequencies and amplitudes.
Sound signals induce fluctuations in pressure along the path of a sound wave. When sound waves fall on a microphone, the microphone converts the sound signal into electrical pulses. These pulses are fed to a sound card which converts these analog signals into digital signals. Using the Fast Fourier Transform (FFT), these digital signals can be effectively broken into its constituent frequencies and amplitudes. These frequencies then can be used to find out the Swaras in a Raga rendered by a musician. A schematic diagram of Raga identification process is shown in Fig 4.1.

4.1 Acquiring Data with a Sound Card

A multimedia personal computer is used to identify Raga. Nowadays all personal computers are having built-in sound card. A microphone can be externally...
connected to the sound card. The sound card converts the analog signals to digital signals, so that the computer can handle these signals. Using Fast Fourier Transform, a mathematical tool, we can convert digital signals into its constituents, namely, frequency and amplitude. Many computer languages offer FFT functions or we can write a program in C or C++ for this purpose. Matlab, Mathematica etc. have built in FFT functions. The FFT based programs help us to sort out the frequency and amplitude parts of the signal. Program 1 and Program 2 in the figure are two computer programs, one used for the analysis of male voice and male instruments and the other for female voice and female instruments. These programs can be used to find out the Swara-positions and hence the Raga in a sound signal. After scanning the signal for about 3 minutes, the detected frequency, amplitude, Swaras and the Raga will be displayed on the screen. A graph with frequency on the x-axis and amplitude on the y-axis also can be displayed. After analyzing the Swara positions the best match among the Ragas will be identified and the graph is plotted. The Swaras belonging to the Raga will be displayed as green dots and the Swaras which does not belong to the Raga as red dots.

4.2 Fourier theorem [22]

According to this theorem, any periodic function \( F(t) \) of the period \( T \), however complex it may be, can be represented by a unique combination of the function \( f_n(t) \) and \( g_n(t) \)

where \( f_n(t) = \sin(2\pi nt / T) \) and \( g_n(t) = \cos(2\pi nt / T) \) where, \( n=0,1,2,3, \ldots \)

Mathematically, Fourier theorem can be written as

\[
\hat{f}(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left( a_n \cos nt + b_n \sin nt \right)
\]
4.2.1 The Fourier Transform [24]

The *Fourier transform*, in essence, decomposes or separates a waveform or function into sinusoids of different frequency which sum to the original waveform. It identifies or distinguishes the different frequency sinusoids and their respective amplitudes.

4.3 Program to find the frequency and amplitude of a sound signal

To find the frequency and amplitude of a sound signal, a program developed for this, *Sruti.m*, is used and is given below.

```matlab
AI=analoginput( 'winsound' );
chan=addchannel(AI, 1);
du=1;
set(AI,'SampleRate',44100)
ActualRate=get(AI,'SampleRate');
set(AI,'SamplesPerTrigger',du*ActualRate);
set(AI,'TriggerType','Manual');
blocksize=get(AI,'SamplesPerTrigger');
Fs=ActualRate;
start(AI);
trigger(AI);
data=getdata(AI);
delete(AI)
clear AI
[f,mag]=daqdocfft(data,Fs,blocksize);
plot(f, mag)
grid on
ylabel('Magnitude (dB)')
xlabel('Frequency (Hz)')
```
title('Frequency Components of Tuning Fork')

Here, winsound is the name of the analog input hardware driver adaptor. The number of channels used is one. Time duration is 1 sec. Sample rate is 44100/sec. TriggerType can be Immediate, Manual, or Software. If TriggerType is Manual, the trigger occurs immediately after the trigger function is issued.

[f,mag] = daqdocfft(data,Fs,blocksize) calculates the FFT of data, using sampling frequency FS and the SamplesPerTrigger provided in blocksize.

daqdocfft() outputs the frequency and magnitude of data. The details of Daqdocfft() is shown below.

\[
\begin{align*}
\text{xfft} &= \text{abs}(\text{fft}(\text{data})); \\
\text{index} &= \text{find}(\text{xfft} = = 0); \\
\text{xfft}(\text{index}) &= 1e-17; \\
\text{mag} &= 20*\log10(\text{xfft}); \\
\text{mag} &= \text{mag}(1:\text{floor}(\text{blocksize}/2)); \\
\text{f} &= (0:\text{length}(\text{mag})-1)*\text{Fs}/\text{blocksize};
\end{align*}
\]

4.3.1 An experiment for finding the frequency of a turning fork

Hold the turning fork in front of the microphone attached to the computer. The sound generated by the tuning fork will be input into the program which in turn will identify the frequency and amplitude. For example, let us consider a turning fork of frequency 256 Hz. We will acquire data for 1 second through one channel of the sound card. Because the tuning fork vibrates at a frequency of 256 Hz, we can configure the sound card to its lowest sampling rate of 8000 Hz. Even at this lowest rate, we should not experience any aliasing effects because the tuning fork will not have significant spectral content above 4000 Hz, which is the Nyquist frequency.
Fig 4.2 Experimentally finding the frequency of a turning fork using the program Sruti

After we set the tuning fork vibrating, place it near the microphone, and run the program Sruti.m. Now the analog data will be scanned for 1 Sec. Then it will be converted into digital data. By applying the FFT, the data will be split into frequency and amplitude. A graph is plotted with Frequency on the x-axis and Amplitude on the y-axis. The frequency with maximum amplitude can be found out and normally, it represents the frequency of the turning fork. But it is not necessary that it is the fundamental frequency. The graph shows the frequency obtained, with a turning fork of frequency 256 Hz in the actual run.

From the graph we can see that the fundamental frequency is 256 Hz and it is having maximum amplitude. The second maximum amplitude is for 511 Hz which is the first overtone.
4.4 The Fundamentals of Raga rendering

A Raga is always rendered with reference to a fixed frequency called the base frequency. Different Swaras are achieved by varying the frequency with respect to the base frequency. One Swara is attained from another Swara by a smooth transition between the corresponding frequencies. Due to the limitations of the human ear and also due to the lack of expertise in detecting changes of one or two frequencies, smaller deviations from the theoretical position in the rendering of Swaras often escape unnoticed. However, the program mrag and frag has no such limitations. It invariably helps us to identify the best performer. Moreover the Swaras in the sequence of rendering will be captured by the program and the corresponding match from the list of Ragas as per the theoretical details available, will be computed. Since human limitations and inconsistencies have no room in this analysis, the outcome is perfect to the core. The program will bring out even the smallest deviations from the theoretical model of the Ragas. By improving the sample rate and the duration for analysis, the accuracy can be improved to any desired level.

4.5 Identifying the Raga in a rendering

Raga can be identified either using a live input through a microphone attached to the sound card or a recorded performance from a compact disk or hard disk. Depending on the gender of the performer either the mrag or the frag program is used. Let us assume that we use mrag.

The base frequency, the number of scans and the duration of the scans are given as the initial input to the program. Let us scan the input signal continuously for 300 times, with a scan duration of 0.5 sec. The sound is input to the program and the analysis begins.

The program first identifies the most frequently occurring frequencies with significant amplitude in each scan. At first the program picks up the most
significant frequencies among the observed frequencies and close to the given base frequency. This is considered as the actual base frequency and all further analysis is based on it.

The frequency ranges corresponding to all the 12 Swaras are calculated, a priori, in relation to the observed base frequency. The base frequency of sa is also termed as base Sruti. It may be noted that the range of frequencies corresponding to a Swara is a new concept introduced by us and extensively used in all our analysis. This has no parallel in the literature. However, this has proved to be substantially important and useful in our analysis.

From each of the range of frequencies of Swaras, the most frequently occurring frequencies with significant amplitude among the observed frequencies are identified. In all there will be 12 such frequencies. From among these 12 the most prominent 7 frequencies are identified. While picking the 7 Swaras, the sequence relevant to the 72 Melakartha Ragas alone are considered. Hence, the program will always identify a Raga from among the 72 Melakartha Ragas which is the closest match to the observed sequence of Swaras. If there are mistakes in the rendering of a Raga then the program will either identify the rendering as a different raga or the same Raga with one or more red spots in the graphical representation of the result of the analysis. The red spots in the graph correspond to significant Swaras among the 12 Swaras in excess to the required 7 Swaras for a Raga. The seven Swaras picked up from among the observed significant frequencies as a result of the analysis are represented by green spots in the graphical display of the output.

The absence of any red spot in the graphical display of the entire scan is a clear indication of the perfection in rendering. The lesser the number of red spots, the more perfect is the rendering. Our study has gone further to evaluate the level of artistes' perfection by devising two scales, namely, Sruti Consistency Coefficient (SCC) and Raga Consistency Coefficient (RCC). The details are given in chapter 6.
4.6 Algorithm of Raga identification program

The algorithm of the Raga identification program is given below:

1) Input the base Sruti (c, C, d corresponding to western notation C, C#, D): s
   Match the range of frequency for this Sruti: (f1, f2)
2) Input the number of scans, n
3) Scan: read the analog input signal for a period of 0.5 sec.
4) Repeat scanning n times
5) Calculate the frequency and amplitude of n scans.
6) Identify the base Sruti – s from the scanned data and update s.
7) Using the value of s, calculate the frequency of all the 12 Swaras and the
   range of frequencies of each Swara.
8) Find out the Swaras of the frequencies stored from the data.
9) Select 7 Swaras from among the 12 Swaras, which constitute the Raga.
10) Identify the Raga from the Swara - Raga - table.

4.7 The Relevance of Scan Duration

For Raga identification, usually we take a minimum of 300 scans from the
input signal with scan duration of 0.5 sec. The time taken to scan each sample is
very important in the Raga detection process. A study was made to determine the
most suitable scan duration. We have found that 300 scans of duration 0.5 sec.
would result in data corresponding to 300 x 0.5 = 150 sec. of actual performance.
This data is sufficient and the duration is adequate in unfolding the specifics of a
Raga. There are exceptional cases when an artiste takes more than 2.5 minutes to
reveal the raga he renders. This is due to the individual’s style of elaborating a
Raga which is permissible in Classical Music. We have also conducted analysis of
Ragas using the program by taking different number of scans and durations.
Raga Kamavardhini, sung by M. S. Subhalakshmy was used for the study with different scan durations and 300 scans. Time durations 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0 sec. were considered for the study. It was found that for 0.1 and 0.2 sec. durations the detection was not consistent and for all other durations, the Raga was identified.

![Fig 4.3 No. of sample frequencies obtained in the range 0 to 600 Hz for a scan duration of 0.1 sec.](image)

The sensitivity of the scan is dependant on the scan duration and is the inverse of the scan duration. The unit of sensitivity is Hertz. For 1.0 sec. scan duration, the sensitivity is $1/1.0\text{sec} = 1\text{Hz}$. and for 0.1 sec. the sensitivity is $1/0.1\text{sec} = 10\text{Hz}$. This means that the frequencies detected will be 10, 20, 30, 40, 50, etc. But if the sensitivity is 1, the frequencies detected will be 1, 2, 3 .... etc. If the scan duration is 0.5 sec., the sensitivity will be 2 Hz. For 300 scans the time taken for detection is 150 sec. If sensitivity is 1 Hz, total time for scan will be 300 sec. If we want more accuracy in the detection, scan duration must be increased. As a compromise, we have taken 0.5 sec. as the scan time for our study. This means
that a singer has to sing a Raga for at least 1.5 minutes in order for the program to identify the Raga.

The graphs showing amplitude and frequency for scan duration of 0.1 sec., 1.0 sec. and 0.5 sec. are given in Fig 4.3, Fig 4.4 and Fig 4.5.

**Fig 4.4** Sample frequencies obtained for duration of 1.0 sec.

Figure 4.3 corresponding to 0.1 seconds duration contain very less no of scan points and is found to be inadequate to arrive at a meaning full conclusion. The identification of Ragas with 0.1 seconds is not advisable.

Figure 4.4 corresponding to 1.0 sec. duration have a very large number of scan points and is very good for Raga analysis. The Raga identification in this case will be very accurate. However the total time for the scan is 300 sec. which is very high and is not recommended for the analysis.
Figure 4.5 contains scan points corresponding to 0.5 sec. scan duration. The number of points is adequate to arrive at a consistently good result. Therefore for practical purposes 0.5 sec. of scan duration is ideal for the Raga identification programs.

A research paper on Identification of ragas played on Musical Instruments was presented in the National Seminar held at Sree Saraswathi Thyagaraja College, Pollachi.