Tanpura se - An Algorithmic Approach To Indian classical Music Using Tanpura Feature Extraction.

Purusottam Samal
samalpuru@gmail.com
Columbia College Chicago
Chicago, Illinois, USA

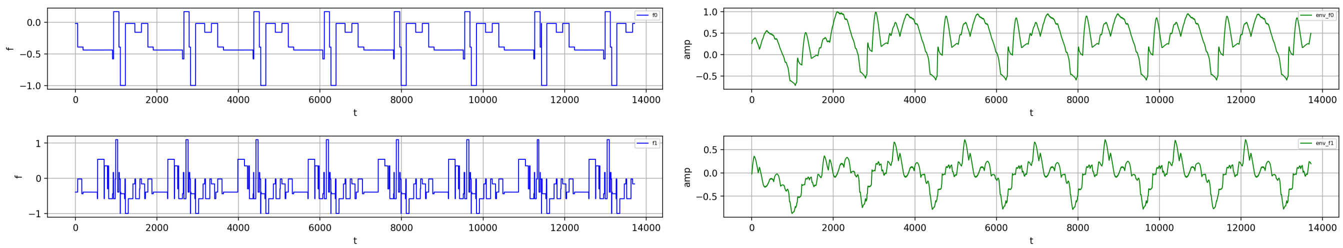

Figure 1: Frequency and amplitude of a tanpura signal over time.

, ,

1 ABSTRACT

The tanpura is an instrument that sets the harmonic background in Indian classical music, in context of which the melody evolves. It is known for its rich harmonic structure leading to the perception of several micro-tonal notes for which there are no corresponding tuned strings, and which correlate well with "shrutis" (micro-tonal note positions) in Indian classical music. This paper describes an approach to generative music in the Indian classical system by harnessing the role of the tanpura as a primary melodic background feature by using features extracted from a tanpura signal for the control of seven sound synthesis modules. The combination of control signals generated by the time-varying frequencies and amplitudes of the prominent partials of a tanpura signal with a modulo-based trigger system facilitated the creation of an interactive generative composition system with evolving melodic and rhythmic phrases. This approach has the potential to be extended further by implementing timbral features from sounds with similarly complex harmonic structures in order to explore the generalisability of the algorithms described in this paper.

2 INTRODUCTION

The tanpura is a long-necked plucked string instrument found in various forms in Indian music. It does not play melody but rather supports and sustains the melody of another instrument or singer by providing a continuous harmonic drone. It is widely held that the tanpura sets the harmonic background in context of which the melody evolves.

It is known for its rich harmonic structure leading to the perception of several micro-tonal notes for which there are no corresponding tuned strings. An acoustical analysis of the tanpura reveals several harmonic partials that correlate well with these micro-tones. Musicians claim the existence of great consonance between the notes used in Indian ragas (melodic modes) and the sound of the tanpura. Indian classical vocalists are routinely advised to practice singing with the tanpura which helps in purifying their ability to produce correct ”shrutis” (micro-tonal note positions) [1] by feeling the consonance between their singing and the tanpura sound.

Because of it’s role as an important melodic background feature, it was discovered that the the frequency content of a tanpura signal can be well-suited to generating melodic sequences in the Indian classical music system. By re-purposing a pitch-detection algorithm based on the Fast Fourier transform, The trajectory of seven harmonic partials and their corresponding amplitudes envelopes (in the order of decreasing magnitude) are analyzed and then re-synthesized. They are then used as control parameters to drive a sound synthesis system written in Max/MSP [6]. This paper will describe the algorithms used to carry out these processes.

3 RELATED WORK

Alpern [2] describes an algorithmic composition as "the process of using some formal process to make music with minimal human intervention". Several studies have been done on generative music in the Indian classical Music system using probabilistic modes to generate raga note-sequences and patterns. Chakraborty et. al. [18] describe an algorithm for semi-natural composition that utilizes a first order Markov chain for generating arbitrary sequences of notes of raga Bageshree. Gosain et. al. [12] use a transition probability matrix to create the note sequences of Raga Bageshree, and then use first and second order Markov chain simulations to generate arbitrary sequences of notes of Raga. By contrast, this paper describes an approach based on signal processing and sound feature extraction due to the effectiveness of the tanpura as a supporting instrument.

Audio Feature Extraction has a long history of use for the generation of data structures for composition and music performance[13–16]. Many different audio features are able to be extracted, such as brightness, spectral centroid, harmonic flux, noisiness, etc. One approach to the use of audio features towards the control of synthesis involves analyzing audio signals for specific characteristics such as spectral centroid or harmonicity, which are then mapped directly to parameters of sound synthesis. The algorithm described in this paper is based on this approach.

The phenomenon of virtual notes in a tanpura drone was reported by Pandya [17]. Virtual notes are complex tones with a missing fundamental. Sengupta et. al. [11] explore this phenomenon further by investigating the waxing and waning of different prominent harmonics at different rates and their relationship to the virtual notes. The approach described in this paper is inspired by the findings from these studies to establish an algorithm to efficiently use the prominent harmonics in a tanpura signal to create a musical composition.

4 METHODOLOGY

In order to use the prominent harmonics in a tanpura signal as control parameters an FFT-based pitch extraction algorithm [19] is repurposed to estimate the frequency of 7 prominent harmonics of a tanpura sound segment and follow their frequency and amplitude over time. Seven audio analysis and synthesis modules are programmed in Max/MSP using the frequency and amplitude features derived from the tanpura sound segment as control parameters. Figure 6 depicts a block diagram of the entire system.

4.1 Audio Feature Extraction

The frequencies of prominent harmonics were calculated to create control signals. the calculation of frequency candidates is achieved with the FFT of a tanpura signal segment using the phase information.

The main structure of the algorithm is depicted in (Fig. 3), where a segment of length N is extracted every R samples and then applied to FFTs.

Usually in an extracted spectrum there are many mathematical peaks and troughs. All of these may not represent the real harmonics. Local maxima and minima may complicate the scene. Sengupta et. al. [11] propose the use of prominent harmonics instead to improve the estimates of the pitch. Without any loss of generality one may take the peak point A (Fig. 2) as a prominent harmonic where:

$$P = V_A/max(V_B, V_C), \tag{1}$$

where V_A is the magnitude of peak point A; V_B and V_C are the magnitudes of left valley point B and right valley point C, respectively and P is 5 dB.

Considering the calculation of an N-point FFT [3], the frequency resolution of the FFT is

$$\Delta f = \frac{f_s}{N}, \tag{2}$$

with the sampling frequency $f_s = 1/T_s$. A block of N samples is used from the input signal $x(n)$,

$$x_1(n) = x(n_0 + n).\ n = 0, \dots, N-1 \tag{3}$$

After applying an appropriate window, the FFT yields $X_1(k)$ with $k = 0, \dots, N-1$. At FFT index k_0 a prominent harmonic of the FFT magnitude $|X_1(k)|$ is detected. From this FFT maximum, the initial

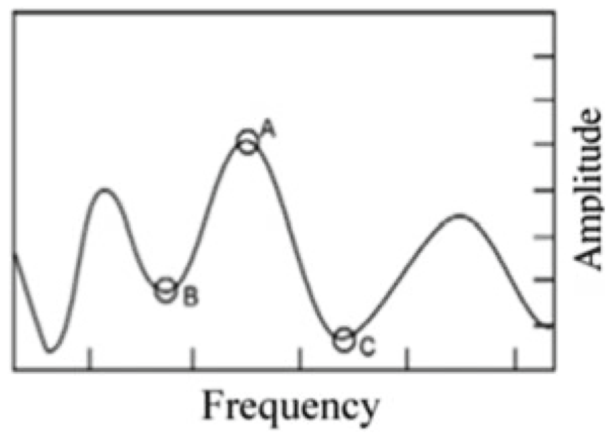

Figure 2: Illustration of Prominent Harmonics (Pg. 118)[11]

estimate of the frequency is

$$\tilde{f}_0 = k_0 \cdot \Delta f = k_0 \frac{f_s}{N}. \tag{4}$$

The corresponding normalized frequency is

$$\tilde{\Omega} = 2\pi \tilde{f}_0 T_s = k_0 \frac{2\pi}{N}. \tag{5}$$

Phase information is used to improve the frequency resolution, since for a harmonic signal $x_h = cos(\Omega_0 n + \varphi) = cos(\phi(n))$ the fundamental frequency can by computed by the derivative

$$\Omega_0 = \frac{d\phi(n)}{dn}. \tag{6}$$

The derivative can be approximated by computing the phases of two FFTs separated by a hop size of R samples leading to

$$\tilde{\Omega} = \frac{\Delta\phi}{R}, \tag{7}$$

where $\Delta\phi$ is the phase difference between the two FFTs evaluated at the FFT index k_0. The second FFT of the signal segment

$$x_2(n) = x(n_0 + R + n),\ n = 0, \dots, N-1, \tag{8}$$

leads to $X_2(k)$. For the two FFTs, the phases at frequency $\tilde{f}_0$ are given by

$$\varphi_1 = \angle\{X_1(k_0)\} \tag{9}$$

$$\varphi_2 = \angle\{X_2(k_0)\} \tag{10}$$

Both phases are obtained in the range $[-\pi, \pi]$. We now calculate an 'unwrapped' ϕ_2 value corresponding to the value of an instantaneous phase. Assuming that the signal contains a harmonic component with a frequency $\tilde{f}_0 = k_0 \cdot \Delta f$, the expected target phase after a hop size of R samples is

$$\varphi_{2t} = \varphi_1 + \tilde{\Omega} R = \varphi + \frac{2\pi}{N} k_0 R. \tag{11}$$

The phase error between the unwrapped value φ_2 and the targer phase is computed by

$$\varphi_{2err} = princarg(\varphi_2 - \varphi_{2t}). \tag{12}$$

The function 'princarg' computes the principal phase argument in the range $[-\pi, \pi]$. It is assumed that the unwrapped phase differs from the target phase by a maximum of π. The unwrapped phase is obtained by

$$\varphi_{2u} = \varphi_{2t} - \varphi_{2err}. \tag{13}$$

The final estimate of the frequency is then obtained by

$$\hat{f_0} = \frac{1}{2\pi}\hat{\Omega_0} \cdot f_s = \frac{1}{2\pi} \cdot \frac{\varphi_{2u} - \varphi_1}{R} \cdot f_s. \tag{14}$$

Using this algorithm, The estimated frequencies of seven prominent harmonics $\hat{f_1}, \ldots, \hat{f_7}$ and their corresponding FFT magnitudes $|X_1(k)|_1, \ldots, |X_1(k)|_7$ is calculated from a segment of length 8192 samples extracted every 128 samples from the tanpura signal. Figure 1. visualizes the trajectories of the frequencies of two prominent harmonics (in blue) and their corresponding FFT magnitudes (in green) throughout the entire tanpura signal. The data is encoded into 32-bit float WAVE files to serve as control signals for sound synthesis in Max/MSP.

4.2 Sound Synthesis

The WAVE files containing the features from the tanpura signal are stored in buffer objects[4] in Max/MSP[6]. While there are several objects in Max that allow one to read the data stored in a buffer object, the groove object [5] was chosen because of the additional flexibility it provides. Multiple groove objects refer to the samples stored in the buffer object and allow for variable-rate looping and sample-playback. Since the algorithm described in section 3.2 extracts a frequency and amplitude value from a segment of 8192 samples taken every 128 samples, The variable-rate sample-playback feature was especially useful to sync the playback of the data with the original tanpura signal. Cubic interpolation was used to interpolate between the data points.

The primary goal of the sound synthesis was to highlight the perception of the frequency and amplitude of the prominent partials in the tanpura signal. To that end, the frequency and amplitude stream pairs (figure 1) served as inputs to seven sound synthesis modules ($SS_1, \ldots, SS_7$). Additionally, the modules incorporate a trigger input in order to facilitate the process of creating a composition (Section 3.3). The sound synthesis algorithms incorporated in the modules are as follows:

Sound Synthesis Modules 1-3: These modules take the frequency and amplitude streams of the first three prominent partials in the tanpura signal as inputs. Figure 3 depicts a generalized block diagram of modules 1-3. The playback of a short percussive audio sample is initiated every time a trigger is received. The output of the audio sample is passed through a resonant bandpass filter with a high Q-factor(250). This causes the audio sample to resonate at the center frequency of the bandpass filter. The frequency ($f(t)$) of partials 1-3 set the center frequency of the bandpass filter. The output of the bandpass filter is multiplied by normalized amplitude values $A(t)$) of the prominent harmonics. Module 1 uses an audio sample with a low spectral centroid , module 2 uses an audio sample with a high spectral centroid while the spectral centroid of the audio sample used for module 3 is between that of 1 and 2. This demarcation ensures a clear separation between the timbres of the sound synthesis module in an attempt to aid their perception.

Sound Synthesis Modules 4-7: Figure 4 depicts a generalized block diagram of sound synthesis modules 4-7. A variable-rate

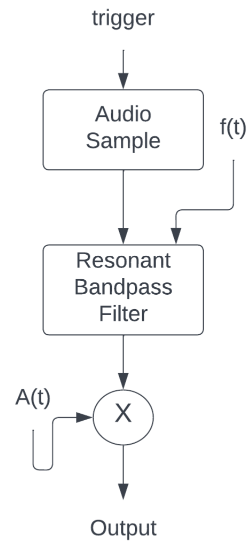

Figure 3: Generalized block diagram of Sound synthesis 1-3.

wavetable synthesis algorithm [9] is used in modules 4-7. It is fundamentally based on periodic reproduction of an arbitrary, single-cycle waveform. In Max/MSP, a buffer object is used to store the samples of the tanpura signal. The wave [10] object is used to access the data stored in buffer as a wavetable for an oscillator. The object allows one to repeatedly read any section of the samples stored in a buffer in order to play a periodically repeating tone with detailed control over the endpoints. Four wavetables with slight deviations in their endpoints and frequencies are created with the base frequency of each wavetable set by the frequency ($f(t)$) of partials 4-7. The output of the oscillators are multiplied by the normalized amplitude values ($A(t)$) of the partials and are further shaped by amplitude envelopes with slight deviations in their attack times. They are then routed into four svf [8] objects which is an implementation of a state-variable filter algorithm. A unique feature of this filter object is that it produces lowpass, highpass, bandpass, and bandreject output simultaneously with all four are available as outputs. The centre frequency of each svf object is additionally shaped by an envelope. The lowpass outputs are summed together to form the final output for modules 4-6 while the highpass output is used for sound synthesis module 7. The envelopes for the wavetable oscillators and the svf filters are initialized every time a trigger is received in the trigger input.

4.3 Compositional Method

All sound synthesis modules take a trigger input. A trigger is a message that tells a Max object to start doing what it does. An algorithmic composition is facilitated through a series of triggers generated by the outputs of a modulo function. A master time interval (in ms) is initialized as a numerator and is individually routed

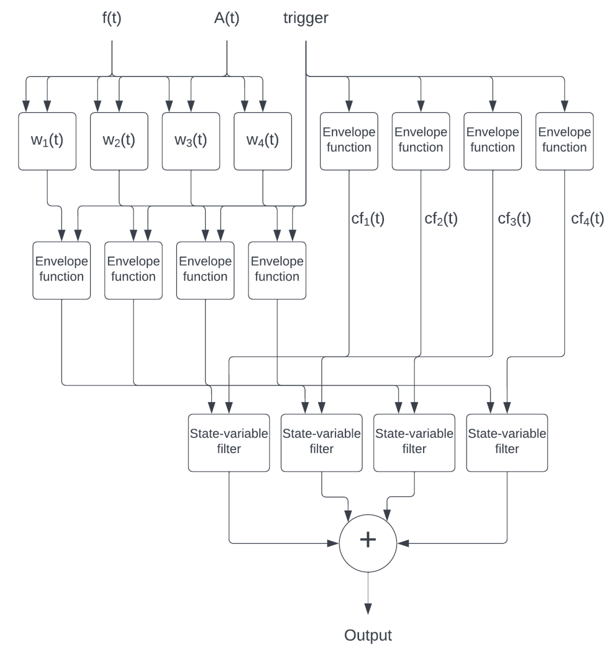

Figure 4: Generalized block diagram of Sound synthesis 1-3.

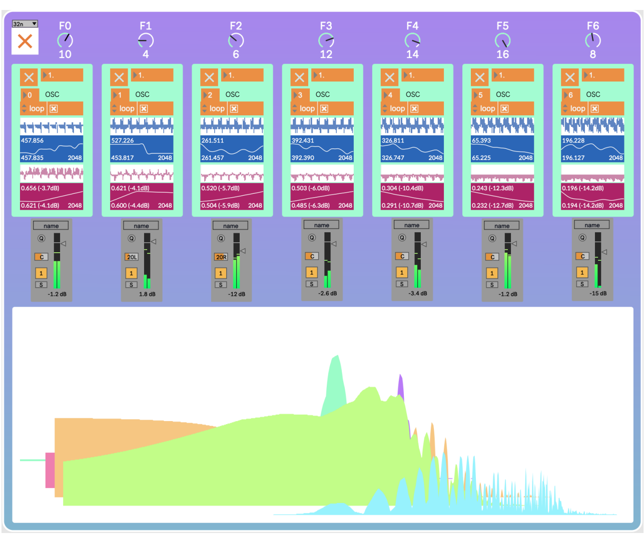

Figure 5: Generalized block diagram of Sound synthesis 1-3.

to seven denominators via Max's % object. Seven subdivisions of the master time interval are generated as outputs and are routed to seven metro object [7] which outputs a series of triggers at the specified time interval set by the output of the modulo function. The denominators associated with the modulo function are varied over time to generate the composition.

5 GRAPHICAL USER INTERFACE

A simple graphical user interface was created for the system that allows one to interact with the output of each sound synthesis module. Each sound synthesis module can be heard individually or as a combination of multiple modules. The trigger speeds of each module can be changed to generated different rhythmic patterns. The interface also incorporates seven color-coded spectograms overlayed on top of one another allowing one to visualize the output of each sound synthesis module as well as the waxing and waning of the different harmonics over time.

6 DISCUSSION

The extraction of features from the tanpura signal is based on an FFT-based pitch detection algorithm which is very accurate. However, the the CPU load of the analysis algorithms has been a point of concern and is something that must be improved upon. An alogorithm based on the Harmonic Product Spectrum (HPS) or Cepstrum-Biased HPS (CBHPS) might serve prove to be faster, more efficient and hopefully enable a real-time implementation.

The graphical user visualization interface provides a minimal yet useful tool to interact with the composition, allowing one to monitor it's individual parts and change their course. However, there is room for further development of this aspect of the system. For instance, incorporating vibrating strings as an input for interaction to simulate the act of plucking strings in a tanpura might be a welcome addition.

This approach to a composition in the Indian classical music system was successful. It was initially assumed that the results of the analysis would require some smoothing to make them usable however the control signals produced by audio feature extraction proved to be very easy to work with and required no processing whatsoever. The frequencies of the prominent partials of the tanpura were of great consonance with the micro-tones used in the Indian classical system. The combination of tracking the time-varying frequencies and amplitudes of the partials paired with the modulo-based trigger system facilitated the creation of several unique and interesting melodic phrases. The work can be extended further by implementing timbral features from sounds with similarly complex harmonic structures in order to explore the generalisability of the algorithms described above. A composition created by the interaction with this system can be viewed here [20].

REFERENCES

[1] Kalyani Akant. 2019. Measuring Frequencies of Shrutis in Indian Classical Music. In *2019 9th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-19)*. 1–4. https://doi.org/10.1109/ICETET-SIP-1946815.2019.9092007
[2] Adam Alpern. 1995. Techniques for Algorithmic Compositionof MusicAdam.
[3] James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.* 19, 90 (1965), 297–301. https://doi.org/10.1090/s0025-5718-1965-0178586-1
[4] Cycling74. 2022. buffer . https://docs.cycling74.com/max8/refpages/buffer~. Accessed: 2022-05-11.
[5] Cycling74. 2022. groove . https://docs.cycling74.com/max8/refpages/groove~. Accessed: 2022-05-11.
[6] Cycling74. 2022. Max/MSP. https://cycling74.com/products/max. Accessed: 2022-05-11.
[7] Cycling74. 2022. metro. https://docs.cycling74.com/max8/refpages/metro. Accessed: 2022-05-11.

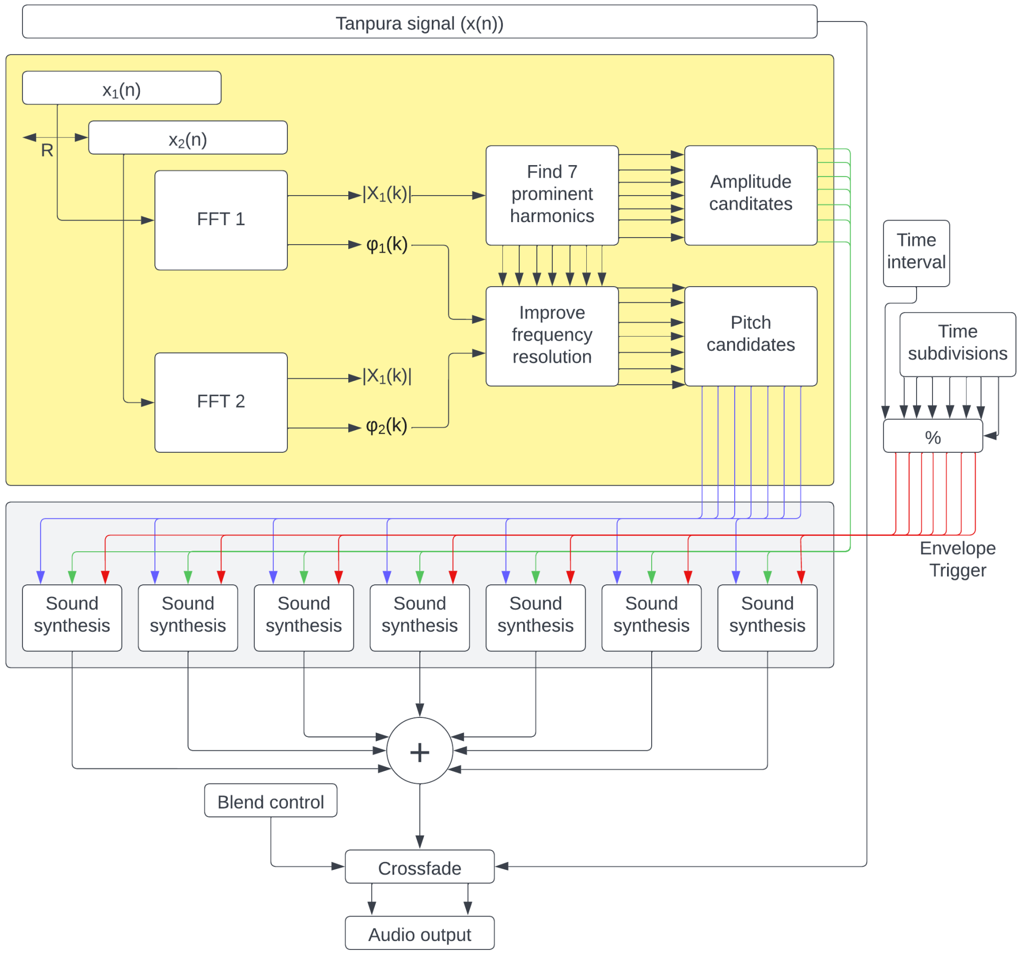

Figure 6: Block Diagram of entire system.

[8] Cycling74. 2022. svf . https://docs.cycling74.com/max8/refpages/svf~. Accessed: 2022-05-11.

[9] Cycling74. 2022. Variable-length Wavetable. https://docs.cycling74.com/max7/tutorials/07_samplingchapter04. Accessed: 2022-05-11.

[10] Cycling74. 2022. wave . https://docs.cycling74.com/max8/refpages/wave~. Accessed: 2022-05-11.

[11] Asoke Datta, Ranjan Sengupta, Kaushik Banerjee, and Dipak Ghosh. 2019. *Acoustic Analysis of Tanpura Signals: Indian Plucked String Instrument.* 107–119. https://doi.org/10.1007/978-981-13-2610-3_6

[12] Devashish Gosain, Soubhik Chakraborty, and Mohit Sajwan. 2018. Simulating Raga Notes with a Markov Chain of Order 1-2. *CoRR* abs/1808.01603 (2018). arXiv:1808.01603 http://arxiv.org/abs/1808.01603

[13] Matt Hoffman and Perry Cook. 2006. Feature-Based Synthesis: Mapping Acoustic and Perceptual Features onto Synthesis Parameters. (01 2006).

[14] Sylvain Le Groux and Paul FMJ Verschure. 2008. Perceptsynth: mapping perceptual musical features to sound synthesis parameters. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. 125–128. https://doi.org/10.1109/ICASSP.2008.4517562

[15] Thomas Lidy, Georg Pölzlbauer, and Andreas Rauber. 2005. Sound Re-synthesis from rhythm Pattern Features - audible insight into a Music Feature Extraction Process. In *ICMC*.

[16] Anastasia Natsiou and Sean O'Leary. 2022. Audio representations for deep learning in sound synthesis: A review. https://doi.org/10.48550/ARXIV.2201.02490

[17] Paritosh K. Pandya. 2005. Beyond Swayambhu Gandhar: an analysis of perceived tanpura notes. *J ITC Sangeet Res Acad* 19, 1 (Dec. 2005).

[18] Mrityunjay Kalita Soubhik Chakraborty and Neha Kumari. 2011. Semi-Natural Composition: An Experiment with North Indian Ragas. *International Journal of Computational Cognition* 9, 2 (2011).

[19] V. Verfaille, D. Arfib, F. Keiler, A. von dem Knesebeck, and U. Zölzer. 2011. *Adaptive Digital Audio Effects.* John Wiley Sons, Ltd, Chapter 9, 321–391. https://doi.org/10.1002/9781119991298.ch9 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119991298.ch9

[20] xxx. 2022. Tanpura se performance. https://ufile.io/l79cvidx. Accessed: 2022-05-11.