Digital sound synthesis of the Tongali and Kolitongimplemented as a virtual instrument plugin

Undergraduate Student Project

by

Christian April P. Bagaforo
2014-43147
B.S. Electronics and Communications Engineering

Jhaezminne S. Gayo
2015-13921
B.S. Electronics and Communications Engineering

Advisers:

Franz A. de Leon, Ph.D
Carl Timothy Tolentino
Crisron Rudolf G. Lucas

University of the Philippines, Diliman
May 2020

Abstract

Digital sound synthesis of the Tongali and Kolitongimplemented as a virtual instrument plugin

The musical instruments of the Kalinga ethnolingustic group are most representative of those found in the Cordillera region. Development of accessible virtual emulations of these instruments is necessary to address the declining interest in the traditional music of the region and aid its preservation. This study aims to synthesize the *tongali* and *kolitong* for use in a virtual instrument plugin. Musical signal analysis methods were used to extract time and frequency features of the instruments. The Modal Distribution (MD) was used to extract the amplitude and frequency estimates. Synthesis was implemented in MATLAB using the extracted features of the MD through Sum of Sinusoids (SoS) synthesis for both the *tongali* and *kolitong*. The output audio files were used to develop a virtual instrument plugin using HISE, implemented in the VST2 standard. A 2IFC listening test was conducted to to determine the perceptual similarity of the synthesized and recorded instrument sounds and calculate the sensitivity index d'. The tongali and kolitong had a d' value of 7.0711 x 10^{-6} and -3.5355 x 10^{-5} respectively, proving that the synthesized sound is aurally indistinguishable from the recorded sound.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Philippines, as a country of many ethnolinguistic groups, offers a wide variety of indigenous music. Historically, music has been an integral part Filipino life, playing a large part in various rituals and ceremonies. Together with its instruments, the music in a certain region defines the historical identity of its ethnolinguistic groups.

Culture, including music tradition, is known to be dynamic with respect to changes in society. Today, many parts of the Philippines are influenced by modern western music, causing the appreciation of indigenous music to decline, and the number of its practitioners to decrease, especially in the lowlands of Luzon. Also, accessibility to handcrafted indigenous instruments, especially bamboo instruments, is limited due to a decrease in instrument manufacturers as well as the inherent durability of bamboo as a material, which is affected by abrupt changes in weather.

Despite this, there are still some regions in the Philippines that practice traditional music and produce indigenous instruments, one of which is the Cordillera region. The region is home to eight ethnolinguistic groups - namely the Banglao, Bontok, Ibaloi, Ifugao, Ilongot, Isneg, Kalinga, and the Kankana-ey.

The region, and the Kalinga in particular, is involved in many music-related activities throughout the country and the world. For instance, Benicio Sokkong is a Kalinga cultural practicioner who teaches and performs indigenous Kalinga music around the world [4]. This suggests that there are active networks of individuals who teach regular workshops on the music of the Cordillera region.

A compilation of the instruments of the Cordillera region by Maceda [5] shows that certain types of instruments are represented in multiple groups and share similar characteristics. Among the eight ethnolinguistic groups, the Kalinga has the highest number of representative instruments.

1.1 Background of the Project

To address the issue of declining interest in the traditional music of the Philippines and aid in its preservation, there is a need for freely available and accessible virtual replications of the sounds of Philippine indigenous musical instruments. These virtual instruments would allow more musicians to incorporate their sounds in musical compositions and arrangements. For this reason, this project aims to implement a virtual instrument plugin that can emulate the sound and expression of the bamboo instruments of the Cordillera region, particularly of the Kalinga.

1.1.1 The Kalinga Group of Instruments

The Kalinga use bamboo instruments of three different types, namely aerophones, chordophones and idiophones. Aerophones are instruments that produce sound when blown due to the vibration of air. Chordophones are instruments that produce sound through the vibration of plucked strings. Idiophones are instruments played by hitting or striking with a mallet or against a surface such as the palm of the hand. In this project, only one of the region's aerophones and chordophones will be synthesized. The selected instrument types and the metric in choosing the instruments to be synthesized will be discussed in the following subsections.

1.1.1.1 Aerophones

There are five types of aerophones used by the Kalinga: the noseflute (*tongali*), plugflute (*olimong*), lip-valley notch flute (*paldong*), panpipes (*saggeypo*), and reed pipe (*patottot*). The nose flute is the most common type of aerophone used by the ethnolinguistic groups in the Cordillera region[5], with different variants used by the Banglao, Bontok, Ibaloi, Ifugao, Isneg and the Kankana-ey. Because of this, the *tongali* shown in Figure 1.1 is selected as one of the instruments to be synthesized in the project.

Figure 1.1: Image of the *tongali* [1]

The *tongali* is a bamboo nose flute that has a blowing hole through its node, which is placed at the player's nostril, while the other nostril is plugged with a leaf. The *tongali* usually has three finger holes and a thumb hole use to change the pitch of the played note.

1.1.1.2 Chordophones

The chordophones used by the Kalinga are primarily zithers, which are instruments with strings that are positioned parallel to the length of its body. The body of the zither serves as a resonator and comes in many shapes and sizes [6]. Zithers are classified by their body types, namely: the board zither, the half-tube zither, and the tube zither. The Kalinga primarily use tube zithers, of which there are two types: the paired-string or parallel-stringed tube zither (*tambi*) and the polychordal tube zither (*kolitong*). The *kolitong*, shown in Figure 1.2, is selected as the chordophone to be synthesized in the project due to its representation of other tube zithers used in the Cordillera region, and because of its ability to play melodies made up of interlocking pitches.

Figure 1.2: Image of the *kolitong* [2]

The *kolitong* has six strings, two of which are found on the front side, which are plucked by the thumb. The four other strings are found on the dorsal side, which are plucked by the index and middle finger.

1.1.2 Software Synthesizers

A software synthesizer uses digital audio signal processing techniques to generate musical sounds. Software synthesizers are typically used to generate temporally and spectrally complex timbres otherwise unattainable by physical instruments. However, they can also be used to approximate the sound of existing instruments. This allows musicians to arrange or perform musical pieces without having to possess or have the ability to play the instruments involved.

Software synthesizers can exist as either a standalone application or a plugin that utilizes standard software interfaces for use with compliant host applications. These can then receive musical performance information in existing encoding standards from other hardware or software.

1.1.2.1 Musical Instrument Digital Interface (MIDI)

The Musical Instrument Digital Interface (MIDI) protocol is a commonly used digital encoding standard that represents musical performance information as non-synchronous real-time instructions called *MIDI messages* that contain playing parameters per note such as a note number indicating the specific key pressed, note on/off (whether a note is being pressed or released), note velocity (how hard the note is pressed) and note aftertouch (maintained pressure applied to a key after being pressed).

MIDI messages can be received by a MIDI-compliant external hardware or software sound synthesizer in order to reproduce the performance. A hardware MIDI controller can be used to provide MIDI messages in order to play and manipulate the parameters of a sound synthesizer in real-time.

1.1.2.2 Virtual Studio Technology (VST)

Virtual Studio Technology (VST) is a widely-adopted software standard for digital audio plugins developed by Steinberg Media Technologies AG. The VST standard allows for plugins to be used with any compatible host software. VST plugins may take the form of audio effects plugins or VST instruments (VSTi). VSTi instruments are capable of being played using MIDI data provided by the host software.

1.2 Project Overview

The block diagram for the project flow is shown in Figure 1.3.

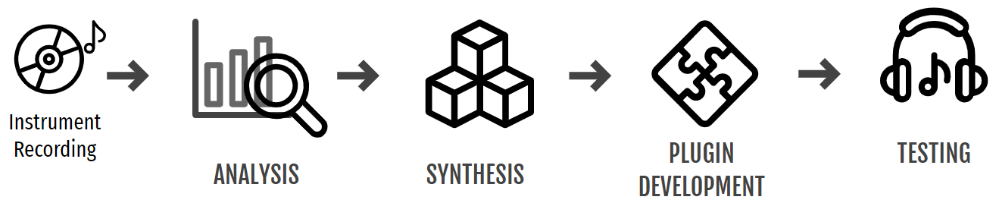


Figure 1.3: Project flow

1.3 Documentation Flow and Organization

Chapter 2 discusses the analysis and synthesis techniques to be used in the study, as well as their use in previous studies conducted on Philippine indigenous instruments.

Chapter 3 describes the problem statement, objectives, scope and limitations of the project.

Chapter 4 describes the techniques for analysis and synthesis of the instruments, and the development resources and procedures used to build the virtual instrument plugin. Subjective and objective testing methods are also described.

Chapter 5 presents the results of the project.

Chapter 6 presents the conclusion and recommendations of the project.

Chapter 2

Related Work

Previous studies were able to analyze and synthesize other Philippine indigenous instruments of different types using different analysis and synthesis techniques. A number of these studies also implemented synthesis in virtual instruments with varying methodologies and results.

In this chapter, various musical signal analysis and synthesis methods used in past studies on Philippine indigenous instruments will be discussed.

2.1 Musical Signal Analysis Methods

Musical signal analysis is the use of signal processing techniques on musical sounds in order to quantitatively describe their perceptual features such as loudness, timbre and pitch.

Loudness is the perceived intensity or volume of a sound and is directly related to the amplitude of the signal. Pitch refers to the "highness" or "lowness" of a sound and is determined by the frequency components present in a signal. Timbre refers to the quality or texture of sound which distinguishes one sound from another given the same pitch and loudness. It is determined by both the signal amplitude and frequency components, as well as changes in both over time. Timbral features can be correlated with audio semantics such as the roughness, brightness, and fullness of the signal [7].

Analysis of a musical signal can be performed in the time and frequency domains separately, or simultaneously in both domains using time-frequency representations of the signal. Specific analysis methods will be discussed in the following subsections.

2.1.1 Time Domain Analysis

Musical signals are typically described in the time domain using an amplitude envelope. It describes the contours of the peak amplitude of the signal over time. An *Attack, Decay, Sustain, Release (ADSR)* envelope, shown in Figure 2.1, can be used to describe the amplitude envelope of any signal. Described in terms of a note being played, an ADSR envelope consists of four sections:

1. *Attack*, the shape and duration of the envelope from the moment the note is initially played to when it reaches peak amplitude;

2. *Decay*, the shape and duration of the envelope between the peak amplitude and sustain amplitude;

3. *Sustain*, the amplitude of the signal as the note is held;

4. *Release*, the shape and duration of the envelope over which the note reaches zero amplitude from the sustain amplitude after the note is released.

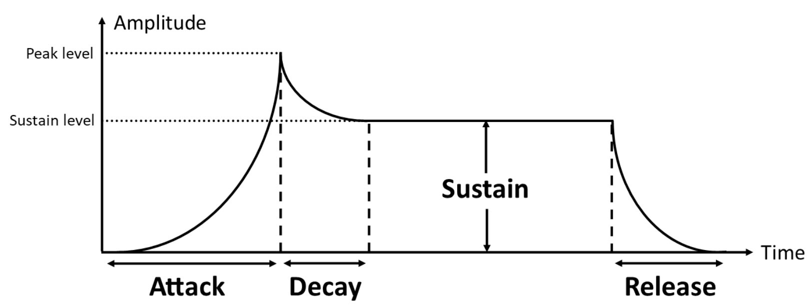


Figure 2.1: Attack, Decay, Sustain, Release (ADSR) envelope

The shape and duration of the attack slope of an instrument is considered as a feature of timbre [3]. For example, the attack slopes of six different instruments - namely the piano, flute, guitar, violin, and the trumpet - are shown in Figure 2.2.

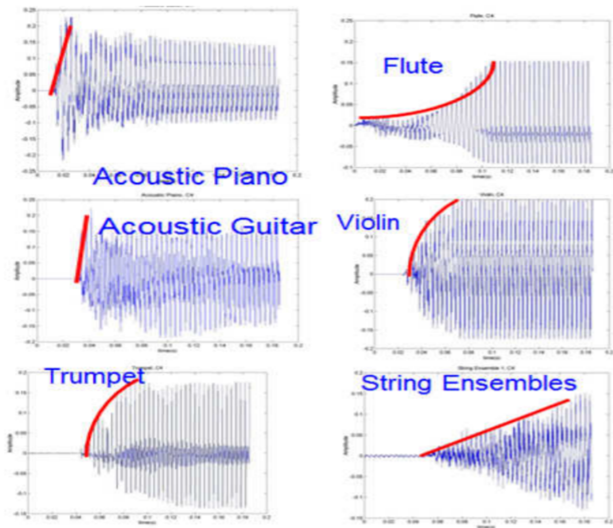


Figure 2.2: Attack slopes of six different instruments [3]

In a study conducted by Macalisang and Reyes [8], the time-frequency distribution of the Angklung was analyzed in order to determine the ADSR envelopes of the spectral components of the instrument. It was then determined that the properties of the bamboo material used in the Angklung is related to its ADSR envelope.

Some signals may not have the basic ADSR envelope such as the Diwas which only contains an attack, sustain and release. For the kolitong, an amplitude modulation is present in the time domain plot. Additionally, the sound made from touching the string while playing is evident. [9]

2.1.2 Frequency Domain Analysis

The timbre of a sound is determined in part by its frequency domain characteristics. Figure 2.3 shows a comparison of the magnitude spectra of timbrally dissimilar instruments, namely the violin, trumpet, clarinet, and the french horn.

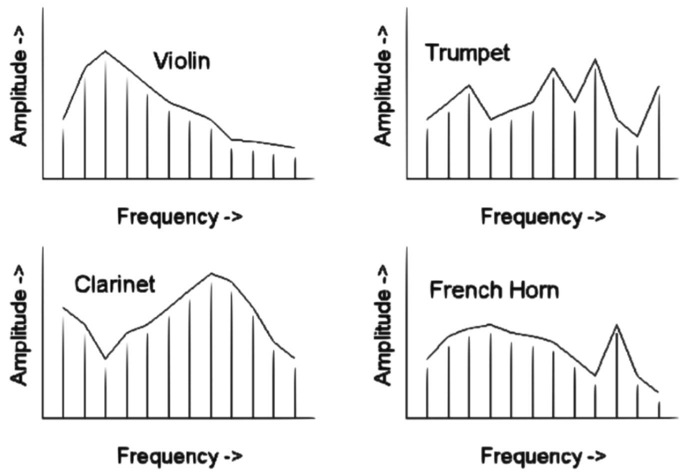


Figure 2.3: Magnitude spectra of different musical instruments [3]

Frequency domain analysis of a musical signal can be used to determine the composition and relative amplitudes of its frequency components. The Discrete Fourier Transform (DFT) of a periodically sampled signal is a vector of the same length as the signal, containing samples of its Discrete-time Fourier Transform (DTFT). The magnitude and phase spectra of the signal can then be computed from the DFT.

2.1.3 Time-frequency Analysis

Since frequency components of a musical signal typically vary over the time domain and frequency domain, using these techniques independently may not provide enough information to adequately characterize the signal. A more useful approach is to analyze a sound signal is by applying simultaneous analysis in the time and frequency domain.

Time-frequency analysis techniques can be used to determine the amplitude and frequency characteristics of the partials of a signal over time. The time-frequency analysis shows the energy distribution of the partials of the signal across time.

2.1.3.1 Short-time Fourier Transform (STFT)

The short-time Fourier transform (STFT) is the most commonly used tool for analyzing non-periodic signals. It can be used to generate the *spectrogram* of a signal, which visually describes the time-varying energy across different frequency bands over short-time analysis windows [10]. Given a discrete-time signal x obtained by uniform sampling of a waveform at a given sampling rate of F_s Hz, the STFT is obtained by multiplying x to an N-point tapered window w with an overlap of half a window length

$$X(t,k) = \sum_{n=0}^{N-1} w(n)x(n + t \cdot \frac{N}{2})^{\frac{-j2\pi kn}{N}} \tag{2.1}$$

With t ϵ [0: T-1] and k ϵ [0:K] where T is the number of frames and K = N/2 which is the index of the last unique frequency value. X(t,k) refers to the window beginning at time

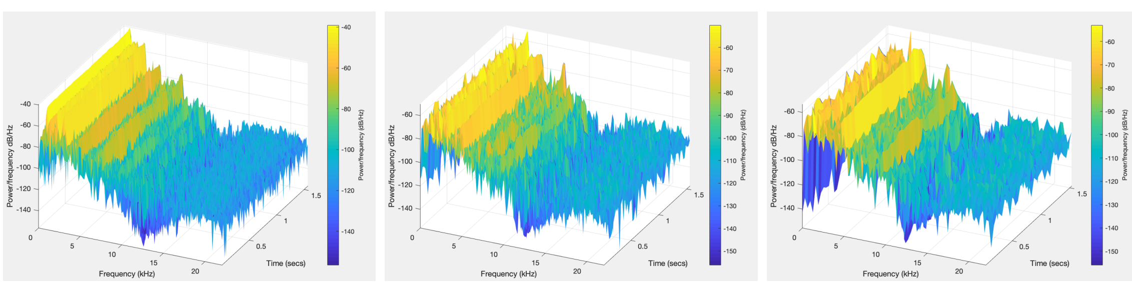
$$time = t \cdot \frac{N}{2F_s} \tag{2.2}$$

and frequency

$$f_{coeff}(k) = \frac{k}{N} \cdot F_s \tag{2.3}$$

in Hertz (Hz).

Observing the formulas for STFT, it can be observed that there is a trade-off in terms of time and frequency resolution in analyzing signals using this tool. In order to achieve a better resolution in the time domain, a narrow window is required which results in poor resolution in the frequency domain. A wide window provides a better resolution in the frequency domain but gives the time domain a lower resolution. [11].



(a) 1024 samples (b) 2048 samples (c) 4096 samples

Figure 2.4: A trade-off of STFT: Using a smaller window size such as 2.4a increases time resolution but decreases frequency resolution. Using a large window such as 2.4c shows the opposite effect.

In order to resolve the mentioned trade-off, a TFD representation based on Cohen's class has been developed [12]. This representation, discussed in the next section, has made it possible to treat the frequency-resolution and time-resolution problems separately.

2.1.3.2 Modal Distribution Analysis (MDA)

In extracting the time-frequency characteristics of a signal, a more useful approach in analyzing the physics behind musical signal sounds has been proven by Guevara [13]. This approach is by analyzing the Modal Distribution (MD), which is a time-frequency distribution developed by Pielemeier and Wakefield [14].

The MD is a time-frequency distribution (TFD) defined as a member of Cohen's [15] class of time-distribution functions consisting of all linear transformations of the Wigner time-frequency distribution of the form:

$$C(t,\omega;\varphi) = \int_{-\infty}^{\inf}\int_{-\infty}^{\inf} W_s(\tau,\xi)\varphi(t-\tau,\omega-\xi;t,\omega)d\tau d\xi \tag{2.4}$$

where $\varphi(\tau,\xi,t,\omega)$ is the *kernel* that specifies the linear transformation. The MD kernel contains low-pass filter functions $h_{LP}(\tau)$ and $G_{LP}(\xi)$ for suppression of cross terms of the Wigner distribution (WD) in time and in cases of frequency modulation respectively, and is given by:

$$\varphi_M(\tau,\xi) = h_{LP}(\tau)G_{LP}(\xi) \tag{2.5}$$

The MD $M(t,\omega)$ is defined as follows:

$$M(t,\omega) = C(t,\omega;\varphi_M) = \int_{-\infty}^{\inf}\int_{-\infty}^{\inf} W_s(\tau,\xi)h_{LP}\times(\tau)G_{LP}(\xi) \tag{2.6}$$

The MD minimizes the smoothing of the time and frequency characteristics of an analyzed signal present in a window-based TFD such as the STFT, while also minimizing the contribution of cross terms from partials in a signal on the frequency and magnitude parameters of unrelated partials that is emphasized in other TFD's such as the Wigner distribution.

Unlike other TFDs that require a conversion of coefficients to units of energy by computing the squared magnitude, the coefficients of the MD are already expressed in units of energy. This shows that the MD is more useful for characterizing the features of a sound signal [16].

Modal Distribution Analysis (MDA) can be used to extract and visualize information about the partials of a signal. It estimates the form of the the amplitude and instantaneous frequency functions of each partial over time. The MD assumes a general signal model given by:

$$\overline{s}(t) = \sum_{l=1}^{M} \overline{B}_l(t)e^{j\phi_l(t)} \tag{2.7}$$

where $\overline{s}(t)$ can contain one or more instruments, M is the number of positive partial frequencies from all instruments present. Each partial is represented as a complex exponential with time-varying amplitude $\overline{B}_l(t)$ and phase $\phi_l(t)$. Additionally, instantaneous frequency $\omega_l^i(t)$ of the lth partial is given by:

$$\omega_l^i(t) = \frac{d}{dt}\phi_l(t) \tag{2.8}$$

A discrete-time implementation of the MD described by Pielemeier and Wakefield [14] was implemented by Guevara [13] and used to perform MDA in order to extract the time and frequency estimates of the partials of piano tones over time. The estimates were used to synthesize piano using Sum of Sinusoids (SoS) synthesis. The methodology and software implemented by Guevara was used to analyze and synthesize the Kulintang [17], [18]. The total number of partials of the Kulintang as well as their frequency and amplitude estimates were extracted from the generated MD's using MATLAB scripts.

In the `Modal Distribution Program` implemented by Guevara [13], the following parameters are specified: transform length, N, cutoff frequency of the cross term filter (CTF), f_c, display interval, t_d, and display length, t_L. Given the sampling frequency f_s, number of frequency bins and time samples are obtained using the parameters:

$$N_c = \frac{4f_s}{f_c} \tag{2.9}$$

$$N_d = t_d f_s \tag{2.10}$$

$$N_L = t_L f_s \tag{2.11}$$

The transform length determines the frequency resolution[1] d_f described in Equation 2.12. A higher value of N gives a smaller width for the frequency bin.

$$d_f = \frac{f_s}{2N} \tag{2.12}$$

The accuracy in computing the amplitude and frequency estimates depends on the number of bins over which to average. In comparing the amplitude estimates for different number of estimation bins, it has been shown that a decrease in number gives an increase in error. When there are at least 4 estimation bins, the maximum error is less than 0.9 cents [13]. Therefore a higher transform length gives a better frequency resolution. Partial frequencies are not necessarily integer multiples of d_f.

[1] The frequency resolution refers to the width of the frequency bin of the distribution.

Since the PW smooths the partials by about N_{MLW} along the frequency axis, NF_{min} should be chosen such that

$$NF_{min} > N_{MLW} \tag{2.13}$$

where NF_{min} is the minimum number of frequency bins between any two partials and N_{MLW} is the number of bins within the main lobe width (MLW).

For instruments that produce sound with a well-defined pitch, the minimum distance between any two partials is approximately the fundamental. Thus, f_c can initially be slightly lower than the fundamental to compensate for the filter roll-off. For the optimum resolution, f_c and N are to be adjusted.

Since the CTF is a lowpass filter (LPF) along the time axis, we could assume that the highest frequency component present in the MD is less than f_c. Thus, based on the Sampling Theorem, it is required to sample along the time axis such that t_d is less than half of f_c. N_L, which is the total number of time points will be computed as:

$$N_L = \frac{t_L}{t_d} \tag{2.14}$$

The three axes of the MD of a musical signal are time, frequency, and magnitude as shown in Figure 2.5. Due to the reduction of cross terms from the WD, the MD of the signal now has no spectral content at negative frequencies. Thus, only the positive terms in the magnitude have physical interpretation. Each time slice represents the energy distribution with respect to frequency. Meanwhile, each frequency slice represents the variation of every with respect to time. Partials are seen as ridges that run along the time axis. Each partial has a width which is defined by the PW and variations its amplitude. The partial width increases at onsets and at points in time where there is a large rate of change in amplitude.

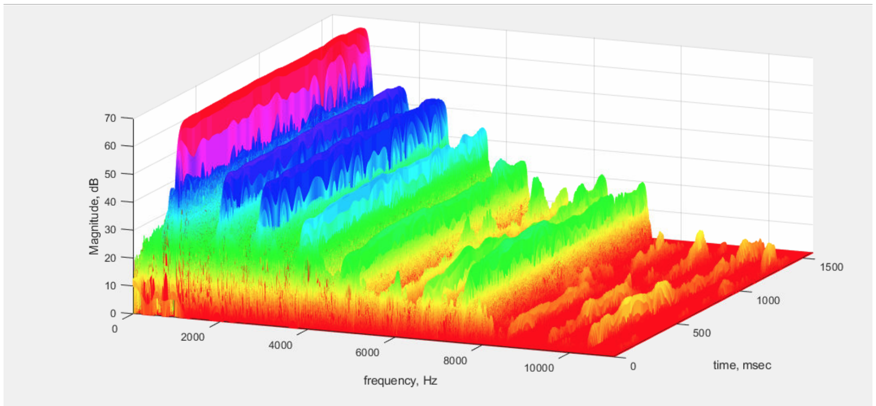


Figure 2.5: Modal Distribution of a Kolitong Note

2.2 Instrument Synthesis

Features of musical instrument sounds extracted using musical signal analysis techniques can be used to synthesize signals that can approximate the original sounds. Synthesis methods can either emulate the actual mechanisms for sound generation of an instrument using *physical modeling*, or approximate the time-domain behaviour and frequency characteristics of the instrument directly using *spectral modeling* techniques.

2.2.1 Physical Modeling

Physical modeling techniques synthesize the sound of an instrument by simulating the sound production events of the physical instrument [19]. It is acheived by either using existing formulas and differential equations derived from the physical structure of the instrument, or by designing a discrete-time model of the instrument.

Macalisang and Reyes [8] used physical modeling to synthesize the Angklung for the iPad and desktop platforms.

2.2.2 Spectral Modeling

Spectral modeling techniques approximate the sound of an instrument by directly synthesizing the frequency components and time domain behaviour of the synthesized signal through the superposition or manipulation of spectrally simple and temporally periodic signals [19]. Features extracted from the actual instrument sound are used to determine the parameters for synthesis.

2.2.2.1 Additive Synthesis

Additive synthesis is a sound synthesis technique wherein complex signals are described as the addition of simpler signals, typically sinusoids, with independent amplitudes, frequencies and phases [20].

The continuous-time function $f(t)$ of a signal can be generally described as the summation of K sinusoidal components:

$$f(t) = \sum_{k=1}^{K} \left[A_k[n] \cdot \sin(\phi_k(t) + 2\pi \int_0^t f_k(\tau)d\tau) \right] \tag{2.15}$$

with time-varying functions per k_th component describing amplitude $A_k(t)$, frequency $f_k(t)$ and initial phase angle $\phi_k(t)$.

Guevara used sum of sinusoids (SoS) to synthesize piano tones [13]. In theory, additive synthesis can synthesize any sound given an infinite number of sinusoid oscillators [21]. This, however, becomes impractical for signals that have a wide range of frequencies such as white and unpitched instruments.

2.2.2.2 Subtractive Synthesis

Subtractive synthesis is a sound synthesis technique produced by filtering a harmonic-rich waveform in order to attenuate or amplify certain frequency components [22]. The original waveform contains energies of the desired frequency components. The simplest case of subtractive synthesis involves the use of filters with a constant shape and cutoff frequency.

Chapter 3

Problem Statement and Objectives

The continued practice of the indigenous music of the Cordillera region is limited by the accessibility of handcrafted instruments and the lack of skilled practitioners. Because of this, there is a need for freely available and accessible virtual emulations that can replicate the full range of sound and expression of the instruments in the hands of a skilled player.

The objectives of this project are as follows:

- Synthesize the Kalinga *tongali* and *kolitong* using features extracted from actual recordings of the instruments. Synthesized sounds must be timbrally similar to the original instrument based on subjective and objective testing metrics.

- Develop a virtual instrument plugin for the twoinstruments using open-source and freely-licensed development resources, and widely-adopted standards.

The focus of the project will be limited to the Kalinga *tongali* and *kolitong*. The developed plugin will be built and tested as a VSTi using the VST2 standard for macOS systems.

Chapter 4

Methodology

Recordings of the *tongali* and *kolitong* are provided by the University of the Philippines Digital Signal Processing Laboratory (UP DSP Lab). These recordings were analyzed to extract the features required to synthesize the instrument sounds. Methods for analysis include amplitude envelope extraction using Hilbert transform, Fourier analysis and modal distribution analysis [13][17]. Sinusoidal additive synthesis was used to synthesize the *tongali* and *kolitong*. Plugin development was done on HISE using the VST2 standard.

4.1 Preparation of Data

Recordings of the different sounds produced by the *tongali* and *kolitong* were provided by the UP DSP Lab. The audio recordings were made in the pseudo-anechoic chamber of the UP DSP Lab and sampled to 44.1 kHz in stereo. The recordings for the tongali were segmented into each played note using Audacity; the kolitong was already segmented. Both were normalized using FL studio. Noise reduction was also done in FL studio. The recordings contain single played notes on each string of the *kolitong* and single continuously blown notes on the *tongali*.

4.2 Analysis of Recordings

The recordings of the different instruments were analyzed using MATLAB and the Modal Distribution Program by Guevara [13] to extract the features used to synthesize the instrument sounds.

As discussed in section 2.1.3.2, the fundamental frequency of the note was needed as it affects the parameter f_c. The fast fourier transform (FFT) was done in order to obtain the fundamental frequency of the note. MDA was then implemented in order to extract the amplitude and frequency estimates of each partial present in the signal. For the *kolitong*, the amplitude envelope was extracted in order to be convolved with the pre-synthesized signal[1] for onset compensation. The flow of the analysis of both instruments is shown in Figure 4.1.

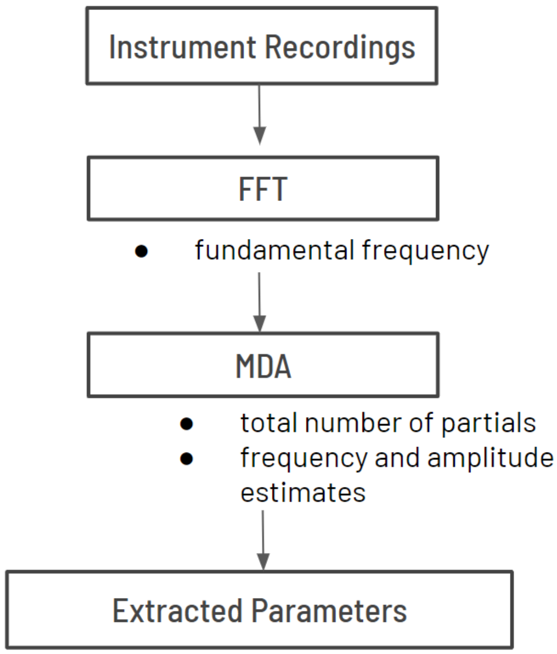
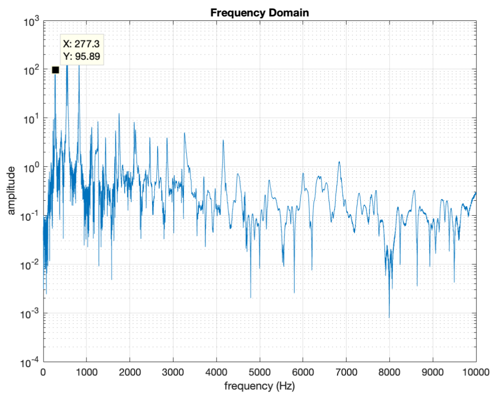


Figure 4.1: Analysis flowchart

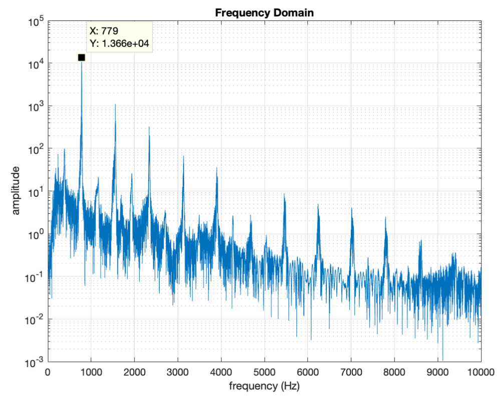
4.2.1 Fourier Analysis

Fourier analysis was used in this project to compare the frequencies and amplitudes of the partials of each instrument. The Fast Fourier Transform (FFT) implemented in MATLAB using `ampfreq.m` computes the Discrete Fourier Transform (DFT) of individual played notes of the original and synthesized signals. The magnitude spectrum of the note is shown in Figure 4.2a from the magnitude of the DFT. The frequencies and amplitudes of the partials were determined based on the locations and peaks present in the magnitude spectrum. For determining f_c, the minimum distance of any two partials Δf_{min} needs to be known. Also, the fundamental frequency of each original and synthesized signal of the *tongali* and *kolitong* were obtained from this spectrum for the computation of their difference in cents.

[1]The pre-synthesized signal refers to the synthesized output from the obtained partial estimates in the MD. Due to time smoothing at the onset, compensation is needed in order to characterize the attack slope.



(a) 1st String of the Kolitong



(b) 10th Note of the Tongali

Figure 4.2: Magnitude Spectrum of the Kolitong and Tongali with their respective fundamental frequencies

4.2.2 Modal Distribution Analysis

Modal distribution analysis was used to determine amplitude and frequency estimates of each instrument. The `Modal Distribution Program` created by Guevara [13] was used to retrieve the modal distribution of the signals. The inputs of this program are the audio signal `.wav` files and the modal distribution parameter settings. The output is the modal distribution `.bin` file.

The following parameters shown in Table 4.1 were applied to all sound samples. The cross term cutoff frequency f_c differs for the *tongali* and *kolitong*.

Parameter	Value
Transform Length	8192
Display Interval	default $(\frac{1}{2*f_c})$
Length of Display	*equal to Data Length*
Display Offset	0

Table 4.1: Modal Distribution Parameters

In the analysis of MD, choosing N and f_c is very crucial for the best resolution. The transform length N was set to 8192 which gives the best frequency bin resolution as discussed in 2.1.3.2. Aside from f_c, the rest of the parameters were set to default. In choosing the value for the CTF cutoff frequency, f_c must be less than Δf_{min}. Recall that Δf_{min} is the minimum distance of any two partials in the TF domain. Thus, for a signal with a well defined pitch such as the *tongali* and *kolitong*, the ideal f_c would be a value that is close to f_o. For all samples of both instruments, the set f_c was fundamental frequency f_o minus 5 Hz.

The MD of the signal can be plotted using `modplot.m`, a MATLAB function created by Guevara [13]. The function displays a user interface shown in Figure 4.3 which accepts the `.bin` file to be plotted. The sub window of `modplot.m` displays the MD plot with magnitude in dB as shown in Figure 4.4

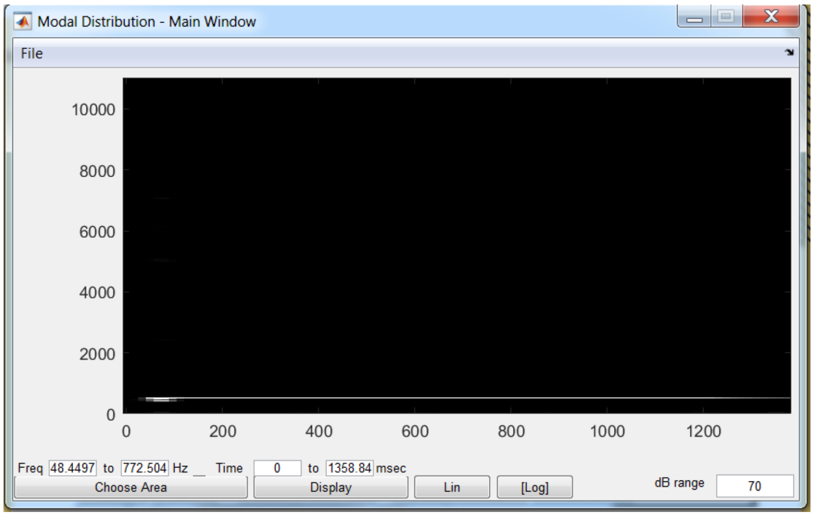


Figure 4.3: MD User Interface

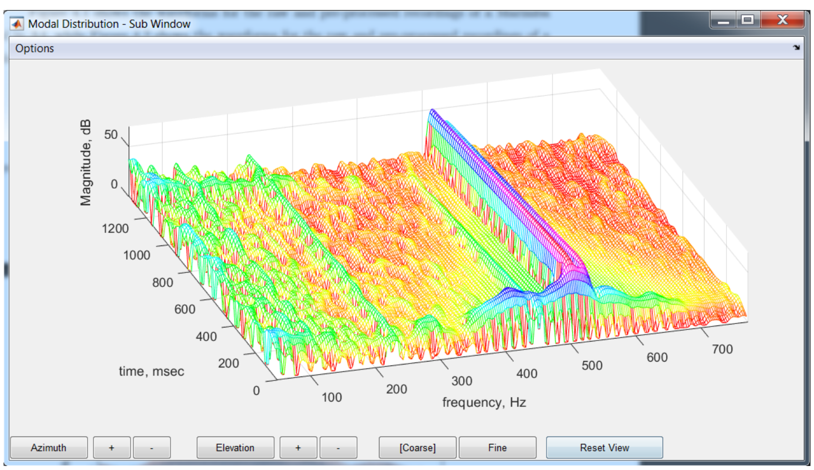


Figure 4.4: MD Plot Window

4.2.2.1 Obtaining Partials and Frequency and Amplitude Estimates

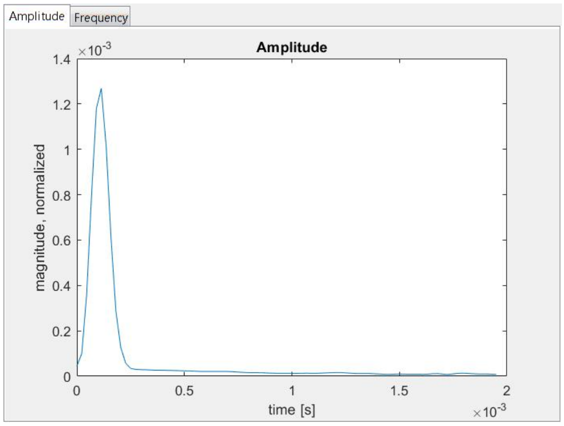
The outputs of the `Modal Distribution Program` were used as an input to MATLAB functions created by and adapted from Agsaway [17]. These functions obtained the total number of partials as well as the frequency and amplitude estimates of the *tongali* and *kolitong*.

The obtained MD from the sound samples are then put in the function `partial.m`. It is a function that determines the number of partials as well as the amplitude and frequency estimates of a signal. This contains `getboundaries.m` and `modalest.m` which are functions created by Agsaway [17]. The function `getboundaries.m` is responsible for acquiring the partials of the signal. The inputs to the function are the instrument recording and the threshold (in dB) which sets the maximum magnitude of the partials that the function will accept. The threshold value should be chosen carefully as it dictates which partials will be considered for synthesis. The threshold value is obtained by taking the logarithmic equivalent of the maximum value in the amplitude of the MD. The threshold values for both instruments are shown in the following table.

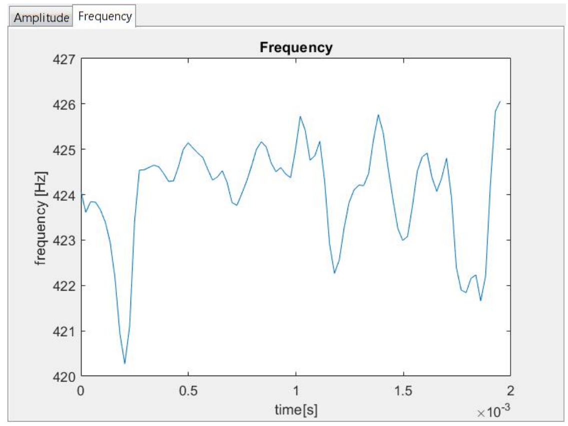
The function `modalest.m` extracted the amplitude and frequency estimates of the signal. The input to this function were the `.bin` file generated by the `Modal Distribution Program` and the start and end boundaries of the partial ridge obtained from `getboundaries.m` [17]. The number of partials from each of the analyzed notes of the different instruments are shown in the Table 4.2.

Instrument	Note	Threshold value (dB)	Number of Partials	Number of Time Slices
Tongali	1	70	157	822
	2	70	239	892
	3	80	164	1163
	4	90	109	1040
	5	80	118	1168
	6	90	202	1742
	7	80	98	1661
	8	90	184	1832
	9	90	130	2171
	10	90	188	2185
	11	100	150	2664
	12	90	196	3041
Kolitong	A1	50	169	1341
	A2	40	200	1613
	A6	40	41	2714
	B1	30	90	1821
	B5	40	262	3011
	C1	30	82	1506
	C4	35	79	2576
	C6	30	110	2933
	D1	37	218	1702
	D3	29	148	2610
	D4	28	223	2610
	D5	30	45	3011
	D6	33	60	3174
	E2	27	48	1884
	E4	30	110	1884
	F3	30	107	1884
	F4	26	147	2638
	F5	28	246	2638

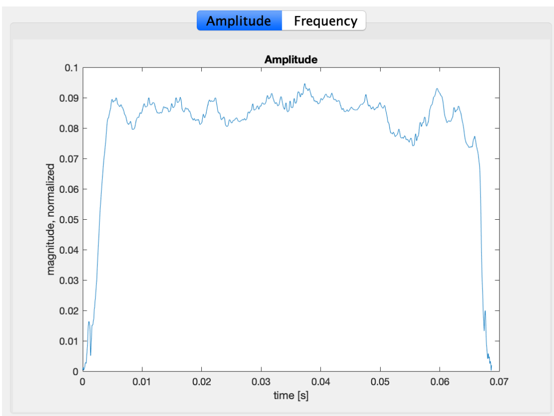
Table 4.2: Number of partials for each sound sample analyzed



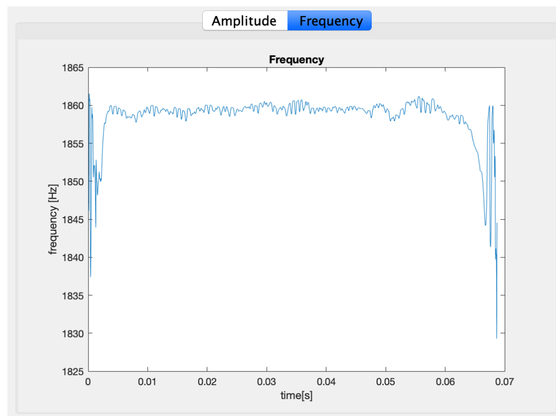
(a) Amplitude Estimate of a Kolitong Sample



(b) Frequency Estimate of a Kolitong Sample



(c) Amplitude Estimate of a Tongali Sample



(d) Frequency Estimate of a Tongali Sample

Figure 4.5: Amplitude and Frequency Estimates obtained from `partial.m`

Table 4.2 shows the number of partials from each of the analyzed notes of the tongali and kolitong. Figure 4.5 provides a sample of the amplitude and frequency estimates of each instrument.

4.3 Synthesis

The *tongali* and *kolitong* was synthesized using sinusoid additive synthesis with amplitude envelope, and was followed by subtractive synthesis for onset compensation. The synthesized outputs were in `.wav` format and was used to build a sample-based VSTi using HISE.

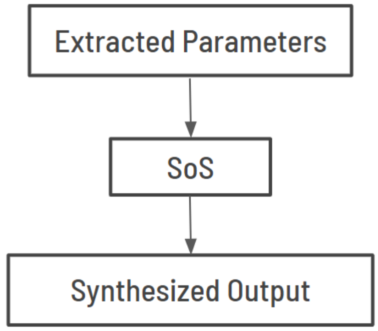


Figure 4.6: Synthesis flowchart

4.3.1 Sinusoid Additive Synthesis with Amplitude Envelope

The obtained partials of the *tongali* and *kolitong* from `Modal Distribution Analysis` were used in a sum of sinusoids (SoS) synthesis to create the synthesized signal. The synthesis algorithm was originally planned to be developed in MATLAB and implemented in Faust to develop a plugin using the JUCE framework. However, it was decided that all the synthesis will be performed solely in MATLAB.

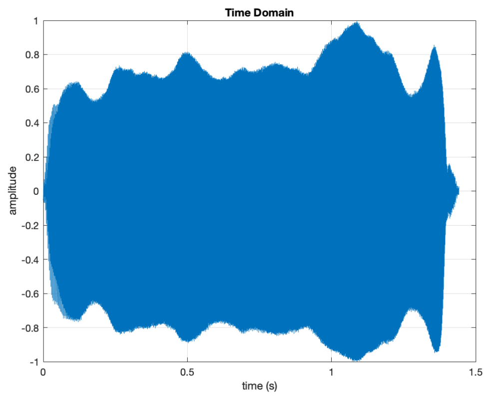
The synthesis algorithm follows the formula:

$$s(n) = \sum_{m=1}^{M} p_m(n) \tag{4.1}$$

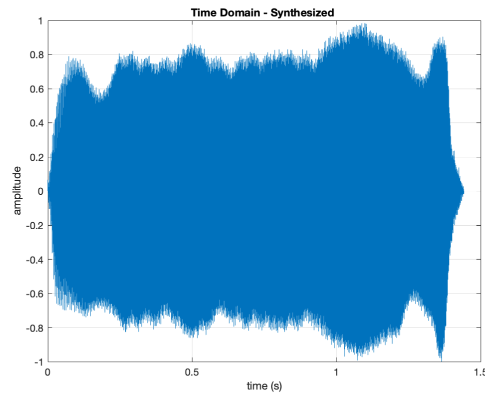
where $s(n)$ is the synthesized signal, M is the total number of partials from the original signal, and $p_m(n)$ is the m-th partial [17].

The function `synthesize.m` was created to synthesize the tongali and kolitong, utilizing previous work from Agsaway [17]. The function accepts a `.bin` file and asks the user for the dB threshold and cutoff frequency of the signal to be synthesized. It adds the partials containing the amplitude and frequency estimates and outputs a `.wav` file of the synthesized signal.

The output developed in MATLAB was used in HISE to build a plugin using the JUCE framework.



(a) Original Signal



(b) Synthesized Signal

Figure 4.7: Synthesis of the 10th note of the tongali using `synthesize.m`

4.4 Plugin Design

The developed synthesis algorithms were supposed to be implemented in the Faust programming language and compiled as an audio plugin project implementing the JUCE framework for the Projucer IDE. A sample-based VSTi plug-in was developed using Hart Instruments Sampler Engine (HISE) instead. HISE is an open-source framework for building virtual instruments based on JUCE and outsources the compilation to the system compiler [23]. The plugin was developed primarily for macOS and Xcode was used to compile the plugin.

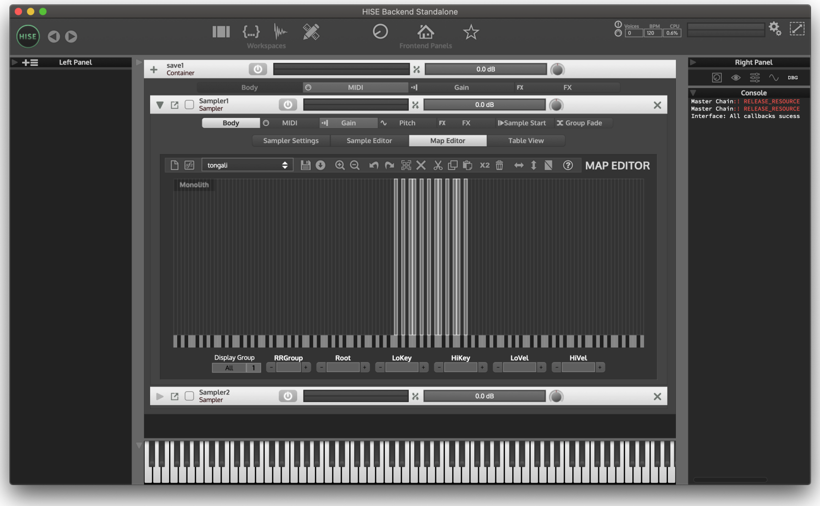


Figure 4.8: User interface of HISE. Here you can see the Musical Instrument Digital Interface (MIDI) mapping of the tongali

The synthesized signals were mapped to MIDI using the HISE interface. The tongali and kolitong were mapped from the lowest to the highest frequency starting at an octave lower than middle C (C3). The VSTi plugin was built using the VST2 standard contained in the latest distribution of the VST SDK supported by Steinberg, as well as any additional required files from version 3.6.10 (build 37, dated 11/06/2018) of the SDK.

A simple user interface (UI) was created which enables the user to choose between the two instruments.

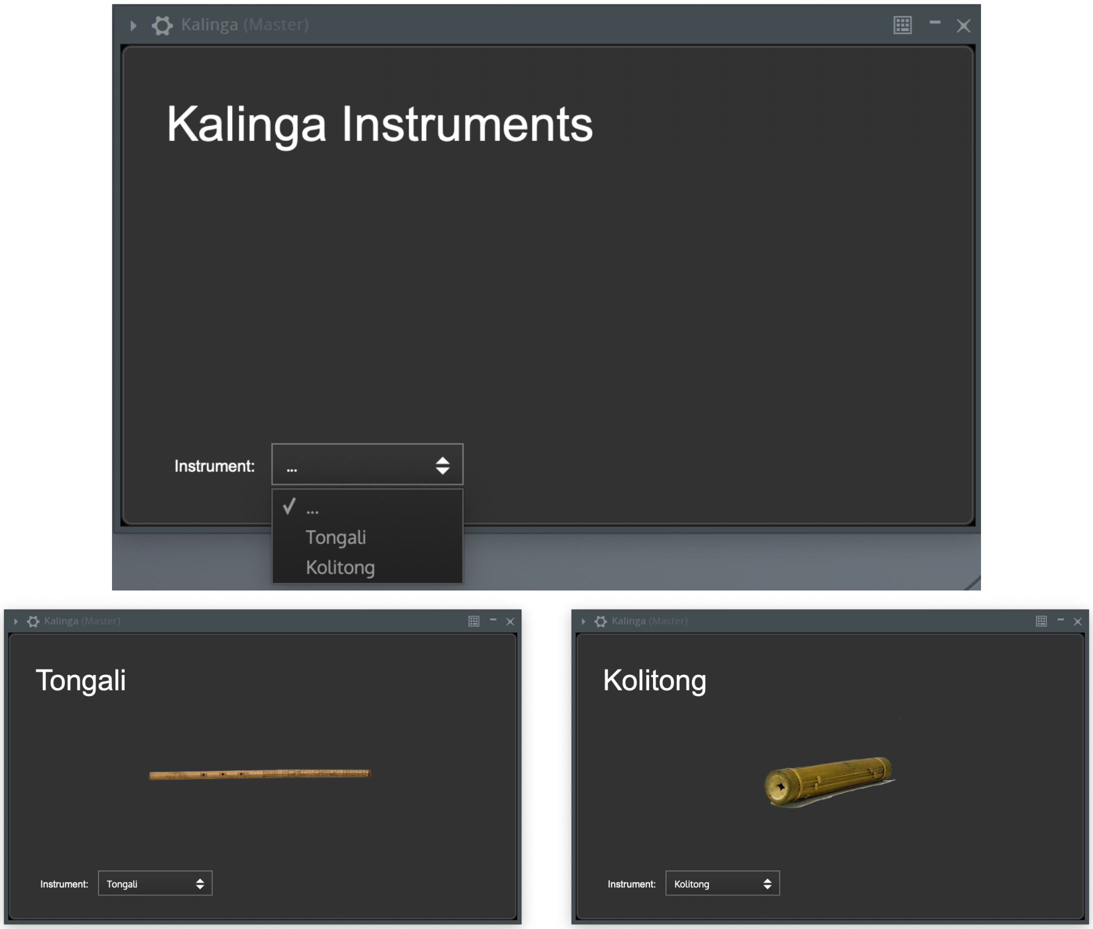


Figure 4.9: UI of the VSTi plugin for the synthesized Kalinga instruments

4.4.1 Playback

The playback is done by pressing the 12 white keys after C3 for the tongali, and the 18 white and black keys after C3 for the kolitong. The playback of the tongali starts once the key is pressed and stops at approximately 2 seconds, which is the average length of the note, or when the key is released. The playing note stops once another note is played, similar to the nature of a tongali which only plays one note at a time. The playback of the kolitong also starts once the key is pressed and stops the current playback and retriggers the signal once the key is repeated.

Chapter 5

Testing

For testing the similarity of the synthesized sounds, two forms of testing has been done:

1. A *subjective test*, employing a Two-interval Forced Choice listening test;
2. An *objective test*, using Fourier analysis to compare the frequency domain characteristics, zero crossing rate, attack time, and correlation of the real and synthesized instrument sounds.

5.1 Subjective test

The subjective test involved a listening test that was done online through Google Forms. A total of 59 volunteers answered the online form for testing.

5.1.1 Two-interval Forced Choice (2IFC)

Aural imperceptibility of synthesized and recorded instrument sounds were tested using a modified Two-interval Forced Choice listening test. The testing procedure was adapted from the tests conducted by Agsaway [17] and Guevara [13].

Each trial of the listening test was conducted in three steps:

1. The listener was presented with 3 sets of recorded and synthesized instrument sounds. There were 2 sets of kolitong sounds, each with 10 pairs, and 1 set of tongali sounds with 12 pairs.
2. The listener was presented with a pair of recorded and synthesized instrument sounds. The listener was required to choose which of the pair was the recorded sound. The order in which the two sounds were played was randomized with equal probability of occurring.

3. The listener was given visual feedback on their response. The listener receives a point for each time the recorded signal was correctly identified.

Instrument	Mean score	Total Score	Range		Accuracy (%)	Standard deviation
			Min	Max		
Tongali	5.1186	12	0	11	42.6554	2.2749
Kolitong	5.2712	10	2	10	52.7119	1.8367

Table 5.1: Listening test results

Table 5.1 summarizes the results of the listening test. The volunteers were able to identify the original signal 44.71% and 52.71% of the time for the tongali and kolitong, respectively. Table 5.2 breaks down the accuracy of the volunteers based on their musical background. Out of the 59 volunteers, 47 people stated having a background in music. 15 test subjects declared to have been studying or have studied in the College of Music in the University of the Philippines Diliman.

Instrument	Accuracy (%)	
	with musical background	without musical background
Tongali	43.8095	44.7059
Kolitong	52.8571	52.3529

Table 5.2: Listening test accuracy based on musical background

The results of the listening test were evaluated using the signal detection model as described in [24]. A *hit* is defined as correctly identifying the first sound played as the recorded sound given the order $\langle Recorded, Synthesized \rangle$. A *false-alarm* is defined as incorrectly identifying the first sound played as the recorded sound given the order $\langle Synthesized, Recorded \rangle$. Hit rate H and False-alarm rate F given h hits and f false-alarms for N trials were calculated as follows:

$$H = P(\text{ ``}Recorded\ first\text{''} \mid \langle Recorded, Synthesized \rangle\ ) = \frac{h}{N} \tag{5.1}$$

$$F = P(\text{ ``}Recorded\ first\text{''} \mid \langle Synthesized, Recorded \rangle\ ) = \frac{f}{N} \tag{5.2}$$

The sensitivity index d' and maximum proportion of correct responses corrected for bias $p(c)_{max}$ were computed as metrics to determine the imperceptibility of the recorded and synthesized instrument sounds from one another.

Given H and F, d' and $p(c)_{max}$ for a Two-interval Forced Choice test were computed as follows:

$$d' = \frac{1}{\sqrt{2}}\left[z(H) - z(F)\right] \tag{5.3}$$

$$p(c)_{max} = \Phi\left(\frac{d'}{\sqrt{2}}\right) \tag{5.4}$$

where z is a transformation to z-score in standard deviation units.

Instrument	d'	$p(c)_{max}$
Tongali	7.0711 x 10^{-6}	0.500002
Kolitong	-3.5355 x 10^{-5}	0.499990

Table 5.3: d' and $p(c)_{max}$ of the tongali and kolitong

The value of d' should be minimized in order to determine the extent to which the recorded and synthesized sounds are indistinguishable from one another. At $d' = 0$, it can be concluded that listeners are unable to discriminate between the recorded and synthesized instrument sounds. This corresponds to a $p(c)_max$ of 0.5, indicating that the subject could not have done better than if they had guessed on each trial [13].

The results show that the synthesized sounds are aurally imperceptible from the original. As a comparison, Agsaway recorded a d' average of -0.077089 for the agong, 0.026759 for the babandir, -0.185041 for the dabakan, 0.125559 for the gandingan, and 0.043725 for the kulintang in his study [17].

5.2 Objective test

Fourier analysis was used to compare the frequency spectra of the synthesized instrument and actual instrument. The Fast Fourier Transform (FFT) was implemented by using the *fft()* function in MATLAB to compute the Discrete Fourier Transform (DFT).

The resulting frequency spectra was evaluated using the cents notation as described by Benson in [25]. This is a logarithmic scale with 1200 cents per octave. A frequency ratio of *r:1* when converted to cents is equal to:

$$1200log_2(r) = \frac{1200ln(r)}{ln(2)}[25] \tag{5.5}$$

On a modern equal tempered scale, the value of a semitone is 100 cents and a whole tone is 200 cents [25]. Using equation 5.5, the cent would be equal to:

$$¢ = 1200log_2(\frac{f_2}{f_1}) \tag{5.6}$$

where f_2 is the given frequency and f_1 is the reference frequency. For the case of objective testing, f_2 was the fundamental frequency of the synthesized instrument sound and f_1 was the fundamental frequency of the recorded instrument sound. The value of ¢ should be minimized in order to determine that the fundamental frequency of the synthesized instrument and of the recorded instrument are comparable. At ¢ $= 0$, it can be concluded that the synthesized and recorded sound have the same fundamental frequency. At 7¢ or greater, the difference between the compared sounds is perceivable.

Acoustic features of the original and synthesized signals were also compared using the `mirtoolbox` in MATLAB. These features are the zero crossing rate, spectral centroid, and attack time. The zero crossing rate describes the frequency of the sign changes in the amplitude of the signal. It helps in estimating the dominant frequency and the spectral centroid of the signal. The spectral centroid describes the brightness of the sound in a signal. The attack time is a feature that greatly affects the timbre of a musical signal. It also describes the dynamic characteristic of the sound. Percent error of each of the acoustic features was computed as follows:

$$\%error = \frac{original - synthesized}{original} \cdot 100\% \tag{5.7}$$

The cross-correlation between synthesized and recorded audio played at the same note frequency was also computed in MATLAB as an additional quantitative metric for similarity. The cross-correlation was normalized such that auto-correlation given zero time delay equals 1.

Instrument	Note number	Fundamental Frequency f_o (Hz)		Cents
		Original	Synthesized	
Tongali	1	217.6	217.2	-3.185344135
	2	265.3	264.5	-5.228343873
	3	293.8	293.8	0
	4	336.7	336.1	-3.087814224
	5	383.2	383.3	0.4517244772
	6	440.1	440.3	0.7865670508
	7	525.3	525.3	0
	8	590	589.4	-1.761472815
	9	671.5	671.9	1.030956737
	10	779	779.1	0.2222237406
	11	895.9	895.8	-0.1932504382
	12	929.9	930.2	0.5584325781
			Average:	1.3755
			Stdev:	1.6431
Kolitong	A1	269.2	267.6	-10.32035276
	A2	341	342.2	6.081624693
	A6	556.8	550.1	-20.95836984
	B1	306.7	306.7	0
	B5	578.9	572.9	-18.03698256
	C1	539.1	533.6	-17.75309166
	C4	329.7	330	1.574564955
	C6	674.6	669.1	-14.17256895
	D1	328.3	325	-17.49003812
	D3	432	431.5	-2.004903581
	D4	521.2	522.3	3.649943765
	D5	598.4	593.2	-15.10989292
	D6	644.8	640.3	-12.12448068
	E2	354.8	352.4	-11.75050246
	E4	494.2	495.2	3.499564663
	F3	395	396.2	5.251472325
	F4	503.1	503.4	1.032032251
	F5	574.4	574.8	1.205175161
			Average:	9.0009
			Stdev:	7.059773262

Table 5.4: Cents difference

Instrument	Note number	Zero Cross		% error
		Original	Synthesized	
Tongali	1	231.667	229.2308	-1.0516
	2	267.8901	265.1163	-1.0354
	3	294.3339	293.3498	-0.3343
	4	336.7305	335.4334	-0.3852
	5	381.3067	382.0513	0.1953
	6	439.8571	437.7976	-0.4682
	7	530.8555	524.0741	-1.2774
	8	589.6677	588.1473	-0.2578
	9	671.4754	671.8666	0.0583
	10	778.3495	777.3338	-0.1305
	11	895.9078	893.5273	-0.2657
	12	931.1468	930.2383	-0.0976
			Average:	0.4631
			Stdev:	0.4179
Kolitong	A1	405.8975	495.4678	-22.0672
	A2	314.7736	380.1895	-20.7819
	A6	413.2270	496.0605	-20.0455
	B1	424.4595	421.9810	0.5839
	B5	440.9662	421.9810	4.3054
	C1	995.5302	445.0221	55.2980
	C4	893.8399	574.1318	35.7679
	C6	1886.2970	569.0822	69.8307
	D1	347.6668	352.9863	-1.5300
	D3	484.8399	456.7039	5.8032
	D4	422.3814	446.0690	-5.6081
	D5	384.4610	301.1880	21.6597
	D6	409.2407	382.6521	6.4971
	E2	395.1992	411.1200	-4.0285
	E4	531.8847	721.7357	-35.6940
	F3	467.0705	490.1232	-4.9356
	F4	356.1993	446.1638	-25.2568
	F5	451.6563	636.9686	-41.0295
			Average:	21.1513
			Stdev:	19.8748

Table 5.5: Zero crossing

Instrument	Note number	Spectral Centroid (Hz)		
		Original	Synthesized	**% error**
Tongali	1	1387.2649	518.9072	62.5949
	2	1169.5774	680.2191	41.8406
	3	835.4090	472.4607	43.4456
	4	699.0826	499.3641	28.5687
	5	888.4032	751.5099	15.4089
	6	953.8458	611.5942	35.8812
	7	1037.6863	755.5730	27.1868
	8	1104.5850	973.8815	11.8328
	9	1162.1698	927.4374	20.1978
	10	1248.3053	998.0610	20.0467
	11	1652.0730	1635.9828	0.9739
	12	2310.6288	2183.6250	5.4965
			Average:	26.1229
			Stdev:	17.6548
Kolitong	A1	1750.8171	1881.7947	-7.4809
	A2	3666.9351	3366.242004	8.2001
	A6	1325.4911	2626.4960	-98.1527
	B1	3286.1638	2013.0501	38.7416
	B5	3204.6365	2013.0501	37.1832
	C1	3898.5388	2983.0309	23.4834
	C4	3660.8532	2437.2634	33.4236
	C6	4536.4882	3736.4570	17.6355
	D1	4811.9959	3285.8907	31.7146
	D3	4782.9447	3877.1327	18.9384
	D4	4288.6743	3689.4156	13.9731
	D5	3448.0615	2650.5028	23.1306
	D6	2837.8929	2393.1726	15.6708
	E2	3283.8878	1515.5992	53.8474
	E4	4898.9783	3248.6188	33.6878
	F3	3556.9833	2658.6645	25.2551
	F4	3808.3726	2765.5962	27.3812
	F5	5346.0774	4368.4668	18.2865
			Average:	29.2326
			Stdev:	20.77210578

Table 5.6: Spectral centroid

Instrument	Note number	Attack Time (s)		
		Original	Synthesized	**% error**
Tongali	1	0.2438	0.2450	-0.4922
	2	0.0854	0.0824	3.5129
	3	0.1574	0.1532	2.6684
	4	0.0596	0.0620	-4.0268
	5	0.3182	0.2726	14.3306
	6	0.2732	0.2660	2.6354
	7	0.1982	0.2108	-6.3572
	8	0.1028	0.1058	-2.9183
	9	0.2168	0.2072	4.4280
	10	0.0655	0.0655	0.0000
	11	0.1190	0.1046	12.1008
	12	0.0704	0.0602	14.4886
			Average:	5.6633
			Stdev:	5.1190
Kolitong	A1	0.0210	0.0204	2.8098
	A2	0.0198	0.0180	9.0657
	A6	0.0252	0.0232	7.9228
	B1	0.0168	0.0192	-14.3010
	B5	0.0168	0.0199	-18.5681
	C1	0.0156	0.0186	-19.3321
	C4	0.0212	0.0192	9.4838
	C6	0.0156	0.0168	-7.7046
	D1	0.0150	0.0176	-17.4194
	D3	0.0156	0.0171	-9.4802
	D4	0.0162	0.0180	-11.1454
	D5	0.0252	0.0282	-11.9716
	D6	0.0174	0.0180	-3.5246
	E2	0.0162	0.0186	-14.8487
	E4	0.0156	0.0174	-11.4800
	F3	0.0156	0.0178	-14.0376
	F4	0.0168	0.0204	-21.4545
	F5	0.0156	0.0204	-30.7416
			Average:	13.0718
			Stdev:	6.762569284

Table 5.7: Attack time

Instrument	Note number	Correlation
Tongali	1	0.8245
	2	-0.7076
	3	-0.9723
	4	-0.3823
	5	0.925
	6	0.3686
	7	0.9956
	8	-0.1745
	9	0.2161
	10	-0.3123
	11	-0.112
	12	0.7505
Kolitong	A1	0.4249
	A2	0.2335
	A6	-0.8740
	B1	0.5881
	B5	0.5248
	C1	0.2081
	C4	0.4288
	C6	-0.6351
	D1	-0.1597
	D3	0
	D4	-0.0820
	D5	0.8141
	D6	-0.6324
	E2	0.1130
	E4	0.4508
	F3	-0.0875
	F4	0.4362
	F5	0.0340

Table 5.8: Correlation

Chapter 6

Conclusion and Recommendations

The Kalinga tongali and kolitong features were extracted from recordings of instruments using mainly Modal Distribution Analysis. Synthesis was implemented in MATLAB through Sum of Sinusoids synthesis for both the tongali and kolitong. The synthesized sounds were then used to develop a sample-based virtual instrument plugin in HISE using the VST2 standard by Steinberg.

A 2IFC listening test was conducted to test the aural similarity of the recorded and synthesized sound with the goal of having a minimized sensitivity index d'. The tongali and kolitong had a d' value of 7.0711 x 10^{-6} and -3.5355 x 10^{-5} respectfully, proving that the synthesis of both instruments are aurally indistinguishable from the recorded sounds. Objectively, both the tongali and kolitong had a high percent error. The tongali and kolitong had an average cent difference of 1.3755¢ and 21.1513¢, a zero crossing error of 0.4631% and 21.1513%, spectral centroid error of 26.1229% and 18.2865%, attack time error of 5.6633% and 13.0718%, and correlation average of 0.5618 and 0.3778, respectively. Despite having poor results objectively, the study showed that both instrument synthesis are subjectively imperceptible from the recorded sounds. The study solves the problem of producing an aurally similar sound for synthesis of the tongali and kolitong

A few recommendations for improvement of the study are suggested:

1. It is advisable to develop an algorithm for the VST instead on relying on samples. Using a sample-based VST similar to what was done in this study takes a considerable amount of of space compared to an algorithm-based VST. Aside from the VST, the samples also need to be imported to the DAW to work. An algorithm-based VST depends on on computing power but is efficient in space as opposed to a sample-based VST.

2. Incorporate the ability to add different techniques of playing. A sample-based VST is limited to the samples provided to it. Incorporating other techniques of playing the tongali and

kolitong will improve its naturalness.

3. Utilize the newer VST3 standard. HISE still uses the VST2 standard, but Steinberg has ended support for the latter standard in 2018, opting to focus on VST3. This ensures that the standard that is used by the VST is maintained and updated to fix issues that may arise.

Bibliography

[1] M. M. I. M. Online, *Tongali.* MIMO - Musical Instrument Museums Online, Nov 2013.

[2] "Images of cordillera bamboo instruments."

[3] J. Sirisha Devi, Y. Madhuri Srinivas, N. Murali Krishna, and M. Epstein, "A study : Analysis of music features for musical instrument recognition and music similarity search," *Special Issue of International Journal of Computer Science* & *Informatics*, vol. 2, no. 1,2, pp. 21–24, 2012.

[4] A. Prior, "Na Suklit : Benicio Sokkong and the Bamboo Musical Instruments of the Kalinga," *Thesis*, 2011.

[5] J. Maceda, *Gongs and Bamboo: A Panorama of Philippine Music Instruments.* University of the Philippines Press, 1998.

[6] "", *Cultural Center of the Philippines Encyclopedia of Philippine Art*, vol. 7. Manila, Philippines: Cultural Center of the Philippines in cooperation with the Office of the Chancellor, University of the Philippines Diliman, second edition ed., 2017.

[7] A. de Guzman, F. Go, and F. de Leon, "Timbre and pitch characterization of monophonic samples of philippine indigenous instruments." Undergraduate Student Project, University of the Philippines, December 2018.

[8] N. Macalisang, I. N. S. Reyes, R. C. L. Guevara, and R. D. Cajote, "Angklung synthesis using physical modeling in the ipad and desktop platforms." Undergraduate Student Project, University of the Philippines, October 2012.

[9] M. Credo, E. Pesa, J. Rili, A. Aragon, and L. Zulueta, "Application software for the synthesis of native philippine bamboo musical instruments by physical modeling." Undergraduate Project, University of the Philippines – Diliman, 2013.

[10] M. Muller, D. P. W. Ellis, A. Klapuri, and G. Richard, “Signal processing for music analysis,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 1088–1110, Oct 2011.

[11] F. d. J.A.L Domingo, J.P.R Castillo and C. Lucas, “Vsti implementation of angklung, bumbong, and marimba,” December 2019.

[12] W. J. Pielemeier and G. H. Wakefield, “Multi-component power and frequency estimation for a discrete tfd,” in *Proceedings of IEEE-SP International Symposium on Time- Frequency and Time-Scale Analysis*, pp. 620–623, 1994.

[13] R. C. L. Guevara, “Modal distribution analysis and sum of sinusoids synthesis of piano tones.” Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 1997.

[14] W. Pielemeier and G. Wakefield, “A high-resolution time-frequency representation for musical instrument signals,” *Journal of The Acoustical Society of America - J ACOUST SOC AMER*, vol. 99, pp. 2382–2396, 04 1996.

[15] L. Cohen, “Generalized phase-space distribution functions,” *Journal of Mathematical Physics*, vol. 7, no. 5, pp. 781–786, 1966.

[16] W. J. Pielemeier, G. H. Wakefield, and M. H. Simoni, “Time-frequency analysis of musical signals,” *Proceedings of the IEEE*, vol. 84, pp. 1216–1230, Sep. 1996.

[17] F. Agsaway, M. O. Co, and R. C. L. Guevara, “Modal distribution analysis and sum of sinusoids synthesis of kulintang musical signals.” Undergraduate Research Project, University of the Philippines, April 2005.

[18] A. F. B. Laguna, N. M. P. Valdez, and R. C. L. Guevara, “Midi implementation of a kulintang modal synthesizer using the vst2.4 standard,” in *TENCON 2012 IEEE Region 10 Conference*, pp. 1–5, Nov 2012.

[19] J. Pekonen, “Computationally efficient music synthesis – methods and sound design.” Masterś Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology, June 2007.

[20] R. C. Maher, “Sinewave additive synthesis revisited,” in *Audio Engineering Society Convention 91*, Oct 1991.

[21] T. Tolonen, V. Välimäki, and M. Karjalainen, "Evaluation of modern sound synthesis methods." Report, March 1998.

[22] G. de Poli, "A tutorial on digital sound synthesis techniques," *Computer Music Journal*, vol. 7, no. 4, pp. 8–26, 1983.

[23] "Introduction."

[24] N. A. Macmillan and C. D. Creelman, *Detection theory: A user's guide.* Psychology press, 2004.

[25] D. J. Benson, *Music: A Mathematical Offering.* Cambridge University Press, 2008.

[26] D. O'Shaughnessy, "Linear predictive coding," *IEEE Potentials*, vol. 7, pp. 29–32, Feb 1988.

[27] B. S. Atal, "The history of linear prediction," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 154–161, 2006.

[28] H.-C. Tai and D.-T. Chung, "Stradivari violins exhibit formant frequencies resembling vowels produced by females," *Savart Journal*, vol. 1, no. 2, 2012.

[29] S. P. K. E. Museum, *Kolitong: Idiochorde VollrÖhrenzither.* MIMO - Musical Instrument Museums Online, Nov 2013.

[30] H.-S. Kim, "Linear predictive coding is all-pole resonance modeling," *Center for Computer Research in Music and Acoustics, Stanford University.*

[31] A. F. Laguna and R. Guevara, "Real time karaoke grader using modal distribution for android devices," pp. 1–5, 11 2012.