Digital Sound Synthesis of the Tongali and Kolitong Implemented as a Virtual Instrument Plugin

Bagaforo, Christian April P.
Gayo, Jhaezminne S.

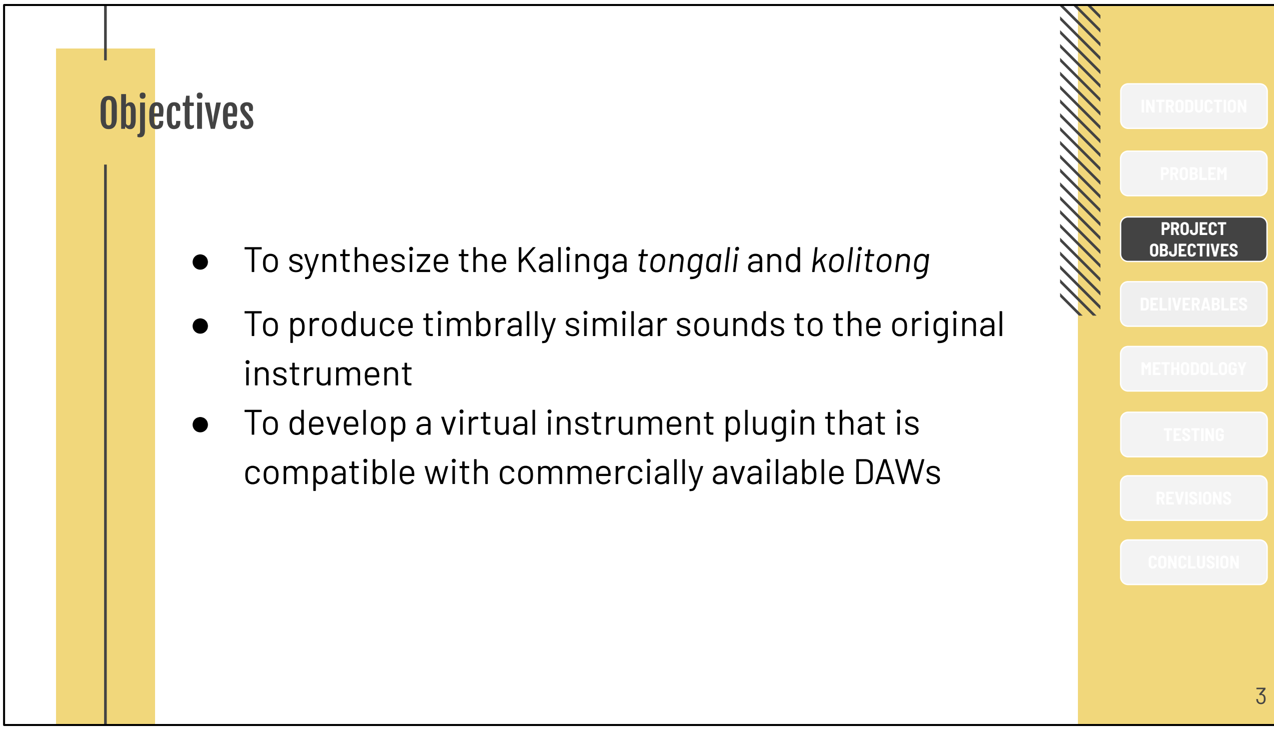
Advisers:
Franz de Leon
Crisron Rudolf Lucas
Carl Timothy Toletino

Today, there is a decline in use and lack of familiarity when it comes to Philippine bamboo instruments. Local bamboo instruments also lack in accessibility because of the small number of manufacturers and the durability of the instrument itself.

In order to provide better access of the instruments, this project aims to recreate the sound of the Kalinga Tongali and Kolitong implemented as a virtual instrument plugin. Instrument synthesis is done to produce aurally similar sounds of the instruments. The development of VST plugins that are compatible with commercially available DAWs will then give access for musicians to be able to arrange music digitally using the synthesized sounds.

Problem Statement

The continued practice of 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 sound of the instruments.

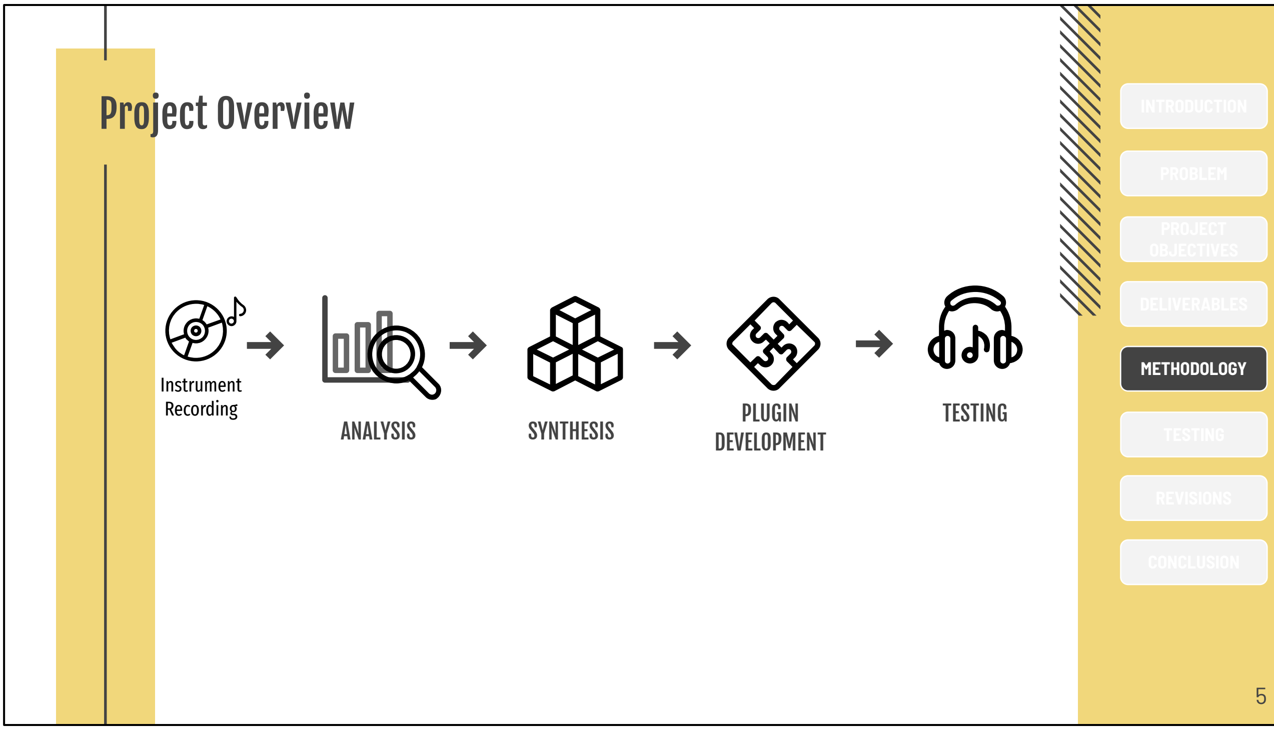


A look back at our objectives:

- To synthesize the Kalinga *tongali* and *kulitong*
    - using features extracted from its actual recordings
    - synthesized sounds must be timbrally similar to the original signal (based on objective and subjective testing)
- To develop a virtual instrument plugin for the three instruments
    - using open-source and freely-licensed development resources, and widely-adopted standards

Deliverables

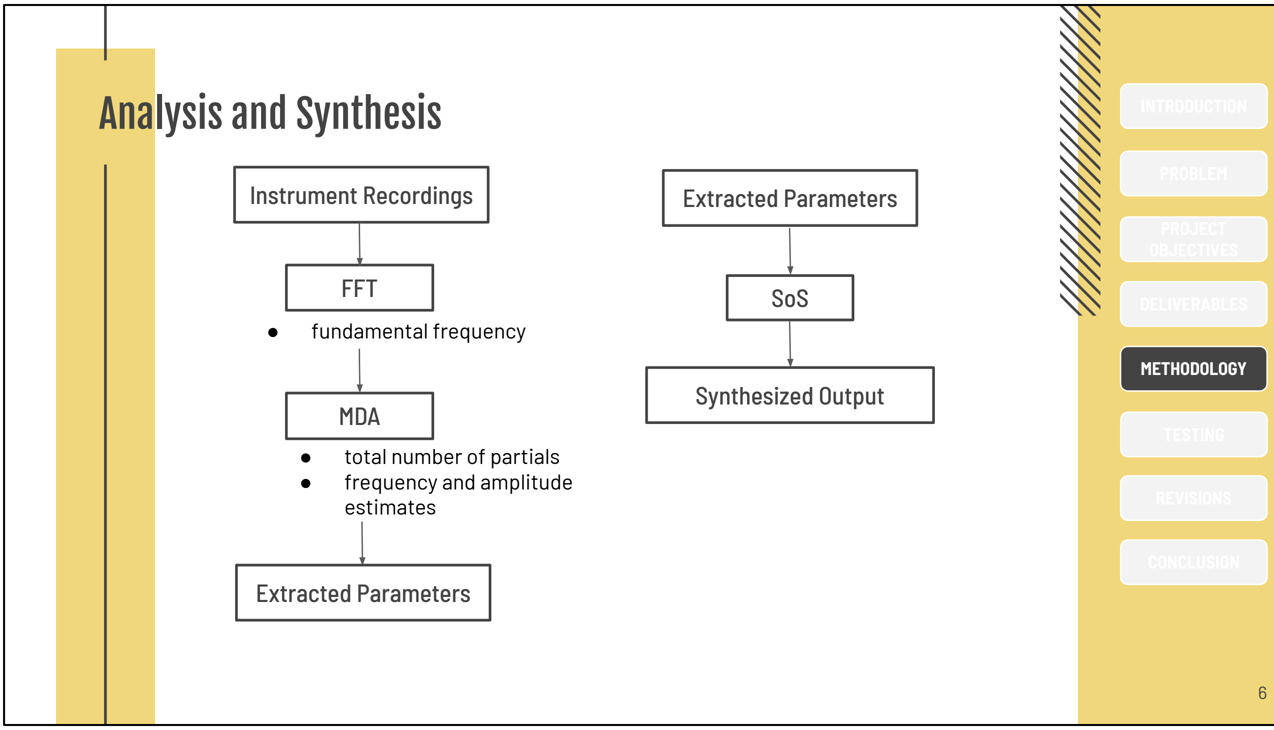
- MATLAB code of the instrument synthesis
- *Virtual instrument plugin using the* WAV files of synthesized tongali and kolitong
- Project documentation, including subjective and objective test results



Recordings are provided by the UP DSP Lab.
The recorded samples were analyzed using time, frequency and time-frequency analysis techniques, particularly the Modal Distribution Analysis (MDA).
The partial estimates from the MD were used for SoS synthesis.

Plugin development using sample-based synthesis was implemented using HISE.

Testing was done in two ways: subjective and objective. Subjective testing involves listening tests using the Two-Interval Forced Choice (2IFC). Objective testing involves the comparison of the acoustic features of the actual and synthesized signals.



Analysis Methods include:

1. Fourier analysis

> to determine the ***fundamental frequency*** of the signal

2. Modal Distribution analysis

> to extract the ***amplitude and frequency estimates*** of each partial of the signal

3. Amplitude extraction using Hilbert transform

> to obtain the amplitude envelope (***ADSR envelope***) of the signal

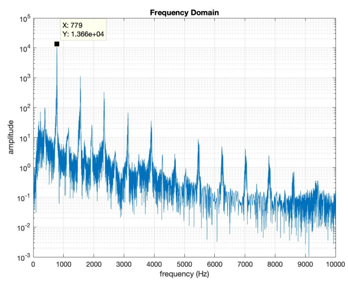
Synthesis Methods include:

1. Sum of Sinusoids (SoS) Synthesis

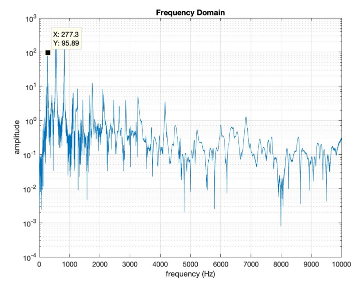
> the extracted features from MDA is used for SoS Synthesis to generate the synthesized signal using the partials extracted from MDA

Analysis

Fourier Analysis



Tongali



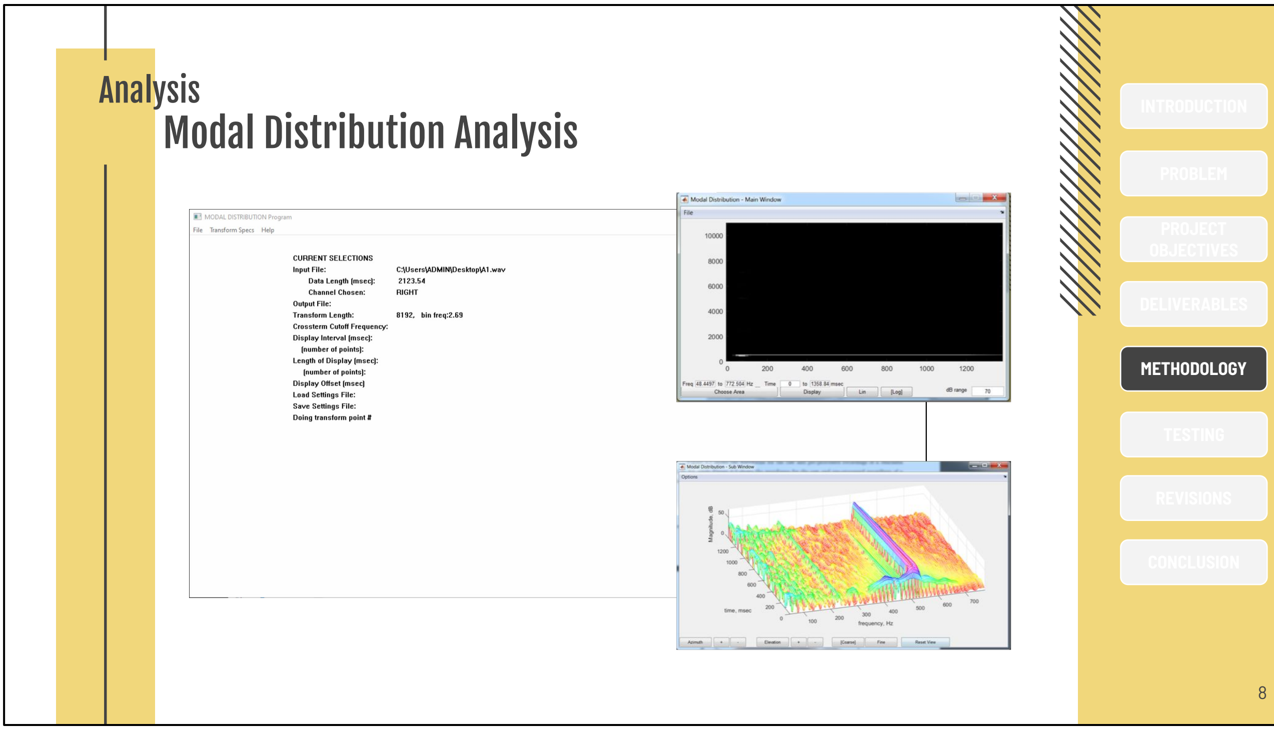
Kolitong

INTRODUCTION
PROBLEM
PROJECT OBJECTIVES
DELIVERABLES
METHODOLOGY
TESTING
REVISIONS
CONCLUSION

Fourier Analysis

- Used to determine the *fundamental frequency* of the signal
- Implemented using the Fast Fourier Transform in MATLAB
- We need to determine the minimum distance of any two partials Δf_{min} . This is the minimum distance from any two frequencies in a signal. For a harmonic signal, it is equal to the fundamental frequency of the harmonics. From this, we can determine the cutoff frequency of the cross term filter (f_c), a needed parameter in Modal Distribution Analysis

The figure above is a semilogarithmic plot of the 1st note of the kolitong and the 10th note of the tongali. The datatip for fundamental frequencies are added (277.3 for the kolitong, 779 for the tongali)



Modal Distribution Analysis

- Used to extract the *amplitude* and *frequency* estimates of each partial of each signal
- A modal distribution (MD) program created by Guevara was used to obtain the modal distribution of the signals. The interface can be seen on the left of the slide
    - A function that is included with the program, *modplot.m* displays the MD plot with the magnitude in dB, which can be seen on the right side of the slide

Analysis

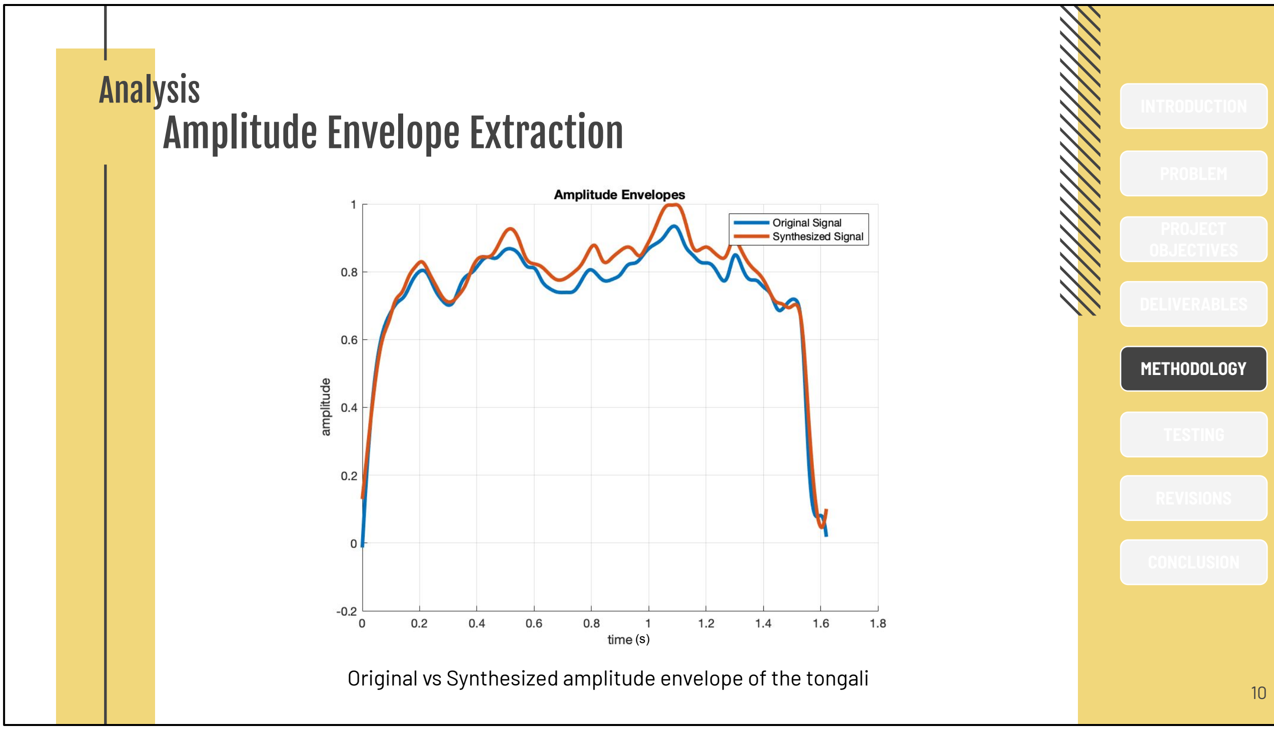
Modal Distribution Analysis

Parameter	Value
Transform Length	8192
Display Interval	Default (1/(2*f_c))
Length of Display	*Equal to Data Length*
Display Offset	0

It is ideal to set the transform length as high as possible to obtain the best frequency resolution. f_c (cutoff frequency of the cross term filter (CTF)) sets the display interval parameter. It is also ideal to set this as near to Δf_{min} as possible because a higher f_c makes the display interval smaller, improving the time resolution.

For a signal with a well-defined pitch, the ideal f_c is a value close to the fundamental (f_o). That is why for the *tongali* and *kolitong*, f_c was set to f_o - 5 Hz. This is to provide a display interval that is as high as possible while giving an allowance to the harmonics.

The partials obtained using Guevara's program are analyzed using a function in MATLAB *(partial.m)* created to obtain the amplitude and frequency estimates of a signal.

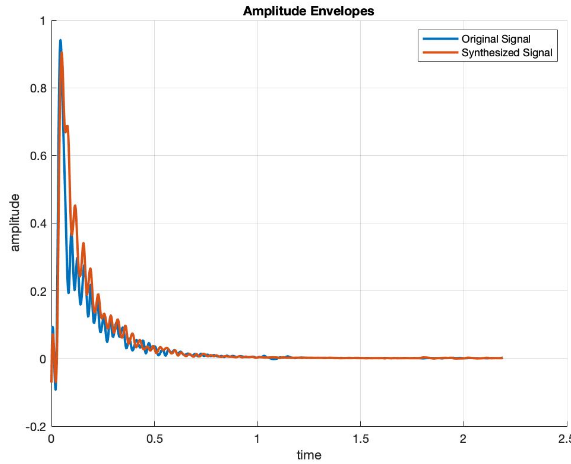


The amplitude envelope of the signal is extracted in order to compare the attack slopes of the synthesized signals. Shown are the synthesized signals from the MD using SoS.

The envelope of synthesized signal of the tongali resembles the original.

Analysis

Amplitude Envelope Extraction



Original vs Synthesized amplitude envelope of the kolitong signal

Having a short attack time, the amplitude envelope of the kolitong signal is slightly smoothened due to the CTF in the MD. Even though the onset smoothing is visible, the sound of both the original and the synthesized samples at the onsets are aurally similar.

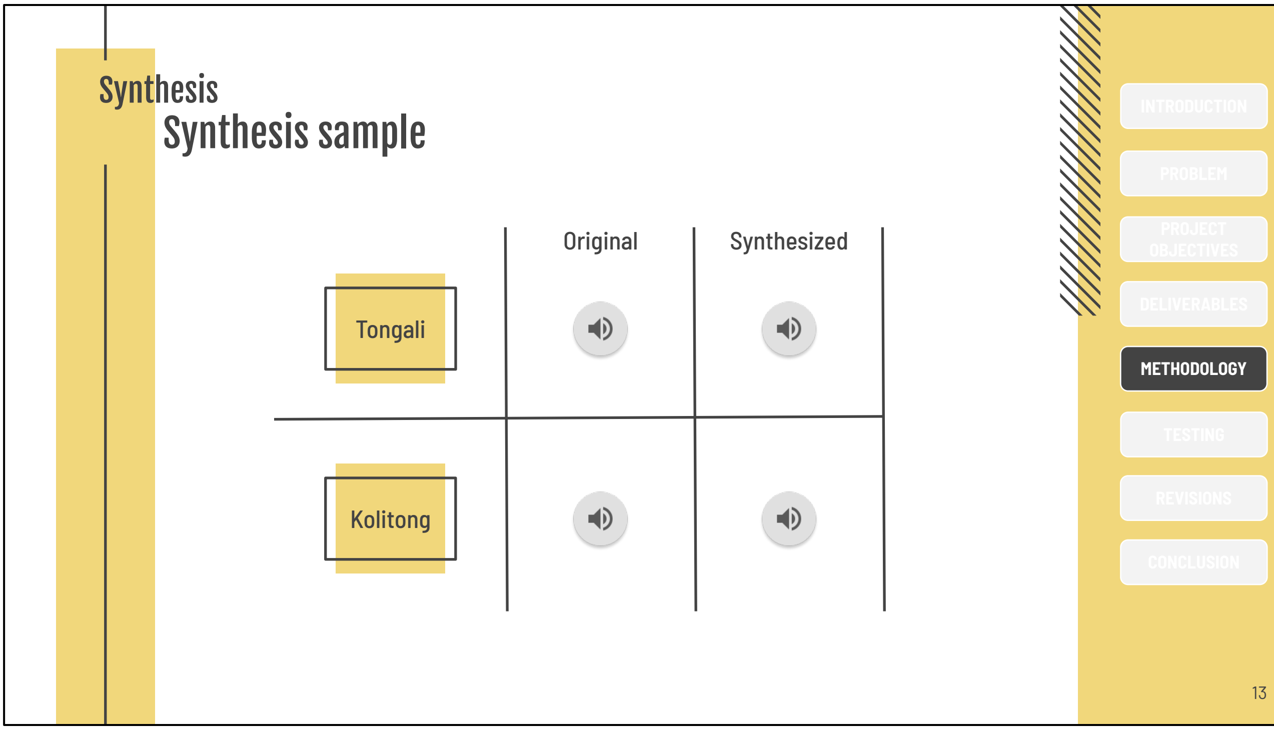
Synthesis

Sinusoid Additive Synthesis

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

s(n) is the synthesized signal
M is the total number of partials from the original signal
p(n) is the m_{th} partial

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. A function 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.



Tongali

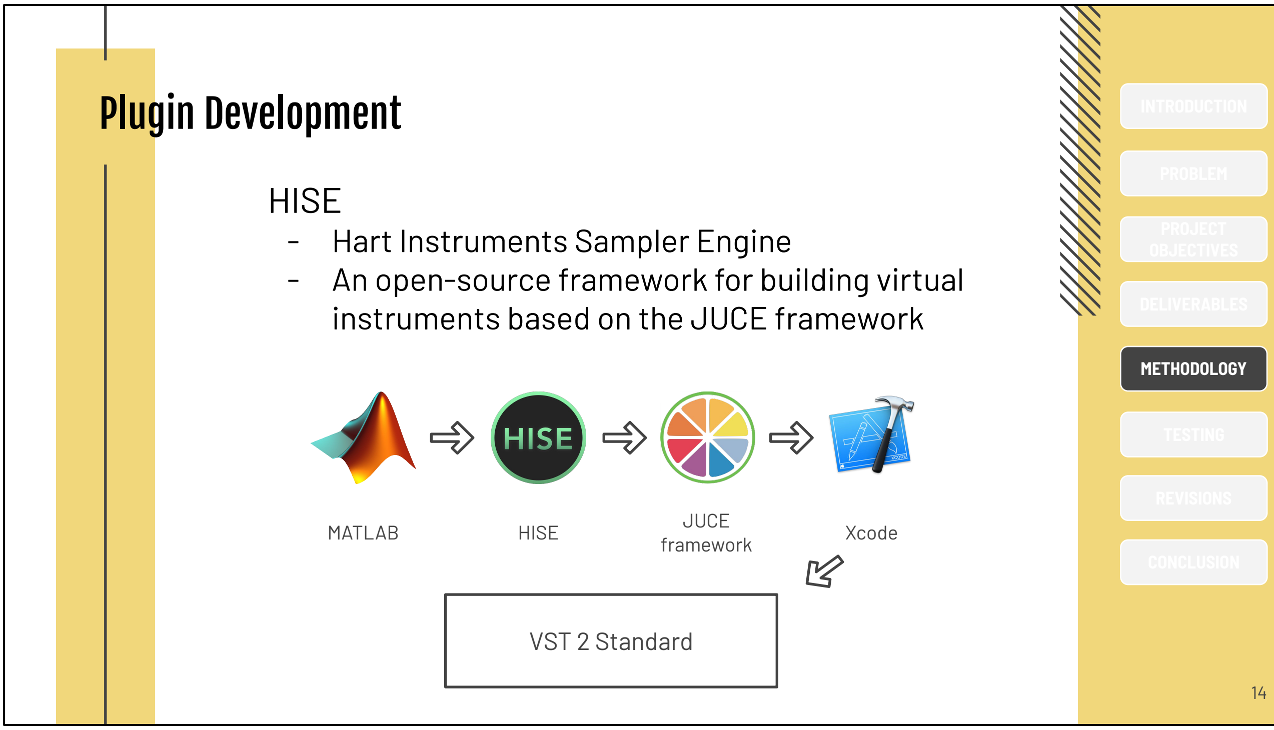
- Original audio: https://drive.google.com/open?id=1qE8uorK3Ind1FyT-1Kdh70CQDrsLJww4
- Synthesized audio: https://drive.google.com/open?id=1uq4hFGGgl-euNjRqOfb4Wv3Ng6ESVoJw

Kolitong

- Original audio: https://drive.google.com/open?id=12idwkKwmTY54LyO3jLnDFcEPAAetfSp5
- Synthesized audio: https://drive.google.com/open?id=1Qe7TEYWEoNYvdtcUmE5OJMaum3wd1IUk

All synthesis audio files can be found here:

- Tongali: https://drive.google.com/open?id=1r_WTgPaGRwGkZpGVZfmlyWvCELUGTqGg
- Kolitong: https://drive.google.com/open?id=1xmJ4Q9G1c3cdc9-fCRkk60R1_GB3mX7f

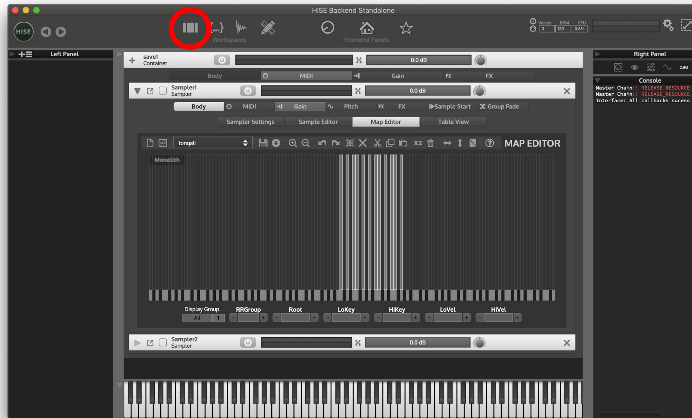


The developed synthesis algorithms were supposed to be implemented in the Faust pro-gramming language and compiled as an audio plugin project implementing the JUCE framework for the Projucer IDE. Instead, a sample-based VSTi plug-in was developed using Hart Instruments Sampler Engine (HISE). HISE is an open-source framework for building virtual instruments based on JUCE [24]. The plugin was developed primarily for macOS.

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.

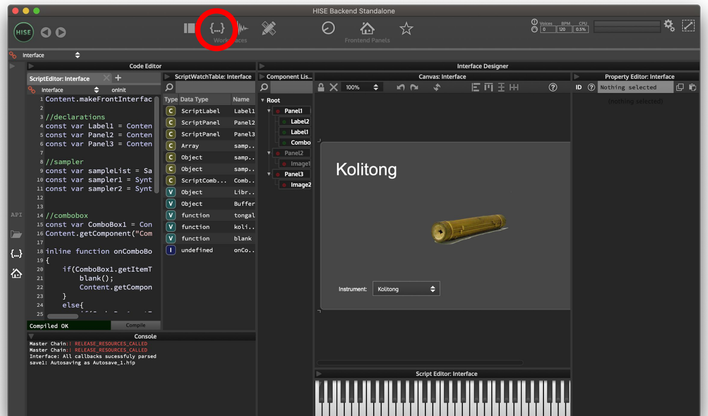
[24] “Introduction.” Obtained from: https://docs.hise.audio/introduction/index.html.

Plugin Development
HISE



This is the user interface of HISE. This view is the main work view, where one can edit the mapping of the audio samples of each instrument in Musical Instrument Digital Interface (MIDI). Here, the tongali and kolitong were sample mapped in increasing frequency starting at an octave lower than middle C (C3). The tongali was mapped on the 12 white keys starting from C3 while the kolitong was mapped to the 18 keys starting from C3.

Plugin Development
HISE

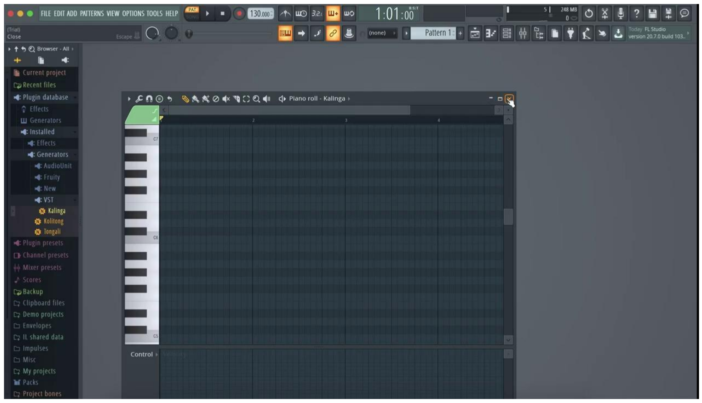


INTRODUCTION
PROBLEM
PROJECT OBJECTIVES
DELIVERABLES
METHODOLOGY
TESTING
REVISIONS
CONCLUSION

This is the interface designer of HISE. Here, one can design the front end of the VST. The sample maps are linked to the front end of the interface in this part.

Plugin Development

VST Plugin Demo



Can also be accessed here:
https://drive.google.com/open?id=1rktx8p-JWmXBVYSj_P6NU9frve2a5jn4

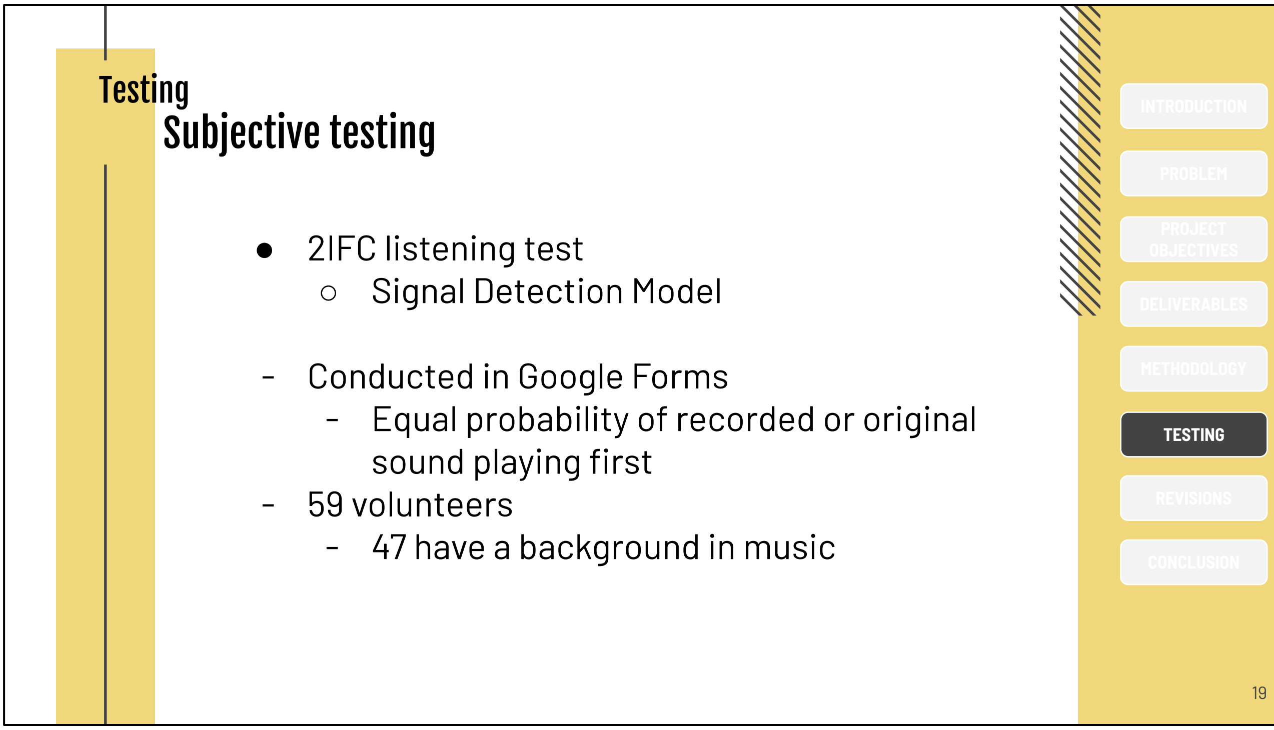
Here we can see the plugin working on FL Studio (v20.5.1 [build 522] - 64 bit). The audio of the video sounds bad because the audio was recorded from broken speakers. To listen to the audio files, please refer to the slide of synthesis samples.

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.

Testing

Two forms of testing was done:

1. Subjective testing
   - Two-interval Forced Choice (2IFC)
2. Objective testing
   - Cents notation
   - Zero crossing
   - Attack time
   - Correlation
   - Spectral Centroid



The listening test was done online through Google forms. A total of 59 volunteers answered the form.

Two-interval First Choice

- Slightly modified to suit the Google Forms platform:

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.

We will be recalling the signal detection model later on.

Testing

Subjective testing

Instrument	Mean Score	Total Score	Range		Accuracy (%)	Standard deviation
			Min	Max		
Tongali	5.1186	12	0	11	42.66	2.27
Kolitong	5.2712	10	2	10	52.71	1.84

Instrument	Accuracy (%)	
	Min	Max
Tongali	43.81	44.71
Kolitong	52.86	52.35

The tables in the slide summarizes the results of the listening test. Note that the mean score in the top table is the average amount of times that the volunteers chose the original signal correctly when given a original signal - synthesized signal pair of the same note frequency. Looking at the accuracy, we can deduce that they were unable to distinguish between the original and synthesized signal. Looking at the bottom table, we can see the breakdown of the accuracy based on their declared musical background. Out of the 59 volunteers, 47 people stated having a background in music, 15 of whom declared that they are studying in the College of Music in UP Diliman.

Testing

Subjective testing

- 2IFC listening test
  - Signal Detection Model
    - H = P(“Recorded First” | ⟨Recorded, Synthesized⟩)
    - F = P(“Recorded First” | ⟨Synthesized, Recorded⟩)

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

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

INTRODUCTION
PROBLEM
PROJECT OBJECTIVES
DELIVERABLES
METHODOLOGY
TESTING
REVISIONS
CONCLUSION

Signal detection model

- A hit (*h*) is defined as correctly identifying the first sound played as the recorded sound given the order ⟨Recorded, Synthesized⟩.
- A false-alarm (*f*) is defined as incorrectly identifying the first sound played as the recorded sound given the order ⟨Synthesized, Recorded⟩. Hit rate H and False-alarm rate F given *h* hits and *f* false-alarms for N trials were calculated shown in the formula above
- Given the hit rate and false-alarm rate, sensitivity index *d’* is computed using the given formula, where z() is the z-score in standard deviation units
  - The goal is to minimize *d’* such than when *d’* = 0
  - At *d’* = 0, the probability of a correct response from an observer ($p(c)_{max}$) is equal to 0.5, indicating that the subject could not have done better than if they had guessed on each trial

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

Testing

Subjective testing

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

We can see in the table above that *d'* is minimized. In turn, $p(c)_{max}$ is near to 0.5.

The results tabulated above show us that the volunteers were unable to distinguish well between the recorded and synthesized signal, thus fulfilling our goal to make the signal timbrally similar subjectively. 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].

Object Testing

Zero Crossing Rate

- Describes the frequency of the sign changes in the amplitude of a signal
- Roughly estimates the average signal frequency

Instrument	Number of Samples	Average % error	Standard Deviation
Tongali	12	0.4631	0.4179
Kolitong	18	21.1513	19.8748

The zero crossing rate provides a rough estimate of the dominant frequency and spectral centroid of the signal. The tongali has a low percent rate due to having a straightforward distribution. The kolitong has a high percent error, in which the higher errors are found in in higher frequency signals.

Object Testing

Spectral Centroid

- Describes the brightness of the sound in a signal

Instrument	Number of Samples	Average % error	Standard Deviation
Tongali	12	26.1229	17.6548
Kolitong	18	29.2326	20.7721

The spectral centroid, in units of Hz shows the frequency at which the signal energy is centered. This study determined that the original sounds had a higher spectral centroid as compared to the synthesized sounds.

Object Testing

Attack Time

- Feature that greatly affects the timbre of a musical signal
- Defines the dynamic characteristic of the sound

Instrument	Number of Samples	Average % error	Standard Deviation
Tongali	12	5.6633	5.1190
Kolitong	18	13.0718	6.7626

INTRODUCTION
PROBLEM
PROJECT OBJECTIVES
DELIVERABLES
METHODOLOGY
TESTING
REVISIONS
CONCLUSION

The percent error of the attack for the tongali is small due to the set CTF cutoff frequency which is approximately its fundamental. The set parameter value for the CTF cutoff preserved the attack slope of the tongali signal.

Greater error in the attack is seen in the Kolitong due to the smaller value for the CTF cutoff. This produces a more time smoothed onset than that of the original.

Object Testing

Correlation

- Mutual relationship which exists between two signals
- Indicates the measure of how a signal resembles another

Instrument	Number of Samples	Average Correlation Coefficient	Standard Deviation
Tongali	12	0.5618	0.3328
Kolitong	18	0.3778	0.2687

INTRODUCTION
PROBLEM
PROJECT OBJECTIVES
DELIVERABLES
METHODOLOGY
TESTING
REVISIONS
CONCLUSION

The correlation coefficient measures the resemblance of the synthesized signal with the recorded sample. It from 0 to 1. A value closer to 1 describes a stronger similarity for the actual and synthesized signal

Object Testing

Cents Difference

- Standard method of representing and comparing musical pitches and intervals with relative accuracy

Instrument	Number of Samples	Average Cents Difference	Standard Deviation
Tongali	12	1.3755	1.6431
Kolitong	18	9.0009	7.0598

The Cents notation determines the difference between the fundamental frequencies of the real and synthesized signal. 1 semitone is equivalent to 100 cents. The perceivable difference in the cents notation is 7 or greater

Revised Objectives

Original

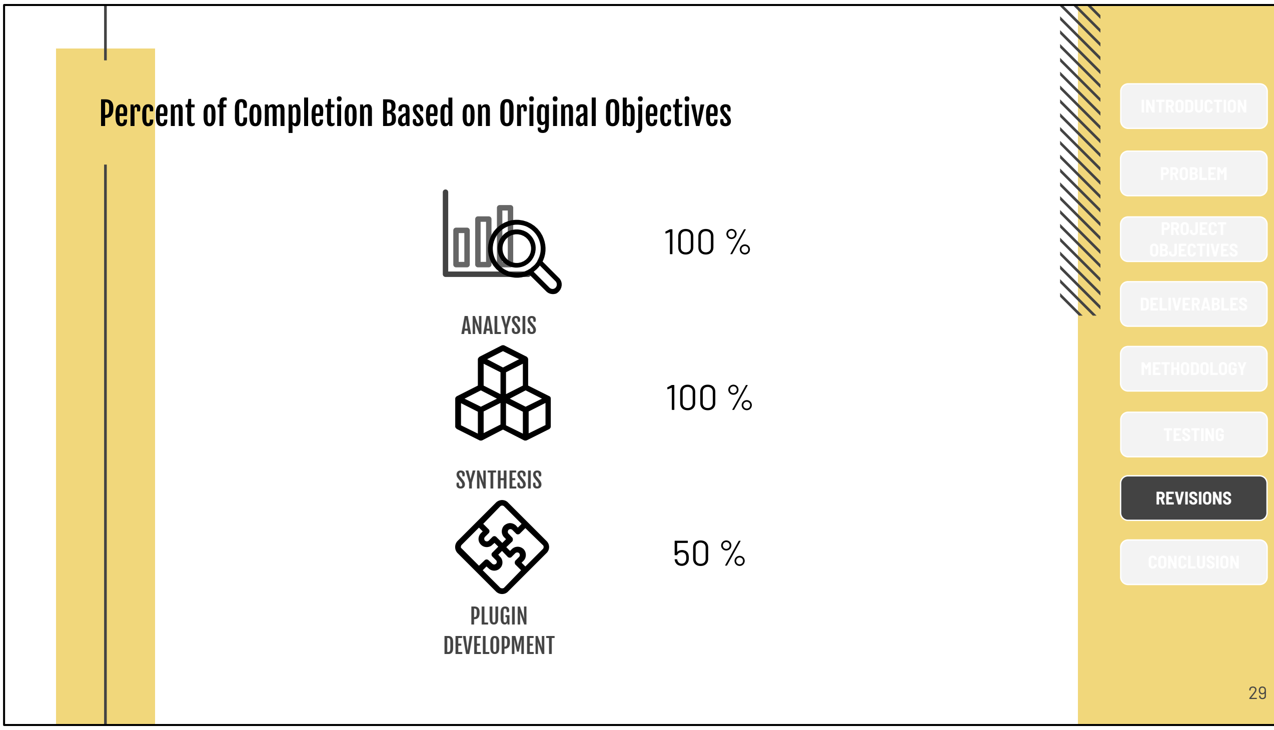
- To synthesize the Kalinga *tongali* and *kolitong*
- To produce timbrally similar sounds to the original instrument
- To develop a virtual instrument plugin that is compatible with commercially available DAWs

Revised

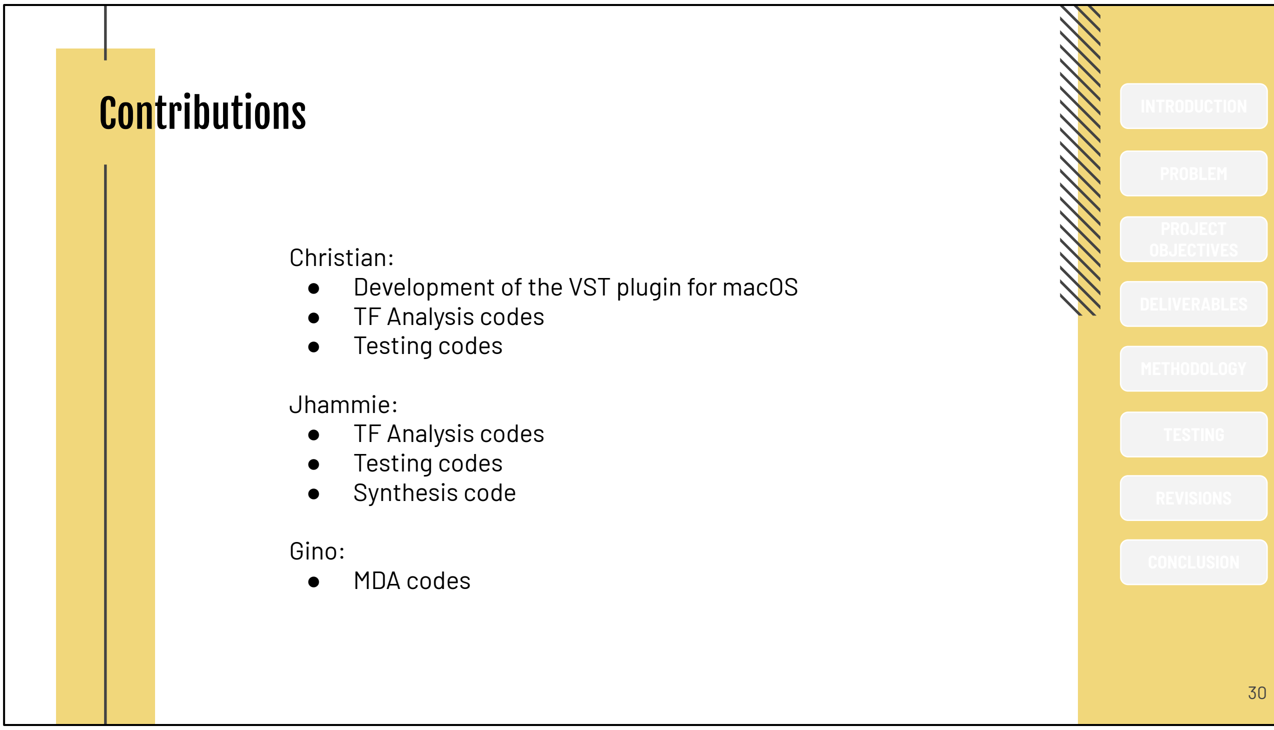
- To synthesize the Kalinga *tongali* and *kolitong*
- To produce timbrally similar sounds to the original instrument
- To develop a **sample-based** virtual instrument plugin that is compatible with commercially available DAWs

INTRODUCTION
PROBLEM
PROJECT OBJECTIVES
DELIVERABLES
METHODOLOGY
TESTING
REVISIONS
CONCLUSION

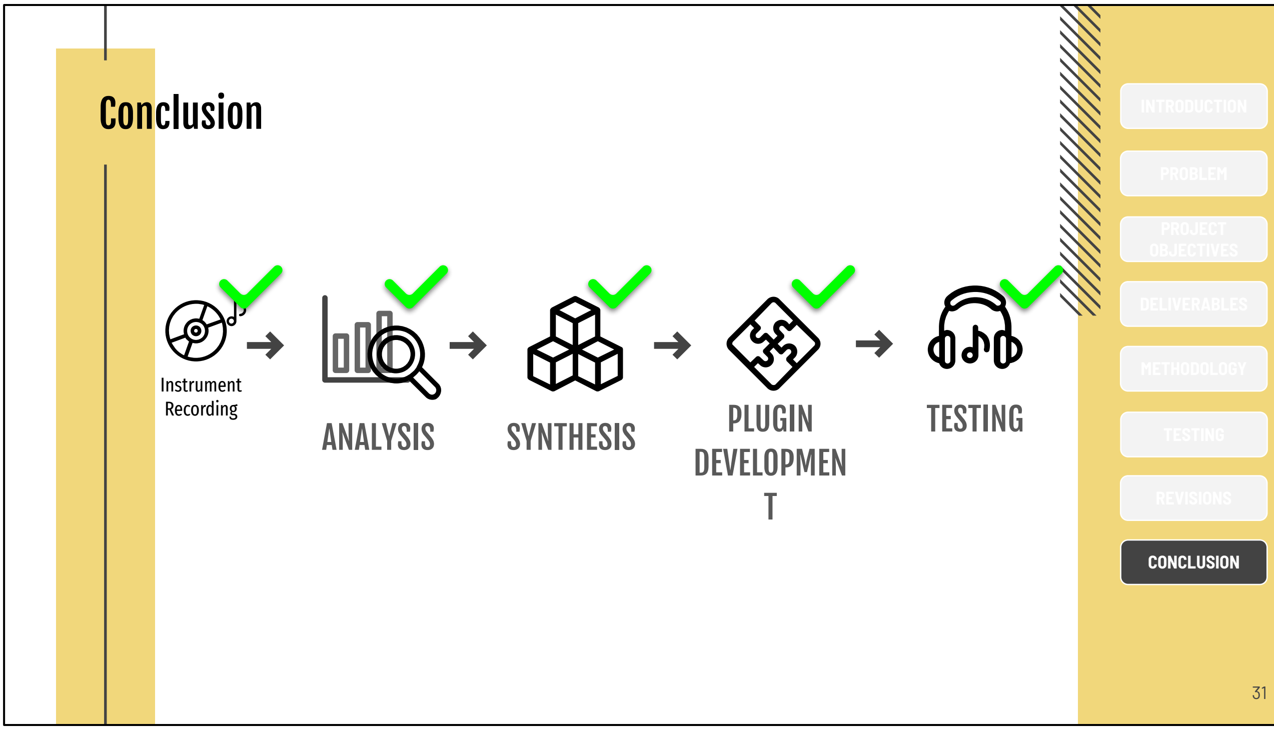
Due to the time constraint, there is no time to implement a synthesis algorithm in Faust to create a VST for the instruments as originally proposed. Instead, the samples created from synthesis in MATLAB are plugged in to HISE to create a sample-based VST. It only requires mapping of the synthesized sounds into MIDI.



Because of the Enhanced Community Quarantine, plugin development was not fully implemented as expected. The initial plan for the plugin was to implement a synthesis algorithm on a VST. While a VST was developed using HISE, it is a sample-based synthesis and is constrained by the sound of the samples plugged into it .



Although Gino did not continue with us in this study, it is worth noting that he has made significant contributions to it. Gino made the MDA code for analysis of the tongali, kolitong, and though discontinued, the ulibaw.



- 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 (SoS) synthesis for both the tongali and kolitong.
- The synthesized sounds were used to develop a sample-based virtual instrument plugin in HISE using the VST2 standard by Steinberg
- Subjective and objective listening tests were conducted
    - 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 7.0711×10^{-6} and -3.5355×10^{-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:
        - Correlation for the tongali and kolitong is 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

Recommendations

1. It is advisable to develop an algorithm for the VST instead on relying on samples.
2. Incorporate the ability to add different techniques of playing.
3. Utilize the newer VST3 standard.

1. 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. 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. HISE still uses the VST2 standard, and most developers still prefer the old standard, but Steinberg has ended support for the latter in 2018, opting to focus on VST3. Using VST3 ensures that the standard that is used by the VST is maintained and updated to fix issues that may arise.

References

[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 : Analysisof music features for musical instrument recognition and music similarity search,”SpecialIssue 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 ofthe Philippines Press, 1998.
[6] "",Cultural Center of the Philippines Encyclopedia of Philippine Art, vol. 7. Manila, Philip-pines: 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 monophonicsamples of philippine indigenous instruments.” Undergraduate Student Project, University ofthe Philippines, December 2018.
[8] N. Macalisang, I. N. S. Reyes, R. C. L. Guevara, and R. D. Cajote, “Angklung synthesisusing 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 synthe-sis of native philippine bamboo musical instruments by physical modeling.” UndergraduateProject, 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, bum-bong, 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 andTime-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 ofPhilosophy, 1997.
[14] W. Pielemeier and G. Wakefield, “A high-resolution time-frequency representation for musicalinstrument 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 musicalsignals,”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 ofsinusoids synthesis of kulintang musical signals.” Undergraduate Research Project, Universityof the Philippines, April 2005.
[18] A. F. B. Laguna, N. M. P. Valdez, and R. C. L. Guevara, “Midi implementation of a kulintangmodal synthesizer using the vst2.4 standard,” inTENCON 2012 IEEE Region 10 Conference,pp. 1–5, Nov 2012.
[19] J. Pekonen, “Computationally efficient music synthesis – methods and sound design.” Master ́sThesis submitted in partial fulfillment of the requirements for the degree of Master of Sciencein Technology, June 2007.
[20] R. C. Maher, “Sinewave additive synthesis revisited,” inAudio Engineering Society Conven-tion 91, Oct 1991.
[21] T. Tolonen, V. V ̈alim ̈aki, and M. Karjalainen, “Evaluation of modern sound synthesis meth-ods.” 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] A. F. Laguna and R. Guevara, “Real time karaoke grader using modal distribution for androiddevices,” pp. 1–5, 11 2012.
[24] “Introduction.”
[25] N. A. Macmillan and C. D. Creelman,Detection theory: A user’s guide. Psychology press,2004.
[26] D. J. Benson,Music: A Mathematical Offering. Cambridge University Press, 2008.
[27] D. O’Shaughnessy, “Linear predictive coding,”IEEE Potentials, vol. 7, pp. 29–32, Feb 1988.
[28] B. S. Atal, “The history of linear prediction,”IEEE Signal Processing Magazine, vol. 23, no. 2,pp. 154–161, 2006.
[29] H.-C. Tai and D.-T. Chung, “Stradivari violins exhibit formant frequencies resembling vowelsproduced by females,”Savart Journal, vol. 1, no. 2, 2012.
[30] S. P. K. E. Museum,Kolitong: Idiochorde Vollr ̈ohrenzither. MIMO - Musical InstrumentMuseums Online, Nov 2013.
[31] H.-S. Kim, “Linear predictive coding is all-pole resonance modeling,”Center for ComputerResearch in Music and Acoustics, Stanford University.