Digital Healing - An Interactive Sound Healing Instrument Synthesizer

Emma Wang
Department of Audio and Music Engineering
University of Rochester
Rochester, NY, USA
zwang180@u.rochester.edu

***Abstract*—This project explores the integration of sound healing using the Solfeggio Frequencies. It is presented on an interactive frequency-modulating synthesizer instrument with real-time visual feedback. Build in Max/Msp, the product consists of a background frequency generator using additive synthesis, an FM instrument synthesizer and an interactive 3D visualizer. The result is an immersive audiovisual interface that supports meditative and internal exploration.**

Keywords—sound healing, Solfeggio Frequencies, FM synthesis, Max/Msp, interactive sound, audiovisual

I. Introduction

The nine Solfeggio Frequencies, used in many ancient wellness practices, refers to specific tones that vibrate at predetermined precise frequencies that are believed to help healing. These frequencies form the foundation of modern sound healing therapy. Sound healing is a therapeutic practice that uses sound to activate relaxation response of the body. Previous studies have suggested that there had been intentional moves in the therapeutic industry to set the instrument reference tone to 444 Hz in order to include 528 Hz, one of the wide-accepted Solfeggio Frequency for healing and transformation, in the musical scale [1]. This move represents the use of Solfeggio Frequencies on dissolving emotional and physical trauma from a subconscious level.

In less professional settings, Solfeggio Frequencies music is produced and shared by online creators. For example, such content has become a popular genre on YouTube, often uploaded as a video series, or live streams. However, these contents are rarely customized and lack interactivity.

Recognizing that the body often signals its own needs, incorporating interactivity into sound healing may improve its effectiveness. This means that people are able to consciously adjust frequencies and rhythms of the music according to their involuntary psychological responses. This objective also aligns with the principle of biofeedback therapy, which teaches the patient to voluntarily control involuntary bodily functions like heart rate to achieve a reduction in stress disorders [2].

With this goal in mind, I aim to develop an interactive instrument synthesizer in MAX that allows users to freely play over a fully customizable background mix of Solfeggio Frequencies. The timbre of the synthesizer is designed to emulate instruments commonly used in sound healing, such as bells and singing bowls, and it's also tuned to the Solfeggio Frequencies to maintain the harmony of the soundscape. There is also an added visual feature that is directed by the performance.

II. Applied Throeies

The product consists of three modules- The instrument synthesizer, the Solfeggio Frequencies slider player, and the visual animation, all customizable and interactive. Each of these features will be discussed in the following sections.

A. *Additive Synthesis - The background Solfeggio Frequencies slider*

This module employed additive synthesis to generate a thickened and rich drone-like tone. The module consists of nine sliders corresponding to the Solfeggio Frequencies, each generated using the mc.cycle~ objects. The multichannel object creates ten parallel sine waves per channel for each frequency, resulting in a fuller and immersive sound.

The frequencies are later scaled and sent to mc.mix~ object to be mixed down to two channels, and sent to stereo output. The collected nine frequencies are again packed and summed, which blends the signals into a single composite output. This whole process allows the users to create a dense and harmonically rich background soundscape through different slider combinations.

B. *FM Synthesis - The Instrument Synthesizer*

Frequency modulation synthesis is the process of using a "modulator" to change the timbre or pitch of the original signal, the "carrier frequency" [3]. In other words, given a base waveform, FM synthesis is about using another waveform to produce a third waveform that's sent to the output. The core idea of this synthesizer is to build one carrier operator and multiple modulators with adjustable parameters. The website Cytamics represents the basic form of an operator/modulator with a block diagram, which goes in the order of input, oscillator, amplifier and output. According to FM synthesis, by adjusting the relationship between the carrier and modulating frequencies, the modulator will create harmonic or inharmonic complex frequencies.

For the carrier operator, the input is received and controlled through a kslider object. The key is tuned by orderly mapping the Solfeggio Frequencies to a coll object with five "octaves" in total. These frequencies are sent to a cycle~ object to generate a signal, which acts as the carrier oscillator. One additional component for the carrier operator is a function envelope that shapes the amplitude of the signal over time. The envelope information is sent through line~ object and gets multiplied with carrier frequency oscillator value. The resulting signal is then sent to the amplifier, a gain slider, to be scaled for output.

There are two modulator operators included in this synthesizer, with the first one being a low-frequency modulator (LFO), which continuously modulates the signal amplitude overtime to produce tremolo effect. In this modulator, a cycle~ object acts as the LFO oscillator. Its frequency is controlled by the "LFO Rate" dial, which determines how fast the modulation cycles. The oscillator generates a signal between 0 to 20 Hz. The other dial "LFO Depth" scales the cycle~ output to control how strongly the LFO affects the carrier signal. Namely, higher LFO depth means more deviation from the carrier frequency. The modulated signal is then added to the carrier frequency oscillator with a +~ object. The summed signal becomes the new input to the main cycle~ object.

The LFO modulator employs the classic operator structure and is fed into the carrier signal to achieve the purpose of modulation. In this case the LFO modulator introduces periodic amplitude variations that create calming or dynamic texture depending on user control. Combined with background soundscape, the LFO modulator adds subtle nuance to the integrated sonic creation without changing the signal's harmonic structure.

The second modulator in this synthesizer is a harmonic modulator, which directly modulates the phase of the carrier frequency. Within the subpatcher, the carrier frequency is multiplied by a harmonic ratio that is controlled by the "Harmonic" dial. The output is sent to another cycle~ object as the oscillator of the harmonic modulator, and then gets scaled and connected to the phase inlet of the carrier's cycle~ object. In this case, by modulating the harmonic dial, the modulator generates additional sidebands that alters the overall timbre. The harmonic modulator is intentionally used to emulate the brighter and dissonant texture emitted by a bell that rides above the base soundscape.

Unlike the LFO modulator that only affects the movement of sound perceptually, this harmonic modulator enables a wider timbre range by reshaping the frequency waveform itself, giving the users more expressive and dramatic control over the instrument.

C. *Reverb unit - gigaverb*

To enhance the massive and immersive feel of the synthesized signal, this project also incorporates gigaverb as a reverb processor. Gigaverb is an open-source reverb algorithm that's written by Olaf Matthes. The natural-sounding effects it gives is particularly suitable for creating ambient and healing soundscapes. In this project, the output of the FM synthesizer is patched into gigaverb, aiming to mimic the spatial environment where instruments like singing bows or gongs are played. Room size, reverb time and damping are the three chosen parameters of gigaverb to control how spacious or diffused the sound feels.

The gigaverb model is based on a combination of Feedback Delay Networks designed to simulate the complex sound reflections in large acoustic spaces. It essentially used a 4x4 householder matrix within its FDN structure. This matrix is derived from the identity matrix and a vector of ones [4], which ensures that each delay line output is evenly distributed and fed back into all other delay paths. As a result, the FDN enhances a uniform and dense reverberation. Together with the FM synthesis unit, gigaverb creates spatially rich ambient sonic quality that allows the synthesized tones to linger and blend in a way that evokes a meditative atmosphere.

D. *FFT and Jitter objects - The interactive visual*

The next step is to address the objective of producing sound-related interactive visuals. Inspiration for this came from Umut Eldem on YouTube, which connects auditory and visual experience through spectrogram creation [5].

To provide a real-time frequency domain visualization, Eldem includes a scrolling spectrogram module built using pfft and Jitter objects. The audio input is captured with the ezadc~ object and analyzed using a pfft~ object with a 512-sample window size. Within the pfft~ subpatch, the incoming signal is analyzed window by window using fftin~, which splits the signal into real and imaginary components that are later converted to amplitude values using cartopol~ object. These values represent how much energy is present at each frequency bin. The output is written into a 2D Jitter matrix using jit.poke~, with X-coordinate being constant zero and Y-coordinate being the bin index. This stores each FFT frame as a vertical slice of data. Fftinfo~ is used to set the matrix dimension dynamically by reporting the window size and helps configure the matrix with the correct height. The frequency-domain data is sent out through the jit.matrix object. Using jit.convolve, each new FFT frame updates the matrix by moving existing data to the right and writing a new column on the left, which creates a scrolling spectrogram effect.

The resulting spectrogram matrix is passed into a jit.gen patcher, where it's spatially mapped onto a 3D sphere. To move beyond this idea of Eldem, a custom color adjustment system is created. The system blends red, the warm palettes, and blue, the cool palettes, using the mix operator. Param object is used to be able to control the palette values outside of the jit.gen subpatcher. The gradient - color blend amount - is calculated based on each pixel's Y-position, namely the amplitude data. By adjusting the meters, the users are able to customize the color display to, for example, have a warmer color for higher frequencies. In addition to color blending, this project also introduced displacement scale parameters, which allows the user to control the intensity of spectral amplitude. This is done by modulating the scale factor of the Z-axis position of each vertex on the sphere, thus determining the average height of spectral amplitude. With all these additional parameters added in, the user is able to make the sonic textures of both their instrument preset and the background soundscape appear to ripple across the sphere surface in real-time. The display is then rendered in jit.world.

The last part of the patch expands to the creation of a geometric surface from a jit.matrix object, which in this case used jit.gl.gridshape. This segment builds on a shape manipulation technique inspired by Andrew Robinson on YouTube [6]. A noise matrix is employed and mapped onto a 3D mesh that responds to it with multiple different shape presets. Beyond this design, this project added the jit.expr objects with an input message expr norm[0]. This object plays a subtle but important role here since it introduces a wave-like evolving distortion across the horizontal axis instead of a random noise pattern. Attrui objects are used to achieve real-time user control over the mesh's appearance such as scale, position and dimension using the jit.gl.handle object.

Additionally, a random color generator patch is created and tied to the kslider input. This means that the mesh's color will change randomly each time a new key is pressed. This feature responds to musical performance with instant visual feedback, thus adding another layer to achieve the biofeedback therapy principle.

III. LIMITATIONS AND CONCLUSION

Throughout the development of this project, all the audio and visual modules contributed to the immersive experience it aimed to create. However, several areas need potential improvements. First, the jit.world window is currently floating and needs to be adjusted to fit into the designated space. Its position is also fixed in relation to the screen rather than the patcher. Embedding it directly into the patcher would improve user experience. Secondly, the color control for the FFT sphere was intended to allow manipulation across the full RGB spectrum. Likely due to improper value scaling or patching, there are currently only limited color outcomes. Only red, blue, black and white were rendered, which limits the visual expression. The issue could not be solved due to time constraints.

In conclusion, despite the limitation, the product is still successfully developed in both auditory and visual domains under a real-time performance context. The program is responsive in both domains to allow user exploration with great degrees of freedom.

References

[1] D. Hulse, "Sound Healing using Solfeggio Frequencies," *ResearchGate*, Jun. 2019. [Online]. Available: https://www.researchgate.net/publication/333852911_Sound_Healing_using_Solfeggio_Frequencies

[2] Cleveland Clinic, "Biofeedback," *Cleveland Clinic*, [Online]. Available: https://my.clevelandclinic.org/health/treatments/13354-biofeedback?utm_source=chatgpt.com

[3] Cymatics.fm, "FM Synthesis: Everything You Need To Know," *Cymatics*, [Online]. Available: https://cymatics.fm/blogs/production/fm-synthesis?srsltid=AfmBOooTFr369_QsDB_tOebmmRwboXivlq6D2r7OTYn5Prs49EmMV2IK

[4] J. Dattorro, "Choice of a Lossless Feedback Matrix," *Digital Signal Processing Related*, [Online]. Available: https://www.dsprelated.com/freebooks/pasp/Choice_Lossless_Feedback_Matrix.html?utm_source=chatgpt.com

[5] U. Eldem, "HEARING GLASS: Audio Reactive Spectrogram in Max MSP Jitter," *YouTube*, 2022. [Online]. Available: https://www.youtube.com/watch?v=QMwMO_2V6V4

[6] A. Robinson, "Max/MSP/Jitter 6 - Displacing mesh geometry with jit.gen," *YouTube*, 2013. [Online]. Available: https://www.youtube.com/watch?v=uzQOiwMkESU&t=883s