y. (n.d.). https://vanhunteradams.com/Pico/Helicopter/Display.html
lay. (n.d.). https://vanhunteradams.com/Pico/Helicopter/Display.html

Audio Spatialization Toolkit for Personal Music Production

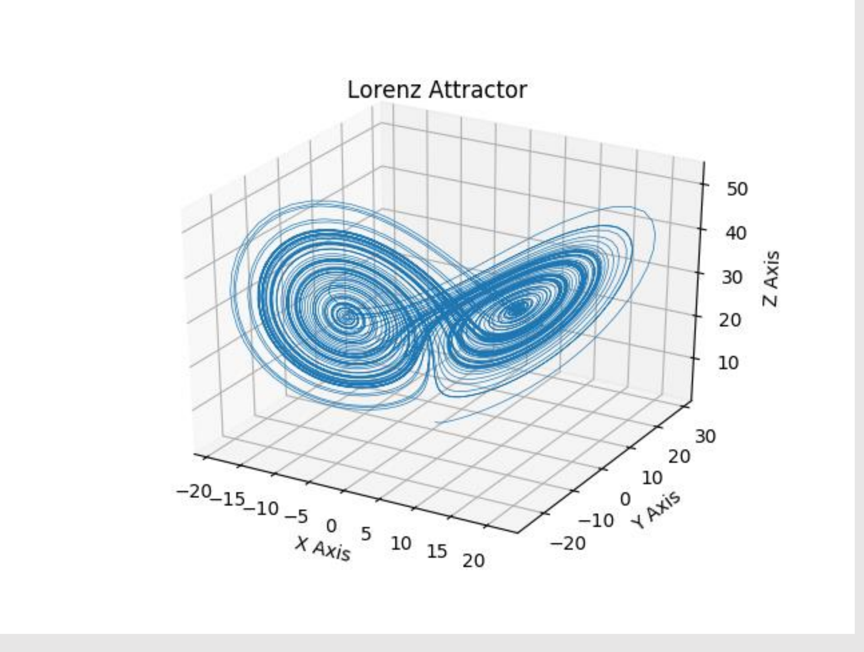
Author(s): Antti Meriluoto; Advisor(s): Hunter Adams

The project was motivated by songwriting

In the age of Digital Audio Workstations, music production has never been more democratic, but purchasing plugins for audio processing can add up quite quickly.

This project aims to develop a parametrizable tool with Python that can manipulate mono tracks with some basic signal processing for creative effects.

The effect I chose to explore as a case study for this project was spatial audio in an effort to encode dynamics into a song I'm producing, namely the famous Lorenz Attractor, shown below.



Implemented entirely in Python

All audio processing was done digitally, so Python and libraries like Wavefile and Numpy proved helpful.

A mono signal can be manipulated to a stereo format for the illusion of spatial characteristics

Based on the dimensions of a user's head, w, the speed of sound, and the sampling frequency of the audio file, I phase shift a mono signal between the left and right channels of the stereo output to simulate the delayed arrival of sound between the user's ears.

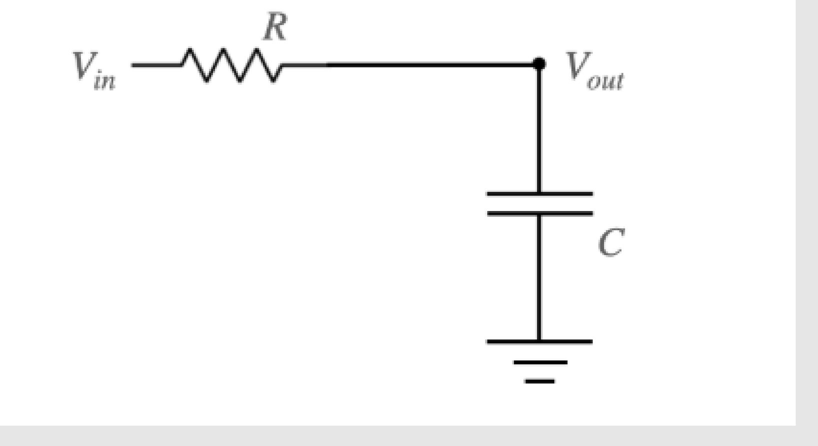
$$sampling\ frequency\ =\ \omega\ =\ 44100\,Hz;\ sampling\ period\ =\ T\ =\ 1/\omega$$

$$speed\ of\ sound\ =\ c\ =\ 343\ m/s;\ w\ =\ 0.1\ m$$

$$channel\ delay\ =\ w\ /\ c\ \approx\ 0.00029\ s$$

$$number\ of\ samples\ to\ shift\ by\ =\ 0.00029\ /\ T\ \approx\ 13$$

I also perform frequency attenuation on the less dominant channel. I use a digital low-pass filter based on a Resistor-Capacitor low-pass filter shown below[1].



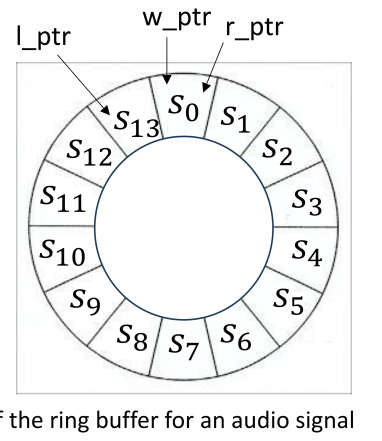
$$V^{N+1} \approx V^N + \frac{\Delta t}{RC}(V_{in} - V^N)$$

Each raw sample represents an input voltage, V^n represents a filtered sample at timestep n; RC is the parameter I can finetune to adjust the strength of the filter. I also perform amplitude attenuation on the channels scaled by the distance of an audio sample from each ear.

A ring buffer automated sample shifts

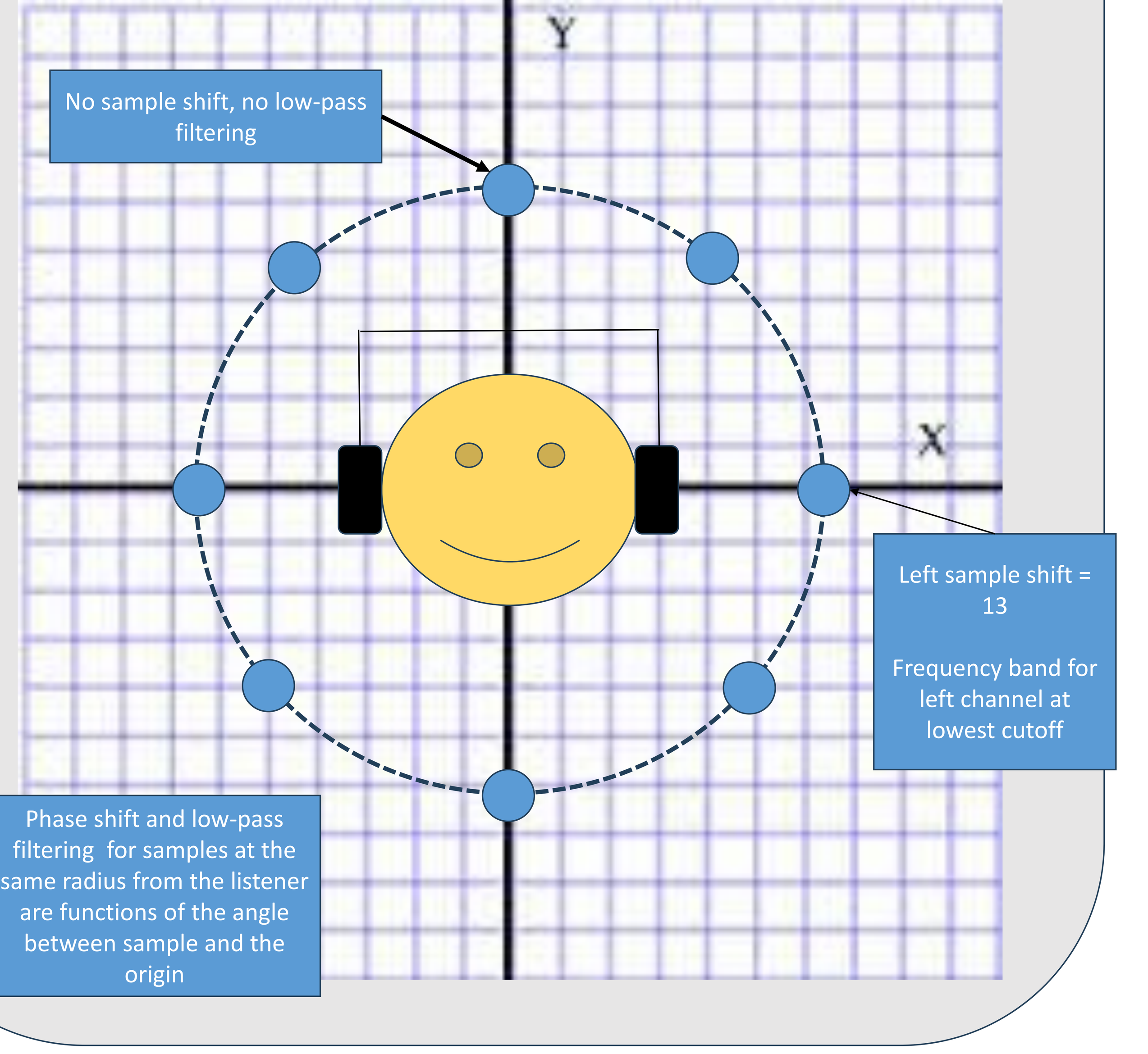
l_ptr and r_ptr point to the sample to insert into the left and right channels at each timestep respectively. They are adjusted based on the "location" of the audio source.

w_ptr shows which index of the buffer to overwrite next.

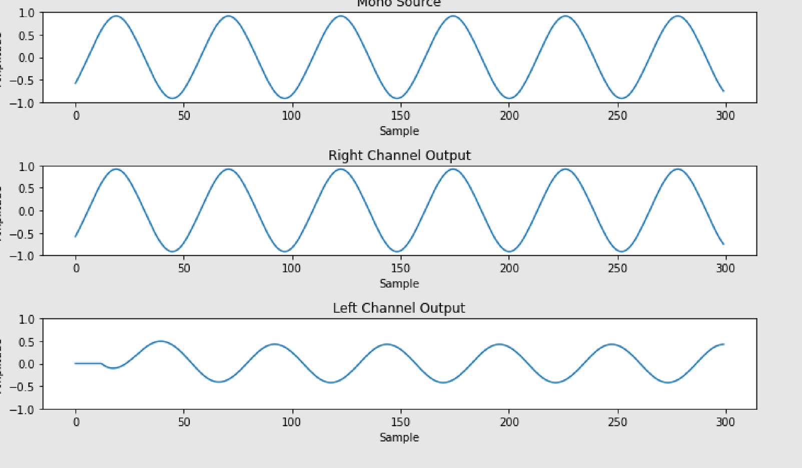


State of the ring buffer for an audio signal directly to the left of the listener at t=13

Parameters are a function of a source's position on xy-plane



The resulting tool satisfies a sense of movement but leaves room for improvement in terms of spatial quality.



The state of each channel's output from an 852 Hz pure tone source to the immediate right of the user

References and Acknowledgments

Thank you very much to Hunter Adams and Bruce Land for their generous help with this project

1. *Adams, H.* (n.d.). *Data Display*. Van Hunter Adams. https://vanhunteradams.com/Pico/Helicopter/Display.html

CornellEngineering
Electrical and Computer Engineering