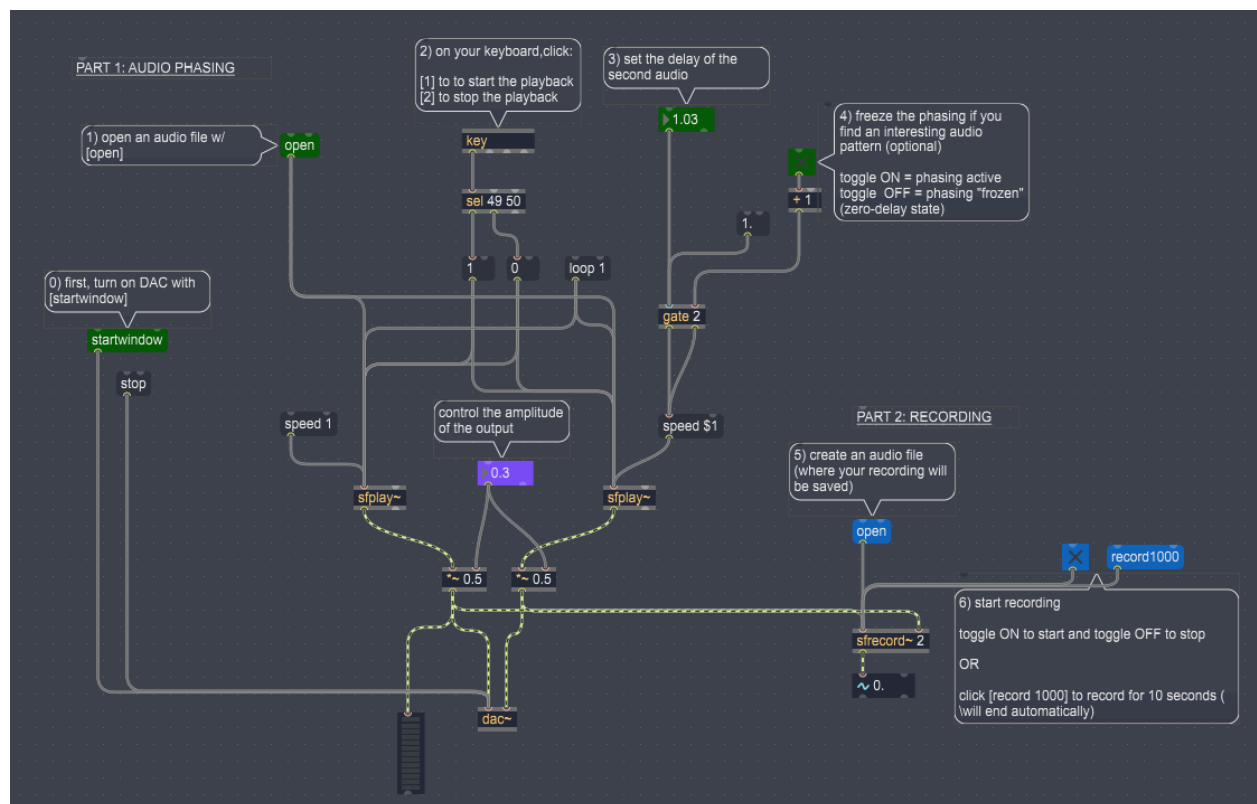


Assignment 3 - Procedural sound phasing device: Emulating Steve Reich's Analog Phasing Process in Max/MSP

Introduction

This patch, designed to digitally recreate Steve Reich's analog tape phasing technique from the 1960s, allows users to load audio files and create phasing effects by gradually shifting one audio stream relative to another, mimicking the process Reich discovered when two identical tape loops played at slightly different speeds.

The patch is organized into two main functional sections: Part 1: Audio Phasing and Part 2: Recording, each handling distinct aspects of the phasing process. This modular design ensures clarity and ease of use while maintaining the core functionality required for authentic phasing effects.

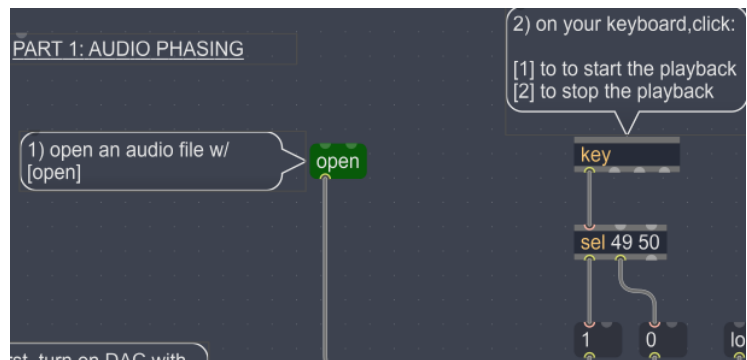


Part 1: Audio Phasing System

Audio File Loading and Playback

The first thing users need to do is load an audio file. I used an **open** message box that opens a file browser when clicked. This connects to objects that load the file into Max's memory so it can be played back.

For playback control, I used the **key** object which detects keyboard presses. When you press "1" on your keyboard, it starts playing the audio file. When you press "2", it stops. The **key** object outputs numbers (49 for "1" and 50 for "2"), so I used **sel 49 50** to separate these commands and trigger the right actions.

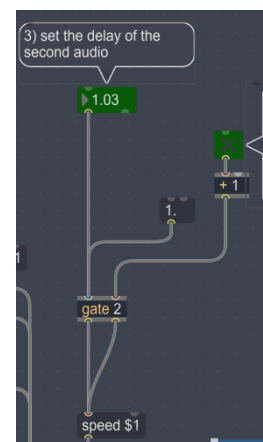


Core Phasing Engine

The main phasing happens by splitting the audio into two identical copies. Both copies come from the same audio file, but one goes straight to the output while the other gets delayed by a tiny amount. This creates the phasing effect because the two copies are slightly out of time with each other.

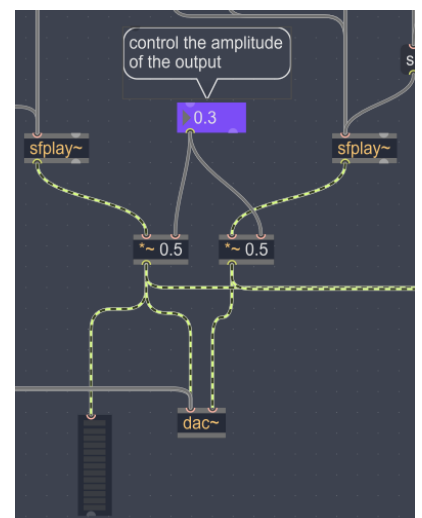
I used **sfplay~** objects to play the audio files and speed to create the phasing effect. One audio output stays at the default speed the the other one can be modified by the user. This makes one copy of the audio run just a tiny bit faster or slower than the other, which is exactly what happened with Reich's tape machines when they ran at different speeds.

The **gate 2** object is used to freeze the audio phasing so that the user could instantly reset the playback speed to default by pressing on the toggle (once they find an interesting audio pattern)



Audio output

Both audio signals get mixed together and sent to the **dac~** object, which is Max's way of sending audio to your speakers. I used **0.3** number boxes to control the volume levels so the mix doesn't get too loud and cause distortion but the user is able to manually control it as well if they wish.



Part 2: Recording System

Setting Up Recording

The recording section uses **sfrecord~ 2** which can record stereo audio (the "2" means two channels). Before recording, you need to click the **open** message in the recording section to tell Max where to save the audio file. This opens a dialog where you can choose the location and filename.

Two Ways to Record

I included two different ways to start recording:

1. Manual Recording: There's a toggle switch that you can turn ON to start recording and OFF to stop recording. This gives you complete control over exactly when to start and stop.
2. Timed Recording: The record 1000 message automatically records for a set time (1000 means 1000 milliseconds, which is 10 seconds). This is helpful when you want to record exactly 10 seconds without having to time it yourself.

What Gets Recorded

The recording captures the same audio that comes out of your speakers - the mixed result of both the original and delayed audio signals. This means you're recording the actual phasing effect, not just the original audio file.

How Everything Works Together

To use the patch, you follow these steps:

1. Turn on audio processing with **startwindow**
2. Load an audio file using the **open** message
3. Set up recording by clicking **open** in the recording section
4. Press "1" to start playback
5. Adjust the phasing speed if needed
6. Use the toggle to freeze interesting patterns
7. Start recording (either manually or with the timed option)
8. Press "2" to stop playback when done

Included Sounds to Experiment With

For this project, I selected three sound files from the open-source audio repository *Freesound.org*. Each of these sounds was chosen for its potential to create distinct rhythmic and textural effects when processed through the phasing patch.

- **Sound01_deepbass.wav** – Created by user HarmonicMess, this sound provides a low-frequency texture that emphasizes pulse and resonance ([HarmonicMess, 2025](#)).
- **Sound02_beepsequence.wav** – Contributed by user AlienXXX, this short, percussive sequence lends itself well to phasing experiments that highlight rhythmic displacement ([AlienXXX, 2015](#)).
- **Sound03_coughing.wav** – Provided by user qubodup, this sample introduces a more organic, irregular texture that contrasts with the mechanical regularity of the other sounds ([qubodup, 2024](#)).

All three sounds are distributed under Creative Commons licenses via *Freesound.org* and were included in the project folder for experimentation.