Trevor LeFebvre

Report on Low-Fidelity Simulation Project

For my project, I focused on implementing a Max MSP patch that would simulate various different qualities of low-fidelity recordings and malfunctioning machines (e.g. turntable). The approaches I took to accomplish this goal were the following:

- Simulating hiss/crackle/pops using Max objects
- A wind-up and wind-down effect to simulate turning on and off a record player while the needle is down
- Skipping record effect
- Pitch/speed warping with randomization to simulate the warping of an LP and/or problems with the motor of a cassette player or turntable

Although my project does not simulate these qualities as accurately or effectively as I would have liked, I can imagine that with some edits and some more focus on being user-friendly, this could become the basis for a useful piece of software to accomplish foley work for independent and/or student film-makers. I will mention some of the various improvements that I had in mind that could turn this into a more sophisticated piece of software.

Hiss/crackle/pop/distortion Simulation

To emulate the different types of noise that can occur in low-fidelity recordings (e.g. tape hiss, surface noise on a vinyl record) I used four different "sub-patches" that use noise~, pink~, and rand~ objects. Traits shared between all these sub-patches is that they all use high-pass-filters using the svf~ object (some of these simulations sound less realistic when they contain lower frequencies), they are all attached to live.gain objects for mixing volume to taste, they all use clip~ objects to prevent audio from getting excessively loud, and they are all fully stereo.

Clicks & Pops Simulation

- This sub-patch is meant to replicate the random clicking and popping that you hear on an LP. I use rand~ objects to send impulses to the left and right channels, use overdrive~ to get them to sound more robust, and random + metro objects for creating variation in the dynamics of the clicks. Since this is controlled by random excitation, there is no uniformity to the clicks and is not good for simulating the types of scratches on

records that cause uniform clicks (determined by the RPM). I included 2 sliders to provide a “fine-tuned” and “coarse” control of the frequency of the clicks.

Hiss Emulation and Crackle Simulation

- These sub-patches are constructed in effectively the same way as each other with the exception of having different ranges for the input parameters. These patches work by multiplying a noise~ signal by two pink~ signals with differing intensities (determined by constants controlled by two sliders). The difference between the two patches is that the hiss emulation allows for the pink noise signals to be multiplied by larger constants (2.0-100.0) whereas the crackle emulation allows for those signals to only be multiplied by smaller constants (0.0-2.0). The high-pass filter comes in handy for these sub-patches especially because without it being used, these sound more like a white noise wall and harsh noise than the type of hiss/crackle that appears in low-fidelity audio recordings.

Distortion/Clipping Simulation

- This sub-patch is implemented using the noise~, pink~ and sah~ objects. The noise~ and pink~ objects send their signals into a sah~ object which is used to allow the pink noise signal to determine when the noise signal gets outputted. This results in a signal that is similar to the clicks & pops simulation but more aggressively nonuniform, making it sound almost like digital distortion or clipping (at least in my opinion -- the concept of simulation often involves subjective perception to a certain degree). I included one parameter to control the magnitude of the pink signal which controls how often this distortion is present. Note that it is rare for distortion to be present when the slider is set to lower values (below approximately 0.8).

Overall Improvements of Noise Simulations

- The main drawback to this approach is that it offers a somewhat limited range of textures to select from. Two different solutions to this problem would be to create a greater variety of noise/distortion simulations to select from or to give the user a larger number of effects/parameters to apply to the simulations. An example of the former idea could be to include more sub-patches with semi-arbitrarily chosen configurations of different noise~ and pink~ objects. The rationale behind this is that noise signals will typically contribute to a signal being

harsh or chaotic and these types of signals can often be leveraged into sounding like a type of machine malfunction or sonic irregularity using effects (e.g. low-pass filter, high-pass filter, etc.). However, since my project goal is slightly more concerned with creating a product to be used by people with little to no experience in sound design (e.g. independent film-makers with no budget to hire foley artists), adding too much complexity to the distortion simulations could make for an overwhelming and overall less desirable product.

- It would have been desirable to implement a different type of clicks & pops simulation for simulating the types of scratches on records that cause uniform clicks (i.e. determined by the RPM).

WIND-UP/WIND-DOWN

This feature is intended to simulate the way that a mechanical motor in a turntable takes time to start up and slow down to a halt when turned on and off. My implementation is fairly simplistic, using a line~ object to create a ramp that changes the playing speed of the sample. For wind-up, we want to start the loop and bring the playing speed from 0 to 1 over a duration of 1 second and for the wind-down, we want to bring the playing speed from 1 to 0 over a duration of 5 seconds and turn off the loop. These durations were selected using my own personal judgement because this is not something that is consistent across machines (i.e. one record player might have a wind-down time of 2 seconds while another might be more like 5 seconds).

Something that I could have put more research into is the shape of the curve for the wind-up/wind-down. I found that a linear curve sounded moderately realistic, but it’s unlikely that a record player or cassette player would follow a perfectly linear curve exactly. Thus, it would have been more flexible to use the curve~ object and to add sliders for easy manipulation of the curvature of the ramp. Another parameter that would be useful to control is the duration of the wind-up/wind-down. These enhancements would give the user much more control over the quality of this feature.

Lastly, it would make for much more sophisticated simulation to have clicks and pops (What I implemented and ones determined by the RPM that I mentioned but did not implement) wind-down as well. This could be accomplished by sending the ramp used for the wind-up/wind-down to the frequency input of the rand~ objects

outputting random clicks and a metro object used for regular clicks. It would also be desirable to have a way of affecting the amplitude and timbre of hiss and crackle simulations during wind-up/wind-down.

SKIPPING RECORD

I implemented the simulation of a skipping record using a metro object connected to the groove~ objects that are used for the sample of the audio that is to be processed and a skipping record sound. Ideally, there should be a default sample bank for the user to select a skipping record sound from as well as an input to allow for the user to provide their own sound. While I managed to get this feature to work on a minimal level, it would have been much more versatile if I had allowed for it to loop over a part of the record other than the beginning of the loop. This would require calculating the time at which the skip record button is clicked and sending the appropriate loop start time to the groove object (or an entirely different approach). Furthermore, it would have been an interesting feature to allow the randomization of the onset and duration of skipping so as to add a generative aspect to this program. Lastly, it would have made more sense to connect the warping to this feature to change the duration of the skips dynamically (i.e. increments between skips are longer when the audio is warped below the original speed).

WARPING

I implemented warping in 2 different ways which can be used simultaneously or independently. Firstly, I gave it a wavering/vibrato effect by processing the chosen sample speed with a cycle object and sending that to the sample playback increment input for groove~ object that is used for the sampled audio. The user is given the option to control the warp depth and frequency of the warping sine wave and also given default times for turntable speed simulation (33RPM, 45RPM, and 78RPM). At higher warp depths and vibrato speeds, the discrete nature of the changes in sample playback speed become more apparent, suggesting that there might be issues with the value representing the desired playback speed getting updated fast enough or the groove object isn’t meant to handle continuous alterations to playback speed (my guess is the former since wind-up/wind-down don’t sound like they have these issues). However, at lower warp depths and warp speeds, this sounds much more fluid and can provide results that range from

slight vibrato to severely warped audio that simulates motor issues in a turntable.

The second way that I implemented warping is with the drunk object, which connects to the groove~ object's sample playback speed and randomly increments it upwards or downwards. I gave the user coarse options for changing the magnitude of these random steps and a number box to change the speed at which these random steps occur. Ideally, it would make more sense to allow for finer control of the step sizes, to add the option to have it tend to vary towards the center (so as to prevent getting stuck at high or low warp depths), and to possibly even have it affect the sample playback speed with ramps rather than discrete changes (so as to allow for smoother changes).

Lastly, it could have been rewarding to go to greater lengths to simulate the behavior of faulty motors in a turntable, more irregular types of warping in vinyl, or the type of warping that can occur with loose tape in a cassette. This could probably be accomplished to some degree of success using amplitude and frequency modulation parameters with cycle~ objects, or we could simply provide default waveforms for the user to select from that come close to replicating these effects.

HISS LOOP

I also gave the user the option to input their own hiss/crackle loops into a groove object so as to provide the user with an easy means of inserting real noise to accompany the simulation patches. It would be particularly useful to provide a sample bank for the user to choose from in this case (similarly to the record skipping noises).

OVERALL IMPROVEMENTS

While this report already borders on providing more insight into the program's limitations rather than its capabilities thus far, I thought I would provide a list of further improvements unrelated to the functions that I have already implemented. Some of these ideas relate to user-experience and ease-of-use but most of the ideas elaborations of the program to make for a more holistic approach to simulating low-fidelity (including cassettes and digital audio).

- Specialized wind-up/wind-down, rewind, and fast-forward simulations for cassette tape (clicking noise + rapid forwards and backwards audio + ramping of speed change).

- A simple and effective way of planning parameters changes (e.g. skips and wind-downs) without real-time parameter manipulation as well as the recording/saving of the output.
- A greater degree of control over whether the audio is stereo or mono (for instance, wouldn't make sense for a mono LP to have stereo surface noise). Ideally, this could be accomplished with a single button.
- Including an easy-to-use filter (possibly with biquad~) with some default options would be useful since low-fidelity recordings often sound muffled (lack of bass or treble frequencies).
- Cassette tapes often sound rather compressed, so a simple compressor with default options to replicate tape compression would be a logical addition.
- Lastly, it would be reasonable to add forms of digital distortion such as sample rate reduction or "bit-crushing" so as to simulate low fidelity in a digital context.

Bibliography

_mark. "Techniques for Generating Clicks, Pops, Hiss and Noise." *Lines*, 16 Apr. 2020, llllllll.co/t/techniques-for-generating-clicks-pops-hiss-and-noise/31314.