MIT Student

21M.380

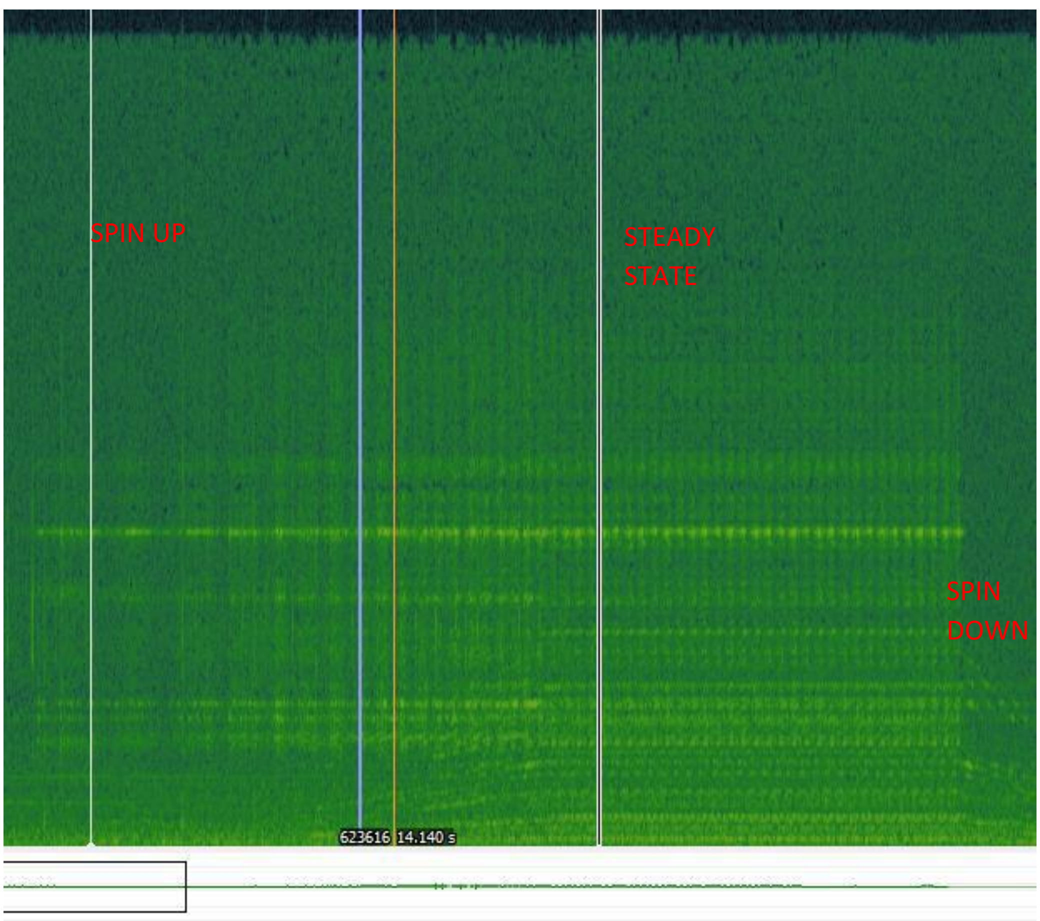
Sound Design

Final Project Part 4: Final Implementation

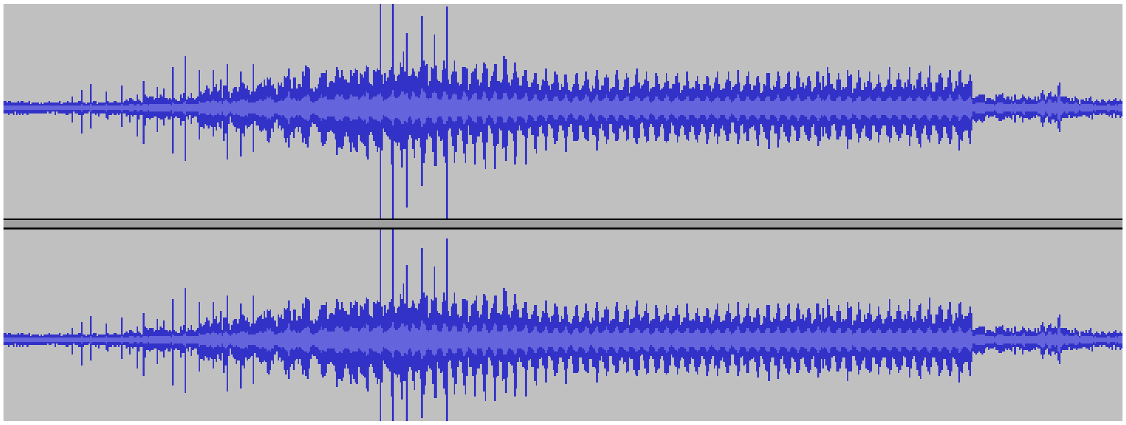
Description of project:

My sound design project focuses on recreating the sound of an electric motor, specifically a Mitsuba 2-5 horsepower brushless hub motor.

Initially, I took a recording of the motor and took the spectrogram of it. In the spectrogram, you can easily the increasing spin up, the steady state, and the spin down. Additionally, you can see that there are several harmonics present, and the sound also pulses in time. Listening to the recording also showed that there was a base pitch or hum of the motor, several harmonics on top, and lots of metallic scrapping and pulsing sounds.




**Spectrogram of Mitsuba motor, depicting Spin Up, Stead State, and Spin Down**



**Stereo waveform plot from Audacity taken from same sound sample as in the spectrogram above, showing pulses in time.**

I wanted to model the distinctive spin up and spin down sound of the motor, as well as the steady state noise. I was especially interested in getting the right combination of harmonics and base frequencies, as well as the pulsing that you can hear as the motor completes its revolutions. The main control parameters I decided on for how the motor are its max speed that it spins up to, how fast it spins up to speed, and how long it is operating in the steady state before it spins down.

The pitch envelope is modeled after the one given in *Designing Sound.* The spin up is modeled as a pseudo-exponential rise, to mimic the fact that the motor has to overcome inertia when getting started. However, unlike the envelope in the book, I wanted to have independent control over the rise time, the steady state time, and the fall time, as these are variables somebody operating the motor in real life would have access to.  I used a RISETIME slider in the main GUI to control how long it took the motor to spin up. The motor will then stay in steady state for as long as the user wants, until the user hits the SHUTOFF_MOTOR switch, causing the motor speed and pitch to linearly decay. The decay is linear, just as in *Designing Sound*, because at this point the motor is simply loaded with a constant value. However, with this envelope, the user can choose to 'turn-off' the motor, causing it to spin down, whenever they want.

After modeling the envelope, I turned towards the pitch of the motor. Initially I was going to copy the motor patch in *Designing Sound*, which had separate patches for rotor sounds and stator sounds. However, the rotor sound was too raw and tinny, and was more suited to a handheld motor device, like a power drill. Additionally, the motors in *Designing Sound* are DC motors, so there was more emphasis on brush noises than on pulsating sounds due to friction on bearings or motor housings. I eventually decided to use the more pitched stator sounds for overtones, rather than the fundamental frequency, and simply added two of them together to get two different overtones. By tuning the amplitude of the outputs correctly, they add a reasonably convincing high pitched whir on top of the rest of the motor noises, and are also shaped by the pitch envelope that models spin up and spin down.

The other main pitched noise on the motor is the rumble, produced in rumble~. This sound was actually based on four-stroke engine patches, but is suitable for this motor because there are lots of pulses audible due to random misalignments or eccentricities in the revolution. I hardcode the rumble to be at a reasonable frequency with respect to the overtones. The pitch is set by the max speed determined by the user, as well as the pitched envelope that also shapes the overtones.

The next step was to create the pulses represented the revolutions of the motor. I did this using a modified FMpulse~ patch. In this patch, the 'modulator frequency' sets the speed of the pulses, while the 'FM index' sets the pitch of the pulses. This is then multiplied with the overtones and the rumble to get the distinctive pulsing sound that shapes the motor whir. The modulator frequency comes from the pitch envelope, so that the pulse speed is synchronized with the pitch rising and falling as the motor spins up and then drops off.

The final step was to add the mechanical scrapping and cogging sounds. I used a modified patch from *Desinging Sound* that was originally intended to mimic a broken engine. The patch output stuttering sounds that were modulated with random noise. By removing the random noise, I was able to just keep the pulses, which were deeper and more mechanical feeling than the shaped sounds produced by the modified FM patch. This is also multiplied by the pitch envelope, which actually serves to speed up and slow down the timing of the pulses to mimic the motor spinning up and slowing down, similar to the FMpulse~ patch.

The parameters exposed to the user are the max speed attainable by the motor, which relates to the highest pitch; the rise time of the motor, which relates to how long it takes the motor to get up to speed (and up to the highest pitch), as well as the speed up of the pulses, and when the motor 'shuts-off', resulting in a linear decay of the overall pitch and sound. There is also a general overall volume control available as well.

I would still like to fine-tune the harmonic overtones to mimic the full spread of the real motor. Additionally, while the mechanical sounds have realistic pulses, I'd have liked them to sound a bit softer and more like scraping – they are a little too deeply pitched. In the real motor sound, they almost sound like sandpaper, rather than hits.

Overall, though, I'm happy with the different layers of pitched and mechanical sound, as well as the independent control of spin up, steady state, and spin down.

**MAIN PATCHES:**
-main (GUI)
-overall_motor_env: creates envelope to shape pitch (of rumble and overtones) and speed of pulses (of FMpulse and cogging)
-rumble~: lower motor pitched sound
-overtones~: higher pitched harmonics on top of rumble~
-FMpulse~: pulse generator to shape pitched sounds into pulses; speed shaping from overall_motor_env
-cogging~: mechanical pulses (due to friction, etc.) , speed shaping from overall_motor_env

**MAIN PARAMETERS:**
**-**risetime: time to spin up, also affects time to spin down linearly (total spin down = ½ * risetime)
-max-speed: highest speed reached by motor (determines highest pitch)
-volume: overall volume control
-shutoff_motor: start spin down, i.e. turn off motor