Quadcopter Simulator

21M.380 Final Project

Description

The goal of this project was to synthesize the sound of a quadcopter's flight, from the perspective of the quadcopter itself, as is common in first person view (FPV) flight.

A quadcopter, shown above, is somewhat similar to a helicopter, but it has four rotors in the same rotational plane. It sounds like a helicopter, like its propellers are chopping the air, but the pitch is much higher because the propellers are spinning much faster than full-size helicopters' propellers. The buzzing sound coming from the propellers sounds almost like a swarm of bees buzzing.

A traditional quadcopter controller has two joysticks which control throttle, roll, pitch, and yaw. The goal was to expose a similar user interface to the user to control the sound in Pure Data.

Design process

Real-world example and analysis

See `sample.wav` for an example of what a quadcopter sounds like. This quadcopter was running with a 4S lithium polymer cell (16.8 volts) and 920kv propellers.

The only sound I replicated was the sound produced directly by the quadcopter itself and not things like the wind (which you can hear in some parts of the recording). In the original proposal, I was planning on incorporating wind effects, but I decided to focus on producing a high-quality motor and propeller sound.

In my acoustic scene, the observer is actually positioned on the quadcopter itself (as in the recording). This isn't realistic in the sense that humans can't actually sit inside a quadcopter. However, many people fly quadcopters using first person view (FPV), where they watch a live video feed from a camera attached to the quadcopter they're flying.

The only sound source is the motor and propeller. The quadcopter control system makes the motor spin at different speeds depending on pilot input, and the differences in speed make a big difference in what is heard (as you can hear in the recording).

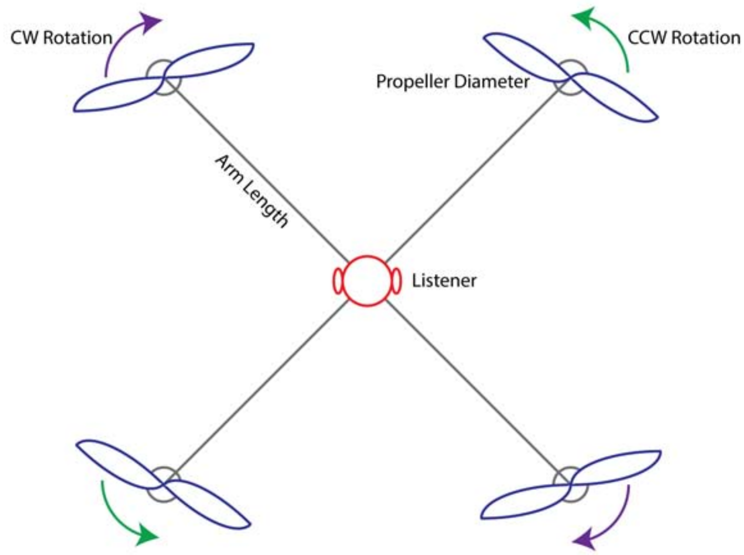
Requirements speci cation

I planned on allowing the user to control all parameters that can be controlled in a real quadcopter. The idea was to expose a joystick-like interface, just like real radio transmitter, allowing the user to specify throttle, roll, pitch, and yaw. In addition, the user should have been able to specify a couple other parameters like the battery type and motor type.

Originally, I was going to allow the user to specify a couple other parameters. I decided not to expose controls for propeller diameter because the diameter is essentially set by motor type: slow motors are paired with large propellers, and fast motors are paired with small propellers. It didn't make sense to expose controls for the two independently when the user could already control motor Kv rating (which essentially translates to speed). I also was planning on allowing the user to specify quadcopter arm length, but that didn't seem to make much of a difference in the sound, especially when the length varied over such a small range (from about 10cm to 30cm).

Research and model

My model was fairly simple.



In order to produce a convincing and realistic sound of a quadcopter, I had to understand the details of the quadcopter control system and how the motors functioned.

The quadcopter controller computes motor commands in order to produce the effect desired by the pilot. In a real system, this is done using closed-loop control (PID loops), but in the case of my sound synthesis, I could get away with using simpler math to produce a realistic-sounding effect (if I connected my controller to real propellers, it definitely wouldn't fly, though).

Quadcopters use brushless DC motors. Unlike brushed DC motors, which use brushes to implement mechanical commutation, brushless DC motors are electronically commutated, so there are no sparking sounds like brushed DC motors. They still do make a whirring sound, though, because of the rotor and stator. Additionally, the propeller, by pushing air around, makes a sound too.

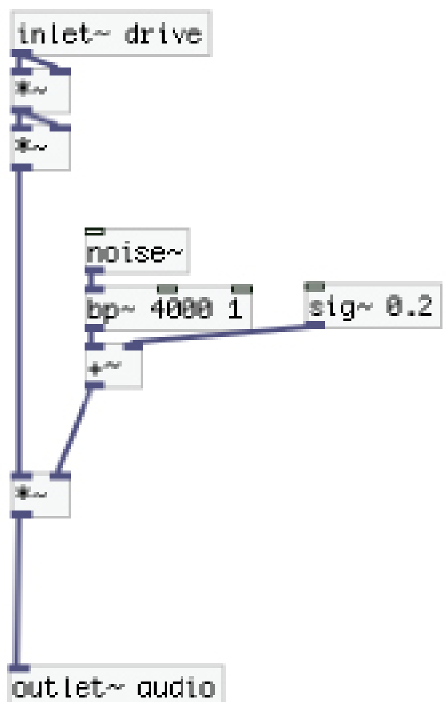
My model turned out to be fairly good. When implementing it, I didn't really have to revisit and change any fundamental things in the model.

There were some small key details that made the implementation sound a lot more realistic. DC motors spin up a lot faster than they slow down, because they can be driven to spin up, but they rely on friction to slow down. Originally, I was not simulating this effect and simply modeling motor speed using a low pass filter at a single chosen frequency, so the motor would spin up just as fast as it slowed down. Once I changed that, though, to spin up faster and slow down slower, it made it sound a lot more realistic.

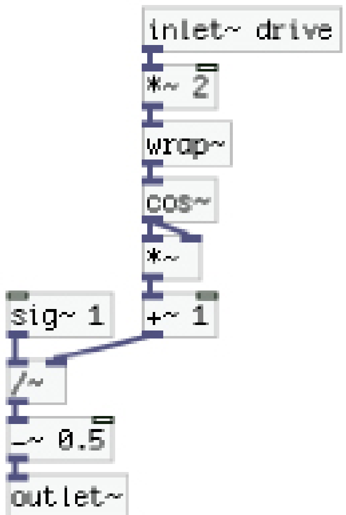
Method selection

I used different sound synthesis techniques for the three parts of the motor / propeller system I was interested in synthesizing: the rotor (makes a rattling sound), the stator (makes a high-frequency whining sound), and the propeller (makes an airy noise).

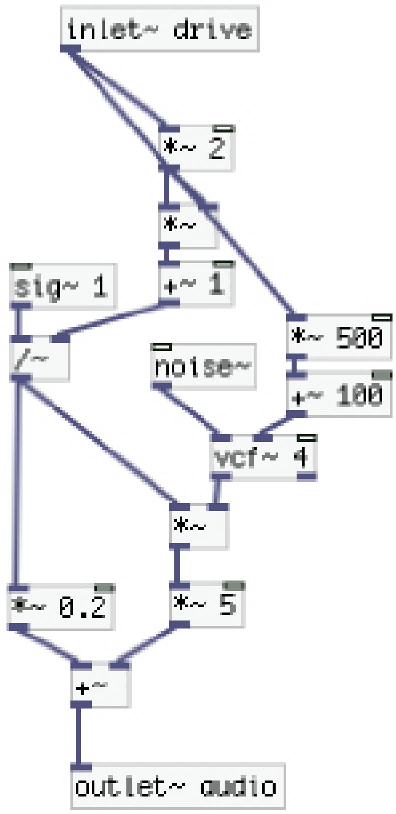
I built the rotor using waveshaping, turning a filtered noise source on and off. The following patch was driven by a phasor at the rotational frequency:



The sound of the stator whine was done using a cosine raised according to $y = 1/(x^2 + 1)$. The system was driven by a phasor at the rotational frequency:



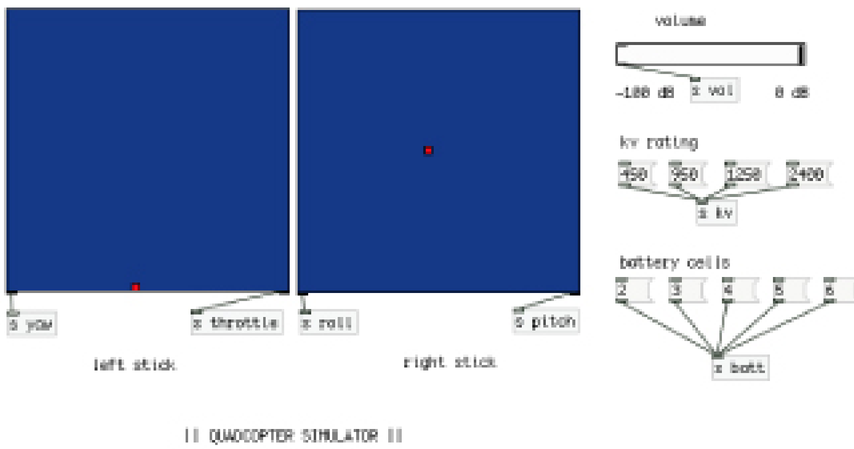
The sound of the propeller was created by shaping a filtered noise source by an oscillator at the rotational frequency.



Implementation

The implementation was done using Pd extended so I could make use of the `grid` object to create the joystick interface.

The interface looks like this:



This is accessible in `main.pd`. The details of the implementation are described in the rest of the Pd patches in the folder.

Conclusion

The implementation was successful! It presents the user with an interface familiar to quadcopter pilots, and it produces a believable sound! One thing that surprised me was how little details make a big difference: for example, implementing the fast speedup and slow spin down made a big difference in the believability of the sound.

I don't believe there is anything particularly bad in the current implementation. I think the propeller sound could use some improvement; I think the rotor and stator sounds are pretty good as-is.

In an improved implementation, I would want to work on modeling the effects of wind interacting with the propeller blades. This would probably involve a more sophisticated model of quadcopter flight itself because wind-related sounds depend on the quadcopter's position and velocity in space. Perhaps this could be worked around by integrating with existing flight simulator software and only doing the sound synthesis portion, getting the position and velocity data from the simulator. In addition to developing a better model of quadcopter flight, it would involve developing a good model of propeller-wind interactions. This could be an interesting challenge, though.

21M.380 Music and Technology: Sound Design
Spring 2016