Final Report: Aerial Robotic Choir

Aditya Ghadiali
Robotics Institute
Pittsburgh, PA 15213
aghadial@andrew.cmu.edu

Saurabh Nair
Robotics Institute
Pittsburgh, PA 15213
snnair@andrew.cmu.edu

Flavian Pegado
Robotics Institute
Pittsburgh, PA 15213
ffp@andrew.cmu.edu

Abstract

The development of robotics has opened new avenues for the entertainment industry. Through this project, we aim to explore one such avenue: the creation of a musical quartet [1] of quad-rotors that "sing" musical tones by controlling their throttle with feedback. The rotation of quad-rotor propeller blades generates an acoustic noise; the fundamental frequency of this sound can be controlled by varying the speed of rotation. While typical applications in robotics would aim to minimize this sound, we look to sense and process it, using it as an input to the controller. The proposed solution involves the use of quad-rotors which are tethered to the ground using an elastic material to allow a wide range of throttles (and therefore musical tones) while keeping the robots within a constrained area. The result would be a visual and auditory spectacle, demonstrating the confluence of robotic control and music.

1 Introduction

Quad-copter controls and dynamics are an interesting and constantly evolving problem in robotics. The proposed project involves the precise control of quad-copters to create a musical sound; bringing together technology and art, a theme which has come to fore with recent advances in robotics and technology.

1.1 Art and Robotics

On Feb. 6, 2017, Intel pulled off an impressive 300 unmanned drone light extravaganza during the Super Bowl LI half-time show [8].This presentation stands witness to the nexus of robotics and the art industry. These demonstrations are just the tip of the iceberg;there is a considerable ongoing research in the field of quad-rotors and entertainment devices. Recently, Norwegian pop star Aurora teamed up with YouTube and Carnegie Mellon for a music video starring some really surprising backup dancers: drones [9]. The quad-copter choreography was programmed by Ali Momeni, associate professor of art at Carnegie Mellon University, and Ellen Cappo, a PhD student in robotics.

1.2 Premise and introduction to music

The pitch of a sound generated by musical instrument is the fundamental frequency of a note. The pitch usually associated to a note. Notes go from A to G in semitone intervals(Except B-C and E-F). An octave contains 12 notes in total. To move an octave higher, we need to double the frequency being played. For example, the note A4 (the note A on the 4th octave) has a fundamental frequency of 440Hz. The fundamental frequency of A5, the A an octave higher, is 880Hz. A melodic phrase is a combination of notes or pitches held for different times in a specific order.The same notes repeat

[1]**Quartet** In music, a quartet is an ensemble of four singers or instrumental performers; or a musical composition for four voices or instruments. For the scope of the course, we will be addressing only a chord or a set of note progressions on one quad-copter played in preprogrammed timing.

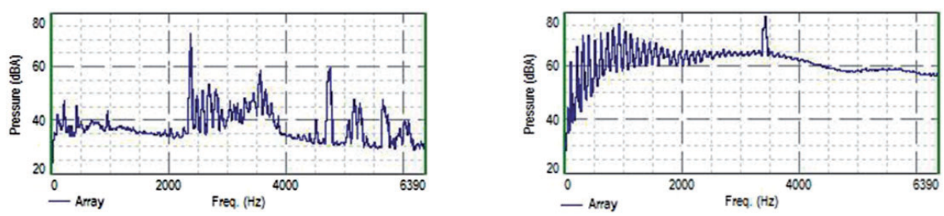

Figure a) Frequency levels for test with all motors running but no propellers. Top of MUAS facing array.

Figure b) Frequency levels for test with all motors and propellers running. Top of MUAS facing array.

Figure 1: Previous Work

themselves every octave, but at fundamental frequencies that are doubled (if the octave is higher) or halved (if the octave is lower). The character or quality of a musical sound or voice as distinct from its pitch and intensity is the timbre. We hypothesized that the quad-rotor will play controlled notes and will have a timbre of its own sounding like an instrument. [2]

2 Related work

There has some previous work in measuring quad-rotor noise, power and the fundamental frequency from different distances, angles and thrusts. Figure 1 illustrates the work done by the self funded research[4].

Some work has been carried out on the auralization of the rotor noise components of a quad-rotor, as described in [5]. Most of this work is directed towards the physics-based prediction of the tonal components of the rotor noise. This paper validates the premise that a particular combination of BLDC motor and propeller at produces distinct tonal noise for different orientations, altitudes and thrust.

This report illustrates the progress made on the project so far, as well as the the updated milestones, work division and proposal feedback.

3 Approach

We start with a brief description of the system architecture covering the Functional and Cyber-Physical Architectures of the system. Next, we present the progress in each of the individual subsystems viz. the Quad-rotor(Drone), Tethering System, Audio interface, and Controls.

3.1 System Architecture

Functional Architecture: The overall functional architecture of the system is illustrated by Figure 2. The key components included are the remote computer, on-board computer, quad-rotor and tethering subsystem. The system consists of a single remote computer and a single drone sub-system (includes the drone, on-board computer and tethering system). The remote computer sends a target note to be played to the drone system. The on-board computer extracts the fundamental frequency of the propeller noise captured via the on-board microphone and calculates the difference between the target frequency and the fundamental frequency. This signal is fed to the controller which generates the control signals. The control signals are communicated to the quad-rotor over UART interface. The drone executes the control commands which produce the desired tone. The resultant propeller noise is feedback to the on-board computer. The tethering system constraints the quad-rotor to the spatial limits specified by the system requirements. For the current implementation, the remote computer and the on-board computer have been combined into a single unit.

Cyber-physical Architecture: The Cyber-physical architecture of the system is illustrated by Figure 2. The remote computer is a standard Linux machine. It runs the sound analyzer blackbox and outputs

[2]**Note** The terms quad-copter, quad-rotor and drone have been used interchangeably.

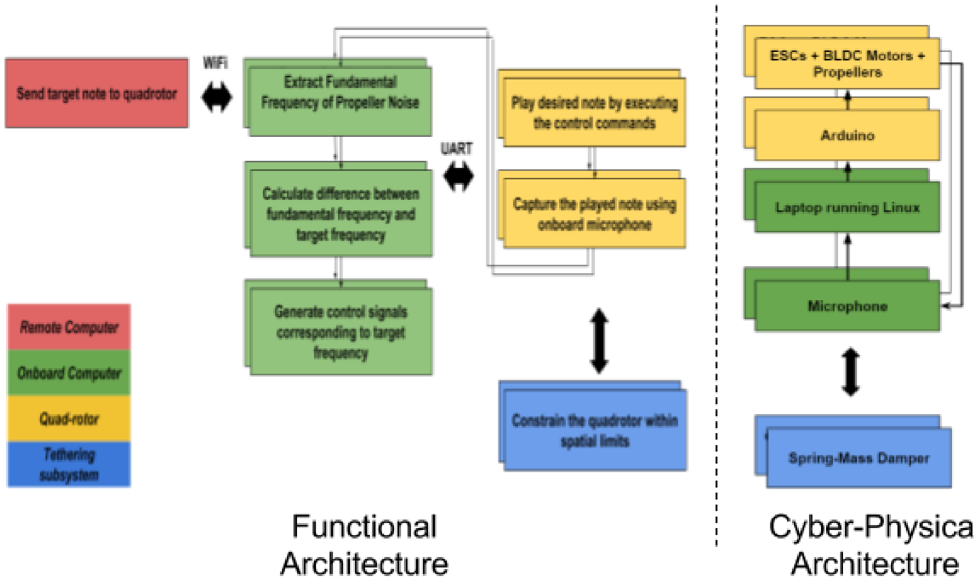

Figure 2: System Architecture

the target notes to be sent to each of the individual drones. The drone subsystem has an Odroid Single Board Computer as the onboard computer. The ODROID is a powerful SBC powered by an Exynos 5422 Octa big.LITTLE ARM Cortex-A15 @ 2.0 GHz quad-core and Cortex-A7 quad-core CPUs and had 2GB of LPDDR3 RAM. The signal processing and controls are handled by the ODROID. We are using Arduino SBC as the flight controller. The drone is powered by four BLDC motors (2300 KV) with 5 inch propellers and a 3S power train. The tethering system is basically a spring-mass-damper system, where the spring allows for the required constrained motion of the drone along the vertical axis and the damping is provided by friction generated by the 80/20 bracket pieces attached to the springs.

f_t = Vehicle Thrust; T = Tension

$$T = \sum_{i=1}^{4} K_i * sin(\theta_i) \tag{1}$$

$$f_t - T = m * g \tag{2}$$

$$T_i = K_i * \Delta x_i \tag{3}$$

$$T = \sum_{i=1}^{4} K_i * \Delta x_i * sin(\theta_i) \tag{4}$$

$$f_t = m * g + \sum_{i=1}^{4} K_i * \Delta x_i * sin(\theta_i) \tag{5}$$

Figure 3: Final Structure

3.2 Setup

Drone:

The drone we were provided with was initially controlled using a Raspberry Pi. Our team along with Professor Momeni came to the agreement that it was likely the Pi would not be able to handle sound processing in real time. So, we opted for an ODROID as our computation platform. This also gives us an option of using ROS in the future. The fitting of the ODROID required a new mount to be fixed to the bottom of the quad-rotor. The design for the base was provided by Professor Momeni. We laser-cut a ply-wood plank using the .dxf design file provided. Refer to the Fig.3 to check out the laser-cut ply wood.

The drone setup comprised the steps of fitting a base the ODROID mounts to, re-wiring the ESCs, new TTL-telemetry connection and sensor calibration. The assembled drone can be seen in fig. 3.

Tethering system: Another milestone we have met is assembling the mechanical base for tethering. The key idea was that the four corners of the base would have pulleys with four steel wire ropes that are attached to the drone at one end and a set of springs at the other, as we can see from the figure. A simple, scalable calculation was carried using the maximum vehicle thrust, vehicle weight and the maximum allowable/required displacement was carried out for the spring constant of the springs used, as we can see from equation 5. The variation in altitude of the quad-rotor will be limited to between 1-4 feet during operation due to the tethering and the size of the brackets used, which meets our sponsor's requirements for visuals. Fig.3 shows the current state of the mechanical structure.

Calibration: Once the initial mechanical setup was done and the drone was assembled, we began calibration of the sensors and the ESCs. The flight controller in use was the Pixhawk-4. The softwares explored included Mission planner and QGround control. The accelerometers, gyroscopes and barometer are calibrated. Unfortunately, we were unable to calibrate the ESCs as we cannot rotate the motors at the minimum and maximum throttle during calibration without a remote controller. This cost us time and is still in progress.

Software control: Our attempts to get the pixhawk controller working failed due to the need for the propriety firmware of the Ariel-Robotics Lab and incompatible RC files. As a result, we bypassed the PX4 with an Arduino that takes feedback from the laptop over serial. The laptop is running a software designed using Pure-Data. The architecture can be seen from figure 2

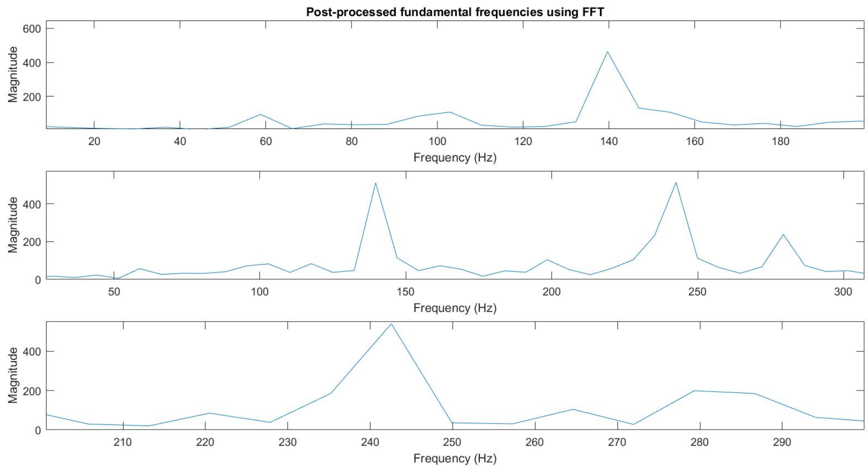

Figure 4: Fundamental Frequency Extraction

3.3 Audio interface

On the audio interface front, we tried to record the sound produced by a BLDC motor held in place by a Panavise using a different ESC so parallel work could be carried out on the drone. We wrote an Arduino sketch to ramp up the motor speed in steps of 0.01 in both clockwise and anti-clockwise directions, printing the speed at each step and listening for changes in the acoustic noise generated by the motor. We have also worked on writing pulse-width modulated square signals of a set of fixed duty cycles to the ESC, and observing the motor's response.

Initially, we used a simple piezo element to record these vibrations, however, this proved insufficient, with the sound being inaudible in the recording. Professor Momeni then provided us with another microphone, pre-amplifying circuit and mixer with which the sound tones generated were recorded using an appropriate, adjusted gain. We processed the recorded sound using Audacity (simple noise suppression, compression and signal amplification) and then used Fast Fourier transform to extract the fundamental frequency of the notes as seen in figure 3.

The recorded and processed audio used was of the ramping duty cycle input, and in the figure we can see two distinct fundamental frequencies in subplots 1 and 3, with subplot 2 showing the transition phase.

In flight, the drone features the same combination of a microphone with a pre-amplifier circuit to record the audio signal. This signal will be fed into the PureData program provided by Professor Momeni running on the ODROID, which will extract the fundamental frequency. We use the difference between this frequency and the frequency of the tone we intended to generate as an error signal which the control algorithm looks to minimize.

Table 1 shows the frequencies and difference in frequencies as we move along musical notes in an octave. The frequency difference scales in multiples of 2 as we go from one octave to the next. For a known acoustic noise to control command/thrust mapping, we can expand this table to span notes over every octave that the drone may generate. The mapping function can be then used to calculate the duty cycle of PWM that corresponds to a particular note. We must note that the range of notes the drone can produce is dependent on a number of parameters, including but not limited to: propeller size and geometry, motor size, drone size and even altitude. Demonstration of the octave mapping can be found here: https://youtu.be/uOV2gfzFpIw [6]

3.4 Controls

Both open and closed loop control were implemented to generate the musical tones. The tone mapping given in section 3.3 corresponds to open loop. Open loop control works fine provided we tune the PWM targets each time the drone is used. The tone generated depends on the propellers,

Note	Octave	Frequency	Wavelength	Frequency Step	Wavelength Step
C	0	16.351	20.812		1.169
C# / Db	0	17.324	19.643	0.973	1.103
D	0	18.354	18.54	1.03	1.04
D# / Eb	0	19.445	17.5	1.091	0.982
E	0	20.601	16.518	1.156	0.928
F	0	21.827	15.59	1.226	0.874
F# / Gb	0	23.124	14.716	1.297	0.826
G	0	24.499	13.89	1.375	0.78
G# / Ab	0	25.956	13.11	1.457	0.736
A	0	27.5	12.374	1.544	0.694
A# / Bb	0	29.135	11.68	1.635	0.656
B	0	30.868	11.024	1.733	0.619
C	1	32.703	10.405	1.835	

Table 1: Musical Note Frequency Table

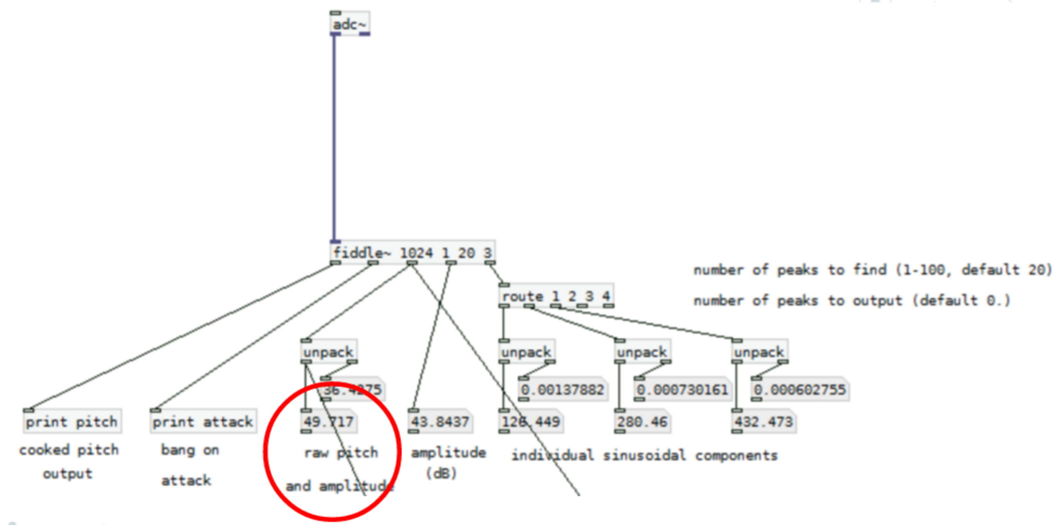

Table 2: Pure Data Structure

battery level as well as the power consumed by the motor. The use of a closed loop controller allowed us to bypass the constant need for tuning. The feedback is given by a Pure-Data based sound analysis software that measures the fundamental frequency at a fixed output frequency (currently set to 11 Hertz, but can be set up to run at upto 20 Hz). The Arduino then calculates the error in the target note and the fundamental frequency which is fed to a PID controller that calculates the new PWM to the ESCs. The Arduino has limitations with regards to computation power, which makes our system response slow. The system can play notes at 60 beats per minute currently. With an Odroid as the main controller, we should be able to generate higher tempos. The difference between open and closed loop control has been demonstrated in our final video [7].

4 Work Division

The work division for the project is shown in Table 3. All the tasks in green have been completed, orange are not relevant to the project anymore, and red implies these tasks were de-scoped.

Task	Saurabh	Aditya	Flavian
Assembly (quad-rotor, odroid base frame, tether support)			
Audio Mapping and Interface			
Pixhawk setup and Sensor calibration			
Odroid Setup			
Software library Exploration			
ESC Calibration			
ROS Setup			
Software Architecture Setup			
Tethering			
Mapping of tones & mic integration			
Build and tune Controller			
Setup second quad-rotor			
Integration			

Table 3: Task Distribution (Green > Complete, Orange > Not relevant for the final system anymore, Red > De-scoped) We set up a ROS architecture with another quadrotor(Danaus) that works on commands sent from a Matlab node. As this system required motion capture set up in the Aerial Robotics Lab which was not acceptable to our sponsor, we had to abandon this line of inquiry and continue development with the single drone as we have.

5 Results

We have successfully generated fifteen semitones using a single quad-rotor, at a speed of 60 beats per minute, with the notes chained together to form a tonal sequence or a song. Higher beat-per-minute tempos have been achieved using the ODROIDs GPIOs as PWM inputs to the ESCs. Currently, we play the root note representatives of the chords that form a song, as seen from the demonstration of 'Smoke on the Water' (link below). The mechanical structure has been tested and verified for a variety of thrusts, as we can see from the videos below. The code implemented is portable to other drones of the same specifications, and can be scaled to replicate the results with other drones as well.

6 Demonstration

A demonstration of our robot choir setup playing Beethoveen's Moonlight Sonata, and Deep Purple's Smoke on the Water can be found here: https://youtu.be/fNxGle8U9os [7]

References

[1] Article title: Welcome to DroneKit-Python's documentation! Website title: Python.dronekit.io URL: http://python.dronekit.io/

[2] Website Title: Mavros Wiki URL: http://wiki.ros.org/mavros

[3] Website Title: Embedded ROS URL: http://wiki.ros.org/rosseria_embeddedlinux

[4] N Kloet, S Watkins and R Clothier: Acoustic signature measurement of small multi-rotor unmanned aircraft systems

[5] Andrew Christian, D. Douglas Boyd, et al Auralization of tonal rotor noise components of a quadcopter flyover

[6] Website: Initial motor sound mapping https://drive.google.com/open?id=0B2TRi3T8Btdkb2N1R3loNXFwMXc

[7] Website: Final System Demo video https://youtu.be/fNxGle8U9os

[8] Website: https://www.usatoday.com/story/tech/talkingtech/2017/02/06/check-out-drones-super-bowl-51-halftime-show/97545800/

[9]Website: https://www.theverge.com/2016/12/20/14030706/aurora-drones-winter-bird-music-video