AFRL-AFOSR-UK-TR-2024-0007

Multimode Bio-Inspired Sense and Avoid

Bomphrey, Richard
THE ROYAL VETERINARY COLLEGE
4 ROYAL COLLEGE STREET
LONDON, , NW1 0TU
GBR

12/11/2023
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515 Box 14, APO AE 09421

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
20231211	Final	START DATE 20190915	END DATE 20230914

4. TITLE AND SUBTITLE
Multimode Bio-Inspired Sense and Avoid

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
	FA9550-19-1-7040	
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Richard Bomphrey

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
THE ROYAL VETERINARY COLLEGE
4 ROYAL COLLEGE STREET
LONDON NW1 0TU
GBR

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
EOARD UNIT 4515 APO AE 09421-4515	AFRL/AFOSR IOE	AFRL-AFOSR-UK-TR-2024-0007

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The Royal Veterinary College (RVC) and Centeye have, in parallel, developed devices that offer capability enhancement in the sphere of sense-and-avoid technology for micro- and nano-rotorcraft. Specifically, the RVC devices use information measured from changes in the induced flow field when the craft enters ground or wall effect, while Centeye's devices use visual information, including optical flow, stereo vision, and active illumination. Each sensory mode has been studied, flight tested, and characterized extensively in recent efforts.
Here, we exploit these capabilities more deeply by feeding the sensor array measurements into a deep learning network to estimate bearing and distance to nearby obstacles in a range of conditions, including those that are challenging for one or the other sensory mode. Avoidance behaviour will be superior to that which is possible by each mode independently. Our objectives included integration of these two technologies onto a suitable test platform for flight testing, simultaneous acquisition of data from at least two sensor modes, and
first-order attempts at algorithm fusion in supervisory flight control software. The integrated sensory mode platform demonstrates obstacle avoidance greater than the sum of its parts. Additionally, we developed 'attention-switching' algorithms to reduce power consumption under differing environmental conditions, and the identification of new capabilities other than obstacle avoidance using the same apparatus, such as the measurement of a wind vector. Our machine learning method matches the physical arrangement of the sensors but is fundamentally agnostic to the sensor type and is applicable to acoustic range finding using inherent vehicle noise as the signal.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	48

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
NANDINI IYER	314-235-6161

Standard Form 298 (Rev.5/2020)
Prescribed by ANSI Std. Z39.18

Report Coversheet

Award Number

FA9550-19-1-7040

Report Type

Final Performance

Reporting Period

Start Date: 9/15/2019

End Date: 9/14/2023

Distribution statement

Distribution A – Approved For Public Release

Program Officer Name

Dr Nandini Iyer

Principal Investigator

Professor Richard Bomphrey

Title

Multimode Bio-Inspired Sense and Avoid

Abstract

The Royal Veterinary College (RVC) and Centeye have, in parallel, developed devices that offer capability enhancement in the sphere of sense-and-avoid technology for micro- and nano-rotorcraft. Specifically, the RVC devices use information measured from changes in the induced flow field when the craft enters ground or wall effect, while Centeye's devices use visual information, including optical flow, stereo vision, and active illumination. Each sensory mode has been studied, flight tested, and characterized extensively in recent efforts.

Here, we exploit these capabilities more deeply by feeding the sensor array measurements into a deep learning network to estimate bearing and distance to nearby obstacles in a range of conditions, including those that are challenging for one or the other sensory mode. Avoidance behaviour will be superior to that which is possible by each mode independently. Our objectives included integration of these two technologies onto a suitable test platform for flight testing, simultaneous acquisition of data from at least two sensor modes, and first-order attempts at algorithm fusion in supervisory flight control software. The integrated sensory mode platform demonstrates obstacle avoidance greater than the sum of its parts. Additionally, we developed ‘attention-switching’ algorithms to reduce power consumption under differing environmental conditions, and the identification of new capabilities other than obstacle avoidance using the same apparatus, such as the measurement of a wind vector. Our machine learning method matches the physical arrangement of the sensors but is fundamentally agnostic to the sensor type and is applicable to acoustic range finding using inherent vehicle noise as the signal.

Bio-inspired sense-and-avoid strategies for flight near surfaces

Final Year Report to AFOSR

Simon Wilshin, Richard Bomphrey

December 11, 2023

Contents

1 Introduction

This project has six major goals:

1. Basic Science
2. Integration of sensors onto common platform
3. Combination of sense-and-avoid modalities
4. Power optimisation
5. Wind vector estimation
6. Further expansion

Of these the first, second and fourth have formed the backbone of our first year's progress, with additional progress in the third. In the second year we made progress on the second, fourth and sixth. In the third year we made progress on the second, third and fifth and sixth. This year we have now completed all core targets in the above categories.

The major new innovations presented in this report update are the new control strategy for sense and avoid operations, the basic design of which is covered in section 2.3, an experimental demonstration of our drones capability to use sensory integration to detect difficult surfaces in section 4.1, an experiments demonstrating sense and avoid capabilities in free flight in section 4.2, power usage optimisation strategies described in section 6 and estimation of air speed as described in section 7.

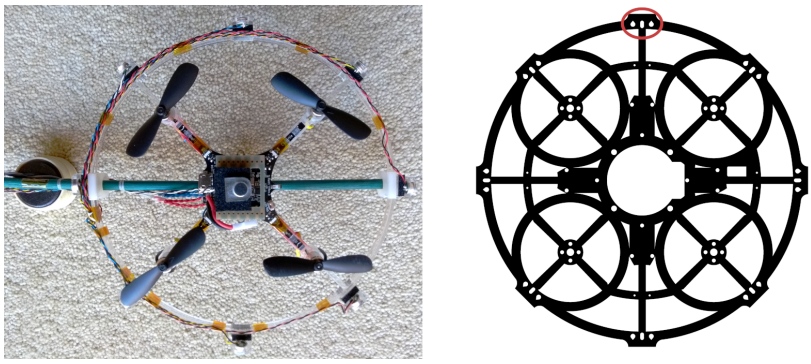

Figure 1: Microphone array mounted on a Crazyflie, and compatible mounting points on the new drone chassis (red highlight). The array has mountings for ten microphones, eight located in a ring around the drone, one located above the centre of the drones chasis, and one located below.

2 Basic Science

The "Basic Science" section of our proposal began with a core question "At what range can fluid dynamics-based mechanosensors detect surfaces without touching them directly?"

We have previously built an array of pressure sensors capable of detecting surfaces at close range, and have developed this work to incorporate the addition of a microphone array. The control architecture we propose to build is agnostic to the sensor type used, and the goal is to fuse different types of mechanosensor with vision sensors to provide an obstacle detector that is greater than the sum of its parts. We have previously addressed this question by operating a drone in an anechoic chamber and using acoustic recordings to measure the position of a planar surface using machine learning. To this end we have constructed a drone mounted microphone array compatible with both Bitcraze's Crazyflie 2.1 platform and the new drone developed as part of the proposal (progress on this section of the proposal is detailed in Sec. 3) (Fig. 1).

The microphone array mounted on the Crazyflie drone was placed in an anechoic chamber with an adjustable planar surface (Fig. 2).Recordings were taken in two configurations with the intent of determining from the microphone recordings the location of a planar obstacle. One of these conditions had the planar obstacle below the drone at a distance of 23, 46, 92, 184, 368, 552 and 736mm below the base of the drone. Recordings were also taken with the same plane with the same plane in a vertical orientation, so the closest point on the plane to the drone was at a distance of 50, 90, 150 and 250mm, and at an angle of 50, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 300 and 315 deg measured counter-clockwise looking at the drone from above relative to the boom which tethered the drone at the rear (hence the lack of observations at 0 deg, the boom obstructs such observation). For the purposes of comparison

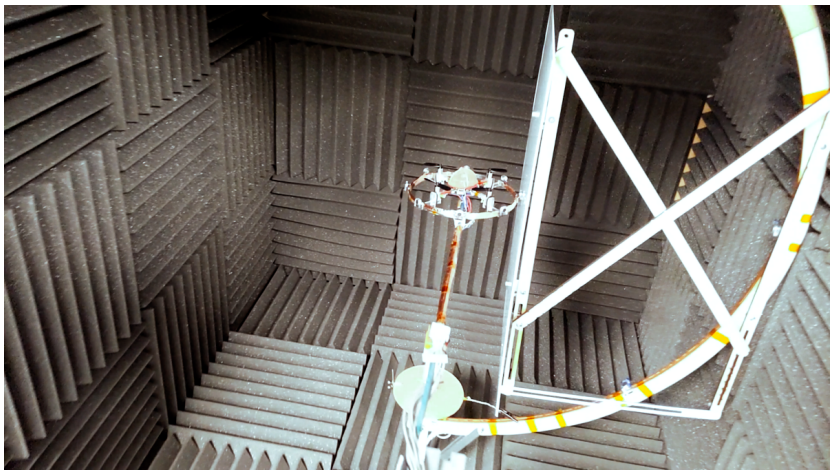

Figure 2: A tethered Crazyflie drone with the microphone array inside the anechoic chamber with a planar obstacle, the obstacle is mounted on a swivel to allow repositioning to any angle excluding those obstructed by the tether boom, and telescoping to permit distances up to the edge of the chamber. The surface can also be mounted below the drone, and the height of the drone adjusted above this surface.

of scale and consideration of ground effect, the Crazyflie with the microphone array is 65mm in diameter, and the rotor diameter is 45mm. We are therefore operating both well within ground effect, and well outside.

A voltage was applied to the drone's motors sufficient to generate enough lift to allow it to hover under free-flight conditions. Note that this stable drive voltage, along with the anechoic environment, and the short time scale over which the observations were taken (minimising thermal and pressure variance) represent somewhat ideal conditions for the task of inferring the position of the plane. An example recording is provided in Fig. 3

With this data we were able to differentiate the different conditions and estimate the position and orientation of the plane in space using a machine learning classifier, the architecture of which will be agnostic to the mechanosensory array used on the drone.

2.1 A machine learning algorithm for obstacle detection

We previously built a machine learning classifier using pytorch. The classification network was built from a configurable input layer, two convolutional layers, two fully connected layers and an output layer.

The input layer can be adjusted to match the sensor suite being used (further discussion in Sec. 3.1.6) via a human readable YAML configuration file. In this configuration the data is split into blocks.

For the test system implemented using the microphones these blocks are the

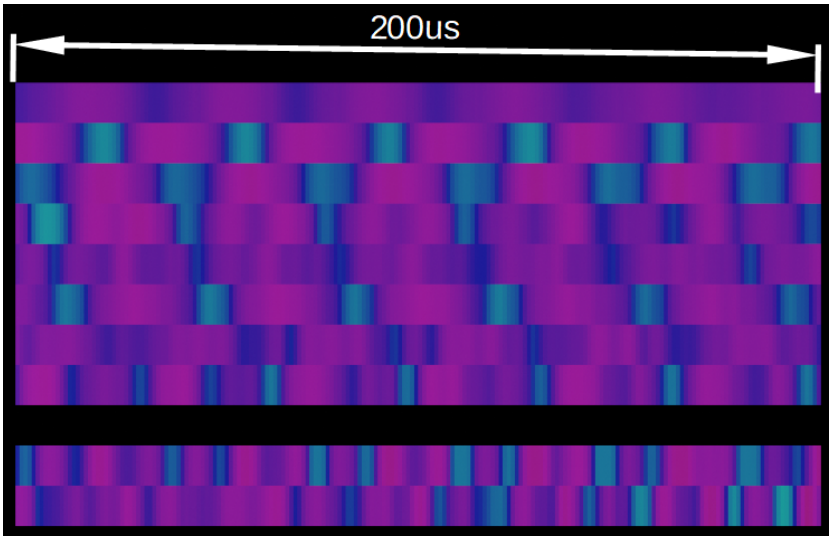

Figure 3: An example microphone recording of the drone inside the anechoic chamber. This images represents 200 microseconds of recording along the x-axis, regions that are magenta are periods of low intensity, cyan are periods of high intensity. The ten rows are the ten different sensors, the block of eight in the upper portion of the figure being the microphones mounted in the ring around the drone, the two below being above and below the drone.

microphone recordings from the eight sensors in a ring, the two microphone on the top and bottom of the drone, and two additional blocks formed from the Fast Fourier Transform (FFT) of these two sets of signals.

Two convolutional layers are then used. The number of convolutional elements as well as the input topology can be specified in the YAML. The surrounding sensors (here microphones, but compatible with pressure sensors, cameras, etc.) form a ring around the drone, and any classification network should respect this topology, that is there should be no 'start' or 'end' sensor, and any mathematical operation should be invariant to a transformation which shifts all the sensors around this ring.

As such we use a circular convolution for the sensor input elements on the ring. Since the top and bottom elements of the microphone are completely included in the convolution along the sensor number axis, these can be a regular convolution.

If x_ij is an element of the blocks, x, discussed above, with i running from one to the sample size N, j running from one to the number of elements in this block (for example eight if we are dealing with the eight microphones around the drone), then this layer performs the following operation:

$$y_\mu = b + w_\mu \star x \tag{1}$$

Where $\star$ is the cross-correlation operator, w_μ is the kernel associated with the μth convolution, of which there are M total, and b is a constant matrix. In the language of convolutional neural networks this first layer assumes inputs with only one channel, and outputs M channels.

All units in the network are Rectified Linear Units (ReLU), and so this response y is modified by:

$$y'_\mu = \max(0, y_\mu) \tag{2}$$

This operation is performed element-wise. The reason that ReLU units are used is that while non-linear transformation units (like sigmoids or hyperbolic tangents) often perform very well, they are notoriously hard to train. This is because they saturate and therefore have very small gradients that are hard to manage when performing back-propagation. A unit which is non-linear (as the max function is), while having non-zero gradient everywhere where the unit is active, generally converge more rapidly. When prototyping this is a desirable property.

Our network pools after every convolutional layer. This reduces the number of parameters in the network, which improves training time and reduces the risk of over-fitting. It also makes the network less sensitive to the precise locations of features, which makes the network more robust. If the difference between the intensity at $1000Hz$ and $2000Hz$ might be important to the network, then a change in conditions that shifts both these frequencies up slightly (say to $1050Hz$ and $2100Hz$) should also be important.

We use 2D max pooling. This splits the layer activations into $K \times L$ blocks and transforms the layer by downsampling to only the maximum value in these blocks.

Our network then proceeds with another convolutional layer with ReLU and pooling. Next we have two fully connected layers. Here we treat the output of the last convolutional layer as a single vector a_i. The activation, b_j of the fully connected layer is just

$$b_j = c_j + \sum_i m_{ji} a_i \tag{3}$$

The units in the fully connected layer are again ReLU units, for the same reason as previously discussed. No pooling was performed here as the fully connected layer respects the topology of the input only in the most trivial way (every unit is connected to every input), meaning neighbouring units are likely to be unrelated.

Another fully connected layer was then used followed by the output layer. The output layer was the azimuthal angle of the obstacle, and a vector pointing to the obstacle in the plane. The azimuthal angle can only be 0 or 90 degrees, and was re-interpreted after the network is trained as a class variable indicating wall position.

The network was trained using Stochastic Gradient Descent (SGD) [1] to minimise the mean square error.

2.2 Network performance

The network performance was evaluated on a test data set of ten thousand sections two milliseconds in length that were not using in the training data. Performance was excellent (see Fig. 4 and Fig. 5).

The drone was able to perfectly distinguish the two conditions (wall to the side, floor below) for all examples in the test data set. When estimating the height of the drone above the surface the standard error was 36.1mm. This error was relatively consistent over the range of distances measured, meaning that when deep in ground effect the detection of the ground was rather inaccurate (although sufficient to inform a surface avoidance strategy), and surprisingly good at large distances (although we again note that these constitute close to ideal conditions). The limits of this floor detection capability were not found as we reached the maximum height of the anechoic chamber before such a limit was reached. For practical purposes this is sufficient, although detecting such a limit for a free-flying drone is highly desirable and is a priority future objective.

Similarly the horizontal distance at which performance begins to degrade was not detected, as the algorithm maintained a good performance out to 250mm, with a standard error on the distance to the wall at 39.5mm and an angular standard error (assuming a Gaussian distribution, not von Mises, which is only appropriate for small angular errors like this), of 16.4 deg.

2.3 Sense and avoid control strategy - design

We have designed a control strategy to allow our drone to avoid surfaces in the environment. This controller takes a two tiered approach. A standard controller than manoeuvres the drone through the environment without concern for sensor

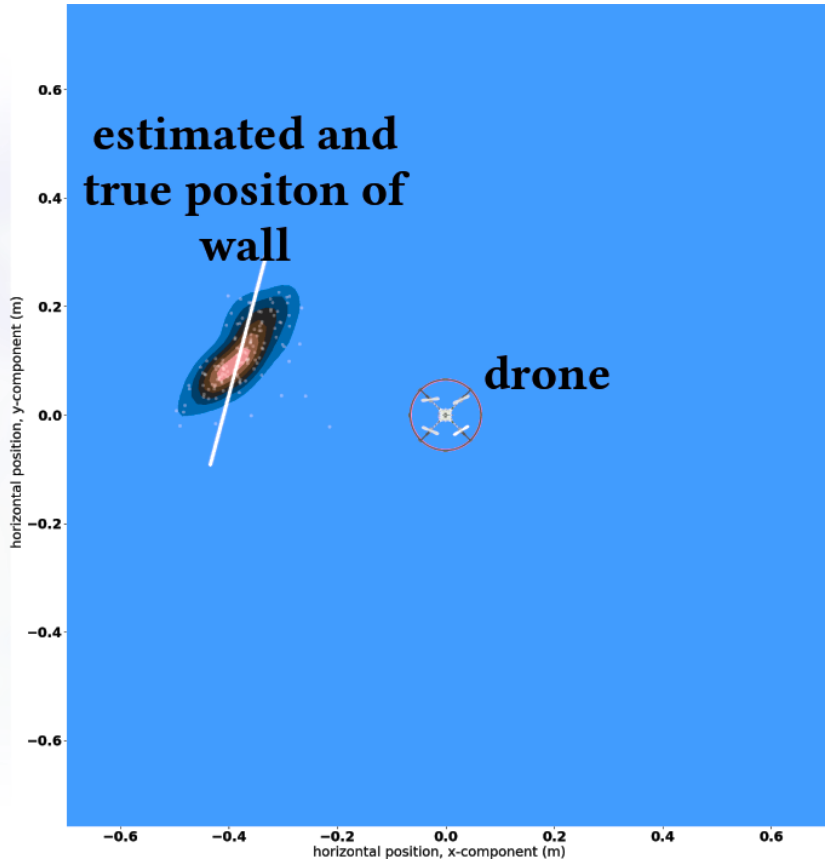

Figure 4: Performance of the machine learning sensory integrator in detecting the position of a surface to the side of the drone. The drone is shown inside the red circle, the wall is the solid white line, the x- and y-axis are position in meters. The white dots show the estimated position of the wall according to the algorithm, with the kernel density plot in blue, black and red constructed from this data. There is strong agreement between these estimates and the actual wall position. The standard error in the estimated position by shortest distance to the wall was 39.5mm, and the standard error on the angle (assuming a Gaussian distribution, not von Mises, which is a reasonable approximation here as the error is small) was 16.4 deg

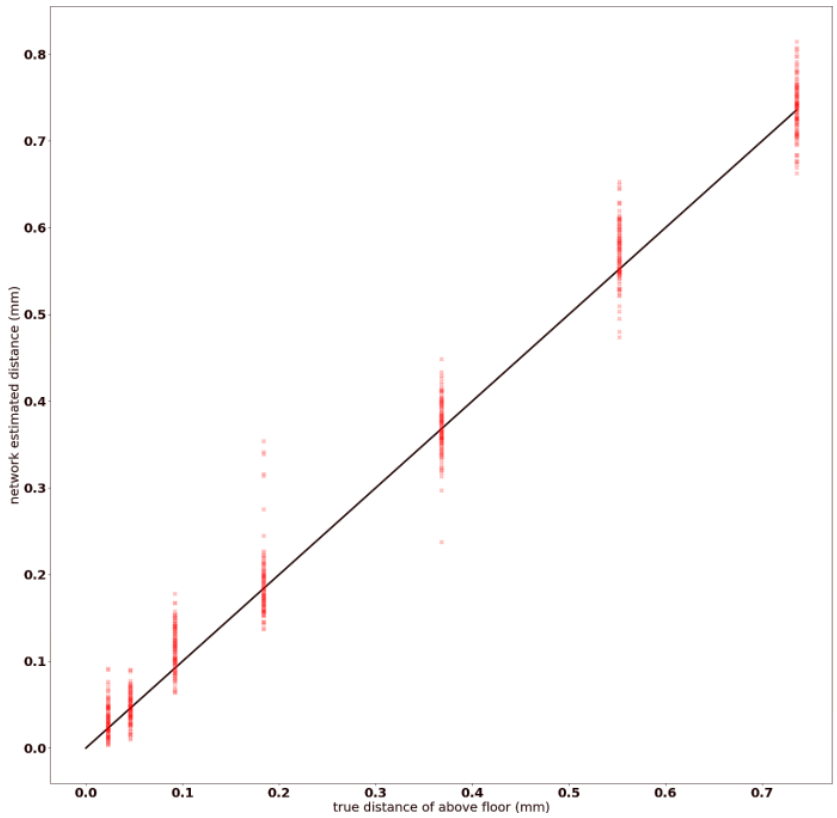

Figure 5: Performance of the machine learning sensory integrator in detecting the position of a surface to the below of the drone. On the x-axis is the true position of the surface below the drone in meters. On the y-axis is the estimated position of the surface according to the algorithm. The standard error on the estimated distance was 36.1mm.

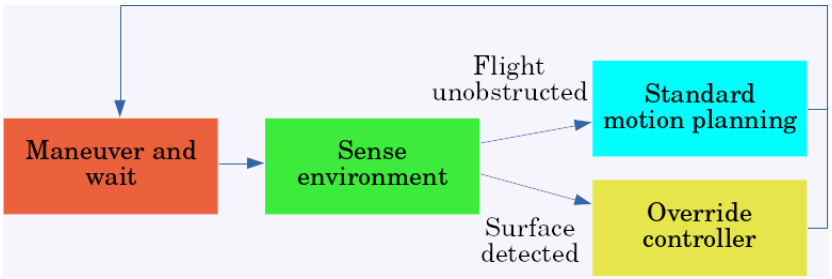

Figure 6: The sense and avoid control design scheme, with a basic controller managing normal motion planning, and an override controller for handling situations wherre the sensors indicate a collision is imminent.

input, and an override controller, which prevents the drone from performing manoeuvres that will result in a collision or accident. The basic schematic for this design is shown in figure 6.

The use of this controller for achieving the sense and avoid modalities of our third major project area is detailed in section 4.2.

Sensing of the environment can be performed by the pressure or optical sensors, with trigger conditions based on the squared deviation of the pressure recordings, or the broad threats detected by the optical flow sensors. In both cases a threshold was used to detect a surface, this threshold had to be selected based on prevailing conditions. The detection was mostly insensitive to lighting conditions or surface type, but see sections 4.1 and 3.3 for some caveats to this.

3 Integration of sensors onto common platform - Mormoops 1.0

We have previously constructed a new drone, the Mormoops capable of supporting the range of sensors and processors required for the project. Here we describe the design decisions, and document our progress in the construction of this drone, which is now flight capable.

3.1 Hardware

3.1.1 Chassis

To ensure ease of replacement the chassis is laser cut from 3mm thick acrylic. In the center there are mounting holes which match those described in the mechanical design section of the technical specifications of the Crazyflie Bolt.

3.1.2 Engines, motors, propellers, electronic speed control

The Mormoops uses four motors and propellers.

As indicated in the proposal, size considerations for the Mormoops were dictated by the need for additional power over that typically achieved in a brushed micro quadcopter such as the Crazyflie. The maximum desired size was such that the prototype could fit in a large thigh positioned pocket on a pair of cargo pants (around eight inches or twenty centimetres).

As such 50.8mm (two inch) propellers were selected. These are slightly larger than the 45 mm / 1.77 inch propellers used on the Crazyflie, which should allow us to make better use of the more powerful brushless motors, while at the same time around 4 cm would be left for mounting a flight controller and payload between the propeller nacelles in the centre, and a few cm left on the exterior for the payload of microphones and vision sensors.

Broadly speaking, more blades on a propeller means more power at a slower rate of rotation, which can in turn give more control, but with the trade-off of reduced efficiency. More blades also means more weight. Typically a drone propeller will have between two and six blades, the former being typical for high-speed drones or drones which have tight power constraints and limited requirements on control responsiveness, the latter being favoured for drones where efficiency is less of an issue, but maximum control authority is required. The Mormoops is intended to be operated both inside of ground effect, and close to walls, where a high degree of control is desirable, and is required to carry a substantial payload. As such a compromise of a four bladed propeller was selected.

In a similar way, pitch trades power draw and control authority for thrust. A more steeply inclined blade will push more air, but at the cost of drawing more power and having less fine control over thrust. The relationship between pitch and efficiency can be complex, and depend on the operating conditions of the drone. Since we are interested in a high degree of control authority for obstacle avoidance, very flat blades were avoided with the Mormoops, but beyond this the other design options were prioritised.

There are many possible materials for propellers, from cheap polycarbonate to expensive carbon fibre. Since the Mormoops is intended to be operated in cluttered environments where collisions are likely, the propellers should be treated as a consumable part. As such cheap, polycarbonate propellers were selected.

All of these design requirements were met by 2035 propellers intended for use with the DYS Shark Mako. Many other compatible propellers exist allowing for ready replacement of parts.

The intention of the nacelles which house the motors is to minimise self-noise of the drone, protect the propellers from the environment, protect the other components of the drone from prop-strikes and help manage airflow.

The nacelles of the drone are four 3D printed tubes with connector tabs to mount to the chassis. These parts are brittle and should be treated as a consumable. A paper composite (paper layers set in epoxy, sometimes referred to as micarta although this is a brand name Industrial Laminates / Norplex, Inc.) was considered for these tubes, but were ultimately too heavy and unreliable.

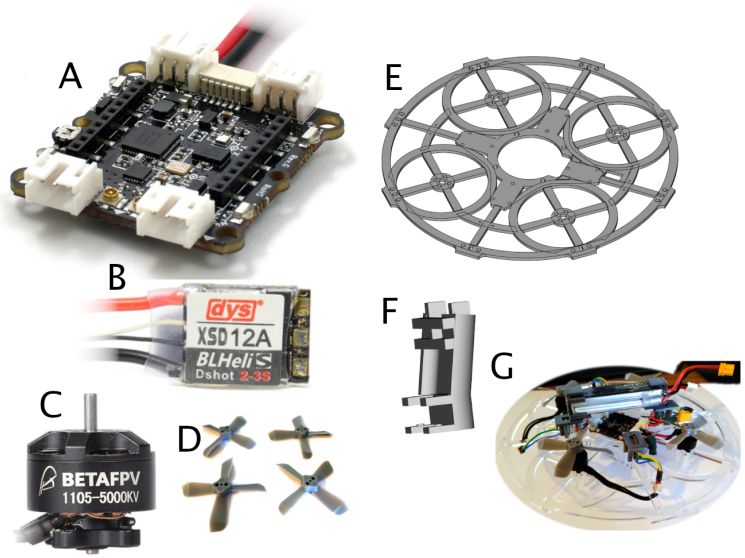

Figure 7: The Mormoops hardware stack for flight operations consists A) the Crazyflie Bolt flight controller, B), C) and D) generic commidity hardware ESC, motors and propellers, E) a custom laser cut acrylic chassis, F) custom 3D printed mountings. G) The assembled drone.

3.1.3 Flight control

The flight controller of the drone is a Crazyflie Bolt, depicted in figure 7. The Bolt is a versatile platform with an integrated power distribution board which can provide up to one amp of power at five volts in addition to a variety of flight control behaviours, enough to power the other onboard components. It can be controlled over radio, via serial pins (UART) and with the GPIO pins and all these configurations have been used.

3.1.4 Processing and storage

The Mormoops controls and co-ordinates the sensors and flight controllers by means of a Teensy 4.0 microcontroller, depicted in Fig. 8. A Teensy 4.1 was used for bench-top testing, although this unit is too long and heavy to be mounted on the drone. We therefore constructed a custom SD-card reader (also show in Fig. 8) which is used to record in flight data, avoiding the need for sometimes unreliable radio transmission from the drone.

3.1.5 Power and battery

Since weight and power optimisation form a core component of this proposal, discussion of the design and operation of the power system for the new drone

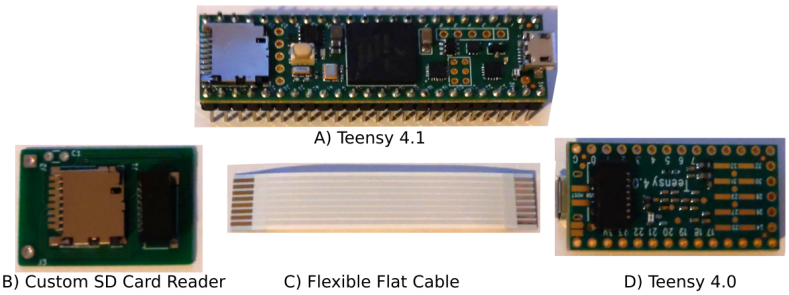

Figure 8: The onboard processing is performed by a Teensy 4.0 (panel D), the 4.1 (panel A) is use for benchtop development due to the ease of access of all necessary pins, and the 4.0 is augmented by a in house designed custom SD card reader (B) connected to the Teensy via flexible flat cable (C).

have been separated and discussed in section 5.

3.1.6 Sensors and communication

The multiple sensor systems require flexibility in the input layer of the neural network used to detect the environment, in particular the sensors are both topologically and temporally distinct (each sensor has a component which forms a loop around the drone of varying resolution, some also have sensors mounted above and below the chassis, and all have different sample rates). To this end, a configurable specification for the input layer has been developed to allow a generic set of convolutional layers to be connected to these varied inputs (an example being provided in Fig. 9).

The routing of on-board communication is depicted in Fig 25, showing which data lines are connected to which units. The central component of the system for processing purposes and high level decision making, including the implementation of our sense and avoid algorithms, will be the Teensy 4.0. Serial (UART) and I2C lines (labelled SLC and SDA in the figure) all eventually route to this unit. Two of the I2C interfaces (SCL0 SDA0, and SCL1 and SDA1) of the Teensy are designated to connect to the differential pressure sensors. These have independent pull up resistors built into the sensor board. One I2C line connects the teensy to the differential pressure sensors to the Centeye (SCL2, SDA2). This is routed via the Bolt which both provides the pull up resistances for these lines, and provides us with a back-up method of communication with the Bolt should it be necessary. The Teensy can issue high level commands for station keeping and manoeuvring to the Bolt via either the I2C line previously mentioned, or via the serial data communication (UART Rx and Tx). This stack has been tested on the bench (see Fig. 10) and communications capability on these lines confirmed for all components. We further note that the Teensy 4.0 has multiple available pins for GPIO and an on-board ADC allowing us to incorporate our microphones. A bare drone can include the full ten microphone

```yaml
net:
  poolers:
    a:
      S: 2
    b:
      S: 2
  y:
    N: 16
    K0: 20
    K1: 3 # Must be odd
  yp:
    N: 8
    K0: 20
    K1: 3 # Must be odd
  yotta:
    N: 4
    K0: 20
    K1: 2
  yottap:
    N: 2
    K0: 20
    K1: 1
  z:
```

Figure 9: An example YAML specification file for configuring the input of the neural network.

array, with both the Centeye and diferential pressure sensors in place, the shared GPIO and data lines cannot be used at the same time, reducing the number of possible microphones to eight.

A new version of a pressure sensor board and microphone array board have been designed and were fabricated this year (schematics in figures 11 and 13).

The differential pressure board is smaller and lighter and has compatible mountings for the new prototype. The microphone array board has capacity to connect and amplify ten collector microphones which can be distributed over the drone. These can then be connected to the analogue channels of the microcontroller.

3.1.7 Concurrent sensor operation

In previous years object detection with pressure sensors and microphones was tested. The drone was operated near a planar surface, and our neural network tasked with detecting the surface. This test was further used to confirm that the Centeye and microphones could be run concurrently. The Centeye was used to measure the height of the drone over one minute trials (the Centeye tracks height reliably over this time scale, this was confirmed by comparing trajectories of drones flying with the sensor measured using infrared motion capture and the Centeye's odometry estimates). The flight controller was asked to maintain constant height based on the odometry and no obvious drift was observed over these short flights.

The drone was flown at three different heights, one deep in ground effect and two outside. Our neural network was then asked to distinguish these states using the sound recordings.

For the trials in ground effect, the sound recordings were clearly different,

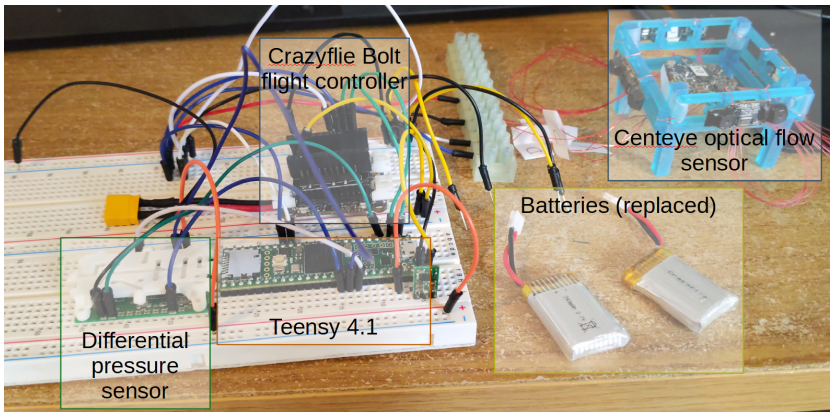

Figure 10: Bench-top test of the new drone electronics stack. The sensors, power, processing and flight control components are highlighted. Here the Teensy 4.1 was used which has a built in SD-card reader. Subsequent tests have incorporated our custom SD-card board for the Teensy 4.0.

meaning close approach to surfaces can be reliably detected. The two trials where the drone was distant from the surface were not distinguishable from one another, suggesting that differences in free-flight away from a surface may be harder to detect using only microphones. The configurations are depicted in figure 12.

Close detection of surfaces with microphones is reliable, especially if integrated over several seconds. This is a result of detecting the change in motor noise while in ground effect. While in ground effect the motors generate more lift for less work, meaning there is a reduction in noise (and frequency) from (and of) the motors. In addition the ground reflection is loud and close. The overall noise is therefore both distinctively increased and changed in character, including spectral content, while hovering.

However, our drone is being controlled via a simple PD controller adjusting height as measured by the Centeye optical flow odometry. Without the odometry, and thus knowledge of the location of the ground, the power of the motors will not automatically be adjusted. This is likely to complicated detection of the ground in scenarios where the drones configuration is not being held constant by external means.

This experiment demonstrates successful integration of the pressure sensing system with the Centeye optical sensor. It also establishes a lower bound on the range at which fluid dynamics based mechanosensors can detect surfaces within the basic science remit when in free-flight (see section 2). Detection when within ground effect is reliably achievable when using only microphones.

A further application of this method immediately presents itself. The odometry measurements from the Centeye have very slow drift, since they are integral estimates of position from the optical flow. This drift could readily be corrected

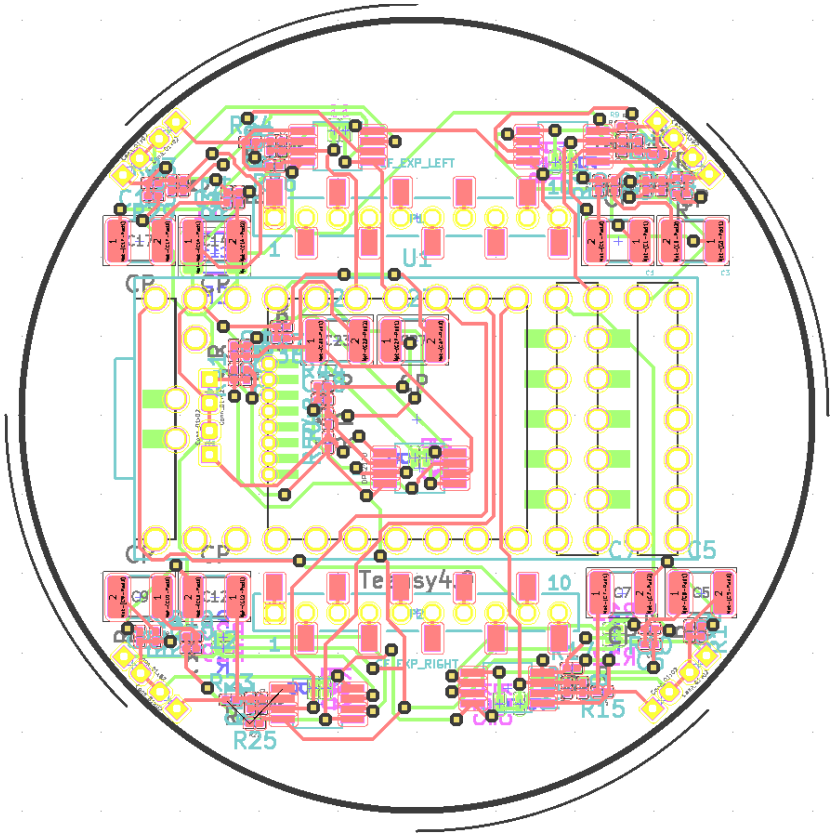

Figure 11: Schematic of the microphone array board. This is a ten channel microphone amplifier board capable of recording sound above, below and all around the drone for surface detection both vertical and horizontal. It includes a mounting point for the teensy 4.0 microcontroller which can record the audio to an SD card or process the audio for control purposes.

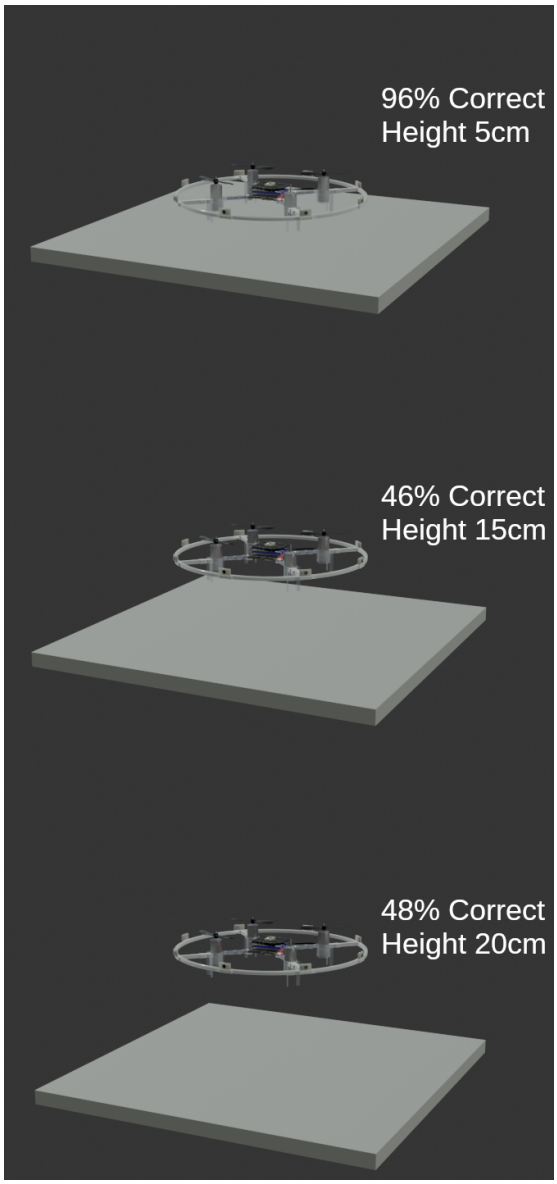

Figure 12: Performance of surface detection in free-flight, the within ground effect condition (approximate height fix of 5cm) was readily detected, the others could not be distinguished from one another. Position was stabilised using simple PD controller using the Centeye odometry demonstrating sensor integration.

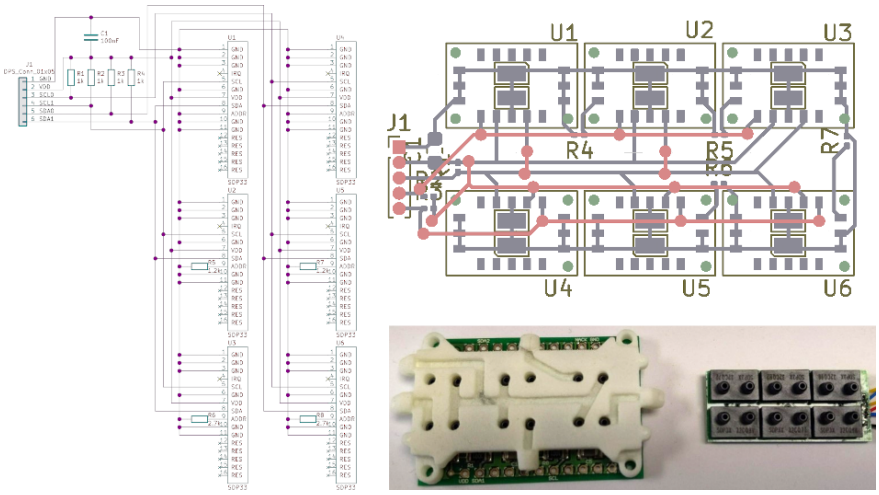

Figure 13: The circuit diagram and board layout of our new, compact differential pressure array. The new unit has one additional channel. A comparison of the size of the two units is shown in the bottom right of the figure, with the new sensor on the right and the old on the left.

for by detecting the location of a known surface in the environment. This would permit longer operation of the sensor with drift correction.

3.2 Surface detection using differential pressure sensors

Previously we made progress in the detection of static surfaces on our platforms. A new miniaturised version of the differential pressure sensor was been built and tested, with up to six channels (one more than the previous five, with the intent to provide better resolution for surfaces below the drone). The circuit diagram, design, and a comparison of the new unit to the old is shown in figure 13. Technical specifications for fabrication are included in the appendix. We previously performed bench top tests of this system on all channels when mounted to both drones and confirmed successful operation. Once successful operation was confirmed we performed a series of experiments to test the sensor system above horizontal planar surfaces in unique configurations. First, a sequence of training data was obtained to determine the relationship between the signal measured by the differential pressure sensor in the presence of a horizontal planar surface. Then a sequence of tests were performed with boxes of varying height. For these initial tests no feedback control was in place, new feedback controlled experiments displaying full sense and avoid capabilities are described in section 4.2.

A crazyflie was mounted with our custom deck for sensor integration and control, which consists of a teensy 4.0 connected to our custom SD card reader. Feedback control was maintained via a Lighthouse Positioning System (LPS) for

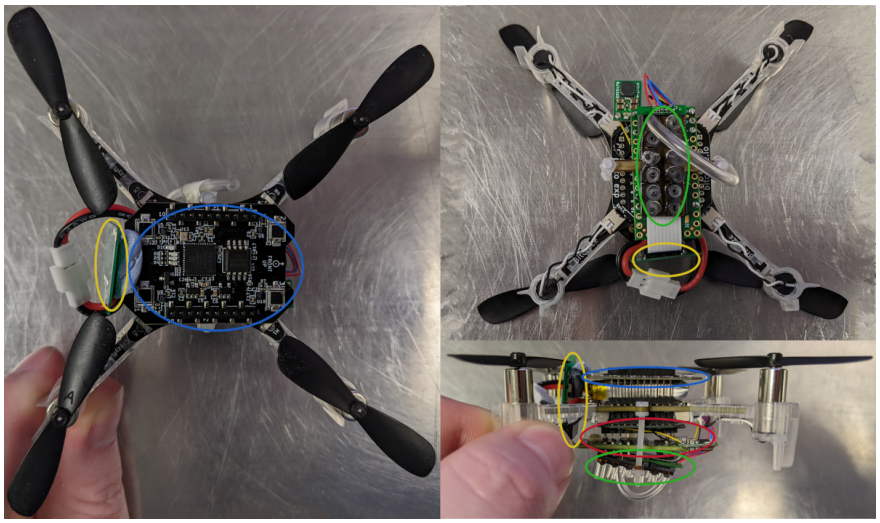

Figure 14: The differential pressure sensor mounted on the crazyflie drone (in the green circle). In the yellow circle is the on-board SD card storage, the red circle contains the teensy microcontroller and the blue circle contains the lighthouse positioning system.

the Crazyflie, using two Valve Index VR base stations. As part of these experiments we discovered that the station keeping the LPS was generally superior to that of Qualisys, requiring less set up time, less frequent calibration, and a most consistent position estimate over time. These advantages are sufficient that this will be our primary method for positioning the drone going forward, as any theoretical (and likely inconsistent) improvement in accuracy that can be obtained from using Qualisys cannot compensate for the drastic improvement in the amount of data of a consistent quality that can be obtained via the LPS.

The differential pressure system was connected to the I2C bus. All channels were tested, but for this experiment only one channel was used at a time, the first channel. Vicon plastic tubing was routed from the differential pressure outlets to point directly down on the drone, and up above the drone on one side (this configuration is depicted in figure 14).

Two experiments were conducted. During the first, calibration data for the estimated height of the drone from the variance of the differential pressure sensors recordings were obtained over 100 samples. This corresponded to 1 second of recording. This was continuously estimated for each new sample. The drone was operated at a variety of fixed heights for between six and ten seconds at a time above a solid surface and recordings from the differential pressure sensors taken on the SD card with a synchronisation timestamp. Positions were recorded from the LPS, again with a timestamp.

In the second experiment the drone was flown above two planar surfaces of a different height to the launch pad, to test if these surfaces could be detected.

The drone was made to perform three dips in altitude. This path is shown in the stills from a video of the video of the drone during the experiment in figure 15. One above open terrain, one above a box of 20.5cm and one above a box of 27cm. Positions and differential pressures were again recorded.

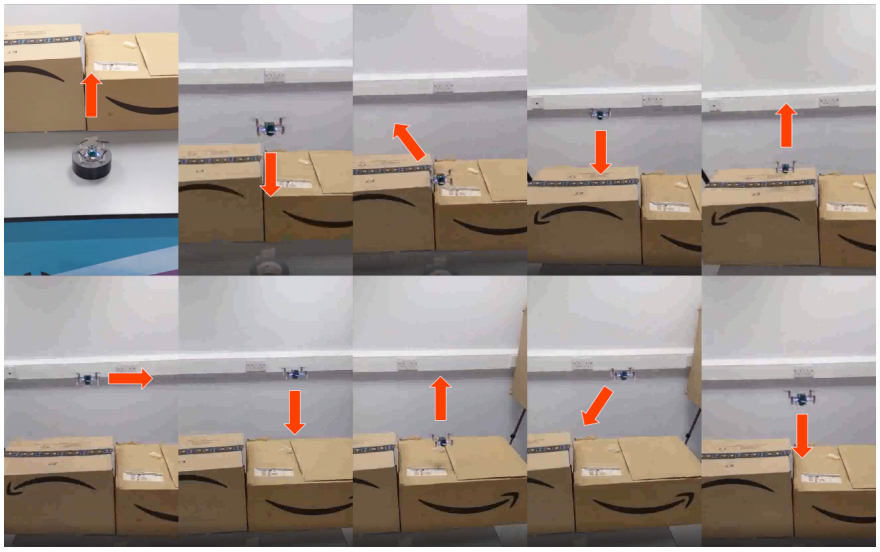

Figure 15: Trajectory of the drone during the height test experiments to determine if the height of two different boxes can be measured using the drones on-board differential pressure sensors.

Two primary experimental results were obtained. First, an approximate proxy for distance during flight was obtained by calculating the variance of the differential pressure sensor around a particular sample. 100 samples (1 second of data) was used. A proxy for height of the form:

$$h_p = \frac{1}{1 + v_p} \tag{4}$$

Where h_p is our height proxy and v_p is our differential pressure variance. This pressure is a noisy, but reliably proxy for height below around 10cm, as seen in figure 16.

In figure 17 is a recording of the vertical position of the drone during this trial, along with a proxy for height derived from the differential pressure sensor.

During this trial the drone takes off first, then performs a dip above open terrain. As can be seen in this figure there is minimal change in the variance of the differential pressure during this manoeuvre. The drone then dips twice, into close proximity of the two surfaces. In both cases the proxy height drops indicating that the drone is approaching a surface. No data is obtained at the end of the trial due to edge effects and disengagement of the rotors during landing.

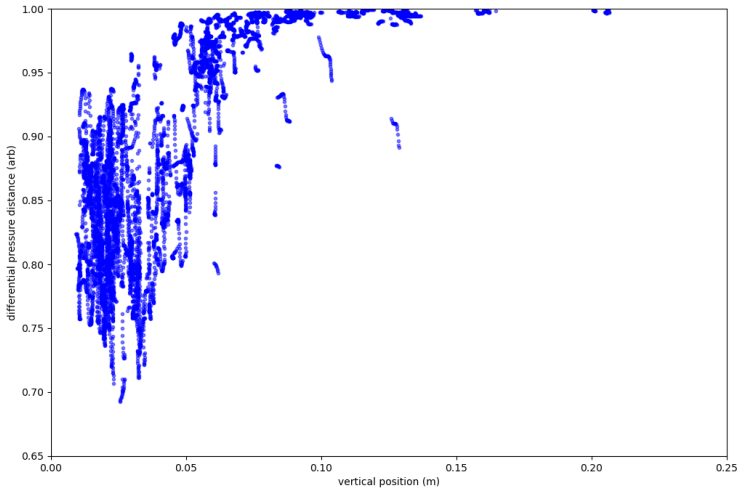

Figure 16: Differential pressure proxy height against actual vertical height in the test runs. Above 10cm this is a poor proxy, but below this scale it decreases noisily but monotonically as the planar surface below the drone approaches.

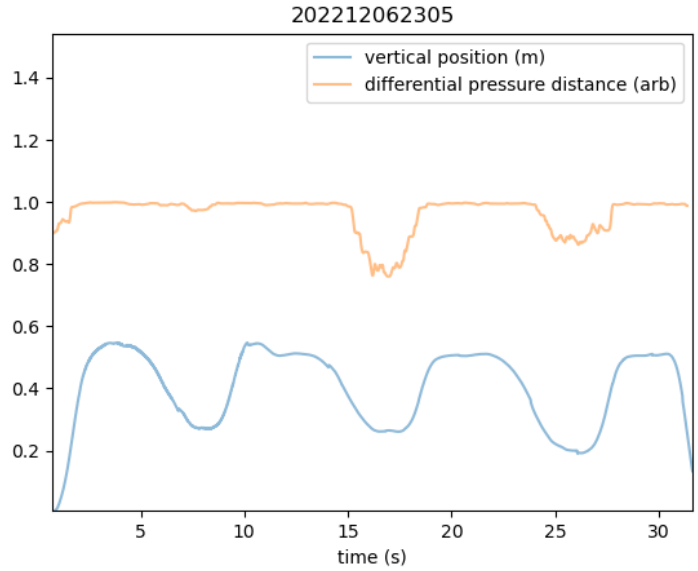

Figure 17: Vertical height of the drone during vertical manoeuvring next to two planar surfaces of distinct heights. In blue the actual height as measured by the LPS. In orange is the differential pressure proxy for height. During the first descent (with lowest height at 8s), there is no ground plane obstacle present. During the two subsequent descents (with their lowest heights at 17s and 26s) there is a solid surface obstacle beneath the drone that the differential pressure sensor detects successfully, as seen in the deflection of the orange line.

3.3 Surface insensitivity

We previously confirmed the surface insensitivity of our pressure sensors, a result which is now used in sense and avoid experiments detailed in section 4.2. In order to demonstrate that differential pressure based systems can be used on a wide variety of surfaces, the pressure signatures of various surfaces were recorded on a free flying drone.

The drone was operated as in the previous section, using the lighthouse positioning system and flown over four surface types; cardboard, smooth paper, thin carpet, and a thick towel. Example recordings are shown in figure 18 along with photographs of the surfaces. The differential pressure sensors are, as expected insensitive to surface type.

This is necessary for operations over featureless surfaces which should induce a failure mode in the Centeye optical flow sensors.

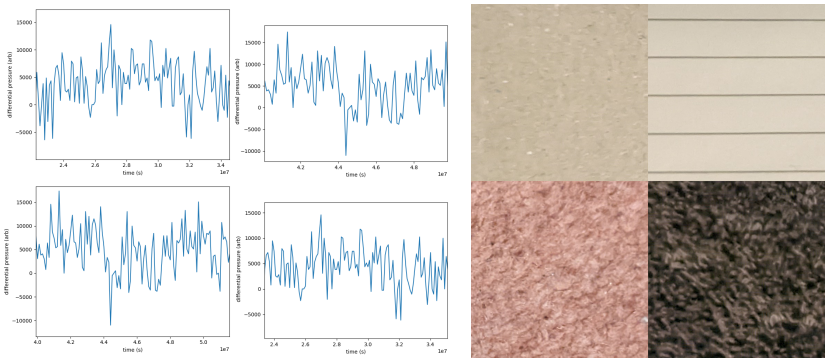

Figure 18: Recordings from the differential pressure sensors hovering 3cm above four different surface types, clockwise from the top left; cardboard, smooth paper, thin carpet, a thick towel. There is no obvious distinguishing difference between the recordings of the four conditions and height extraction performs correctly. A k-means clustering approach to the recordings also does not suggest a difference between these conditions, with a two, three and four cluster classifier performing comparable to chance and no obvious kink in graphs of explanatory power.

3.4 Final revision of the mormoops prototype

The final version of the mormoops capable of mounting the centeye optical flow sensors, differential pressure sensors, and microphones is described with a parts list and description of mounting points, and a chassis layout suitable for laser cutting, in the appendix.

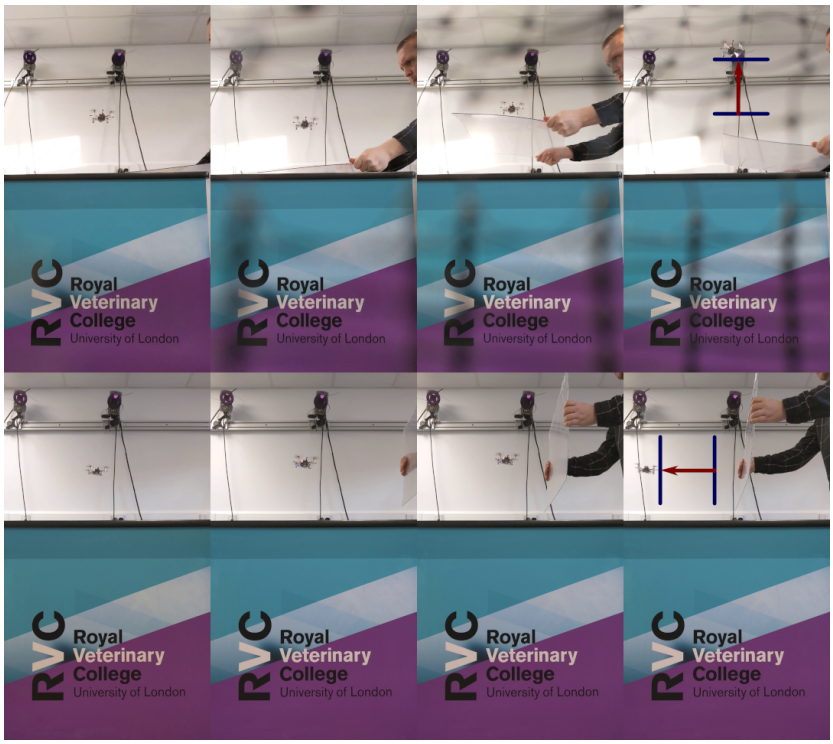

Figure 19: Still images from movie of surface avoidance behaviour. Above: ground avoidance as a horizontal planar surface approaches from beneath. Below: wall avoidance as a vertical planar surface approaches from the right.

4 Sense and avoid modalities

We have previously demonstrated the capability of our drones to avoid planar surfaces in the environment by sensing them with pressure sensors using the basic control scheme. Using the same configuration as with previous surface avoidance behaviour, we programmed a drone to move away from surfaces detected by the on-board microphone array. An example of avoiding both horizontal and vertical planar surfaces is shown in figure 19.

In addition to these previous experiments we have now conducted two additional experiments, one in the lab looking at using sensor integration to detect surfaces via sensory integration, and another sensing and avoiding obstacles in free flight using our more sophisticated control scheme.

4.1 Multimodal surface detection

Our multimodal sensor system has superior detection capabilities than the sum of the capacities of either sensor modality. Both systems have sensor failure

Figure 20: Representation of the three failure modes avoided by our sensor integration method, transparent, low-light conditions and distant surfaces. Note that the optical flow sensors are effective even in moderate light conditions, so the low light illustration is merely indicative.

modes which are common in a typical environment. In this experiment we tested our capability to reliably detect multiple difficult surfaces. These failure modes are depicted schematically in figure 20. Here our surface detection approach uses the algorithm with pressure sensors described in section 3.2, and the broad threats metric of the Centeye system.

For these experiments our drone was tethered and the motors ran at sufficient power to support the drone with various surfaces placed at a distance of 2cm. A reference cardboard surface with substantial texturing was used as a control. At this distance both sensors were able to detect the detect the target in good lighting conditions.

We considered three failure modes depicted in figure 20. The optical flow sensor cannot reliably detect unpatterned and transparent surfaces, since it depends on reflected light. With no reflected light there can be no surface detection. The pressure sensor was able to detect such a surface when sufficiently close.

In the first test case a perspex surface was used. The surface was 4mm thick, flat, unscratched, and clean. Visually this surface was highly transparent when looked at head on (from the side the Fresnel effect would render the surface visible). The optical sensors were not able to see this surface in this configuration. No substantial broad threats were recorded by the Centeye. However, the pressure sensors detected the surface without issue.

The second failure mode is total darkness. Without active sensing (for example using a pulsed light source) the optical flow sensor cannot detect surfaces, for the same reason as above

Finally, the reference surface was moved to $12cm$ away. At this distance the optical flow sensors reported an obstacle, while the pressure sensors did not.

This demonstrates the capability for our drone to operate in diverse environments, in diverse conditions where some, but not all, sensor modes fail.

4.2 Sense and avoid modalities - new control and free flight

This year we have performed free flight experiments avoiding complex surfaces using a more sophisticated control scheme. The control scheme is described

Figure 21: A sequence of still frames showing our drones capabilities when climbing a suburban staircase. The drone moves up the stairs consistently detecting the obstacles and avoiding them by invoking an override controller.

in section 2.3, here we describe a set of experiments designed to challenge our drone in a more complex environment than found in the lab with the previous experiment.

These experiments are feedback controlled. Without this feedback control in all the experiments described in this section the drone would crash into various surfaces in each experiment. They rely on the on board sensors detecting the surfaces to complete the task, and our new controller responding appropriately to those detected surfaces.

We selected a suburban house as our testing environment. This is a more challenging location compared with the lab with surfaces of different textures and orientations. Our objective was to perform three distinct tasks. Regular stair climbing, irregular height surface avoidance, and irregular stair climbing with variable surfaces.

In each case we are using the new control scheme described in section 2.3.

In the first experiment the drone is informed of the location of the first step in a set of stairs, and instructed to move up the stairs. It is told the average height and width of the steps, but nothing more. If the drone follows the instructions purely in the motion planning portion of the controller, it will crash into the staircase (this was confirmed by disabling the override controller completely).

If a surface is detected below the drone by the pressure sensors, the override controller is engaged and the drone is moved until it will not collide with the next step. Control is then returned to the motion planning controller and the drone repeatedly moves up the stairs.

A sequence of frames from this behaviour is shown in figure 21. A plot of the pressure sensor output, drone heights and control configuration is included in figure 22.

Two further experiments were performed along similar lines. In the first, the drone was instructed to fly into a planar surface, the override controller prevented this behaviour for various obstacle positions.

In the last experiment an irregular staircase was constructed out of cardboard boxes of various heights and textures (plain, laminated, etc.). As with the previous stair climbing experiment the drone was instructed to climb this irregular stair case only knowing the start location of the stairs, the direction to

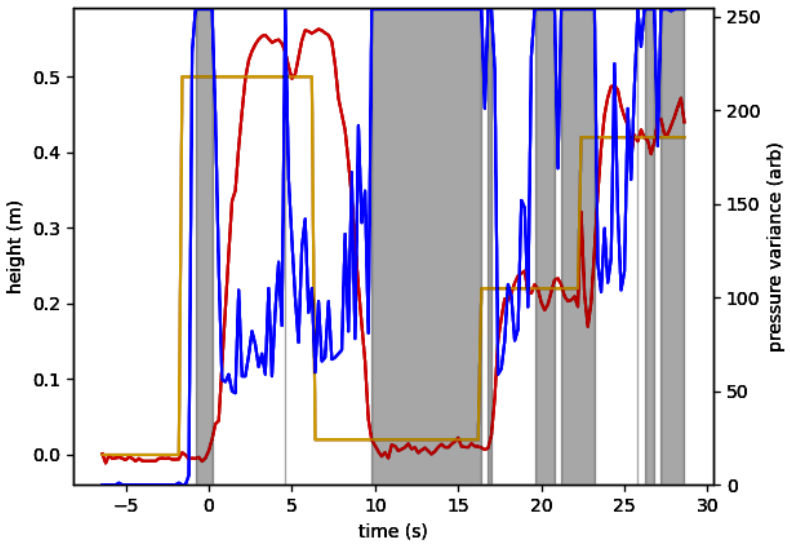

Figure 22: Plot of the drones control configuration (dark regions use the override controller, light standard motion planning), differential pressure variance (blue line), target height of the drone (orange), and actual drone height (red line) through the stair climbing trial. Notice how on the right of the graph the target height is adjusted upwards when the surface below the drone is detected.

move, and the stairs average length and width. This ensured that in this case the staircase could only be climbed if the sensors had robust surface detection. This was indeed the case, and further the drone was instructed to move past the end of the makeshift stairs. When the time came to do so the drone did not engage the override controller, detecting that forward motion was indeed safe.

5 Power optimisation

5.1 Weight reduction

As noted previously, the weight of the prototype drone was substantially reduced in previous years. While the original prototype had sufficient capacity to carry and operate the microphones and Centeye sensors, the drone had lacked manoeuvrability due to a suboptimal thrust-to-weight ratio. This would make operations, both for testing concurrent application of sense and avoid modalities, and measuring power saving through selective sensor use, more challenging.

The benefits of reduced power usage in sensors is proportional to the motor power used in flight, which is in turn proportional to the drone weight. The

motors typically draw several Ampere of current when in flight, contingent on load and desired manoeuvrers, while the primary draw of the sensor system is the Centeye optical flow sensor, which draws approximately 300mA depending on conditions.

Measuring reductions in power usage is difficult on the drone in free flight, however, measuring reduced battery usage should be more straight forward by recording multiple operations of the drone when the battery is at an identical charge level. While a theoretical demonstration of reduced power usage (as described in the sections on future work below), is desirable (simply take the duty factor of the Centeye, multiply it by the bench top power draw, and estimate the reduced power consumption), it is more desirable to demonstrate that this reduce power draw is detectable in practice.

This reduction in power usage for the drive system improves the relative magnitude of the power saved using the Centeye and makes detecting an extended lifespan for the drone more likely with feasible numbers of flight recordings.

The weight reduction was achieved in the following ways:

Chassis Redesign The shape, size and composition of the chassis was previously redesigned and optimized. In figure 23 the removed material from the chassis is illustrated. This alone reduced the weight by fifteen grams from twenty seven grams to twelve grams for the version of the chassis with mounts. For the high performance version the weight is reduced to five grams.

Further weight savings were made by changing the material used for the chassis. The original prototype used 1mm thick acrylic. The new version uses 1mm polycarbonate, which is lighter by five grams for the chassis with mounts (now three grams) and by three grams for the high performance chassis, which is now two grams.

In addition the chassis are more crash resistant. The acrylic chassis were typically the part irrecoverably damaged if the drone crashed. No attrition has occurred with the new polycarbonate chassis.

However, the new chassis is also more flexible, which makes it harder to control and fly. This is addressed by mounting two half gram carbon rods between the motors. This provides the necessary rigidity without adding substantial weight.

Alternative Battery For endurance flying, as is typical in a testing environment, the current battery is ideal with a large capacity. However, it is also heavy. For future experiments where the capacity to perform high-powered manoeuvrers is a higher priority than long flight times, a smaller battery will be preferable. The current batteries weigh thirty grams, while the lighter variant weighs only twenty-two grams, saving eight grams.

Alternative wiring Directly wiring the motors to the ESC, and shortening the wire from the ESC to the flight control board has two advantages. Firstly, it reduces the weight slightly (each connector is a little under half a gram),

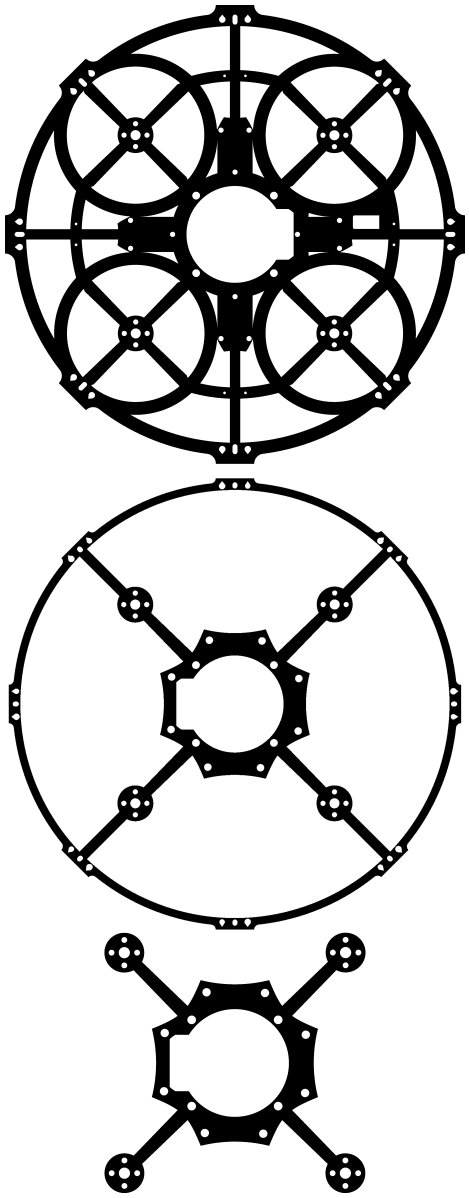

Figure 23: The chassis of the prototype has been redesigned to reduce weight. After significant testing much of the supporting material for the outer ring was identified as either superfluous or excessive in size. Where possible material on the body has also been removed. Three designs are shown, at the top is the original chassis design. The two below are the multi-sensor mounting variant, with mounting points for pressure sensors and microphones and cameras, and below a higher performance design which can only carry a smaller sensor payload.

secondly it reduces the need for the nacelles by reducing the possible travel of the wires connecting the boards and motors. This makes removing the nacelles possible further saving weight.

Nacelle removal While the 3D printed nacelles offer a number of advantages, they also add weight. With the reduction in length of the wires connecting the flight control board and the ESC, and the ESC connecting to the motors we have no further mechanical need for the nacelles. There is no noticeable difference in flight performance with or without the nacelles (suggesting that even if they help with airflow, that is counteracted by their additional weight). Since the drone can operate without them, they will be removed to save weight. Future testing could investigate if some variant of the nacelles improved airflow. The nacelles weighed ten grams.

Total weight savings The original prototype weighed one hundred and eighteen grams. With all of the above weight savings the new high performance drone weighs seventy three grams, a saving of forty five grams. The drone with additional mountings for sensors weighs eighty four grams.

The thrust-to-weight ratio of the original drone was very close to one with 20g of sensors outside of ground effect, as the drone was sluggish to respond and would lose height during moderate manoeuvrers under these condition. With the reduction in weight the thrust to weight ratio of the drone is now 1.6 permitting substantially improved sensor capacity or greater manoeuvrability.

An illustration of the new, reduced weight drone and the original prototype is included in figure 24

The primary aim with improving the flight performance of the drone is to improve our ability to confirm reductions in power requirements empirically for the drone in flight by selectively deactivating the Centeye optical flow sensor (by improving our statistical power through increasing the effect size associated with disabling this sensor). A secondary benefit is increased flight time, which in turn improves the number of recordings we can take of the drone during experimentation.

Improved statistical performance We can estimate the improved statistical performance for detecting energy savings on the drone in flight by means of a simple calculation. The motors for the original prototype drew around 1.3 ampere under steady flight outside of ground effect. The Centeye draws approximately 0.3 ampere. A reasonable proxy for the variability in power usage is the flight time, if we assume constant power draw the power usage ought to be uniform across flights (this is obviously only approximately true). Experience suggests this varies by as much as 50%.

If the duty cycle of the Centeye can be reduced from 1 to 0.5 then we will be attempting to detect a difference in average power draw of 0.15 ampere on 1.6 ampere, or around 9%.

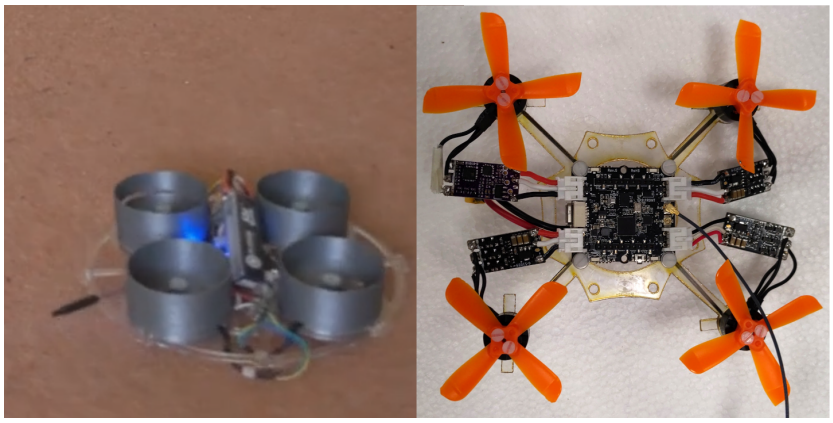

Figure 24: The original prototype and the improved, low weight modification, with a 50% higher thrust to weight ratio. The new drone operates with a lighter battery, removed nacelles, reduce lengths of wire between component, and a completely redesigned chassis which is both smaller, lighter, and made from lighter material.

With the savings in power the total current draw will be closer to 0.85 ampere, meaning we now need to detect a saving closer to 19%, which should be far easier.

To detect a difference of the first magnitude will require around 75 sets of observations, while to detect the second will require only around 19.

We have now performed this experiment to confirm the implied power savings, and considered a nuance associated with the power-up requires of the Centeye which mean restart frequency must be considered. The modelling and performance of this new algorithm is described in section 6.

5.2 Power and battery

The motors, ESC and flight controller are all compatible with either 2S or 3S batteries. All power for sensors and on-board processing is provided at 5V by the flight controllers integrated power management. In figure 25 the stack of power sources and sinks is depicted, excluding the motors which are managed separately by the flight controller.

5.3 Optimisation by novelty detection

We wish to adjust the sample frequency or activation state of a pressure and optical flow sensor in proportion to how unusual the observations are, assuming that the sensor is attached to the drone in steady state far from any obstacle. That is we would like a method for detecting when we are observing conditions which are radically different from steady state free flight.

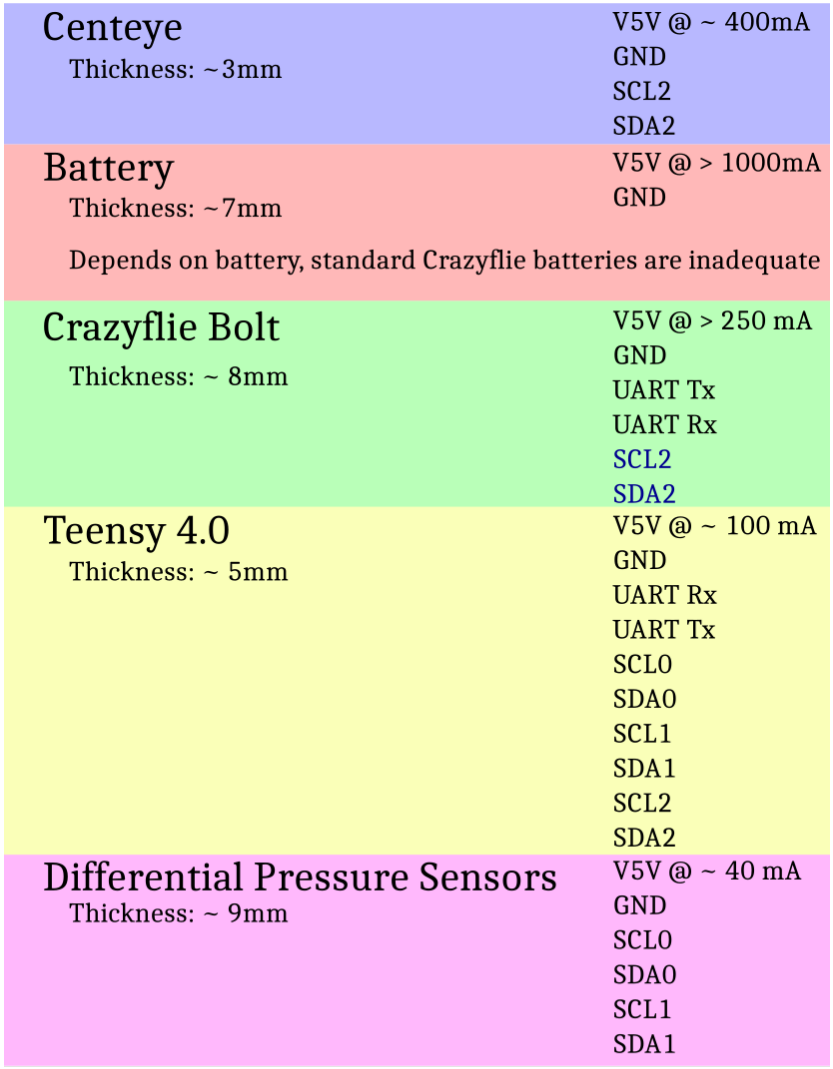

Figure 25: Power and communication systems. Power estimates are on the right, as are the channels used to connect each component. The teensy acts are a communication hub processing data from the sensors through I_2C, and passing high-level flight instructions to the flight controller via UART. We have upgraded to 2S batteries for the new drone, as the standard batteries for the Crazyflie are inadequate.

We model the state of the sensor using a Gaussian process. Samples from a Gaussian process are drawn from a multivariate Gaussian distribution with a covariance matrix specified by the covariance function $\Sigma(t, t')$. Since we can observe large quantities of steady state behaviour in non-obstacle conditions (free-flight), we can characterise Σ very precisely and construct it from a large number of positive definite basis functions.

Let us assume we have a set of observation x_i at times t_i. As the convolution of two Gaussians is itself a Gaussian we can compute the probability density for observing $x = \chi$ at time τ, it is just a Gaussian with an updated mean and standard deviation. As such we can calculate the probability that we obtain an observation as extreme as χ simple by computing the cumulative density function of this Gaussian.

This will allow us to build a classifier, by selecting the acceptable false positive rate T for this classifier, that is how often it identifies a background event as an anomalous signal. If the probability of observing χ given our previous observations xi, $P(\chi|x_i)$ is less than T then we classify this as an anomalous observation.

5.4 Anomaly detection

Anomalies during flight might take the form of obstacles or transient events like changes in air currents. We can model the presence or absence of these entities as a Markov process. This process has two states, an anomaly is present (with probability p), or no anomaly is present (probability $1 - p$).

Between these two states there is a transition matrix Q. The Markov property requires us to solve:

$$\frac{dP}{dt} = QP \tag{5}$$

Where $P = (p, p - 1)$. This has a simple and well known solution of the form:

$$p = p_0 + (p_1 - p_0) \exp(-\lambda t) \tag{6}$$

Here p_0 is the steady state probability for there to be an anomaly present (if the system is left for a long period of time it will tend to have an anomaly present with probability p_0), p_1 is the initial probability of the system having an anomaly, and λ is a decay constant controlling how rapidly the system transitions between these two states.

We have a classifier which can detect the presence of an anomaly. If no anomaly is present it has a probability T of detecting one (false positive rate), and a probability of correctly identifying the absence of an anomaly $1 - T$ (true negative rate). The true positive and false positive rates will depend on the type of anomaly under consideration. One radically different from steady state behaviour might have a high true positive rate, R, and a low false negative rate, $1 - R$, while in one which is barely distinguishable from steady state this might be reversed.

These will have to be parameters which determine how unusual an event we wish to detect. However, once these are specified we can use our sensor to

update our estimate of the state of the system (the presence or absence of an anomaly). If the sensor has detected an anomaly then if we thought there was a probability p that there was an anomaly prior to the observation, then, by Bayes Theorem, we now ought to believe there is an anomaly with probability:

$$p' = \frac{Rp_0}{T + (R - T)\, p_0} \tag{7}$$

Similarly if our sensor detects no anomaly we should now believe there is an anomaly with probability:

$$p' = \frac{(1 - R)\, p_0}{1 - T + (T - R)\, p_0} \tag{8}$$

Using these equations we can update our estimate of the probability that an anomaly is present or not, which will determine the sample rate we us.

5.4.1 Sample frequency control

We adjust the sample rate using a sigmoid. If the probability of an anomaly, p according to our Markov model is low (comparable to the prior probability of an anomaly), then we operate at a base sample rate ρ_b. If it is high then we want to operate at some high frequency ρ_M. We therefore interpolate between these two extreme using a sigmoid function, σ.

$$\rho = \rho_b + (\rho_M - \rho_b)\, \sigma\left(\frac{p - p_c}{s_p}\right) \tag{9}$$

where p_c is some central probability below which the sample rate should be low, and s_p is a scale for the rate at which a change in probability of an anomaly should adjust the sample rate (small s_p corresponds to rapidly cranking the sensor to maximum rate when suspicions are raised, the threshold for "suspicion" being controlled by p_c).

5.5 Results

Processing first simulated data, we create a suitable Gaussian process, generate a sample on which to train some hyper-parameters, and then generate a usage example where a second, different, Gaussian process is used to create an anomalous region with different covariance and noise. The results of applying our algorithm to this data can be seen in figure 26.

The region that is anomalous is obvious to a human observer, due to the slightly higher noise and strange, longer range correlation structure. It is also obvious to our classifier, which assigns a low probability of us being in an anomaly almost everywhere outside of the one we created, and a high probability for the region where the anomaly is present.

As a result the sample rate controller is able to pick suitable sample rates depending on the type of region (anomalous or not) that we are in.

For this test the average sample rate is a quarter of the maximum, but 95% of observations in the anomalous region are still made.

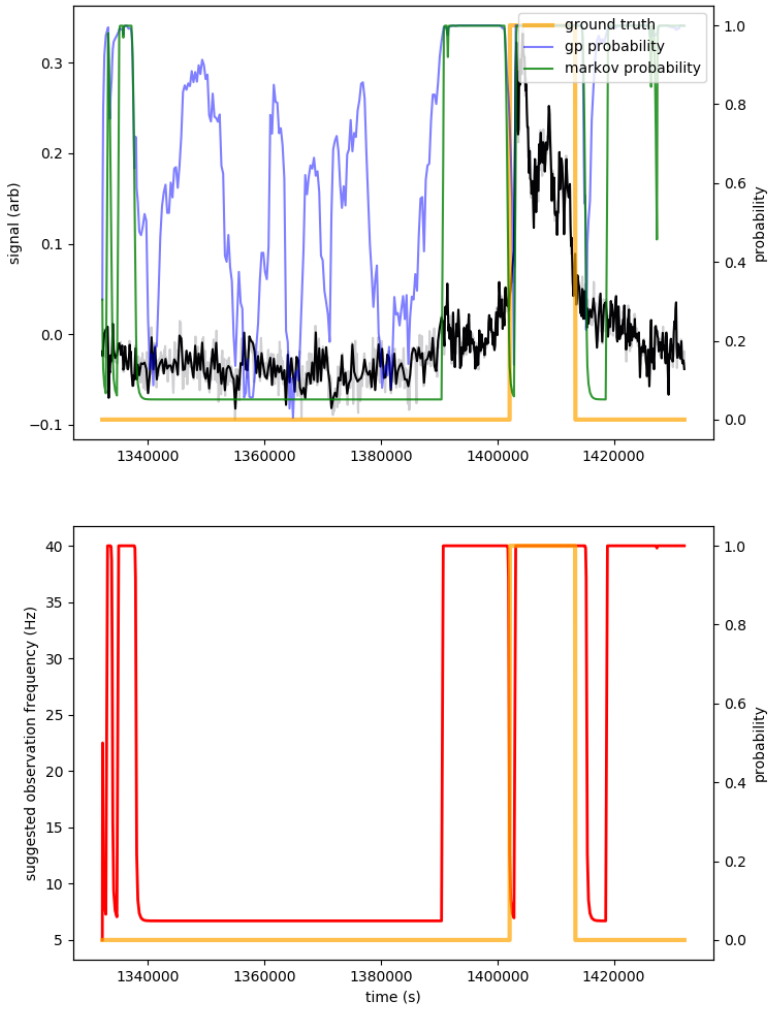

Figure 26: Results from a simulation run on a Gaussian process. In the top figure in dark blue is the signal, the high-alpha behind this is the maximum sample rate observation of the signal, the orange line indicates where the anomaly has been inserted, the pale blue line is the probability the classifier assigns to seeing an observation that extreme assuming the steady state Gaussian process, the green line is the probability we are observing an anomaly according to the Markov process. In the bottom panel is the sample rate the rate controller assigns to each set of observations, there is a clear preference for high sample rates in the anomalous region.

5.6 Modelling the covariance matrix

Modelling the sensor observations asynchronously as a Gaussian process requires us to estimate the covariance matrix both across variable and time. This must be done at least over the time scale that we intend to make observations over, which is no trivial task since such a matrix (technically a matrix function) is likely to be high dimensional. If μ label observation variables (pressure, inertial sensor recording like angular acceleration, etc.), then we can write our covariance matrix $\Sigma_{\mu\nu}(\tau)$, with τ the separation in time between two observations (we assume homogeneity).

Conventional methods to estimate Σ are likely to fail, because they require constructing and inverting the matrix. For example maximum likelihood estimation of a series expansion for Σ will require evaluating Σ for all observations and then inverting the resulting extremely high dimensional matrix (for our sample data this would be a matrix with a rank close to one million). We will therefore have to seek an alternative.

We can build an empirical model of our covariance matrix by computing the covariance of our observations. We shall see later that this has several drawbacks which must be overcome, but has the virtue of being tractable, at least in principle.

To proceed we shall consider the formula for the covariance of two random variables, with covariance homogeneous in time. This is:

$$S(\tau) = \langle X(\tau), Y(\tau) \rangle - \langle X(\tau) \rangle \langle Y(\tau) \rangle \tag{10}$$

In analogy to the method of kernel density estimation[2], we can use a Gaussian kernel to perform these averages over our asynchronously observations and build an estimate of the covariance. Note that we cannot simply use this formula as is, since the asynchronous and irregular observations mean we only have a single observation at most time intervals. Such an estimate would have the value:

$$S(\tau) = \int d\tilde{\tau} \sum_{i,j} (X_i Y_j - \langle X \rangle \langle Y \rangle) \exp\left(- (\tau_{ij} - \tilde{\tau})^2 / 2T \right) \tag{11}$$

Here i and j run over all observations and τ_{ij} is the temporal separation of the observation of X_i and Y_j, and T is a smoothing constant, it should have a scale characteristic of the smallest structure in the covariance matrix we are interested in.

This method is applied in a subsequent section to obtain covariance matrix functions for hovering Crazyflies.

However, there remains a problem which must be solved before such a method can be implemented, the requirement that the covariance function be positive definite. If this covariance matrix is not positive definite then the above integral becomes undefined (the integrand is divergent), and the method cannot be used.

6 Power saving by disabling the optical flow sensors

This year we explored the potential power savings from reducing the duty cycle of the Centeye system, that is powering the system down when the differential pressure sensor is sufficient for navigation.

The Centeye was selected because the power draw of the two primary sensor systems is not comparable. While the Centeye typically draws 300mA at 3.7V, the draw from the differential pressure sensors is around 20mA. Because our drone is configured so that the differential pressure sensors take power directly from the $3V$ line on the teensy, it is not practical to measure this power consumption directly without including the teensy's own power draw. This change in current draw would also be sitting on top of the approximately one Ampere draw of the motors on even our lowest power drone. This renders changes in power consumption from disabling the differential pressure sensors not only trivial, but basically unmeasurable.

Power saving benefits from disabling the Centeye are not only measurable but also potentially substantial. However, they must be carefully managed to realise actual power savings.

In order to quantify the potential power savings, we powered the Centeye separately during recording and considered a power saving algorithm. When the pressure sensors detect a surface, the optical flow sensors can largely be ignored, which means they can be powered down.

However, with this approach there is a catch, start-up for the optical flow sensors draws more power than steady operations, leading to roughly one and a half times the power consumption over a second or so (with a peak power draw of around 500mA). Disabling the optical flow sensor for a short period of time then reactivating it is thus a net power loss.

A simple algorithm to save power thus has to wait until the system is confident that the pressure sensors are detecting surfaces, and can disable the optical flow sensors for a protracted period of time. We have designed a simple system along these lines.

The system begins by waiting for the pressure sensors to detect a surface (such as a wall to be followed). Once this surface has been detected for one second, the Centeye system is disabled. Once the pressure sensor is no longer detecting the surface, the Centeye is reactivated. If power draw from the Centeye is P_0 during steady operation, the energy use for a full restart is E_r, the frequency with which restarts occur is f_r, the duty cycle of the Centeye (the time the Centeye is powered on vs. powered down) is d, then the typical power consumption is:

$$P = P_0 d + E_r f_r \tag{12}$$

The power saving over leaving the Centeye on permanently is:

$$\Delta P = P_0(1 - d) + E_r f_r \tag{13}$$

We can see that this only saves power if:

$$P_0(1-d) > E_r f_r \tag{14}$$

The restart frequency is a function of the environment. If we are performing wall following operations for say, 20 seconds where the pressure sensor will reliably detect the obstacle, then the power savings are substantial (around 6mAh). This power savings have been confirmed in bench top measurements, with the Centeye powered from a bench top power supply.

Note that for cluttered environments this algorithm can lose power. If the restart frequency is too high, then the benefits of powering down the system are lost to the additional current draw during restart.

7 Wind vector estimation

This year we have confirmed that, by using the kalman filtered state estimates of the crazyflie platforms our drones are based on, we are able to estimate airspeed as a proxy for wind vector estimation. The wind vector can be reconstructed by the airspeed using the Centeye to obtain absolute odometry or station keeping.

At a steady state with a fixed velocity (which is up to insignificant inertial correction terms equivalent to being stationary in a moving block of air) the thrust from the drone must match the drag from the air. There is thus a direct, monotonic relationship between thrust and velocity at steady state.

In order to estimate speed in the air, we thus simply require a relationship between some measurable state on the drone, and the acceleration of the drone. Fortunately there is a direct relationship between the orientation of the drone and the thrust if we are close to steady state motion.

We will assume that a model of the drag at fixed airspeed exists, which can be readily obtained by flying the drone at fixed airspeeds. The more difficult challenge is obtaining a vector estimate for the thrust, which we solve here.

In order to perform vector estimation of movement, we will require a means for taking the observed orientation of the drone (which will be in some representation of the special orthogonal group), and converting it into a vector estimate of wind speed.

Fortunately in this case we can solve the trivial case, when the rotation is confined to the roll axis, and then transform any other case to this case by means of another rotation. The relevant transformation is easily estimated by calculating the transformation needed to convert the estimated state of the drone, into one which is purely a result of rotation about this axis.

As such, we only need to find a relationship between roll and acceleration along a single axis. To estimate this we flew our drone moving between two points separated by 80cm repeatedly, with timescales for the transition of 4s, 3s, 2s, 1s and 0.5s. For the two shortest timescales the drone was not able to fully complete this manoeuvre stopping short of the full 80cm transition, but higher accelerations were recorded which was sufficient for our purposes.

From the odometry and kalman filter the position and roll were estimated, the position differentiated twice, and a linear relationship between acceleration and orientation estimated. This result is shown in figure 27.

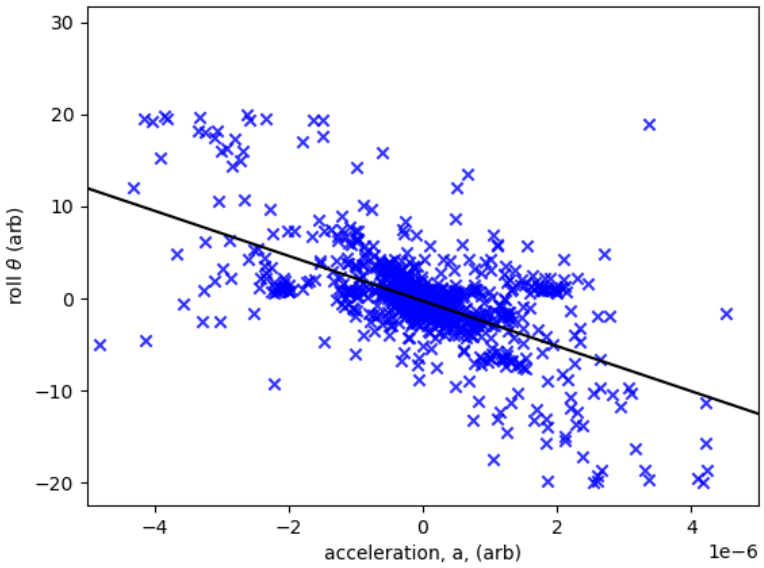

Figure 27: Estimation of airspeed around the drone can be reduced to an exercise in estimation the drones acceleration and odometry. The former requires a linear relationship between a state variable (in this case roll), and the acceleration of the drone. In this figure the acceleration of the drone is plotted on the x-axis, the roll is on the y-axis. A linear regression of these two factors is included, with a significant slope, $p < 0.001$.

Further free flight validation tests were performed. Here a crazyflie was flown in front of a wind tunnel (image in the left panel of figure 29) with station keeping performed as previously. The angular state of the drone was estimated and reported.

However, in the wind tunnel it was found that the kalman filter based state estimator was revising the estimate of the drones orientation with each step based on the equations of motion. This meant that the estimate of the wind strength was very poor, and could only be based on a small number of observations, as shown in figure 28.

To correct for this the magnitude of the wind was estimated from the inertial measurement unit directly, rather than from the output of the kalman filter. The angle of the drone to horizontal was estimated as:

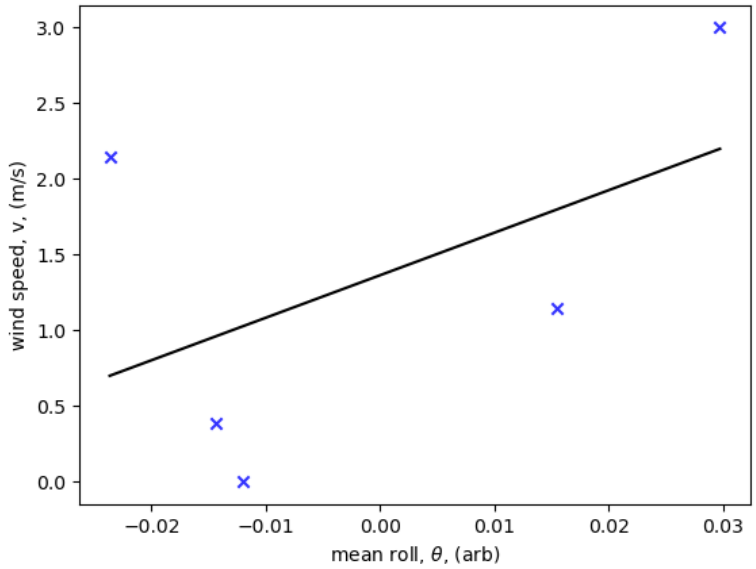

Figure 28: Estimates of the wind speed inside a wind tunnel are extremely poor, even when restricted to a short time period upon entering the tunnel (here 2 seconds). Here the x-axis is the mean roll of the drone, the y-axis is the wind speed inside the wind tunnel. The line is a linear fit to the data, but little to no stock should be placed in this model, it is an extremely poor fit to the data.

$$\theta = \arcsin(a_x / \sqrt{a_x^2 + a_y^2 + a_z^2}) \tag{15}$$

The wind tunnel was operated at a sequence of different wind speeds, three trials at each wind speed, and a plot of wind velocity against mean drone angle is included in the right panel of figure 29, along with a model of wind speed against angle. Further work could be performed to validate the relationship against the previous results using a suitable thrust and drag model, and then the result extended to higher wind speeds.

The wind speed is accurately estimated across a wide range, although the fit is of a slightly lower quality at lower speeds. Note that the direction of gravity is estimated on power on to calibrate the sensor, and this direction does not inherently align with the orientation of the drone. This is the reason for the negative angle at zero wind speed.

During the course of studying this phenomena an immediate use was apparent. Since the Kalman filter was constantly updating the best estimate for "level" for the drone, over the course of the trial the state estimator became increasingly confident that the drone was level during rapid air flow when inclined.

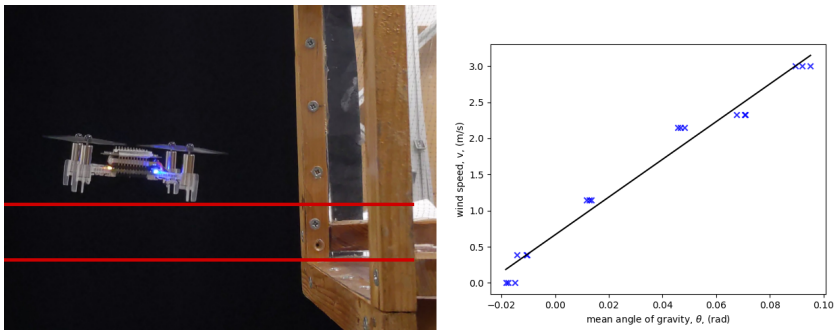

Figure 29: On the left is an image of the drone being operated next to a wind tunnel generating constant airflow. The red lines are horizontal (aligned with the exit to the wind tunnel), the drone is inclined so as to resist the airflow. On the right this wind speed plotted on the y-axis, the roll is on the x-axis. A linear model of these two factors is included, which is an excellent fit, although poorer at lower wind speeds, with significant terms, $p < 0.001$.

When the drone attempted to land, leaving this airflow, the drone immediately began moving counter to the now fictitious wind flow. At low speeds the drone could recover, but at the three maximum speeds in this trial the result was a consistent crash.

This problem could be readily overcome by feeding the wind speed estimate derived here into the kalman filter. This would prevent the filter from misidentifying the inclined state. This would prevent drones operating at higher altitudes moving down out of airflow from correcting for winds after then had exited the increased airflow, preventing crashes.

8 Conclusion

This year we have successfully demonstrated the power of sensory integration on our drone platform, demonstrated sense and avoid strategies with a new control scheme, demonstrated how careful power management can extend drone flight time, and shown how our drone can estimate wind speed from on board sensor data. This represents a completion of all major objectives. We have previously published[3] results from the second and third year in the International Journal of Micro Air Vehicles. We have presented this final year's results at Ubiquitous Robotics 2023, and we will be presenting them at DARC 2023.

References

[1] B. T. Polyak and A. B. Juditsky. "Acceleration of Stochastic Approximation by Averaging". In: *SIAM J. Control Optim.* 30.4 (July 1992), pp. 838–

855. ISSN: 0363-0129. DOI: 10.1137/0330046. URL: https://doi.org/10.1137/0330046.

[2] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. Vol. 383. John Wiley & Sons, 2009.

[3] Simon Wilshin, Stephen Amos, and Richard J Bomphrey. "Seeing with sound; surface detection and avoidance by sensing self-generated noise". In: *International Journal of Micro Air Vehicles* 15 (2023), p. 17568293221148377.

A Appendix - Technologies and prototypes

Here we describe in additional detail two primary technologies, the final version of the mormoops drone, and the refined differential pressure sensor board, including schematics and component lists.

A.1 Mormoops nano-UAV

Here we describe the fabrication and assembly procedure for the mormoops nano-UAV. This vehicle is capable of carrying the revised differential pressure sensors, microphone sensors and Centeye optical flow sensors discussed in the report. The microphone array and differential pressure sensors are described in later segments in this document.

A.1.1 Chassis

Multiple chassis designs for the mormoops platform have been created, the final, multimodal chassis is described here. This platform can carry any two of the microphone array, the Centeye optical flow sensor or the differential pressure sensor. It is cut from 3mm acrylic.

In figure 30 is a annotated laser cutting template indicating the mounting points for each component on the drone. Mounting instructions are included in the caption for this diagram. The fully assembled drone is depicted in figure 31.

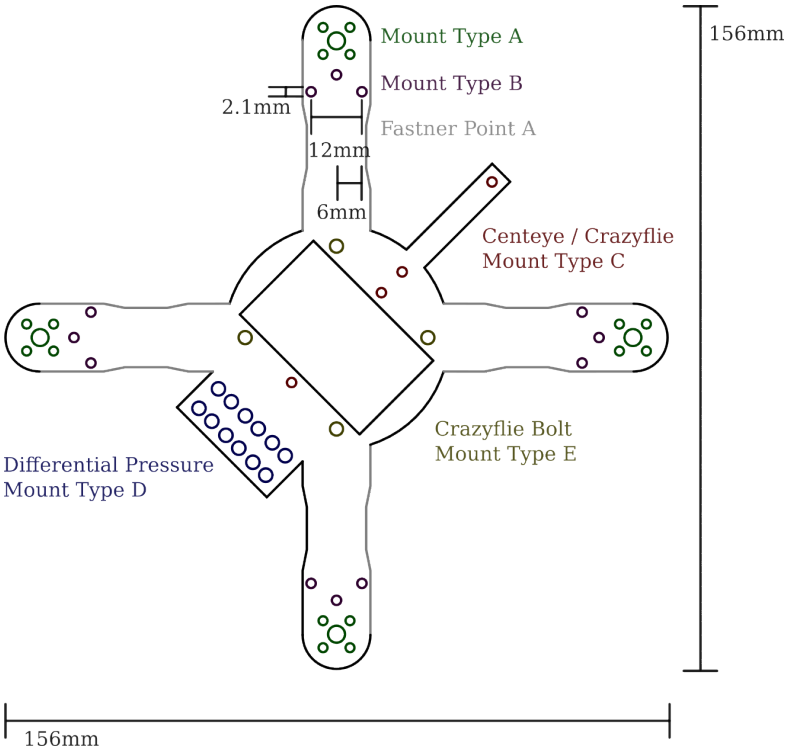

Figure 30: Laser cut template of the final mormoops design with mounting types indicated (this diagram is to scale and the vector version can be used to laser cut the chassis, so not every distance has been included for standard mounts). With the exception of the mounting point (mount type D) for the differential pressure board, and the optical flow mounting points (mount type B), the remaining mounts are standard to the relevant manufacturer and the technical specifications have not been included here, the former is described in the technical drawings for this sensor. Mount type A is a standard BetaFPV mount for 5000KV brushless motors. Mount B is for nylon screws used to hold the Centeye optical flow sensors in place below the drone, three 2mm nylon screws 40mm in length should be used to hold the cameras in place. Optionally a mounting plate with the same layout as this fastener can be attached to the bottom of these screws. Mount type C offers two locations where the Centeye optical flow sensor can be attached, it uses the standard crazyflie 2mm screw mounts. The top right should be used if the lighthouse positioning system is included on the drone, as otherwise the mount location conflict. Otherwise this sensor can be mounted on top of the drone using the central mount location. Mount type D is used for the differential pressure sensor, which is described later in this document. Mount type E is the standard crazyflie Bolt mount from Bitcraze, where the flight control board should be mounted. The fastner points A are for holding the BLHeli-S ESC can be held in place. These can be fastened in place using nylon wire, or with velcro. The latter allows for easier replacement of the ESC and is generally more secure, but the former is lighter.

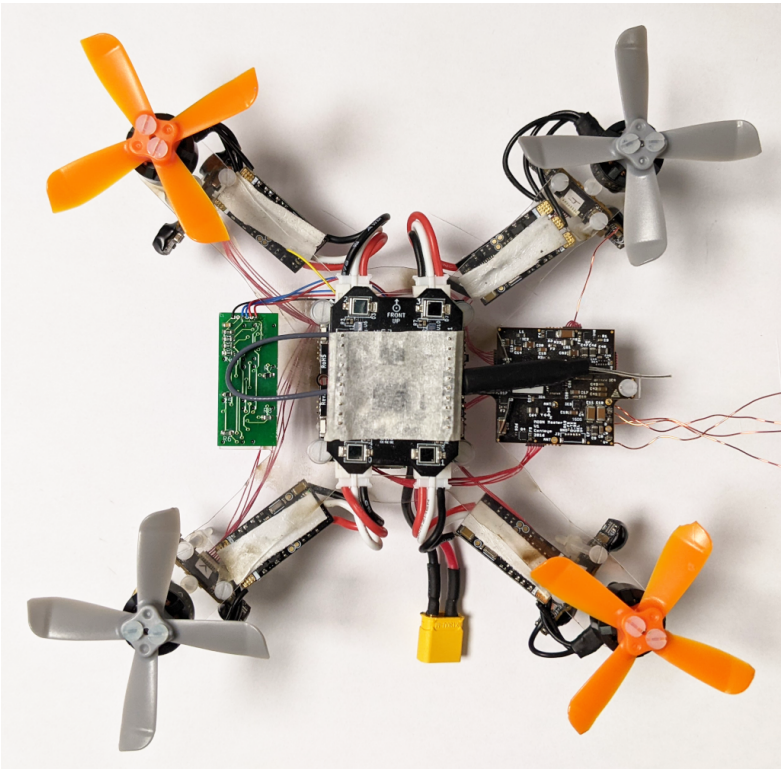

Figure 31: An assembled final mormoops drone in the differential pressure and Centeye optical flow sensors mounted. On top of the drone is the Lighthouse Positioning System (LPS) from Bitcraze that can be used for off-board control or station keeping. Alternative configurations include microcontroller mounted digital microphones (up to eight channels via Inter IC-Sound) below the drone. This configuration is not depicted.

A.1.2 Component List

To assemble the base drone without sensors or external control hardware the following parts are required:

- The included laser cut chassis
- 4 x BetaFPV 1105 5000kv Brushless motors
- 4 x BLHeli-S 12A electronic speed control board
- 1 x crazyflie Bolt flight control board from Bitcraze
- 4 x 3mm nylon screws and bolts, 40mm in length
- 14 x 2mm nylon screws and bolts, 40mm in length
- 1 x 1S Battery (2S, model XT30 from TBS Graphene)
- 4 x 2 inch propellers (DYS Shark Mako Prop 20354B)
- A reel of Nylon wire

A.2 Revised differential pressure sensor board

Here we include the schematic for the layout of the new, compact differential pressure sensor boards. In figure 13 we include the schematic for the board, along with the board design. A photograph of the board mounted to the mormoops is included in figure 31.

The new board is smaller and lighter than the previous design, and includes an additional sixth channel.

A.2.1 Component List

This board requires the following components, labels are as in the schematic:

- C1 is a 100nF bypass capacitor rated for 5V, metric 1608 form factor
- J1 is a mounting for a connector, but no connector should be attached if weight is a concern, wires will be directly soldered
- R1-4 are 1kOhm resistors, 5% tolerance, form factor metric 1608
- R5 and R7 are 1.2kOhm resistors, 5% tolerance, form factor metric 1608
- R6 and R8 are 2.7kOhm resistors, 5% tolerance, form factor metric 1608
- U1-6 are Sensirion SDP33 differential pressure sensor

B Other procedures, software and protocols

Other procedures, software and protocols are described in this technical report in place.