Autonomous Object Tracking and Obstacle Avoiding Multirotor of Team Aeolus, PES University

Sovan Das[a], Akshay Kalkunte Suresh[a]
Department of Telecommunication Engineering
Siddharth R N[a]
Department of Mechanical Engineering
Anush S Kumar[b]
Department of Computer Science & Engineering
Pranav Singhania[a]
Department of Information Science & Engineering

[a]PES Institute of Technology, Bengaluru, India
[b]PES University, Bengaluru, India

ABSTRACT

This paper details the design and assembly of an autonomous micro aerial vehicle with navigation capabilities in GPS denied environment developed by Team Aeolus, PES University. It uses a downward facing optical flow sensor for state estimation and computer vision using an Intel® RealSense™ (R200) [3] camera for localisation. It is capable of tracking and guiding multiple randomly moving ground robots by priority assignment programming while actively making use of dynamic path planning to avoid ground based and aerial obstacles observed by a rotating LIDAR [10] Sensor.

1. INTRODUCTION

1.1 Problem Statement

The 7th mission of International Aerial Robotics Challenge involves interaction of the aerial robots with the constantly moving ground robots, navigation in futile, unsupportive environment with no external aids or stationary point of references which is followed by the interaction of one aerial robot with other aerial robots. In the first mission, the ground robots are to be herd towards the green line in a $20x20m^2$ arena by the aerial vehicle while dodging the obstacle robots. The aerial robot changes the direction of the ground robots by touching the tactile switch present on top of each ground robot. When the switch is tapped once the robot turns 45^0 clockwise & when landed in the front it will turn 180^0. Mission completion is achieved when at least seven ground robots cross the green line in the given time limit. [1]

1.2 Conceptual Approach

1.2.1 Stabilize

The dynamic state of the multirotor is defined by the following parameters:

1) Individual Motor speeds
2) Attitude

3) Pitch, Roll and Yaw rates.

To stabilize the multirotor, the manipulation of these parameters is imperative. PID (Proportional–Integral–Derivative) tuning of different state affecting systems is done to stabilize the multirotor. The on-board inertial measurement unit (IMU) and optical flow sensors provide real time information and feedback on the state of the multirotor. The optical flow sensor provides the position of the multirotor on the projected Cartesian plane along with the Pitch, Roll and Yaw rates. The noise generated by the stabilizing sensors is minimized by use of an Extended Kalman Filter [7].

Ground effect on the multirotor- As per one of the competition objective, the multirotor is required to touch the top of the ground robots to result in different forms of movements for the ground robot. In other words the multirotor is required to fly close to the ground. This results in the development of a high pressure region under the multirotor that leads to wobbling. This has been avoided by increasing the height of the landing gear attached to the multirotor.

Optical flow [2] sensor- The X, Y distances calculated by integrating the X, Y velocities reported by the optical flow sensor were quite different from the actual ground distance values measured. Hence to overcome this we performed a custom calibration of the sensor, by comparing the returned values with the actual values, and developed a mathematical corrective correlation. The SONAR rangefinder on the optical flow also generates a lot of noise leading to numerous junk values, in order to decrease this, an Extended Kalman Filter [7] and Histogram Filter [8] are used. The SONAR also has a minimum height for operation, which is 30cm. In order to negate this aspect, tall landing gears are used.

On-board IMU- The on-board accelerometer and gyros are all MEMS and these sensors are highly susceptible to magnetic interferences. When the NUC and buzzer were in close proximity to the board, junk values were generated by these sensors, leading to serious stability issues. To avoid this, the NUC is mounted on a glass fibre plate and the buzzer is kept as far as possible.

1.2.2 Navigation

Navigation is performed using a vision based system comprising of Intel® RealSense™ (R200) [3] 3D camera and PX4flow [4] optical flow camera.

The arena is navigated with the aid of computer vision algorithms in detecting visual landmarks on the arena floor. Canny Edge Detection [5] is employed in detecting the edges and Hough transform [6] is used to filter out only the lines corresponding to the arena grids. This allows us to accurately pin point the quadrotor's location in the arena and allows it to interact with the ground bots and obstacles.

1.2.3 Object Detection

Two Intel® RealSense™ (R200) [3] cameras are mounted on the quadrotor so as to provide a FOV (Field Of View) of the arena. By analysing the point cloud of the RealSense™ cameras, the RGB and depth streams are correlated and the positions of all objects in the arena relative to the quadrotor are obtained. The PX4flow [4] camera allows for optical flow calculations which are used to perform position estimation thereby working in unison with the RealSense™ cameras.

As there are multiple objects in motion all over the arena which might enter the cameras' frame of view at any point of time, and the camera itself being in motion subjected to varying lighting conditions, there is a need for a visual system that can learn, detect and track moving obstacles quickly and accurately. A HOG (Histogram of Oriented Gradients) based SVM (Support Vector Machine) is used to achieve this. The network is trained offline with positive images of the ground bot shot at different angles, elevations and varying light conditions along with negative samples without the ground bot. The negative samples to positive samples are in the ratio 5:1.

1.2.4 Planning

Identification of the ground bots and obstacles is of primary importance in the competition, which is done using the on-board Intel® RealSense™ (R200) [3] cameras and a LIDAR [10] rangefinder. When the ground robots and obstacles are identified by the on-board image processing, their respective distances from the multirotor are calculated. Each square on the arena serves as a reference point. Each square has two sides which are parallel to the green line. The side closer to the green line is chosen. Each of these sides has two edges. These edges on the square containing the closest ground bot free from close obstacles are identified. The peripheral green line square edges are identified. The algorithm plots straight lines between the two edges from the ground bot square to the peripheral green line edges. In the first iteration, the on board processor uses ground bot and obstacle distances to compute all the lines of least resistance. In the second iteration it chooses one line out of all the lines of zero resistance which has the greatest perpendicular distance from the closest ground bot or obstacle. The ground bot is then rotated into an orientation which is parallel to the chosen line. By using the above algorithm, the margin for obstacle interaction between the ground bot and multirotor is greatly reduced.

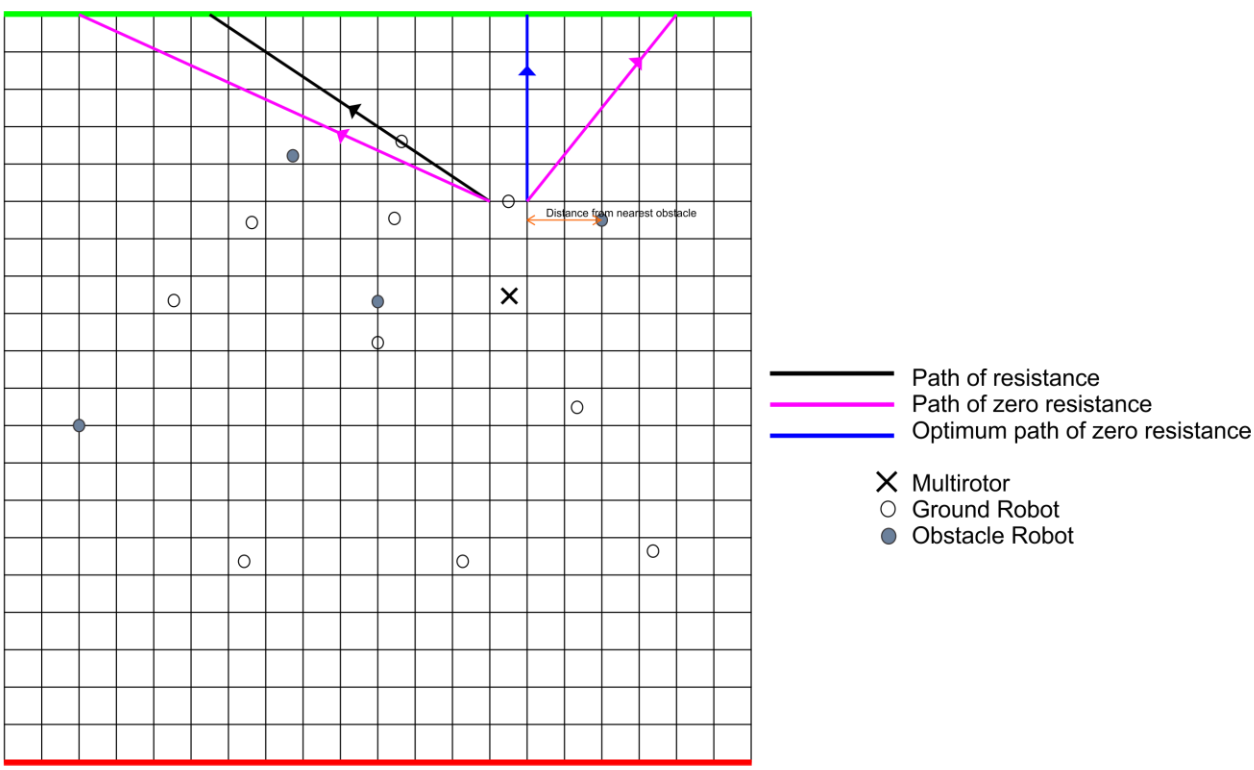

Figure 1. Path Planning

1.3 System Architecture

Figure 2. System Block Diagram

1.4 Yearly Milestone

This is the first time team Aeolus is entering IARC. Based on our own understanding of the problem statement while also taking lessons from the experience of past competitors, we have tried to build a stable and robust multirotor with autonomous capabilities. We have been developing the quadrotor since January, 2015.

2. AERIAL VEHICLE

2.1 Propulsion System

The multirotor used by us is a quadcopter. It consists of four brushless motors which are used to primarily generate lift. An X configuration is used to allow for a larger FOV (Field of View for the Multirotor). Navigator series Tiger MN4010 [9] motors are used. These are pancake motors as they have a low aspect ratio, the motors work at high 6 cell voltages i.e. 22.2 V using very little current since they work at high voltages. The consumption of low amount of current allows us to reduce the battery pack weight since the charge required is low, and also for a given charge rating of a battery these motors give higher flight times. Usage of low amount of current also leads to minimal heat generation, which is an indication of high efficiency.

Figure 3. Tiger Motors MN4010 370KV

Propellers- 15*5.5 Carbon fibre propellers are used to generate lift. Large propellers are used since the motors running are low KV motors. Carbon fibre propellers do not undergo flexing during rotation, hence there is minimal loss of lift. The propellers are mounted on T style mounts.

ESC- the Electronic Speed Controllers used on the multirotor are the primary controllers of motor speed. The ESC work on the Simon K software and help generate different motor speed by an input PWM signal. The ESC's used on the multirotor are Afro 20A ESC's. They were chosen as they work at high voltages (6 cell- 8 cell LiPo) and have a current rating of 20A which is 5% higher than the maximum current used by the motor at maximum throttle. This ensures that the ESC and motor don't overheat and burn.

2.2 Guidance & Navigation

The micro aerial vehicle needs to know its target so as to plan its path.

Initially, the relative position and velocity of every ground robot that is present in the RGB and depth frame of RealSense™ with respect to the multirotor is calculated. From this information, the relative velocity & position in the corresponding NED (North-East-Down) coordinates are generated with respect to the quadrotor's inertial frame and fed to the flight control unit to enable it to follow the ground robots.

Using a feed forward neural network the bot which is moving towards the green line or the bot which requires least manipulation to make it move towards the green line is selected.

With respective to the aerial obstructions, the rotating LIDAR [10] provides the distance of the obstructions at a step size of 1^0 in all directions which is analysed to predict any collision and avoid it.

2.3 Control

The Autonomous Multirotor consists of two main control systems:

1. Flight controller- This system is mainly responsible for the stability of the multirotor. We use the PIXHAWK [11] as our flight controller, and the on-board gyros and accelerometer are used to determine the attitude and automatically stabilize the multirotor by the usage of stock system programs. It is also used to control the motion of the multirotor by converting the input values produced by the on-board processor to PWM values read by the Electronic Speed Controllers (ESC).
2. On-board processor- The multirotor uses the Intel® NUC for all dynamic and static computations. This system commands the flight controller based on the sensor data it receives from the state determining sensors as well as the obstacle and ground robot detection sensors and cameras. The processor communicates with the PIXHAWK [11]

through commands sent through MAV protocols. These commands are compiled and executed on a Robot Operating System and result in the movement of the multirotor.

As per section 1.2.2, 1.2.3 and 2.2 the data obtained is sent to the NUC which with the help of a custom algorithm solves the dynamic maze which is then processed as MAVLink messages understood by PIXHAWK [11] to manipulate multirotor movement. The response of the PIXHAWK [11] is tuned to an acceptable level by PID tuning.

PID tuning- As mentioned earlier in section 1.2.1, it is a method used by control engineers to manipulate the response of a closed loop control system by using a PID transfer function. PID tuning was done on the multirotor to manipulate the response of the body to a published ROS topic related to position and velocity generated by the bot-tracking algorithm. For doing so, the PID gains on the three control axes must be set respectively.
This was done by running a test code which would take the multirotor to a suitable height and perform impulsive positive and negative roll, pitch maneuvers. The response of the code for a given roll, pitch, yaw angle was observed. When the response of the multirotor within close tolerances of the input movements, the multirotor was said to be tuned.
For each axis maneuver, initially the P gain was increased from zero keeping I and D gains at zero. The P gain was done up to a value, where the response was almost equal to the input angle. The attitude of the multirotor was observed from the GCS-Q Ground Control. When the P gain was set, the D gain was increased from zero to a point where the speed of response of the multirotor was acceptable. It was seen to that the multirotor was neither too aggressive nor too slow in the response. Finally I gains were set so that the steady value of the response was within close tolerances of the input.

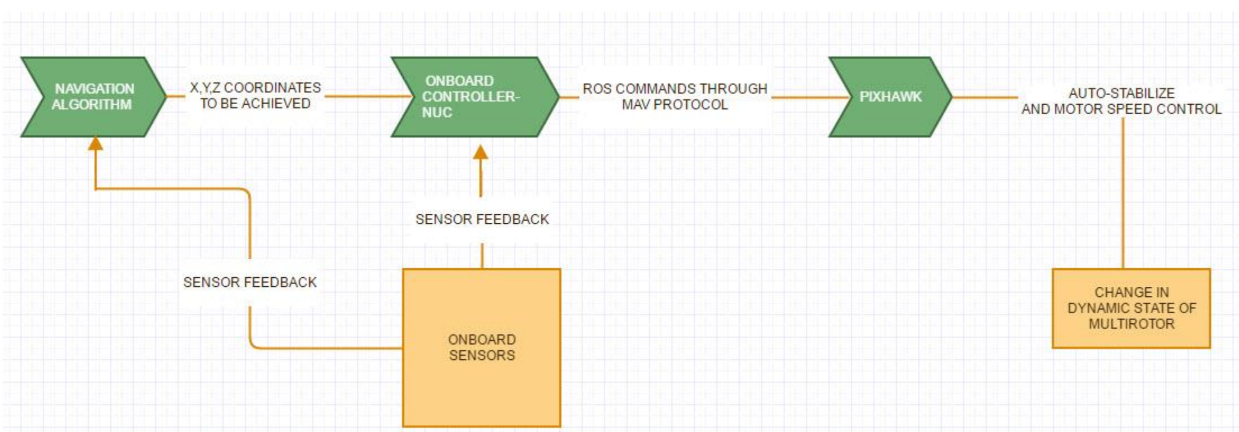

Figure 4. Flow of Control

2.4 Flight Termination System

The vehicle is equipped with three safety switches to terminate the flight in case of emergency. The first one is a RC switch which kills the power supplied to the motor when appropriate PWM is generated from the set channel from the transmitter. A PPM signal sensitive switching circuit is used to perform the above said function. The next switch is a mechanical switch between the power distribution board and the batteries which can be used to completely cut-off the supply to the flight. Another pulse button switch is implemented in between the flight control board and the ESCs which helps in killing the signal sent to the ESCs resulting in flight termination.

3. PAYLOAD

3.1 Sensor on-board

- LIDAR [10] – This device has a distance range of (0.2m-6m) with a distance resolution of <0.5mm and angular range of 360^0 with resolution of $<1^0$. The sampling frequency is >=2000Hz and the laser used has a wavelength of 785nm and 3mW power which is human-safe.
- Optical flow – Manufactured by 3DR robotics, PX4flow [4] sensor uses a camera with focal length of 16mm to provide the flow data from which relative velocity and position of the quadrotor in X-Y direction is estimated. It has a MAXBotix Sonar attached to it which provides the distance of the vehicle from the ground when in the air. The sonar has a lower range limit of 0.3m and the optical flow readings are only valid for certain maximum speeds with respect to the ground distance, for example when flying at a height of 1m, the horizontal velocity cannot be more than 2.4m/S so that the position can be estimated accurately.
- 3D Camera – Intel® RealSense™ R200 is the 3D camera which has maximum depth perception of 4m, FOV of (77^0x43^0x70^0) cone for RGB frame & (70^0x46^0x59^0) cone for depth frame and the pictures are taken at a frame rate of 60fps. It is connected to the processing unit using USB 3.0.

3.2 Communication Systems

The micro aerial vehicle uses all low power communication systems. First, the system which connects the on-board processing unit to ground unit is WiFi 802.11b/g/n. Second the system which connects the flight controller unit to the ground unit is 915MHz telemetry, which uses different bandwidths for uplink and downlink. The radio transmitter connected to flight controller unit for manual control uses radio link over 2.4GHz with 8 channels which is demodulated by the receiver. Flight controller unit accepts PPM (Pulse Position Modulation) hence the PWM (Pulse Width Modulation) signals given out by the receiver is again encoded into PPM signals. The processing unit is connected to the flight controller through USB and uses RS232 protocol hence a USB-TTL circuit is used.

3.3 Processing Unit

This on-board processing unit on the vehicle is Intel® NUC which takes input from the 3D camera and LIDAR [10] connected via USB 3.0 and controls the flight controller unit. The specification of the unit is as follows

- Processor: Intel® Core™-i3-5010U CPU @ 2.10GHz x 4
- Memory: 8GB RAM
- Secondary Memory: 120GB SSD Hard Drive
- Graphics: Intel® HD Graphics 5500 (Broadwell GT2)

3.4 Flight Controller Unit

The 3DR PIXHAWK [11] is the multirotor flight control unit consisting of IMU and input/output module for the multirotor. It houses 168 MHz / 252 MIPS Cortex-M4F processor running NuttX Real-Time Operating System. Over the operating system there exists two main layers: PX4 flight stack, an autopilot software solution and the PX4 middleware, a general robotics middleware which can support any type of autonomous control of the multirotor. The position hold function is an application which is used in the mission with the help of an optical flow sensor. All the sensors and processing unit communicate through I2C and serial data communication (3.3V TTL) protocols respectively.

3.5 Power Management System

The vehicle is powered by Nano-tech Turnigy 6-cell 8000mAh LiPo. The battery is connected to the power module which supplies power to the PIXHAWK [11] (flight controller board) & power distribution board. The power module enables the flight controller board to monitor the current and voltage of the battery during the flight time. All the four ESCs and a DC-DC voltage step down circuit is connected to the power distribution board. As the supply voltage of the battery is 22.2V and the processing unit requires constant supply of only 11.1V, the DC-DC voltage step down circuit performs this operation. The rest of the sensors and devices are powered by the secondary supplies of the flight controller board and processing unit for e.g. USB ports. As per the theoretical calculations with the current configuration of the battery, the vehicle can sustain a flight time of 27minutes.

4. OPERATIONS

4.1 Machine Interface

4.1.1 Robotic Operating System (ROS) [12]

Robotic Operating System (ROS) [12] is a framework for developing robotics software. It helps integrate sensors and their data using high level programming languages. ROS [12] is compatible with around 2000+ software libraries and with various popular robots and sensors. In our integrated system, ROS [12] plays a pivotal role in retrieving the data and publishing values back to respective systems. Every ROS [12] package is considered as a node. ROS [12] nodes use a ROS [12] client library to communicate with other nodes, they can publish or subscribe to a topic (data stream).

MAVLink extendable communication node for ROS [12] with proxy for Ground Control Station (MAVROS) [13] is one such package that helps communicate with PX4 for controlling the multirotor.
Using MAVROS [13] we can subscribe to a particular sensor value (or parameters) located on PX4. MAVROS [13] uses Micro Air Vehicle Communication Protocol (MAVLink) to communicate with PIXHAWK [11], for example, to send commands for arming, changing direction and so on. MAVROS [13] comes with additional packages that can communicate with other sensors which can interface with PIXHAWK [11], like the optical flow sensor.

ROS package for RealSense™ camera (R200) helps us extract separate RGB and depth frames from the cameras. From the frames obtained, further analysis and processing is done using Open Source Computer Vision library (OpenCV).

OpenCV is a library used widely for real-time computer vision. Its enhanced multicore processing helps us to compute faster. Its application in the competition is mainly for object tracking, edge detection and colour detection.

Using vision_opencv [14] and its subpackages namely cv_bridge which is used to brigde between ROS messages and OpenCVand image_geometry which helps to work on the geometry of the image.

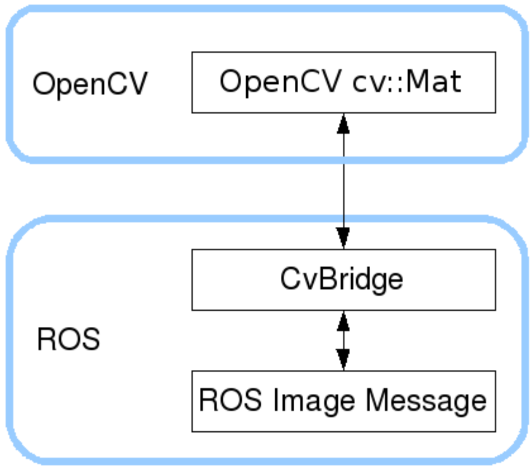

Figure 5. ROS Image conversion

5. RISK MANAGEMENT

5.1 EMI/RCI

As discussed earlier, the multirotor has MEMS gyroscopic & magneto sensors in the IMU, they can be affected by the DC current in the wires around them. As operations of ESCs involve very high current, they generate enough electro-magnetic interference to misguide the sensors. The solution to this is to keep the IMU and ESC's wiring and also placing the ESCs in a plane perpendicular to the IMU plane.
As mentioned earlier, in communication the radio uses 2.4GHz and the telemetry uses 915MHz which will not interfere with each other. Also the 915MHz for telemetry is accepted in the country where the venue is located. The processing unit is connected to the ground unit through WiFi 802.11b/g/n. Hence while setting up the WiFi 802.11b/g/n link care has to be taken as to which channel band the communication is to be configured with so as to avoid interference from the other local WiFi 802.11b/g/n signals. The radio communication interference is taken care of as far as possible.

5.2 Vibration

Rotating motors are the primary source of vibration on a multirotor. When the multirotor is modeled as a 6 DOF (Degree of Freedom) System, the system can resonate at 6 different frequencies. The theoretical values of these frequencies are found by solving the differential equation of motion of the multirotor using MATLAB. It is essential that the frequency of the motor induced vibration should not be equal to any one of these six frequencies. At resonance the multirotor shall vibrate with a theoretically infinite amplitude exhibiting different modes of vibrations. This shall ultimately result in loss of stability of the multirotor. As an effective method to reduce this risk of inflight resonance, the multirotor arms are made of hollow square tubes, since they offer a higher moment of inertia and also higher rigidity when subjected to bending. Use of longer tubes reduces the frequency of vibration. These methods ensure that the frequency of vibrations are really low and below the resonant frequencies.

The PIXHAWK [11], NUC and all sensors are mounted on foam vibration damping pads, ensuring that the vibrations are not transmitted. Vibrations hamper the sensing, producing junk values therefore it is imperative that the vibrations are nullified.

5.3 Propeller Safety

To protect the propeller from external bodies, the periphery of the propeller must be contained in a ducted fan casing, which serves two purposes. Firstly, it secures and protects the propeller. Secondly, it reduces the formation of tip vortices and thereby reducing the vortex drag and increasing the lift.

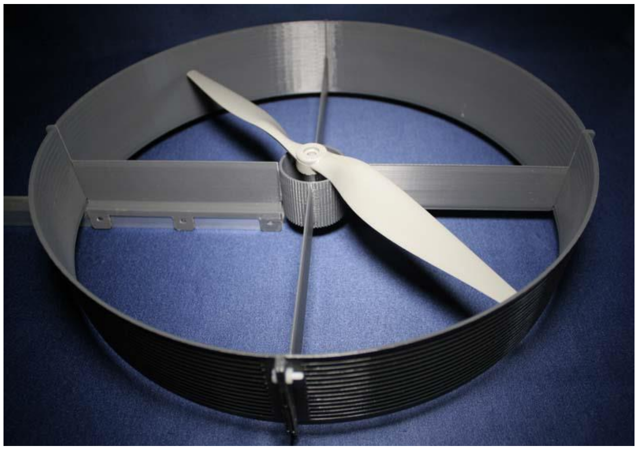

Figure 6. Ducted Propeller

6. REFERENCES

[1] www.aerialroboticscompetition.org/rules.php
[2] en.wikipedia.org/wiki/Optical_flow
[3] www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html
[4] https://pixhawk.org/modules/px4flow
[5] docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_ddetector/canny_detector.html
[6] http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm
[7] http://www.cs.unc.edu/~welch/kalman/
[8] http://liu.diva-portal.org/smash/get/diva2:360818/FULLTEXT01.pdf
[9] http://www.rctigermotor.com/html/2013/Navigator_0910/38.html
[10] http://www.robotshop.com/media/files/pdf/datasheet-rpLIDAR.pdf
[11] https://pixhawk.org
[12] wiki.ros.org
[13] wiki.ros.org/mavros
[14] wiki.ros.org/vision_opencv