Control and Coordination of Customized Drones Using Human-Robot Interaction

Ahana Kumar[1], Himanshu Sharma[1], Nithin Bodanapu[2], Pallavi M P[2], Vaibhav Bajaj[2], Deepali Prabhu[2], Disha Aravind[2], Srikrishna B R[3]

[1]*Department of Mechanical Engineering*
[2]*Department of Computer Science & Engineering*
[3]*Department of Electronics & Communication Engineering*
PES University, Bangalore

ABSTRACT

As we begin to incorporate robots into our lives to assist us in our day to day tasks, it is essential that we must be able to interact with them in a safe and easy manner. In this paper we discuss human computer interaction techniques such as voice recognition that help to interact with a robot, as well as techniques on obstacle avoidance and human detection that allow for safer interactions. These techniques shall be used to control and coordinate a swarm consisting of 3 drones. A customised drone is built for a safe and stable flight.

1. INTRODUCTION

1.1. Problem Statement

Mission 8 of the International Aerial Robotics Competition requires the completion of a task by a human,with the aid of helper drones. The task is to retrieve an item from a sealed box, while being subjected to attack by opposing aerial drones. The helper drones will provide healing and code decoding capabilities. The mission focuses on human-robot interaction, through non-electronic commands, and the use of Swarm Intelligence concepts to handle the coordination of multiple drones.

1.2. Approach

The proposed approach to the task makes use of certain primitive behaviours, that each drone is capable of performing, localization techniques and a communication architecture, that allows the user to control and coordinate the actions of all the drones in the swarm.

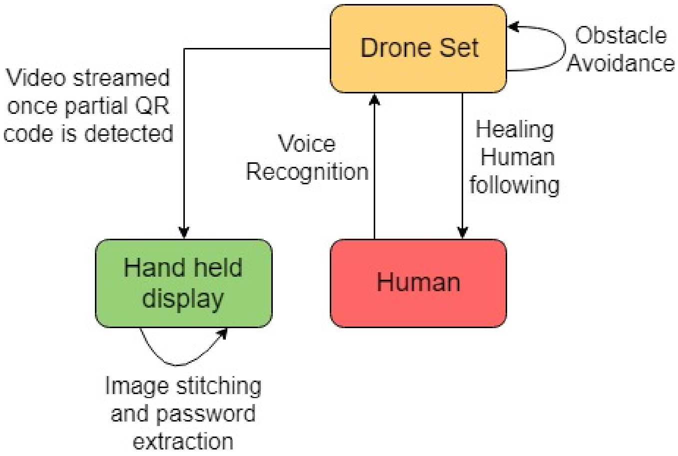


Fig. 1: System Architecture

2. LOCALIZATION

Since the task places the restriction that the drone must fly in a GPS denied environment, Localization is made possible by fusing the visual odometric[1] values from the ZED Stereo camera[2] to the local coordinate values of the pixhawk, using the native ECL-implemented Extended Kalman Filter(EKF)[3] state estimator. This gives us an accurate estimate of a drones position in a frame of reference where the point of take-off is the origin. Hence, each drone has a different frame of reference.

3. COMMUNICATION

3.1. Intra-Drone

Communication between the drones takes place over the ROS[4] network using a master-slave architecture. All the drones in the system are connected to the same WIFI network. An Intel NUC[5] is placed on the master drone, which acts as the master processor on which the roscore is run. Each drone is equipped with a Jetson TX2[6] processor board, which behave as the slave processors and are configured to know the master processors IP. In this manner the master drone can relay the voice commands given by the user to the appropriate slave drones, which can then react accordingly.

3.2. Human to Drone

3.2.1. Voice Recognition: Audio is recorded in .wav format and processed to get the corresponding word. There are 16 words that can be used including go-up, go-down, stop, default and so on. Four template recordings for each word are pre recorded and saved in a folder for future matching. The wav recorded audio file is noise isolated and clipped appropriately to contain only the speech and no silence periods. Features for the audio file to represent the phenomes accurately using Mels frequency cepstral coefficients[7]. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent

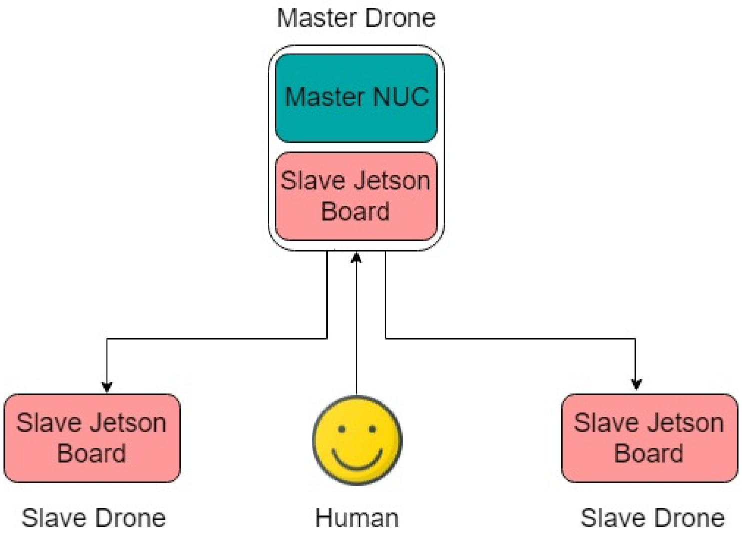


Fig. 2: Master-Slave Communication Architecture

this envelope. We use pythons speech feature module to extract the MFCC vector for the audio file. The matching is done using dynamic time warping. This matches the series regardless of accelerations and decelerations between the two operand series. The fast-dtw module of python is used instead of the original DTW algorithm for the same. Some accuracy loss is incurred due to this decision.

4. NAVIGATION

4.1. Navigation Algorithm

The navigation of the drones, except for safety behaviours such as obstacle avoidance and continuous tracking using Human Detection, is mainly controlled by the users commands. Hence, the drones loiter at their respective positions until they are given a command by the user, based on which the drones take appropriate action. These navigation actions are based on a finite state automaton. The algorithm representing the FSA is as follows:

Algorithm 1 Navigation

```
while ROS OK do
    if $New\_Command$ then
        $Target\_pose = Action(New\_Command)$
    else
        if $Human\_Detection == True$ then
            $Target\_pose = Human\_Tracking_{p}ose()$ # Loiter fixed distance behind the Human
        else
            $Target\_pose = Current\_pose$
    if $Obstacle(Target\_pose) == True$ then
        $Target\_pose = Obstacle\_Avoidance\_Pose()$ # Move to safe position
    $Publish(Target\_pose)$
[1]
```

4.2. Navigation Architecture

The navigation architecture[8] consists of several nodes that communicate over the ROS network using ros topics or ros services. These nodes comprise of programs that are device drivers or programs that process the sensor input from these devices. This can be seen in Figure 3.

The LIDAR[9] sensor's data is consumed by the Lidar_Node, which is a software module in the form of a ROS Node, which provides a wrapper that abstracts the low-level details of serial communication that is performed to obtain data from the LIDAR sensor, and in turn provides the readings of the LIDAR sensor in a programmer friendly manner over ROS strongly typed messaging queues.

The OA_Node is the module responsible for obstacle detection. It uses the Lidar_Node to obtain readings from the LIDAR sensor and performs analysis and computation over multiple instances or snapshots of LIDAR readings to detect obstacles and in turn provides details of obstacle presence. The node also performs data filtering. It restricts the distance to accurate range of the lidar, this parameter is configurable by the user.

The Mavros[10] and Mavros_Extras are ROS Nodes that abstract details of communication over serial or WiFi link with the MAVLink[11] protocol for communication with the FCU.

The FCU receives messages over MAVLink that contain information and commands that it must follow. The FCU runs the PX4 firmware[12] that runs on top of NuttX[13].

The Auto_Nav_Bridge node is the module responsible for low level communication with the FCU via MAVLink. It also takes care of publishing rates requirements from the MAVLink protocol and the PX4 Firmware. It also plays crucial role of continued publishing the current point on failure of planner node.

The Auto_Nav_Planner node is the module responsible for taking high level decisions to enable the aerial robot to navigate to the provided destination coordinates without colliding or touching the obstacles. It interacts with the OA and Bridge nodes to accomplish this. This module is the

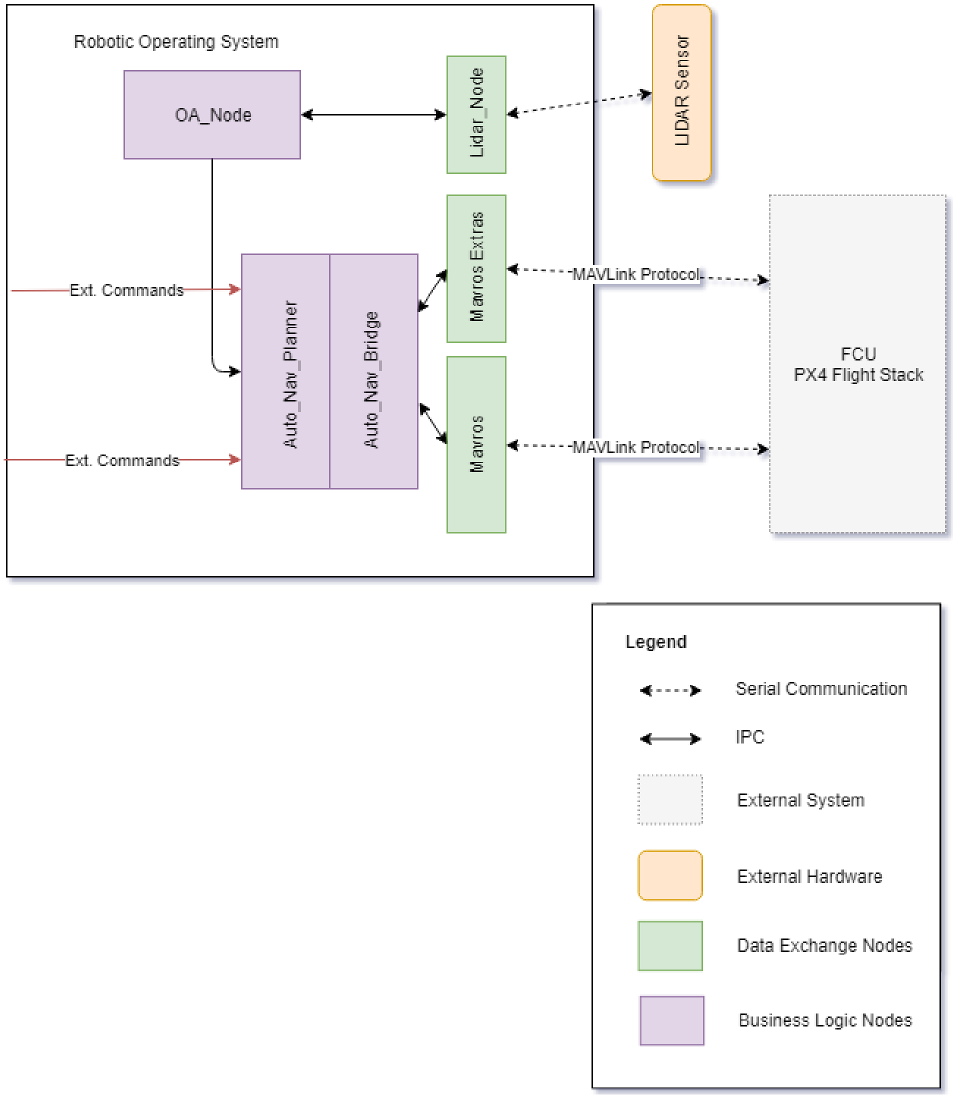


Fig. 3: Navigation System Architecture

entry point for clients/users to use the system. It acts like a facade to the system, hiding its internal complexities and providing a simple interface for the client to use the system.

5. VISION

5.1. Human Detection

Human Detection is a process that we implement to make sure that the master drone always follows the player. This is done to ensure that the user always has a drone in its proximity that can shoot a healing beam. This is achieved by detecting symbols on the users t-shirt, and centering it in the image.

These symbols are used to create a wireframe and this wireframe can give us a good idea of location and movement of human[14]. This tracking will help us predict the movement of the human and hence move the drone in advance and make the overall movement smooth. This will ensure that the number of sudden corrections will be lesser and hence will lead to a much more stable system. This will help us modify the position of the drone. Centring the human with respect to the camera image will help us aim the laser at the human much easier and hence quicker to heal.

5.2. QR code detection

We use the OpenCV library to develop this module using python as the language. We are dividing our QR code detection into 3 main parts :

- Partial QR Code detection
- Image stitching
- Decode QR code

Partial QR Code Detection will involve detecting contours of a square and extracting it in a lossless manner[15]. This involves completion of the square as QR code will not form a complete square. A image completion method like dilation will be used to extract the partial QR code and an erosion is performed to contradict the previous dilation process. This will ensure that the embeddings of the QR code are retained.

Image Stitching is the process of putting together all the different QR code segments such that they connect well together and a proper QR code is formed. Image stitching can be done using a matching algorithms which compares the continuation of an image from every side and chooses the highest value as the common side and stitches them together. This is done in a repeated manner to stitch 4 images together and form the complete QR code

Decode QR code is the process of Decoding the QR code which also involves dealing with noise and appropriate cropping for accurate decoding extracting the passcode out of the QR code that will be used for unlocking the box. We use the Zbar library in python to decode this QR code and extract the passcode out of it.

5.3. Obstacle Avoidance

A basic approach is applied using a SLAMTEC, RPLidar A2[9], which is a 360°2D laser scanner. The drones will operate at low altitudes to avoid collisions with sentry drones. The laser scan from the LIDAR gives the distance and angle of any obstacles with respect to the drone.The trajectories of sentry drones can be calculated using these values[16]. In case of imminent collisions the drone will change its altitude until there are no obstacles within a given safety range.

6. **AERIAL VEHICLE**

We intend to design a drone that is different from the conventional ones, use a new material for manufacturing, analyze its workability on software and finally manufacture it via fused deposition modelling, completely made of Carbon+PETG material, and then check its reliability by flying it.

Carbon fibre+PETG material is deep black with ease of printing whilst offering high working temperature and impact resistance. They give better rigidity, strength, reduced warping, detailed products, great texture and look, extremely good chemical resistance, almost isotropic shrinkage, 3 times lower moisture absorption than ABS and processing range between 230-270°C.

For designing, Solidworks software was used. The design was then analysed on NX 11. Necessary changes were made and then the components of the drone were 3D printed using Carbon fibre+PETG material.

6.1. Design

6.1.1. Base Plate:

There were a couple of visible changes that have been made relative to the normal base plate of a drone:

- **Single plate design:** This is done to decrease the weight and have no height issues for components. Due to the single plate design, the arms must be designed in a flat plate manner instead of a circular shape. Thickness of the plate is kept the same as the 650 mm frame.
- **Shape of the plate:** Keeping in consideration the aerodynamic effect of fluid onto the surface, we have rounded the edges and corners.

The size of the base frame is $18 * 18$ cm which is made in order to have enough space to house components such as Jetson tx2 etc which is $16 * 15.5$ cm approximately.

6.1.2. Arm:

Avoiding the use of a conventional arm, keeping in mind the Centre of Gravity and stability considerations:

- It is a flat plate structure because our frame is single plated.
- Elevation for more stability during flight.
- The protrusion is for fitting the propeller guards for safety purpose.
- The length is as per the propeller size which is 15 inches(=38.1cm)

6.1.3. Landing Gear:

The Landing Gear was designed keeping in mind the sturdiness of the drone on landing and the shock relief, in case of impact landing.

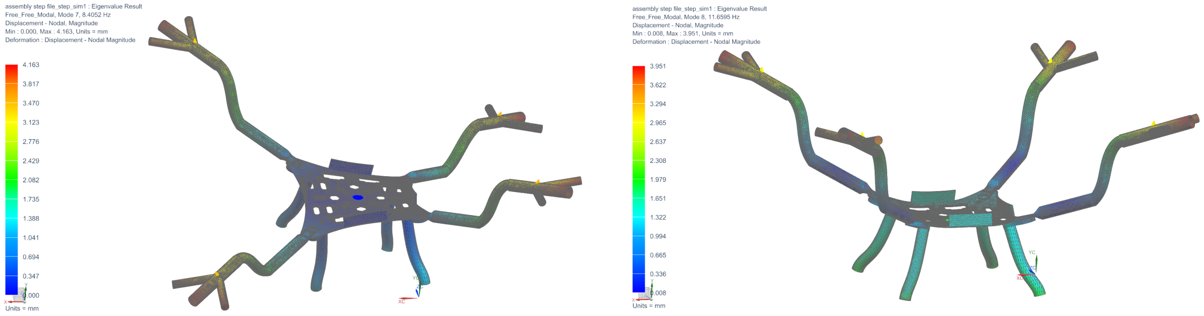
6.1.4. Propeller Guards:

Safety is a serious issue when it comes to drones and especially for this mission in which a human will be on field with them. All sided protection is the aim which is fulfilled by covering from the top and bottom, along with the whole circumference.

6.2. Analysis

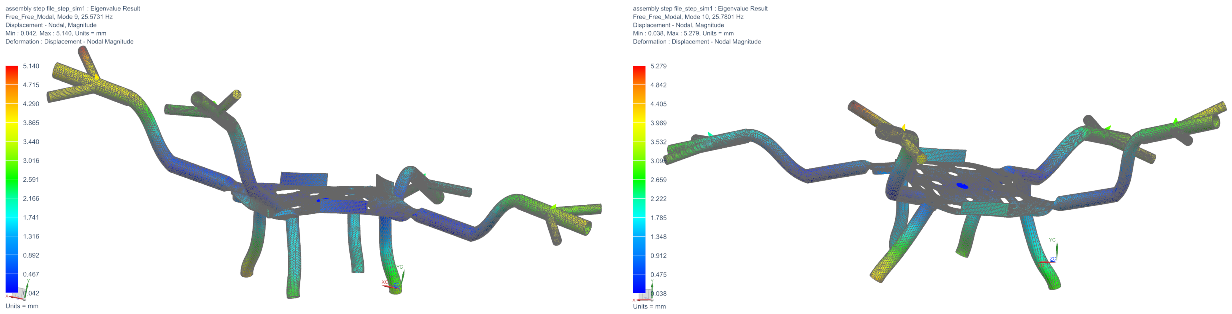
To ensure structural rigidity and strength to weight ratio, we have considered Analysis which includes Modal and Static Analysis[17].

6.2.1. *Modal Analysis:*



(a) 1st mode- 8.4052 Hz (Twisting about Y axis)

(b) 2nd mode-11.6595 Hz (Bending about X axis)



(c) 3rd mode-25.5731 Hz(Twisting about Y axis and bending about Z axis)

(d) 4th mode-25.7801 Hz(Twisting about X axis and bending about Y axis)

Fig. 4: Modal Analysis

6.2.2. *Static Analysis:*

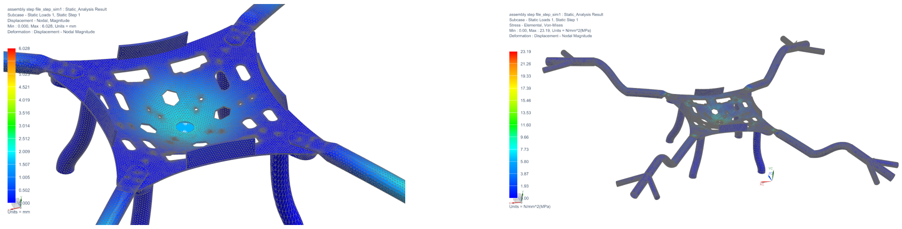
For static analysis of the assembly, vertical loads have been considered; at the centre, weight is 3.5kgs and 200g on each of the arms. The landing gears have been arrested in all degrees of freedom.

Property	**Analysis Value**	**Limit**
Displacement	5 mm	-
Stress	25.81 MPa	56 MPa
Reaction	7.4 N	-

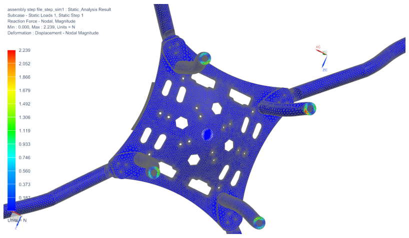
TABLE I: Static Analysis

The displacement will be further reduced when damping is done. The reserved factor, is the term for factor of safety in Aeronautical Industry, is found to be 2.17. No vital structure damaging regions were seen. Displacement, Stress and the Reaction forces that occurred on applying the boundary conditions, all were well within the safe limit.

6.2.3. *Landing Gear Analysis:*



(a) Displacement (b) Stress



(c) Reaction Forces

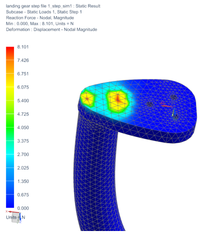
Fig. 5: Static Analysis of frame

The top part which is connected to the base plate, is arrested in all directions. Taking safety and multiplying it with 4g instead of g(acceleration due to gravity), the force on each arm for our study is fixed to 45N.

Property	**Analysis Value**	**Limit**
Displacement	0.312 mm	-
Stress	3 MPa	56 MPa
Reaction	8 N	-

TABLE II: Landing Gear Analysis

(a) Displacement (b) Stress

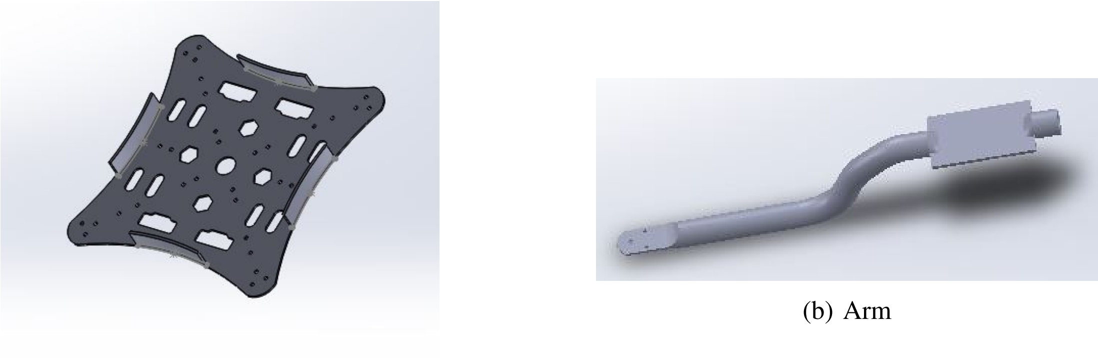


(c) Reaction Forces

Fig. 6: Forces on Landing Gear

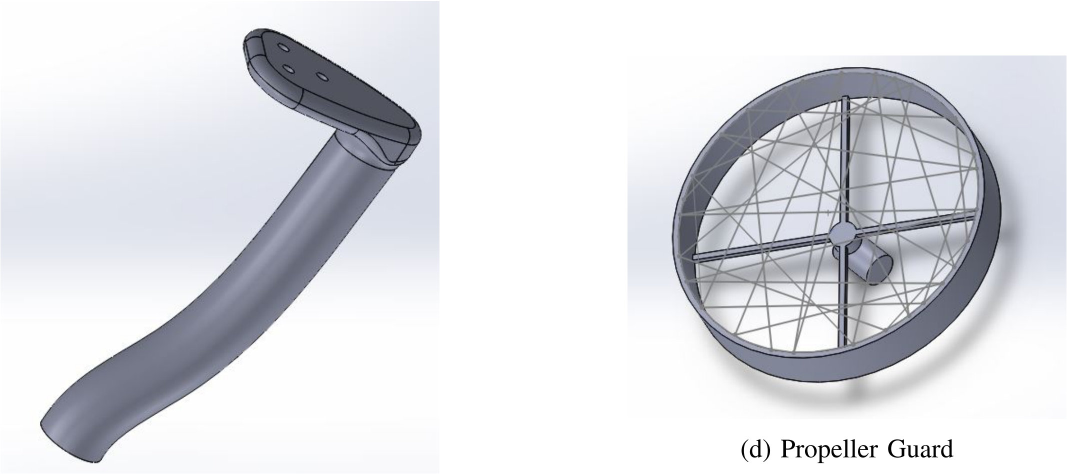
The sharp edge at the top part of circular section has maximum stress. Rounding of the edges

would decrease the stress concentration in one area.
After analysis, we finalised the design as shown in Fig. 7



(a) Base Frame

(b) Arm



(c) Landing Gear

(d) Propeller Guard

Fig. 7: Final Design

6.3. Manufacturing

For the 3D printing of the drone components, we have used Fused Deposition Technique[18]. The parameters set for the process were based on the suggested values of the manufacturer of the material Carbon fibre+PETG which are mentioned in Table III

Property	**Value**
Layer Thickness	200 microns
Infill	100%
Nozzle Diameter	0.4mm
Shell Thickness	1200 microns
Extruder Temperature	235°C
Bed Temperature	90°C

TABLE III: 3D printing parameters

REFERENCES

[1] I. Afanasyev and I. Ibragimov, "Comparison of ros-based visual slam methods in homogeneous indoor environment," https://www.researchgate.net/publication/320623436_Comparison_of_ROS-based_Visual_SLAM_methods_in_homogeneous_indoor_environment.

[2] Stereo Labs, "About zed stereo camera," https://www.stereolabs.com/], 2017.

[3] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," pp. 3068 – 3068 – 12, 1997. [Online]. Available: https://doi.org/10.1117/12.280797

[4] Open Source Robotics Foundation, "About ros," http://wiki.ros.org/, 2017.

[5] Intel, *Intel NUC Board D53427RKE Technical Product Specification*, https://www.intel.com/content/dam/support/us/en/documents/boardsandkits/D53427RKE_TechProdSpec05.pdf, 2014.

[6] Nvidia, *Nvidia Jetson TX2 System-on-Module*, https://www.docdroid.net/yGXIxZu/data-sheet-nvidia-jetson-tx2-system-on-module.pdf, 2014.

[7] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *arXiv preprint arXiv:1003.4083*, 2010.

[8] V. B. S. A. A. S. K. N. B. P. M. P. U. S. H. S. N. P. Syed Hashmath Ulla T A, Shivkumar Umadi, "An autonomous navigation and object tracking system for aerial multirotor robots with obstacle avoidance in gps denied environments, pes university," http://www.aerialroboticscompetition.org/assets/downloads/2018SymposiumPapers/PESUniversity.pdf, 2018.

[9] SLAMTEC, *RPLIDAR A2 Specifications*, http://bucket.download.slamtec.com/004eb70efdaba0d30a559d7efc60b4bc6bc257fc/LD204_SLAMTEC_rplidar_datasheet_A2M4_v1.0_en.pdf, 2016.

[10] O. S. R. Foundation, "About mavros," http://wiki.ros.org/mavros, 2017.

[11] L. Meier, *MAVLink Micro Air Vehicle Communication Protocol*, http://qgroundcontrol.org/mavlink/start.

[12] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6235–6240.

[13] Contributors of www.nuttx.org, "Nuttx real-time operating system — nuttx real-time operating system," 2018, [Online; accessed 31-May-2018]. [Online]. Available: \url{http://www.nuttx.org/doku.php?id=nuttx&rev=1525032444}

[14] A. Rivas, P. Chamoso, A. González-Briones, and J. Corchado, "Detection of cattle using drones and convolutional neural networks," *Sensors*, vol. 18, no. 7, p. 2048, 2018.

[15] Y. Kato, D. Deguchi, T. Takahashi, I. Ide, and H. Murase, "Low resolution qr-code recognition by applying super-resolution using the property of qr-codes," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 992–996.

[16] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011.

[17] A. Javir, K. Pawar, S. Dhudum, N. Patale, and S. Patil, "Design, analysis and fabrication of quadcopter," *Journal of The International Association of Advanced Technology and Science*, vol. 16, 2015.

[18] O. Dandgaval and P. Bichkar, "Rapid prototyping technology-study of fused deposition modeling technique," *International Journal of Mechanical and Production Engineering*, vol. 4, no. 4, pp. 44–47, 2016.