# A Novel Approach to Autonomous Human Safe UAVs for Man-Machine Interaction and Cooperation

Vilde B. Gjærum Peder Mathias W. Teigmo Vegar A. Bergum
Henrik Fauskanger

May 2019

*Ascend NTNU*
*Norwegian University of Science and Technology*
*Trondheim, Norway*

**ABSTRACT**

A novel system for autonomous UAVs capable of verbal interaction with a human pilot while being human safe, in a GPS-denied environment, is presented. The system is designed to solve the 8th mission of the IARC, while also serving as the first iteration of a longer development process. INS sensor measurements as well as pose estimation from a SLAM implementation are passed through an extended Kalman filter to ensure stable control at various speeds. A deep learning approach to human pose estimation as well as voice command recognition is presented and used for man-machine interaction, where a global behaviour module is able to allocate resources and estimate optimal solutions. The physical air vehicles are designed for human safety, where several modeling and simulation techniques such as modal analysis and CFD analysis are employed. Control systems, such as obstacle avoidance, and drone behaviour is also verified by both physical testing and simulations. As a whole, these modules form a full system capable of competing in the 8th IARC.

## INTRODUCTION

This paper describes the student organisation Ascend NTNU's contribution to the eighth mission of the International Aerial Robotics Competition (IARC).

### Problem Statement

In the eighth mission of IARC, the goal is to develop four fully autonomous helper drones, whose objective is to assist a player in an arena retrieve objects in locked bins. Four hostile sentry drones attempt to neutralise the player using lasers. The human player has a total of 10 lives. Additionally, the four friendly drones can each heal the human player once. The human player can only interact with the drones in a non-electronic manner such as voice commands and body gestures. Since there will be both drones and a human in the arena simultaneously, the drones must be safe for humans. The helper drones must avoid colliding with obstacles, as well as sentry drones, at all costs. Furthermore, the drone cannot rely on external positioning systems, such as GPS or visual tracking.

### Conceptual Solution

The solution includes four quad-copters using the NVIDIA Jetson TX2 as their processing platform, capable of performing real time mapping and localisation using computer vision, threat identification and communication with a global ground station. The vehicle's control systems are based on a Pixhawk flight controller, ensuring stable and safe flight. Global behaviour and man-machine interaction is handled by a computationally capable ground station that estimates, and delegates, optimal solutions to tasks given by the human player. A detailed model over the system architecture can be seen in Figure 2.

### Yearly Milestones

This is the first year mission 8 of the IARC competition is held. The goal of this year has been two-folded. The first part aiming on creating a solid basis for next year, the second part aiming on starting to solve parts of the mission. It has also been very important for Ascend NTNU as an organization to build upon what it has learned from the previous mission.

- Show man-machine interaction
- Autonomous flight and mapping of room
- Autonomous takeoff and landing
- Obstacle avoidance

## AIR VEHICLE


*Figure 1. One of the air vehicles*

Ascend created four identical vehicles for IARC 2019. They have a quadrotor layout designed to carry one forward facing RGB-D (Stereolabs ZED) camera for SLAM, one downwards facing camera for QR detection, one 1D lidar for ground distance measurement, one Pixhawk 4 flight controller and one Nvidia TX2 for computation. All propellers are surrounded by ducts with net covering the top and bottom to comply with human safety requirements of the competition. The overall size of the vehicle is 70cm on its diagonal with a total weight of 2100g. The overall system architecture of each drone, as well as the common ground station system, is presented in Figure 2, with the drone itself depicted in Figure 1.

### Propulsion

Thrust is provided by Racerstar BR2312 960KV Brushless Motors along with DJI9450 propellers. Correct gap clearance between duct and propeller is important to gain optimal efficiency [1, p. 38]. Because of this, the propellers are cut down to 9" to fit into the ducted fan frame design. Upon installation the propeller gap is adjusted by sanding the ends, as this

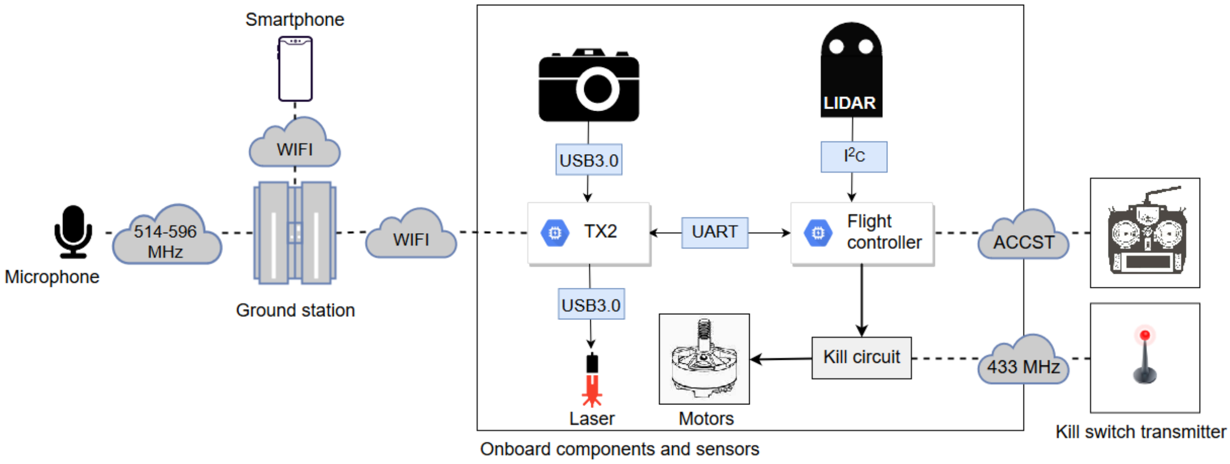


*Figure 2. Overall system architecture.*

method proved the easiest way to achieve the desired tolerances. Lastly the propellers are rebalanced to avoid eccentricity of the rotating mass, thereby reducing vibrations. Power is delivered to the motors through 30A electronic speed controllers with pulse width modulation signals routed through the kill switch.

### Power Management

The motors have an efficiency of 6.0 g/W (1 Watts per 6 grams). This means that the 2100g drone requires 350 W to remain airborne. Our 4 cell LiPo battery of 5000mAh has 74Wh of energy. If only motors are considered we will then get 12 and a half minutes of flight. Because over-discharging the batteries is undesirable, a 20% safety factor will be applied. This gives approximately 10 minutes of flight time, enough time to set up the drones and complete the mission (max 8 minutes) without having to worry about over-discharging the batteries. The remaining electrical components have negligible energy consumption in comparison to the propulsion system (around 10 W) and are considered into the safety factor.

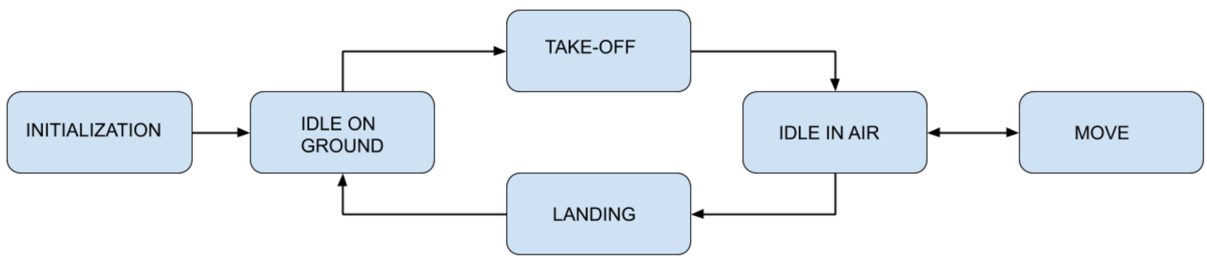


*Figure 3. Diagram over the FSM (Finite State Machine).*

### Flight Control System

The system is running Linux with mainly three different modules communicating with each other using the Robot Operating System (ROS) framework. ROS provides conventions, libraries and easy communication between modules across multiple devices. A model of the overarching architecture is illustrated in Figure 4, showing the three modules: Control, Perception and AI.

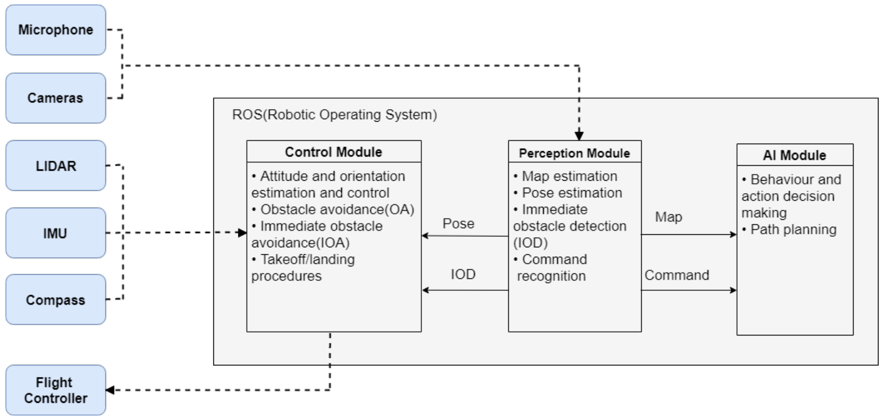


*Figure 4. ROS system architecture.*

*Navigation/State Estimation System*

The navigation needs are motivated by the AI module that computes the most efficient series of waypoints and actions to solve the mission. The drone must be able to both follow waypoint paths and simultaneously avoid known and occurring obstacles.

Figure 3 illustrates the finite state machine (FSM) of the control system. It includes all the states the drones can be in and how they can transition inbetween them. Obstacle avoidance does not belong to any specific state, as it is authorized to override the FSM, regardless of state, if needed.

The obstacle avoidance system is based on Panagou 2014 [6], which uses vector fields. The method exploits that vector fields are easily manipulated to generate a flow the drone can follow. Two vector fields are used. One global field based on where the AI module wants the drone to fly, and one based on obstacles in the map. The global field is an attractive field, and the obstacle map is a repelling field. The vector field is represented by Equation 1.

$$F(\mathbf{r}) = \lambda(\mathbf{p}^T\mathbf{r})\mathbf{r} - \mathbf{p}(\mathbf{r}^T\mathbf{r}) \tag{1}$$

The position vector relative to the global frame is represented by $\mathbf{r} = [x, y]^T$, while $\mathbf{p} = [x, y]^T$ defines the direction of the field. The parameter $\lambda$ determines the property of the vector fields. $\lambda = 1$ for a repelling field, and $\lambda = 2$ for an attractive field. To obtain a vector field the drones can navigate after, the global field, $F_g$, and the obstacle avoidance fields, $F_{oa}$ for each obstacle must be combined. This is done as shown in Equation 2.

$$F = F_g + \sum F_{oa} \tag{2}$$

*Attitude/Position Control System*

To estimate the spacial state of the system, the position and orientation of the drones, the measurements from all the sensors are fused. This is done by an Extended Kalman Filter (EKF) running on the flight controller. The EKF receives measurements from an Inertial Measurement Unit (IMU), a downward facing 1-D LIDAR, as well as visual odometry data from the Perception module, in order to estimate the translation, orientation and velocity of a given drone. In practice, the EKF functions as a 3-degrees-of-freedom estimator of orientation, position and velocity in order to handle noisy sensor data and discrepancies between different sensor systems.

### Flight Termination System

The flight control system uses a custom designed kill switch to cut the power to the motors. The PWM signal from the flight controller to the electronic speed controller (ESC) is routed through the kill switch. The signal goes through an and-gate which is enabled by a 433 MHz receiver. When the kill switch is enabled the and-gate of the kill switch closes the circuit, allowing the PWM signal to pass through to the ESC. With the circuit in a closed state the ESC is able to spin the motor. The default state of the kill switch circuit is open, thus the receiver must continuously receive a 433 MHz signal from its transmitter for the motors to receive power. The transmitter is in the hands of the organizer, in order to comply with the rules of mission 8.

As an additional safety system, a human safety pilot can override the autonomous behaviour of the drone by transmitting the correct signal from the pilot's controller. This allows the pilot to take control of the drone and safely land it. Doing this during the competition will terminate the attempt, but will be performed should the drones threaten to hit the human player or behave erratically. This is the preferred option, as triggering the kill switch will not stop any forward momentum the drone might have.

## MISSION PACKAGE

The mission package is a series of components and modules specifically designed to allow the air vehicles to execute the IARC missions, and operate safely in similar environments. Some of the components are specifically aimed at IARC mission 8, while others are general systems allowing for autonomous operation. The components and modules are generally separated into two categories: local and global, where global systems are executed in a ground station (see Figure 2), while others are implemented locally on each air vehicle.

### Perception System

The AI module relies on the knowledge of the drone's 3D location in the arena to compute the best plan, and as such necessitates a method for estimating the position and a map of the world. One such method is simultaneous localisation and mapping (SLAM), where the ORB-SLAM 2 [5] implementation has been chosen for its compatibility with RGBD-images, loop-closure and its superior feature-based method. Feature-based methods are generally faster than appearance-based methods [7, Chapter 11]. It performs loop closure which ensures more accurate and reliable maps which is essential when navigating without GPS.

*Target Identification and Behaviour*

For the human player to be able to retrieve the items, the drones must find the code that unlocks the bins. On the top of each bin, a quarter of a QR code is displayed. The drones must detect these quarters, find out how they should be connected together and then scan the QR-code to retrieve the encoded data. QR-square candidates are identified by performing segmentation on the image, extracting areas that are likely to be relevant for QR-code detection. Doing this enables detection of QR-codes at various distances and orientations. Distinct corners are detected by a *Harris Corner Detector*, and the candidate is rotated and cropped to a minimal bounding box before further evaluating whether it is a QR quarter segment or not. To be passed as a QR quarter segment the candidate must be approximately squared, big enough, and more than half of the lines detected by the *Probabilistic Hough Transform* must be orthogonal to other lines. The QR quarter segments are naively matched together until the resulting image can be scanned to retrieve a valid result.

One of the weaknesses of this approach is that the method is sensitive to view-point. To address this, the AI module provides a special behaviour pattern to search for the QR-code. The human player can designate a location to a drone by recursively selecting a quadrant from a quadtree-representation [3] of the arena floor, which is done through a *postfix recursive* language built from simple voice commands (See more about this language in the User Interface / Man-Machine Interface section). Once the designated drone has navigated to the assigned location, it will fly in a set outward spiral to try and locate the QR-code, scanning the ground with a downward facing camera. Once a QR-code segment has been detected, the drone will be able to position itself to get an accurate scan.

*Threat Identification and Behaviour*

Any physical obstacle that the drone can crash into and subsequently harming itself or disqualifying itself from the competition, is considered a threat. There are two systems that deal with the behaviour of various types of threats, both receiving data from one of two modules built on the same threat identification algorithm.

On a macro scale, the AI module uses a navigation system built on an evidence grid [4] representation of the world. This can be a 2- or 3D-representation, depending on the needed complexity. In the case of threat identification, the arena is abstracted into a 2D field, where any threat is assumed to have an infinite height. This is an important feature that accommodates the relatively limited field of view of each drone. The evidence grid is used as a high level strategy map for the AI module. It is continuously updated based on the outputted features and point cloud data from ORB-SLAM 2, which has been processed by an obstacle detection algorithm capable of detecting clustered 3D points and output a bounding box and a collision vector for each threat, or obstacle, detected.

The evidence grid is used by a high level navigation system. The navigation system is responsible for setting intermediate waypoints that constructs the most efficient and safest path from a drone's current location, to some target location. With perfect data and exclusively static obstacles, this navigation system should be able to avoid any possibility of collision. As a drone is en-route to a waypoint, the navigation system will recalculate the most efficient and safest path, always making sure that the newest data from the ORB-SLAM 2 system is used. The navigation system and the evidence grid are running on the ground station which means they are susceptible to delays. The second, micro level threat identification system running locally on the drones is designed to tackle this issue.

To deal with the uncertainty of data from ORB-SLAM 2 and the delay that can be experienced from the evidence grid data and the navigation system, each drone has a separate obstacle avoidance system (See the Flight Control System section). This obstacle avoidance system receives bounding boxes from a similar obstacle detection algorithm as the navigation system. This version of the obstacle detection algorithm is designed to run on each drone and have a computational delay that is shorter than a single ORB-SLAM 2 frame. Being able to process incoming data from ORB-SLAM 2 faster than ORB-SLAM 2 can produce more data, allows the flight control system's obstacle avoidance to always have the newest available data. The obstacle avoidance system is authorised to interrupt the path to a waypoint in order to avoid a collision that the high level navigation system was not able to detect.

*Gesture Identification and Behaviour*
The human detector and pose estimator is based on Cao et al. 2016 [2]. The paper proposes a bottom-up approach for realtime multi-person pose estimation by introducing the use of a parametric representation – referred to as Part Affinity Fields (PAFs). In essence, a PAF is a set of vectors that encodes the direction from one part of the limb to the other – each limb is considered as an affinity field between the relevant body parts.

The method uses an branched and iterative prediction architecture, following Wei et al 2016 [8], to predict the confidence maps of the body joints and the PAFs. The architecture refines the predictions of the confidence maps for the body joints and PAFs over successive stages, with intermediate supervision at each stage. At the end of each stage, the corresponding loss function is applied for each branch to guide the network.

To get body part candidate regions, the confidence maps for the humans in the image are aggregated. The algorithm performs non-maximum suppression on the confidence maps to obtain a discrete set of parts locations. During inference the algorithm computes line integrals over all the PAFs along the line segments between the pairs of the detected body-parts. If the candidate limb formed by a connection of certain pair of points is aligned with corresponding PAF then it's considered as a valid limb.

Finally, with the body part candidates and scored pairs of these parts, retrieved by integrating over the respective PAFs, one has to match these parts. This problem can be viewed as a k-partite matching problem, where nodes of the graph are body part detections, and edges are all possible connections between them, i.e. possible limbs. Cao et al., 2017 propose a relaxation where the initial k-partite graph is decomposed into a set of bipartite graphs where the matching task is more feasible to solve. The decomposition is based on the problem domain – e.g. a hip cannot be connected to a foot directly, thus one can, for instance, first connect hip to knee and then knee to foot.

The custom solution can be seen in action in the provided web-video[1].

### Communication Systems

Communication between the drones and the ground station, as well as the pilot, is done wireless over WiFi 802.11ac. The ground station accepts signals over a 5.0Ghz band, while also using a 2.4Ghz band as a fallback. The pilot's microphone transmits audio signals to a dedicated audio system, operating at 514-596Mhz.

### User Interface / Man-Machine Interface

In order for the human player to retrieve information from the drones, a web video stream to transfer live video from the drones and ROS is utilised. This is necessary both for the operator to receive critical information, as well as a formal requirement for the competition. A web video server package opens a local port and waits for incoming HTTP requests. As soon as it receives a request, it subscribes to the corresponding topic, creates an instance of the encoder and transfers the encoded raw video packets to the client.

[1] `https://www.youtube.com/watch?v=d3TfM4yAc7o`

The human player will be carrying a smartphone encapsulated by a Plexiglas cover to ensure one-way communication. This is the only way the friendly drones can communicate with the human player, and it is by this the code for the locked bins will be sent.

*Voice Commands*

Communication from the human player to the drones is done through a predefined language based on a set of voice commands. A deep learning model has been trained to recognise the 10 first letters of the NATO Phonetic Alphabet. This model is running on the ground station, which is capable of receiving an audio signal from the human player. Using these 10 letters, the human player can use a predefined language to string together letters into a command that can initialise some behaviour from the drones. As an example, the arena floor can be split into four quadrants, each assigned to a letter. Once a quadrant has been selected, the human player can continue to specify a new quadrant inside the currently selected quadrant using the same set of four letters, creating a postfix recursive command. This command can then feed into the navigation system that would make the selected drone navigate to the assigned location. With such a language, the human player is able to communicate complex and powerful commands to the drones by simply talking into a microphone. The selected microphone completely covers the mouth of the player, significantly reducing noise from the arena, making the module 95% accurate.

### High level behaviour

The AI behaviour module, GHC (Global Heuristic Control), is responsible for task delegation and execution. GHC is a multi-processed framework that is designed to interpret postfix recursive voice commands, allocate resources to a mission or command (a drone and its flight time is an example of such a resource), and enable the correct systems for executing a mission or command. While resource allocation is based on heuristic functions and drone status, each drone has independent processes that are capable of executing missions - such as searching for a QR-code at an assigned location or follow a human - in parallel.

Controlling four drones through a restricted vocal language while attempting to execute a mission can be a demanding task for a human. GHC is meant to work in tandem with the human player, offsetting parts of the decision making process to a global system that is capable of estimating optimal approaches to smaller missions.

## RISK REDUCTION

With a human player present in the arena, the main focus of risk reduction is the safety of the player. Avoiding damage to critical and expensive components of the drone is a secondary concern but also needs considerable attention in order to perform satisfactorily.

### Vehicle Design

To ensure safe flight, electromagnetic interference and risk of unwanted oscillations have to be taken into account. Measures for EMI/RFI shielding, and shock and vibration reduction are deployed to avoid this.

*EMI/RFI Solution*

The ESC's expose nearby circuits to some amounts of electromagnetic interference. To mitigate this, sensitive electronics (especially the magnetometer) and the ESC are placed as far away from each other as possible within the frame of the drone. Room for electromagnetic shielding is also available. To ensure reliable connectivity between the kill switch and the multi-rotor, the receiver is placed outside the carbon fibre frame, as carbon fibre can weaken or block radio signals.

*Shock/Vibration Reduction*

To reduce shock damage to important components of the drone, the landing gear has been designed to deform in order to absorb the energy of a crash. The ducts are connected to the rest of the frame with nylon bolts. In the event of a hard crash, the duct will break off without damaging anything other than the bolts connecting the elements of the frame. The human player will still be safe, as the energy of a crash will go into breaking inexpensive, easily replaceable components, leaving the ducts intact and still shielding the human from the propellers.

Motors and propellers were tested to run at just under 9000RPM (150Hz) at full throttle. During testing the throttle is at about 50% of max, with an idle speed of about 1000RPM at zero throttle. During flight the speed should therefore always stay way below 9000RPM. If no eigenmodes below 150Hz exist, the build should be free of oscillations. To prevent oscillations, modal analysis (Finite Element Analysis) of the frame was performed. See Modal Analysis under Modeling and Simulation for in depth info on this.

**Safety**

To ensure safety for humans, as well as avoid expensive crashes, extensive component and system testing has been performed in simulated environments. Additionally, precautions such as nets, manual override and external motion tracking have been made during physical testing. A pre-flight checklist is also implemented to ensure all systems are A-OK before flight.

**Modeling and Simulation**

*Gazebo simulator*

The mission simulator is implemented as a ROS package, and Gazebo is used for simulations of physics and sensors. The helper drones are based on the correct 3D-model of the drones hardware. This allows for testing of the control-systems without any uncertainties in position or surroundings, which again allowed for better bug searching and isolated testing of parts of the systems.

In addition to testing and simulating control systems, the behaviour module, GHC, has been simulated and tested in a similar environment. These simulations has allowed for evaluation of different behaviours, heuristic function tuning and has served as a test-environment for new features and additions, without endangering humans and equipment through premature physical testing.

Additionally, Gazebo is capable of simulating package loss during data transmission between the drones and the ground station.

*CFD analysis*

In order to make the ducts of the drone as efficient as possible Computational Fluid Dynamics, or CFD, was performed in the program ANSYS CFX 17.0. The objective with this

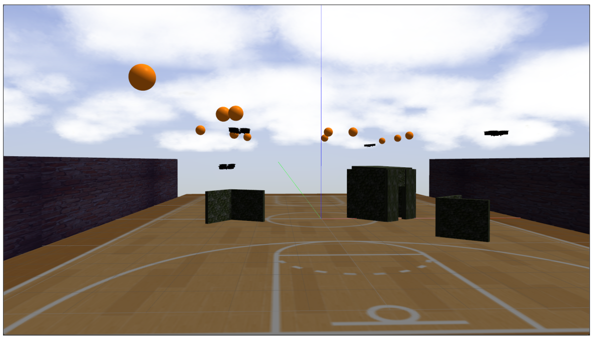

*Figure 5. The mission simulator seen from one of the friendly drones.*

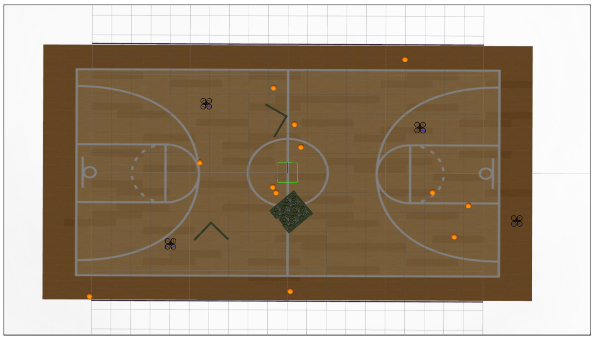

*Figure 6. The mission simulator seen from above.*

study was to find a configuration where the ducts generate extra lift. When air flows over an aerofoil, lift can be generated, something that was exploited to increase the efficiency of the duct by about 20% according to the simulations. Figure 7 shows the contour plot of relative pressure distribution around the duct, with Figure 8 showing a plot of lift generated vs air velocity from the same simulation, both showing the positive effect the duct has on the generated lift.

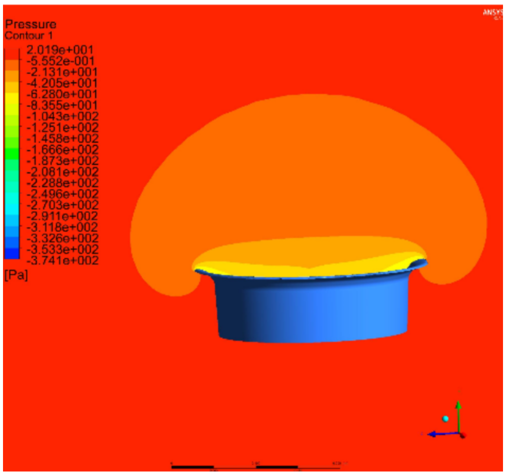


*Figure 7. Postprocessing results of CFD analysis*

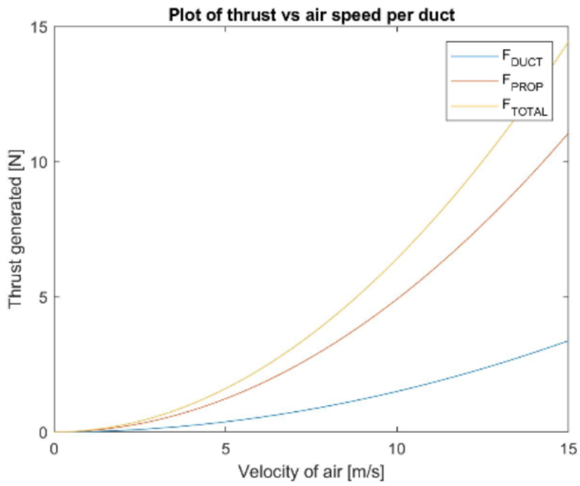


*Figure 8. Plot of thrust/lift generated vs air velocity*

*Modal Analysis*

To prevent oscillations, modal analysis (Finite Element Analysis) of the frame was performed in Siemens NX using the SOL 103 Real Eigenvalues solver, with the goal being to eliminate all eigenmodes below 150Hz.

Analysis of the initial design of the frame yielded 4 modes under 150Hz; at 51Hz, 71Hz and two at 104Hz. Modes at these frequencies is within the range of the motor's frequency during flight, and would lead to oscillations in the frame. Figure 9 shows the lowest, and thereby most critical mode at 51Hz. Stiffening the structure to prevent the motion of this mode is what the upcoming design iteration should do.

By adding a bracket to the top part of the frame (Figure 10) and rerun the analysis, the amount of modes below 150Hz is reduced to two; at 132 and 133Hz. From Figure 11 one can see that the critical part of the frame now is the stiffness of the ducts.

The next step in the process was reinforcing the bottom rim of the duct (Figure 11) as this proved to be the critical area of the structure. Running the analysis on this design proved to satisfy the design criteria, as there were no eigenmodes below 150 Hz.

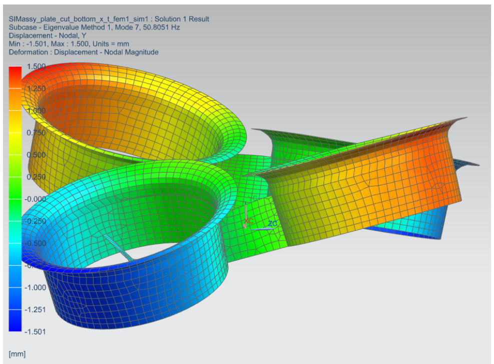

*Figure 9. Initial design: lowest mode at 51Hz*

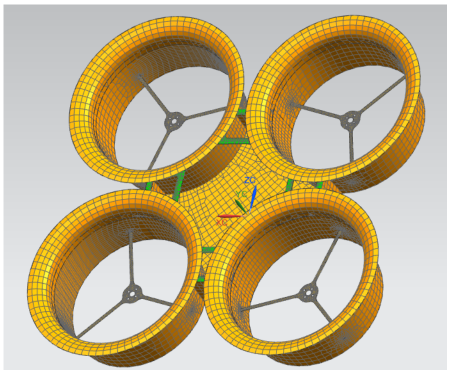

*Figure 10. Design iteration 1 Changes highlighted in green*

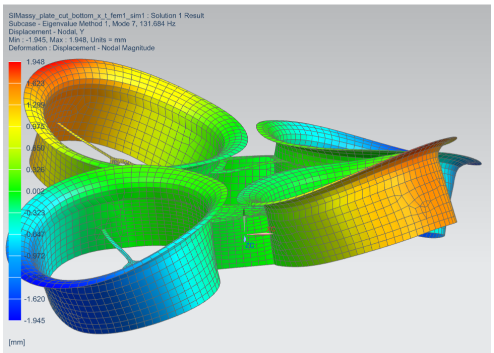

*Figure 11. Design iteration 1: lowest mode at 132Hz*

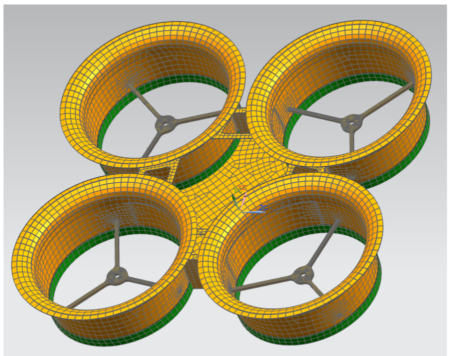

*Figure 12. Design iteration 2 Changes highlighted in green*

## Physical Testing

Physical testing was performed to validate the results obtained from CFD simulations and modal analysis.

*CFD validation*

To validate the performance of the design changes from the CFD-analysis, thrust of the motors was tested. Equal motors were tested with and without the ducts around the them. The motors without ducts were tested on a test bench, while the motors with ducts were tested while hovering under similar conditions by making the drone weight the same as the unducted motors had produced in thrust on the test bench. The power drawn from the motors were then compared. The results indicated a that the ducted design was around 20% more efficient, although further testing should be done to confirm that the recorded improvement was only from the ducts. These results strengthen the credibility of the CFD analysis conducted in the design process, that had shown that ducts whould increase efficiency by about 20%.

*Modal analysis validation*
When the frame had been produced the results of the modal analysis were replicated by flying the drone with and without the reinforcing brackets as seen in Figure 10. Without brackets the drone experienced severe resonance that made loud noise and left the drone very unstable. The motion of the oscillations resembled that of Figure 9. With brackets attached the drone flew perfectly stable during testing, and no resonance was experienced.

## CONCLUSION

The system presented in this paper is a preliminary attempt at solving some of the challenges presented in the IARC Mission 8 statement. Both simulated and physical tests indicate that autonomous takeoff and landing, autonomous flight and mapping, man-machine interaction and obstacle avoidance have a reasonable likelihood of being demonstrated in a satisfactory manner during the 28th rendition of IARC.

In order to complete the mission in it's entirety, the system is likely to require more sophisticated behaviour in some areas, especially in relation to protection and healing of the human player. These challenges, as well as others uncovered during competition, could be the targets of future work.

## REFERENCES


[1] Sethuram Balakrishnan. "Duct fanned shielding design for quadrotors". In: *UiT The Arctic University of Norway* (July 2016). URL: https://hdl.handle.net/10037/9982.

[2] Zhe Cao et al. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: (2016).

[3] R. A. Finkel and J. L. Bentley. "Quad trees a data structure for retrieval on composite keys". In: *Acta Informatica* 4.1 (1974), pp. 1–9. URL: https://doi.org/10.1007/BF00288933.

[4] Martin C. Martin and Hans P. Moravec. "Robot Evidence Grids". In: (1996). URL: https://www.frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1996/RobotEvidenceGrids.abs.html.

[5] Raul Mur-Artal and Juan D Tardos. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262.

[6] D. Panagou. "Motion planning and collision avoidance using navigation vector fields". In: (2014).

[7] Richard Szeliski. *Computer Vision: Algorithms and Applications (Texts in Computer Science)*. Springer, 2010.

[8] Shih-En Wei et al. "Convolutional Pose Machines". In: (2016).