Autonomous Quadrotor for the 2016 IARC by Team Elikos

Christophe Bédard – B.S.E Mechanical Engineering

David Binet – B.S.E Electrical Engineering

Pierre-Yves Lajoie – B.S.E Computer Engineering

Justine Pepin – B.S.E Computer Engineering

Antonio Sanniravong – B. S. E Electrical Engineering

Olivier St-Amour – B.S.E Software Engineering

Eva Terriault – B.S.E Computer Engineering

Team Elikos - École Polytechnique de Montréal

ABSTRACT

This paper presents different aspects of team Elikos' solution attempting to resolve two of the most trending issues of the moment in this domain, being the interaction between robots, and navigation in a sterile environment with no external navigation aids such as GPS. Using various sensors including GNC sensors, camera and lasers, the vehicle shall be able of such interaction while avoiding obstacles, and present autonomous navigation capabilities based on computer vision and sensors analytics work.

INTRODUCTION

Statement of the problem

Mission 7a of the International Aerial Robotics Competition involves a sheep and shepherd problem wherein the team's aerial robot must herd terrestrial robots, hereinafter referred to as targets, by either triggering the top touch paddle or the bump sensor on their front side. The targets must be herded towards a green line within a 20x20 meter arena while dodging obstacle robots made of large PVC piping roaming the arena in a circular motion. Mission completion is achieved when at least 7 targets which have been interacted with cross the green line.

Yearly Milestones

Team Elikos entered operations in November of 2013 with the initial plan of getting an aerial platform up and running using off the shelf products for IARC 2014. The result was a Turnigy Talon V2 quadrotor frame modified to hold cameras, an embedded computer and some wireless communication peripherals. The system was capable of relative position estimation through optical flow and was controlled by an automated ground station.

In the 2015 IARC edition, we presented an aerial vehicle capable of target identification and pursuit, as well as an improved positioning system using SLAM and a completely revisited platform.

This year, following last year's work, we present a vehicle with an improved platform, target identification and pursuit, as well as a revisited positioning system. We also intend to offer obstacle avoidance, interaction with ground robots, and functional artificial intelligence modules, as described in the present paper.

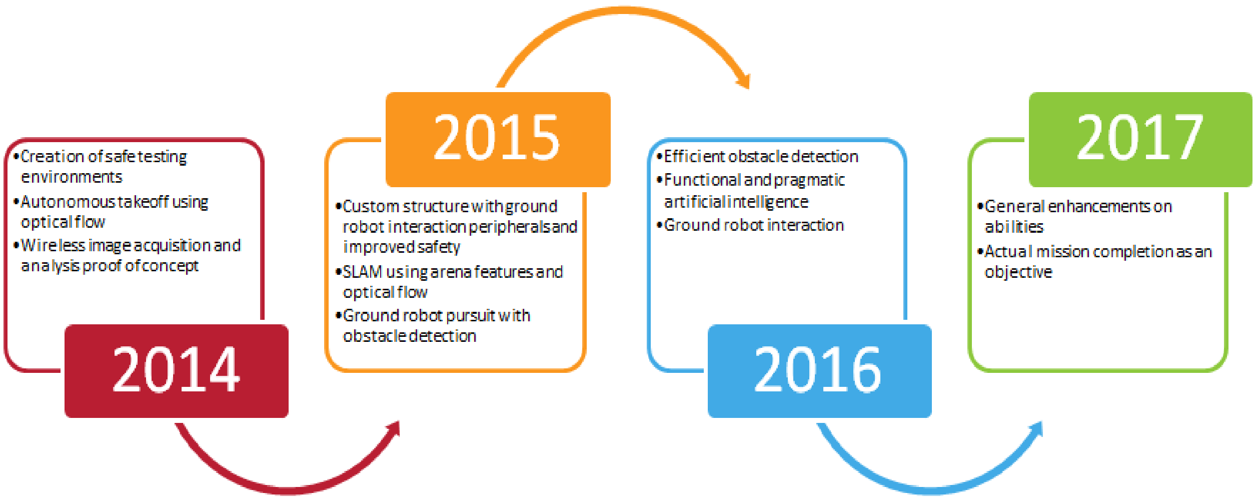

Figure 1. Yearly Milestones

Conceptual solution to solve the problem

Before outlining the details of our latest solution to mission 7, we must first go through the set of assumptions we are making about the competition. Though we do not claim them as being absolute facts, they are what we used to guide us in building a consistent solution for mission 7.

Those assumptions have initially been made before last year's competition, and have been slightly modified throughout this past year.

Vehicle movement within the arena

Whether it be to land in front of a target to trigger the bump sensor or to get close to the top touch paddle, having the vehicle make precise vertical movements is an inevitable part of the competition. The corollary to this statement is that the system must have valid position or velocity estimation at all times, including when very close to the ground to allow for stable takeoff.

Limitations of optical flow integration over time

Position estimation from optical flow integration can quickly diverge from ground truth since at every iteration, small errors are added to the estimation. However, this can be mitigated by fusing the estimation in a Kalman filter with a secondary positioning system, as described further in this paper.

The last 30 centimeters

Another limitation of our optical flow approach based on the px4flow sensor was that the measurements were nearly useless when close to the ground. Since the lens of the camera is fixed focus and the Maxbotix sonar onboard the px4flow has a minimum range of 30 cm, there is a dead zone where the landing area and the targets reside. The dead zone is further aggravated by the ground effect which can quickly make the vehicle drift with no means of it knowing so [1] [2].

Competition area sterility

The competition rules indicate that the spirit of the competition is to develop navigation tools which can function without external navigation aids such as a GNSS or large stationary points of reference. To compensate for this, the arena is filled with a 1x1 meter grid to help in the development of visual odometry. Furthermore, a direct consequence of having the competition indoors is the obvious presence of a ceiling and four walls which can be rich in robust visual features for tracking and mapping, more so than the competition arena which is only guaranteed to contain weaker high frequency features. Our combined solution of those two approaches will be presented in this paper.

Our solution

This year's solution is similar in some aspect to last year's vehicle, but also present significant changes. The vehicle is now composed of an improved carbon-fiber plate supporting a single front-facing camera used with ROVIO, a VN-100 Rugged VectorNav IMU, an onboard NVIDIA Jetson TX1 computing platform, a LIDAR-Lite laser down-facing for altitude measurement, a px4flow smart camera for optical flow, as well as Intel RealSense R200 cameras for obstacle avoidance.

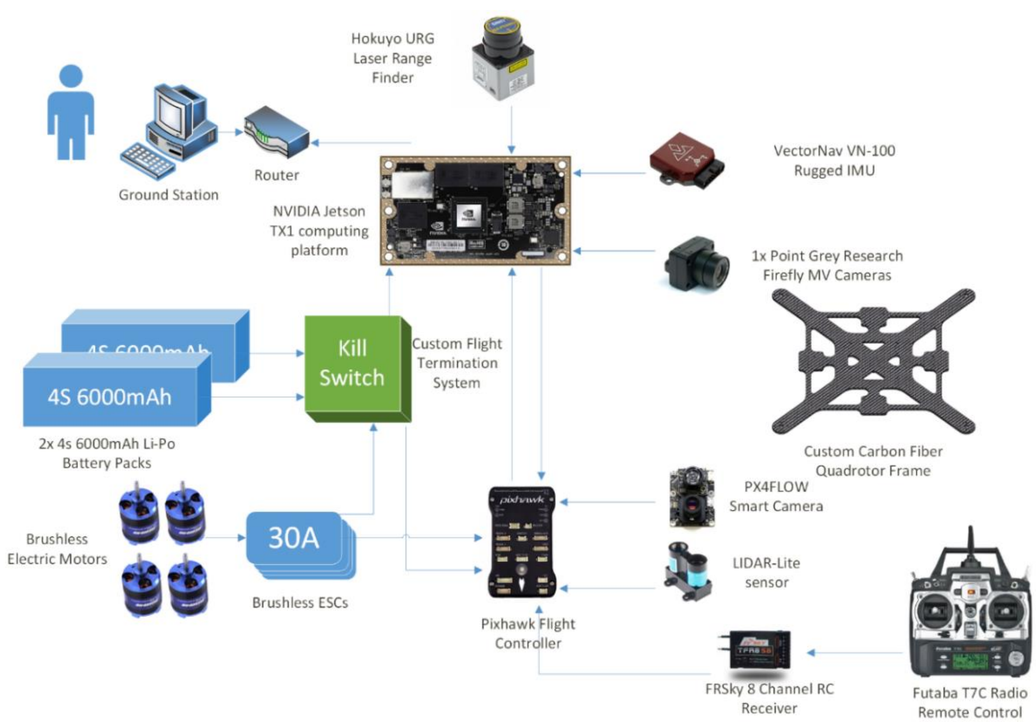

Figure 2: Overall system architecture

AIR VEHICLE

Propulsion and lift system

The basis for the UAV platform that was designed for this iteration remains structurally similar to our previous platform. The quad-rotor X configuration was kept because of its mechanical simplicity and its stability. To further increase manoeuverability and easily allow for additional payload, the lift system has been upgraded to 14”x5.5 APC propellers, 41mm diameter Turnigy Multistar brushless motors, 30A T-Motor ESCs and 6Ah 4S Lithium-Polymer batteries. This setup comfortably accommodates payloads from 2.5kg up to 3kg while remaining efficient.

Guidance, Navigation and Control

Stability Augmentation System

Various localization algorithms have been looked upon and experimented with in order to counter the sterile navigation aspect of the competition. In fact, many of the SLAM algorithms investigated have problems with either feature detection or loop closure in a repetitive and homogeneous environment such as the competition arena. This experimentation work lead us to settle on the ROVIO (Robust Visual Inertial Odometry) monocular framework [3], with the following setup: the VectorNav VN-100 Rugged IMU hardware synchronized with one Point Grey Firefly MV front facing Camera.

The ROVIO framework uses direct intensity errors of image patches as visual measurements within the extended Kalman filter update step. ROVIO makes use of a robocentric approach for the tracked features and offers a highly robust position estimation. Furthermore, with that approach, no initialization procedure is required. All those characteristics make this framework a promising candidate for resolving the many challenges to be encountered: fast and frequent motions, sterile environment and moving objects, from take-off to landing.

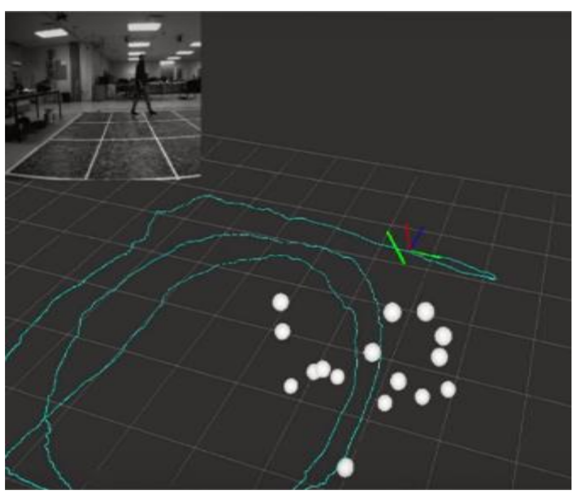

Figure 3. ROVIO experiment with moving objects

A second positioning system is present on the vehicle, using the px4flow smart camera. This camera integrates optical flow calculations [4] to estimate the position on the X-Y plane. A LIDAR-Lite laser facing downward is used for local altitude estimation.

Navigation

Our previous navigation strategy was based on a basic tracking of the target closest to the UAV. The danger of collision by interacting with the target was not evaluable by the robot and thus the action was considered completely unsafe. Hence, since our robot was unable to perform efficient obstacle avoidance, a safe distance was kept from the ground to avoid collision with any of the obstacles. With the addition of obstacle detection, navigation can be allowed to be much more elaborate. It can be broken down into three parts which consist of target selection, action planning and path planning.

Target selection is based on target detection and inference and on a high level strategy, which for this trial consists of interacting with the closest ground robot. Once the target is chosen, the desired actions are queued and executed.

Action planning is implemented with the strategy design pattern which allows switching from one type of behavior to another based on external factors such as the positions of the selected target and the desired outcome. Those behaviors can be, for example, interacting with the robot using the front side bumper or the top touch paddle.

The action planning outputs position setpoints which are then fed to the path planner. Path planning is simplified as a two dimensional problem which has as its two main goals to avoid obstacles and to stay within the arena boundaries. This is achieved using the move_base package from the ROS (Robot Operating System) navigation stack. The altitude is commanded separately to allow for simpler control. The obstacle avoidance method is discussed in more depth in the *threat avoidance* section below.

Control System Architecture

Our software architecture is made of different sub-systems that run on concurrent processes hosted on ROS. These sub-systems are each responsible for achieving individual and decoupled tasks, but still need to communicate information. Most of the sub-systems use the event driven subscriber/publisher network provided by the ROS API, while some use the TF listener/publisher for transforming 3D coordinate frames of reference.

The Detection & Tracking nodes for both target robots and obstacles uses sensors in order to gather information from the real world and send it further into the pipeline. The Detection & Tracking nodes for targets sends an unique ID, a color, a position and an orientation for every target detected by the target selection node. The latter will use this information coupled with his own position sent by the localization node to find the most desired robot to interact with based on the current strategy. The position of this targeted robot is translated into a setpoint sent to the navigation node alongside a cost map produced by the Detection & Tracking node for obstacles. The Destination node will then produce a path planning in order to send commands to the Flight Control Unit.

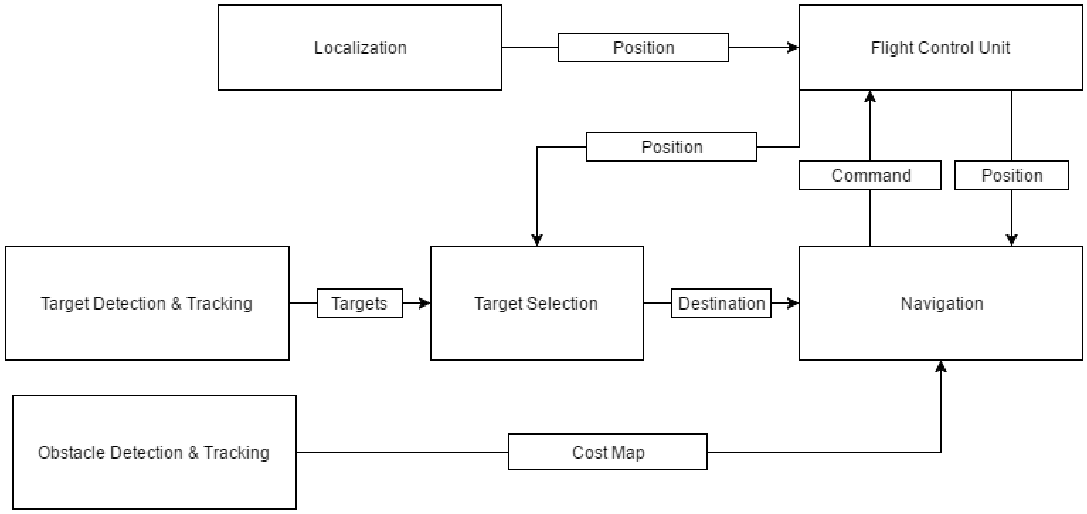

Figure 4. Software Architecture

Flight Termination System

The flight termination system, commonly called kill switch, is heavily based on the solution we presented last year. However, our new design only cuts off the power of our propulsion system to avoid restarting the onboard computer every time we need to use the kill switch. Last year, we faced some problems with our kill switch's design mainly because we were cutting the ground of the propulsion system. This caused some instabilities with other components on the quadcopter. Furthermore, we had to deal simultaneously with two grounds, which was obviously not ideal from a design perspective.

Our new design simply cuts off the positive voltage to the propulsion system to avoid any issues. This requires minimal changes on the design: switching from low-end to high-end switches. The kill switch module is controlled by a RF module which was made on a different board, in order to easily change it without modifying the whole design of the kill switch. We can now bypass the signal coming from the RF module to easily do some independent testing.

PAYLOAD

Sensor Suite

GNC Sensors

The GNC sensor suite used for control is comprised of the Pixhawk`s inertial measurement unit, a px4flow optical flow camera and a LIDAR-Lite laser altitude sensor. An additional VectorNav VN-100 IMU is used for more precise inertial measurement as well as easier hardware synchronization with the front facing Point Grey Firefly Camera.

Mission Sensors

Sensors used specifically for mission objectives include multiple Point Grey Firefly MV cameras for target identification and a Hokuyo URG-04LX LIDAR for obstacle avoidance; they are described in depth in the two following sections.

Target Identification

One of the key features needed to achieve the mission is the target robot's identification. For each target detected, we extract the absolute position and orientation in the arena coordinate frame, a unique ID and its color. The process to get all the information about the targets splits in three major steps.

The first step is to detect the targets on the images grabbed by the cameras. This year again we chose an approach which uses HSV filters and morphological transformation algorithms included in the OpenCV computer vision algorithm library. The reason why we chose this approach is that the colors of the target robots (red and green) are easy to isolate from the hue and saturation spectrum of the arena floor. Then, through morphological operations, we retain only the significant blobs by excluding the smallest ones and eliminating the noise. A disadvantage of this technique is that it requires a very accurate color calibration which needs to be done in the exact same conditions than the ones during the flight.

The second step is to give a unique ID to each target in order to differentiate the known ones from the new ones. It also provides an easy way to estimate the direction of targets by comparing the positions of each target sharing its ID with another target from a previous detection.

The last step is the extraction of the position of the target robots in the arena. To facilitate this computation, we use the TF package of ROS which offers a coordinate frames tree with which we can perform the 3D transformations between the coordinates of the targets on the images to their coordinates in the arena.

Threat Avoidance

Once the target robots identified, the quadrotor can't head straight to them without facing a serious threat from the cylinders carried by the obstacle robots. To avoid any collision with those robots we use a LIDAR sensor, the Hokuyo URG-04LX, which has a maximum measurement distance of 4m and a 240° field of view. Since the laser output is a 2D plane showing a section of the local environment, we can track the obstacles robots and identify the areas where the quadrotor can fly without hazards. To extract the obstacles positions on that kind of map, we apply a RANSAC algorithm to find sets of points that form convex half-ellipse, with a 4 inches' diameter, from the laser's point of view which are characteristic of the cylinders used in the mission.

Although that method is very effective to find obstacles with a height higher than the altitude of the quadrotor, it cannot detect the ones below the laser. To face this challenge, we expect to simplify this 3D problem into 2D by flying low enough below the smallest obstacle. Nevertheless, we look forward to use an Intel RealSense R200 camera to identify the obstacles near the quadrotor. With that camera, we can use the depth map given in output to locate any threats and even some target robots.

The threat avoidance is one of the last operations that we perform before the path planning in order to insure that the path planner module receives up to date information about the obstacles with a minimal latency. With the LIDAR or the 3D camera, we produce an occupancy grid, near the quadrotor, that is used to compute a cost map which is mainly the addition of the occupied areas and a security radius around them. The costmap_2d package of ROS is used to build that map which will be used by the path planner to react to the evolution of the local environment.

Communications

We use ROS to coordinate the communication between all our sensors and our data modules treatment since it is extremely useful to manage dataflow. Furthermore, ROS permits running nodes in a decentralized fashion. It allows us to directly link a ground station, connected through Wi-Fi, where compressed images may be received from the on-board computer and further processed either for monitoring or for CPU load relief.

Power Management System

Our current system is powered up by one battery containing 4 lithium-ion polymer cells. It's possible to use up to 2 of those batteries to increase to autonomous flight time to conduct longer tests.

The batteries power 3 main systems: the propulsion system, the flight controller and sensors system, and the onboard micro computer system. The propulsion system is powered through our kill switch so we can power-off properly this system when needed. The Pixhawk, the cameras and the laser are still powered by a 5v regulator (LM2678 from Texas Instruments).

We observed some instabilities on the output of that regulator last year which were mainly caused by the fact that the output is unstable until a load is applied. Also, the fact that we were cutting off the ground of the propulsion system. Those two issues have been fixed in the current design.

The onboard computer is still powered trough a 12V flyback regulator commercial power supply. The main change of this year’s design is that the power supply is not soldered directly on the board of the kill switch. Since we are using a different board, we can easily change it to another solution if a problem is discovered. Although having multiples boards can increase the complexity of the system, it makes it easier for every team member to work on a specific module.

OPERATIONS

Flight Preparations

While many automated mechanisms implemented in our flight controller ensure that the most crucial sensor calibrations are checked before each flight, manual verifications are still required to validate these checks and also to verify unmonitored levels and system states. Mechanical integrity is one example of an aspect that cannot be monitored automatically but that represents a great risk for human safety and for the protection of the system components in the event of a heavy impact or a crash.

The following is a checklist of elements that must be verified, at the very least, once before each extended flight session.

Pre-flight checklist

- Kill switch is functional
- Safety switch is enabled before any manipulation

- Overall mechanical structure is undamaged and propellers spin in the right directions
- Onboard computer, flight controller and peripherals power up and are mounted securely
- Li-Po batteries are sufficiently charged and well fastened
- Battery alarm is connected and functional
- Radio calibration and presets are correct
- Optical flow and computer vision cameras lenses are focused
- Inertial sensors are calibrated properly (magnetometers, gyroscopes, accelerometers)
- Telemetry is functional

After each flight attempt, the following checks have to be made.

Post-flight checklist

- Battery levels are over the safety threshold
- All component temperatures are within their normal operating temperatures
- Every mount, propeller and screw is properly tightened (especially after a hard landing or a crash).

Man/Machine Interface

Color calibration for target detection module

The calibration of the color detection is usually a slow and painful process, therefore we have developed a simple tool to make it faster and easier. First, we produce images, in the detection module, on which we display green and red circles where we find corresponding colored blobs. While viewing those images, our tool shows one track bar for each parameter to allow changes in runtime. Once the detection is good enough, we can save the values of all parameters in a file by pressing a single key. The calibration files are then produced and can be applied later to the detection module.

RISK REDUCTION

Vehicle Status

Shock/Vibration Isolation

Pixhawk's flight controller is affected by the motors' vibration during flight. In order to solve this issue, we used MATLAB/Simulink software to simulate real-time vibrations in order to find a way to reduce the effects of these vibrations on the Pixhawk's inertial sensors. After sizing calculations and simulation, it was found that the motors' vibration is reduced by 40 to 60 percent by adding 4 small cylindrical pieces of sorbothane polymer directly under our flight controller.

EMI/RFI Solutions

The EM noise generated by the propulsion system can be troublesome to the flight controller`s internal compass. For this reason, care has been taken in the final layout to keep the flight controller as far away as possible from any power circuitry.

Safety

Last year's prop guard was aesthetically pleasing and functional, but its 500 g left a lot to be desired. Therefore, our new goal was to design a lightweight prop guard. We looked at what was used – by both professionals and hobbyists – and decided on individual, 3D printed prop guards. Figure 5 shows this new design, which obviously covers less area around the quadcopter, but it weighs less, at approximately 360 g. We have also considered joining the individual pieces with small, rigid metal rods to protect the inside between two motors.

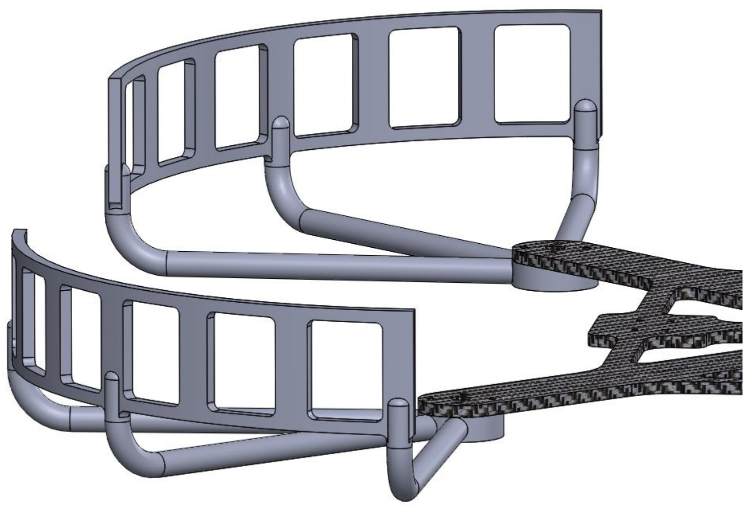

Figure 5. New prop guard prototype

Modeling and Simulation

Environment simulation

Simulating the competition environment play a big part in the development of the software solution to the mission. The Gazebo simulator has been used to reproduce the 7th mission's arena and to model the physics of a quadcopter controlled by a SITL version of the Pixhawk controller. Target and obstacle robots can be added to the environment to simulate the interactions on a visual and physical level. This way, computer vision, navigation and control logic can all be tested before being ported to the real platform.

Finite element analysis

Building on previous work, we embarked on a new challenge: manufacturing our own carbon fiber panels with vacuum assisted resin transfer molding (VARTM). To validate our new design and study the added rigidity of a sandwich panel, we used a finite element software (ANSYS Workbench). We used the explicit dynamics module to simulate the impact of a 4-meter fall on a quarter of the frame. This proved that minimal deflection was easily attainable with a core, with the important benefit of reducing weight.

Testing

Reproducing the competition's environment

In order to properly test and prepare for the competition environment, we acquired a similar carpet to the one we are expecting to face in the American Venue for this year's competition. This allows us to test the majority of our modules, including localization, navigation, target identification and pursuit as well as threat avoidance.

Past year's incertitude about the competition's environment had proven us the importance of being able to test our system in a proper environment. This investment has and will be proven more than worth it.

Figure 6. Reproduction of the competition's environment

CONCLUSION

Elikos' solution to step forward into the untying of mission 7a translates into several enhancements of our design: enhanced platform and safety guards, improved ground robot detection and tracking, as well as interaction with those objects, obstacle avoidance, revisited localization and navigation algorithms and improvements in the electrical system and materials. Our growing experience at the IARC allows us to be confident of the capabilities of our new quadrotor, and the achievement of this year's objectives.

REFERENCES

[1] I. Sharf, M. Nahon, A. Harmat, W. Khan, M. Michini, N. Speal, M. Trentini, T. Tsadok and T. Wang, "Ground effect experiments and model validation with Draganflyer X8 rotorcraft," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on on Unmanned Aircraft Systems*, Orlando, FL, IEEE, 2014, pp. 27-30.

[2] S. Aich, C. Ahuja, T. Gupta and P. Arulmozhivarman, "Analysis of ground effect on multi-rotors," in *Electronics,Communication and Computational Engineering (ICECCE), 2014 International Conference on Human-Robot Interaction*, IEEE, 2014, pp. 236-241.

[3] S. O. M. H. R. S. Michael Bloesch, "Robust Visual Inertial Odometry Using a Direct EKF-Based Approach," ETH-Zürich, Zürich, 2015.

[4] D. Honegger, L. Meier, P. Tanskanen and M. Pollefeys, "An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on* , Orlando, 2014.

[5] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann and V. Kumar, "Influence of Aerodynamics and Proximity Effects in Quadrotor Flight," in *Experimental Robotics*, Québec, Springer International Publishing, 2013, pp. 289-302.

[6] M. Orsag and S. Bogdan, "Influence of Forward and Descent Flight.," Recent Advances in Aircraft Technology, 2012.

[7] T. Madani and A. Benallegue, "Backstepping Control for a Quadrotor Helicopter," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Beijing, 2006.

[8] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proc. Sixth IEEE and ACM International Symposium*, Nara, Japan, IEEE, 2007.

[9] B. Johann and K. Yoram, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278-288, 1991.