Technical Paper for the International Aerial Robotics Competition

Thomas Rostrup Andersen Brage Eikanger Håkon Flatval
Simen Haugo Martin Lysvand Sollie Tobias Stene Hansteen
Christian Wilhelmsen

Ascend NTNU
Norwegian University of Science and Technology
Trondheim, Norway

ABSTRACT

The aim of this paper is to describe a system for a fully autonomous MAV capable of solving the seventh IARC mission. The system is designed to perform on-line strategic planning, collision avoidance, robot-to-robot interaction and navigation, relying on INS sensors and camera vision in a GPS-denied environment.

INTRODUCTION

Statement of the problem

In the seventh mission of IARC, the goal is to develop a fully autonomous drone, whose objective is to guide at least 7 out of 10 wheeled robots across a green line in a 20x20 meter flat arena, through physical interaction. The drone must accomplish its objective within a given time constraint, whilst detecting and avoiding moving obstacles. Furthermore, the drone cannot rely on external measurements, such as GPS or camera tracking systems. This description constitutes part A of the mission, while in part B the drone will additionally compete against another drone simultaneously.

Yearly milestones

This is the first time that NTNU has taken part in IARC. At the outset, we did not know what would turn out to be the most challenging aspects of the problem. Our goal this year was to attempt to solve part A of the mission, which involves several key problems such as pose estimation and navigation in the arena, detection of targets, obstacle avoidance, and creating strategic plans to herd the robots efficiently.

Conceptual solution to solve the problem

We broke down the problem into a modular structure, listed below, and assigned smaller groups of team members to work on or across of these modules.

- *Perception*: Estimate the absolute position of the drone in the grid, as well as the position of moving targets and obstacles.
- *Planning*: Compute a plan of waypoints and actions so as to efficiently interact with the targets to solve the objective in time.
- *Control*: Compute and follow collision-free paths between waypoints, and perform necessary maneuvers to interact with targets whilst avoiding nearby obstacles.

Our vehicle this year has been developed internally, with a custom frame that aims to be robust against vibrations and allow for easy replacement of key parts and quick access to sensors. We use an off-the-shelf flight controller, Pixhawk, providing low-level flight stabilization and INS measurements, and augment the drone with a laser rangefinder as an altimeter, a Hokuyo LIDAR for obstacle identification, and several cameras for both target tracking and localization of the drone in the arena.

Overall system architecture

A figure of the overall system architecture is shown in figure 1.

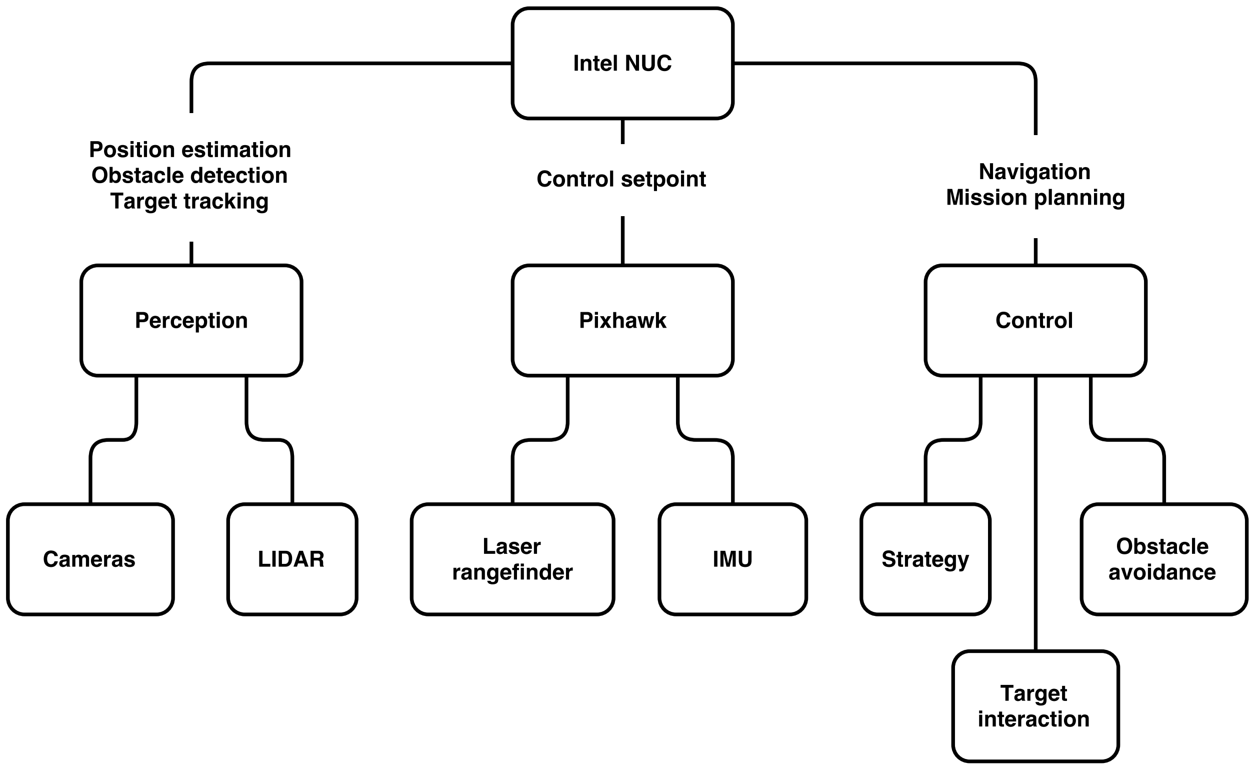

Figure 1. System architecture

AIR VEHICLE

Our vehicle was designed to accommodate the housing of multiple cameras and an on-board computer that is capable of running image processing tasks. We chose a typical quadrotor design, with brushless DC motors. With battery, payload and hull, our drone measures 110 cm (with 14" propeller setup) and weighs 3000 grams. With full payload, the operational flying time is about 13-18 minutes.

Propulsion and Lift System

Four Multistar 4225, 610kv motors powered by Afro 30A electronic speed controllers provide lift for the quadrotor. The maximum total thrust generated is 4800 grams, well above the weight of our drone.

Guidance, Navigation, and Control

Our guidance and navigation needs are motivated by our planning module that computes an efficient plan of waypoints and actions to solve the mission. The planner relies on the knowledge of the drone's 2D location in the arena to compute the best plan, and as such necessitates a method of estimating our position. Furthermore, the drone must be able to follow waypoint paths while avoiding obstacles.

Stability System

We use the Pixhawk flight controller, with the PX4 flight stack, to perform low-level stabilization and flying. Commands to the flight controller can be sent over a serial connection from the on-board Intel NUC.

Navigation

The Pixhawk provides a basic inertial navigation system consisting of an accelerometer, gyroscope and magnetometer. This allows us to obtain measurements of the drone's orientation and acceleration. If it were not for noise in these measurements, we could integrate acceleration to obtain displacement, and thereafter compute the absolute position in the arena. To cope with the inevitable drift, we have augmented the drone with four side facing Logitech C920 cameras and one downward facing fisheye camera, and employ multiple computer vision techniques for estimating pose from video streams.

Grid localization

To compute the position of the drone, we rely on the square grid pattern on the competition arena. The grid enables us to estimate the drone's position without optical-flow based algorithms, which would be prone to drift.

Our algorithm for localizing the drone in the grid requires three input components: Video from a downward facing fisheye camera, the height of the vehicle, provided by a laser rangefinder, and the last measured pose. The output of the algorithm will be the position of the drone in the horizontal plane defined by the input height, and its yaw. The algorithm is divided in two: First, a set of lines are extracted from a single image in the video stream, then our position is computed from the detected lines.

The image first undergoes an algorithm for edge detection. The resulting binary image is then warped to remove the geometric distortion caused by the fisheye lens, producing an appropriate rectilinear image. Since mapping the full field of view of the fisheye lens onto a finite plane is impossible, we crop the field of view from 180 degrees to 144 degrees. Pixels outside this range suffer from too heavy distortion to be useful for processing. Finally, a probabilistic Hough Transform is used to connect the edges into lines.

To compute our position, the lines are sorted into two categories, corresponding to the two directions the lines may be pointing. Lines that are not well-fit are discarded. Then the drone's yaw is computed. The orientations of the lines on the floor are known, limiting our possibilities for yaw. We set the new orientation equal to the orientation whose difference from the last estimated yaw is minimal. Using the height, our camera's field of view, and the distance each line has from the middle of the screen, we may compute the drone's position in terms of the square directly below the drone separately for the x- and y-coordinates. The new estimation of the drone's position is taken to be the one minimizing the distance from previous estimation.

Additionally, we use four side cameras to locate the corners of the arena. Since each corner of the grid is unique, we can obtain yet another measurement of the absolute position, and combine it with the line-based estimator for additional robustness. This is done by matching four observed intersection points near a corner with the corresponding four points in a world-scale grid model, and estimating the planar homography between them. The result is an estimate of the rotation and translation of the camera relative to the corner, containing the planar displacement with correct scale. Since the corners are not always in sight, these observations arrive at lower rates than the downward camera system.

A downward facing rangefinder is used to compute the drone's height above the arena. The height is necessary to resolve the scale ambiguity in going from image-space to world-space displacements.

Sensor fusion

We use the Pixhawk measurements of linear acceleration and orientation in our own Kalman filter, to estimate the rotation and translation from the grid origin frame to the drone frame. We augment the filter state vector with a model of the IMU noise to estimate bias in the double-integrated linear acceleration vector, and include measurements from all corner detections, and the displacement measured from lines, to reliably estimate the drone's absolute position in the grid.

Strategic planner

With knowledge of the drone's absolute position, and estimates of each target position, a strategy to solve the objective can be computed. A strategy amounts to a sequence of 2D waypoints that the drone must navigate toward, and an action to perform at each point. For example, that the drone is to search for targets near position (x, y) in the grid, and land on top of the target that best matches a given orientation and position.

The planner was developed by two master students on our team as their thesis work [1]. The herding problem was modelled mathematically and formulated as an optimization problem called Time-Dependent Orienteering Problem with Time Windows. The goal in the optimization is to find a sequence of waypoints and actions that maximize expected reward. Metrics such as time-of-travel, likelihood of a successful interaction, or a target's distance to an obstacle or the edges of the arena, can be used in the design of the reward function.

Flight Termination System

The IARC Common Kill Switch is designed with two assumptions: "The motors are powered from no more than a 3-cell series connected Li-Poly battery pack" and "the main motors draw 35-amps continuous with peak pulses of 100-amps". We have made a few changes to the reference design, as our drone is powered by two 4-cell batteries, and may exceed 35-amps continous draw. We have added a 4th N-channel MOSFET to allow higher current flow through the kill switch. In addition we have added bullet connectors to make the kill switch act like a power distribution board as well.

PAYLOAD

Sensor Suite

GNC Sensors

The Pixhawk flight controller includes an affordable IMU sensor, consisting of an accelerometer, magnetometer and gyro. Additional sensors are supported as plug-ins. We also include the laser rangefinder LIDAR-Lite v2 for height measurement. The rangefinder has a range of 40m, well above the designated operational limits for the mission.

Four side facing Logitech C920 cameras, and a downward facing ELP-USBFHD01M-L180 fisheye camera provide additional motion measurements. The downward facing camera has a field of view of 180 degrees, though we crop it to 144 degrees. Pixels outside this range suffer from heavy geometric distortion, and are unsuited for image processing.

Mission Sensors

The sensors chosen are motivated by the need to avoid obstacles and to identify and track several targets. Obstacle identification is performed by usage of the Hokuyo UST-10LX, a planar scanning LIDAR, attached to the top of the drone. The LIDAR has a functional range of 0.02-10m meters in a 270 degree angle span. The position of the obstacles is computed by trigonometry using the estimated drone position and orientation. We compute collision-free paths between waypoints by connecting straight line segments that minimize time of travel, constrained to not be near any obstacle within a specified radius. For additional security, we allow intervention by a higher-priority controller if the drone gets sufficiently close to any object.

Targets located beneath the drone in a 4x4 meter vicinity are detected by a blob-tracking algorithm running on the video from the downward facing camera. Detecting blobs is done by segmenting shapes in the image that have the distinct color plate, common to all ground robots in the competition. The grid position of detected targets is inferred from the image coordinates of the blob by inverse projection, using the estimated orientation and position of the camera. Targets located outside the visible area beneath the drone are detected by the side cameras whenever possible.

Communications

The Pixhawk is connected to the on-board Intel NUC through UART over a FTDI USB to UART adapter. The on-board computer communicates with the ground station through 802.11ac WiFi using its integrated Intel 8260 WiFi adapter. Our UAV has two antennas, and the ground station computer is connected to a Asus RT-AC66U router. The maximum WiFi bandwidth possible for this combination is 867 Mbps at 5 GHz.

Communication between the Pixhawk, the on-board Intel NUC and the ground station is handled primarily by the Robot Operating System (ROS), but with custom transmission for video streaming. MAVROS, a ROS package that wraps around the MAVLINK protocol, provides communication between the Pixhawk and the on-board Intel NUC. IMU measurements are transmitted to the NUC, while control commands are transmitted to the Pixhawk. Heartbeat messages must be sent regularly to establish link connectivity, and is handled automatically by ROS.

The downward facing camera is connected to the NUC via USB. To minimize latency, we perform image processing for this camera on-board, by directly memory-mapping buffers from the internal memory of the camera to the NUC RAM.

The remaining four cameras are also connected to the NUC via USB. Due to performance requirements, processing these video streams is performed by the ground station. Video frames are transmitted over WiFi using the GStreamer library. Computations performed by the ground station are sent back to the on-board computer over the same WiFi.

Telemetry data is also sent over WiFi through ROS messages. This allows live visualization and debugging of the mission.

Power Management System

The drone is powered using two ZIPPY Flightmax 4000mAh 4S1P 20C batteries. Two FrSky FLVSS LiPo Voltage Sensors are used to monitor the cell voltages. The current from the batteries flow through our custom made safety shutdown switch circuit board, and is distributed to the ESCs using side-mounted bullet connectors.

OPERATIONS

Flight Preparations

To ensure the safe operation of our vehicle, the following checklists are used for every flight:

Preflight checklist

1. Verify that the vehicle is in good physical condition, with no loose parts and with no objects in danger of being hit by any propeller
2. Turn on killswitch transmitter, ensure it is set to the kill position
3. Turn on RC controller
4. Connect batteries to the killswitch input connector and the voltage monitors
5. Verify that the battery voltages are visible on the RC controller screen, and that the batteries are in a charged state
6. *If the on-board computer is to be used:* Power it on and connect to it using SSH over WiFi from the ground station computer, start any necessary software

Takeoff checklist

1. Verify that all persons are at a safe distance, and behind a protective net if available
2. Verify that all switches on the RC controller are in the correct position, and throttle set to minimum
3. Enable killswitch to power motors, wait for the correct audible response from the motor controllers
4. Arm flight controller using the RC controller and take off

Landing and postflight checklist

1. Gently land the vehicle
2. Disarm the flight controller using the RC controller
3. Set the killswitch transmitter to the kill position
4. *If the on-board computer is running:* Gracefully shut it down
5. Remove the batteries

Man/Machine Interface

Transitioning between manual and autonomous flight is done using the RC controller. After a manual takeoff, control can be given to the on-board computer by switching to the *offboard* control mode on the RC controller. This will only be allowed by the Pixhawk flight controller if the requirements for entering offboard mode are satisfied; if they are not, the mode change will be rejected and the vehicle will remain in manual control. Manual control can be regained at any time by exiting offboard mode on the RC controller.

For debugging and live-visualization purposes, we have designed a tool that visualizes the estimated state of the moving targets, the location of the drone, and overlay the planned path of waypoints and actions.

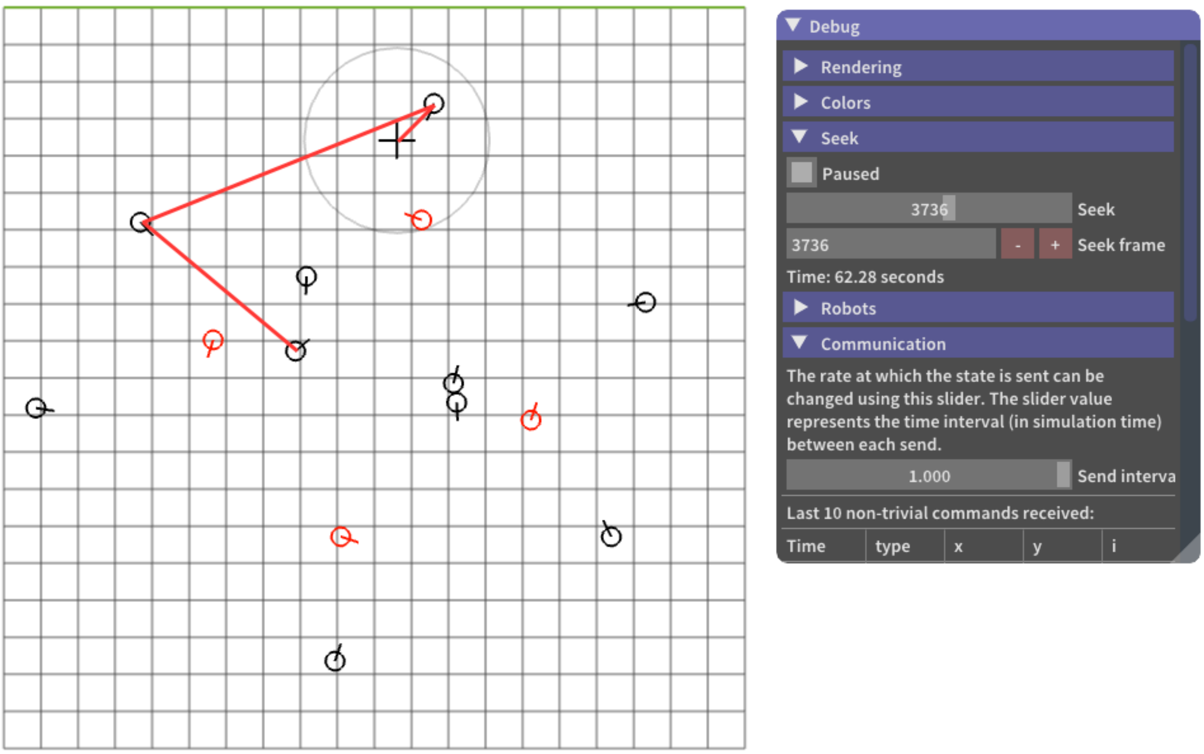

Figure 2. We visualize the estimated state of the moving targets (black) and the obstacles (red), and the location of the drone (cross), and overlay the computed path from the planner (red lines).

RISK REDUCTION

Vehicle Status

Shock/Vibration Isolation

Shock and vibrations can easily disturb the sensors on the control board, in addition to making the video from the cameras blurred. Vibrations should therefore be reduced as much as possible.

Several methods were used to reduce the vibrations. The vibration reduction starts in the propellers and motors, they have been carefully measured and balanced to reduce the production of vibrations, from unbalanced weight distribution. The arms of the drones are made of carbon fiber, which helps to keep the frame lightweight and strong, but it is also a good material to absorb the vibrations from the motors. The material and the geometry of the frame also reduce deflection in the frame and reduce asynchronous vibration. The control board is mounted with shock absorbing tape to reduce high frequency vibrations, but still allows the control board to get quick feedback from the movement of the drone.

EMI/RFI Solutions

The electronic speed controllers (ESC) have potential to generate a lot of noise to nearby circuits. To avoid this the ESC is placed as far away as possible from the sensitive electronics. Especially the magnetometer. There is also made space for EM shielding if needed. To ensure good connection between kill switch and the multirotor, a standard frequency hopping radio signal is used.

Safety

To ensure the safety of the drone, we use prop guards and large slow-rotating plastic propellers. The plastic propellers cause a bit more vibrations, but they are less dangerous than hard and sharp carbon propellers.

Modeling and Simulation

Validation of the frame

Finite element analysis was used to optimize and validate the structural integrity of the drone. In addition, MATLAB was used to calculate the optimal thickness of the carbon fiber arms. The mounting points between the arms and the center of the frame are designed with planned yielding in place. In a crash, the joint breaks at a planned point to ensure minimal damage to the drone. The joint is a small part made out of plastic and is cheap and easy to replace. The other parts of the drone were designed far stronger than the yielding point for added rigidity and reduction of vibrations and deflection between the control board and the motors.

Simulation of high-level plans

The planning algorithm was tested heavily by the use of computer simulations. We had initially tried Gazebo as the platform for performing simulation, but found it to be far too heavyweight for our purposes. Therefore, we wrote our own simulation, copying the behaviour of the targets and obstacles, and performing simple collision handling and robot-to-robot interaction.

The physical drone-to-robot interactions, such as landing on the targets, were abstracted away as timed events. For example, if the drone is commanded to land on a specific target, our simulated drone will follow a straight path towards it at a constant speed. Once it is within a specified radius, a timer will count down a programmable amount of seconds, after which the landing will have been performed.

To keep the simulation fast and simple, we abstracted away the drone dynamics and the collision-free path generation, and instead made the drone move at a programmable top speed, following the shortest possible straight-line segment towards its target.

The simplicity of the simulation allows us to run many thousands of simulations simultaneously. We were therefore able to test a myriad of different scenarios, and visualize the output of the planner in each.

Testing

Our drone pose estimation algorithms have been tested and compared to ground truth measurements from an Optitrack camera motion tracking system. Thus we have been able to tune and improve our algorithms with continuous testing.

The physical parts of our vehicle, in particular the motor and arm mounts have been developed with several design iterations and rapid prototyping, allowing us to test the strength of our parts and improve the design when needed.

CONCLUSION

This report was written before all our components were finalized. We have yet to begin the work on target interaction, and have yet to finalize target identification and tracking. Generation of high-level strategic plans and collision-free paths has been implemented, though not integrated with the control system.

The position estimation system for the drone has received significant attention from multiple team members, and achieves usable performance at the time of writing. We consider this module to be of utmost critical importance, as it underlies the functionality of all other modules. We have had to solve multiple problems within this module, including video-rate detection of grid lines in a 60fps fisheye camera, detection of arena corners and estimation of a 6 DOF pose from these, and streaming of several video streams from the on-board computer to the ground station. Finally, we developed our own Kalman filter that fuses camera and IMU measurements. Further testing in a large-scale arena, and integration with the control system, remains as work to be done until the competition.

We foresee that our position estimation approach based on detecting lines may be difficult to use in the real competition, due to the underlying assumptions we have made. For example, we assume that the grid pattern is in fact rigid; an assumption that may be broken if the surface consists of soft material that can be affected by wind. The success of our approach is also highly dependent on the underlying background texture, which is unknown to participants until they arrive at the venue. We would like to compare our position estimation scheme with other camera-based navigation techniques, such as SLAM [2] or the use of optical flow and stereo vision [3].

REFERENCES

[1] Bakke, A. S. and Rugsveen, N. I. (2016). Handling autonomous robot scheduling as an optimization Problem. Norwegian University of Science and Technology.

[2] Engel, J., Sturm, J. and Cremers, D. (2014). Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems*, 62(11), 1646-1656.

[3] Badino, H., Yamamoto, A. and Kanade, T. (2013). Visual odometry by multi-frame feature integration. *In Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 222-229).