PCC's Autonomous Air Vehicle System for IARC 2017

1 June 2017

Stephanie Guzman
University of Arizona
Alberto Heras
University of Arizona
Stephen Jewell
University of Arizona
Colin LaSharr
University of Arizona
Ryan Province
University of Arizona
Frank Manning
Pima Community College

[1] Abstract

The Pima Community College UAV Club has designed an air vehicle system to compete in the International Aerial Robotics Competition (IARC). The rules require an autonomous air vehicle to herd a group of 10 ground robots while avoiding collisions with a second group of 4 obstacle robots. All 14 ground-based robots are themselves autonomous and move according to their own internal algorithms, including specified responses to external collisions and mechanical forces. The air vehicle is designed to use machine vision as well as lidar and sonar scanning to sense the positions of ground robots, and to navigate relative to a 20 m x 20 m arena. The arena is marked with a known grid pattern.

[2] Introduction

[2.a] Statement of the Problem

The mission requires an autonomous aerial robot to herd a group of 10 ground robots ("mission robots") within a square 20 m arena. An additional 4 obstacle robots are also present and must be avoided by the air vehicle. All 14 ground-based robots are autonomous and move according to known, internal algorithms.

The arena is indoors on a floor marked by a pattern of 1 m white grid lines internally. The 4 outer edges of the square arena are marked by a green line on one end and red line on the opposite end. The other 2 outer edges are white lines. The overall objective is to herd all mission robots across the green boundary. Each mission robot can be steered to a limited extent by applying a small mechanical force to the robot – either to a paddle on top or a bumper in front. In parallel with these activities the aerial robot must also avoid colliding with the obstacle robots, each of which has a cylinder extending from the top. The height of each cylinder has a 2 m upper limit but the height is otherwise undefined. Note that all ground-based robots will generally be colliding with each other, thus complicating their movements.

The mission terminates at a 10 minute deadline. Other factors may also terminate the mission prior to the time limit, such as a collision with an obstacle robot, or if all mission robots either go out of bounds or reach the green boundary.

[2.b] Conceptual Solution to Solve the Problem

The air vehicle uses on-board machine vision to detect the grid lines and boundary lines of the arena in order to keep track of the position of the vehicle with respect to the arena. In addition, the vision system determines the positions of the ground robots. For obstacle robot avoidance, a lidar scanner generates a 240 degree horizontal scan pattern in order to sense obstacle robots within a range of 4 m. Four additional sonar sensors are used to cover the 120 degree blind spot of the lidar. Lidar, sonar and camera data are combined in order to avoid obstacle robots.

Disclaimer – this paper describes a conceptual solution that is intended to perform the full IARC mission at a future date. Only a small part of the solution has actually been implemented in hardware or software as of this writing.

[2.b.1] Figure of Overall System Architecture

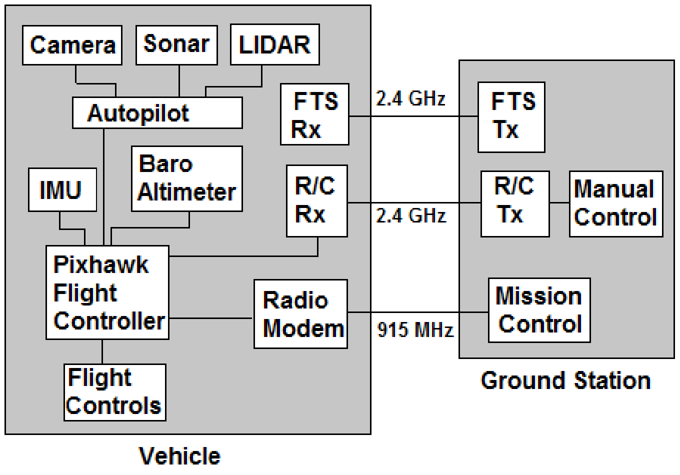


Figure 1. Overall system architecture.

[2.c] Yearly Milestones

Vehicle and machine vision development will be emphasized in 2013-17. Enhanced maneuverability and controllability will be done in 2017-18.

[3] AIR VEHICLE

The vehicle consists of a highly modified 3DRobotics Y6, which has a tricopter configuration. Most of the Y6 internal electronics and sensors are retained, as are 2 of the 3 motor pylons, plus 4 of the 6 motors.

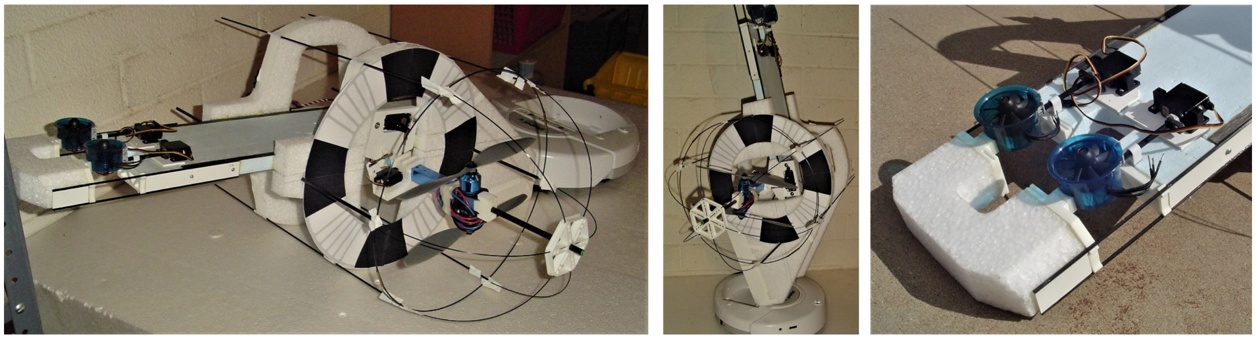


Figure 2. (L-R) Air vehicle landing attitude for activating robot front bumper, attitude for top switch, twin tail rotors.

The Y6 main body and tail pylon are replaced with a 20 mm thick styrofoam plank reinforced with carbon strips. The original landing gear is replaced with structures of expanded polypropylene (EPP). Blocks of EPP are also located in the nose and tail to minimize crash damage.

The vehicle has a total of 6 rotors for propulsion and attitude control. The Y6 is modified such that all 6 rotors have some form of Thrust Vector Control (TVC). All rotors can tilt through a 180º sweep angle. Of the original 6 main rotors in the Y6, 4 are retained as-is, each with a diameter of 254 mm. The main rotors are grouped in 2 countrarotating pairs. The 4 rotors are also modified to tilt as a single unit about the pitch axis for increased maneuverability. The remaining 2 rotors in the Y6 tail are replaced with a thruster that consists of 2 small electric ducted fans (EDF) with 40 mm diameter rotors. Each EDF implements TVC with independently-controllable tilt angles of 180º about the roll axis.

[3.a] Propulsion and Lift System

Altitude controlled by total thrust produced by the 4 main rotors. Horizontal translation in the X-direction is controlled by tilting the 4 main rotors about the pitch axis. This allows the lift vector to be tilted without rotating the fuselage, which by comparison allows rapid changes to longitudinal acceleration. Horizontal translation in the Y-direction depends on the mode:

Mode 1 -- Low airspeed
The thruster TVC generates a thrust component in the Y-direction for vernier control of lateral acceleration. Again this can be done rapidly and without rotating the fuselage.

Mode 2 -- High airspeed
The roll angle of the entire vehicle can be varied, which tilts the lift vector for lateral acceleration. This type of control is typical of helicopters and multirotor vehicles.

[3.b] Guidance, Navigation and Control

Attitude control
Pitch -- Controlled by thrusters in the tail.

Yaw – Controlled by differential torque on the 4 main rotors.
Roll – Controlled by the thrust difference between left and right main rotors.

Note that the tail thrusters generate an unwanted yawing moment whenever they generate a force component in the Y direction. This yawing moment is automatically countered by differential torque on the main lift propellers. Alternatively, thruster control is flexible enough that the thrusters can theoretically be used for yaw control if needed.

Also note that, depending on the strategy used for maneuvering, the fuselage can be held at a level attitude (roll angle = pitch angle = 0), independent of maneuvering. Alternatively, the fuselage can similarly be held at a zero roll angle and constant pitch angle within a pitch range of 0º to 90º. As an example, the vehicle can hover with the fuselage in a 90º nose-down attitude.

[3.b.1] Stability Augmentation System

The flight controller, based on an off-the-shelf Pixhawk unit, reads IMU sensors, including accelerometers, gyros and magnetometers. The processor uses an Extended Kalman Filter to calculate Euler angles. Various PID controls are used to actively control rotation rates, Euler angles, rotation rates and altitude.

[3.b.2] Navigation

Navigation is performed primarily by a machine vision system that tracks grid lines in the arena. A Hokuyo URG-04LX scanning laser rangefinder is also used to detect obstacle robots. The lidar sensor is augmented by sonar sensors that cover the lidar's blind spot. The primary altimeter is a barometric pressure sensor, augmented by a Sharp IR rangefinder that is used to compensate for variations in barometric pressure.

[3.b.3] Figure of Control System Architecture

The intended autopilot is a bare-metal ARM Cortex M4 (STM32F407) programmed in Ada 2012. The Ravenscar subset of the language would be used. We are investigating the use of the Crazyflie open source drone project. Anthony Gracio, an intern at the AdaCore company, has rewritten Crazyflie software from C to SPARK, which is a high-reliability subset of Ada. In the process several bugs were uncovered in the original C code.

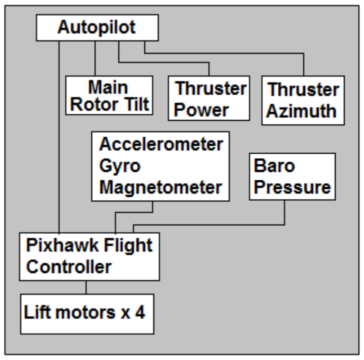


Figure 3. Control system architecture.

[3.c] Flight Termination System

The flight termination system (FTS) allows an operator to remotely cut power to the lift motors and thruster in an emergency. This is a crucial safety feature intended to prevent injury and property damage. The FTS consists of a conventional R/C system on 2.4 GHz spread spectrum. The R/C system is connected through an electronic interface to the IARC Common Safety Switch, as described in the Mission 7 rules.

[4] PAYLOAD

[4.a] Sensor Suite

[4.a.1] GNC Sensors

The vehicle is cannibalized from a 3DRobotics Y6 tricopter with the following equipment:

Flight Controller (FC): Pixhawk
- 32-bit STM32F427 Cortex M4 core with floating point unit.
- 168 MHz/256 KB RAM/2 MB Flash
- 32 bit STM32F103 failsafe co-processor
- Firmware: APM:Copter 3.1 (open source)
- OS: NuttX RTOS

Sensors:
- ST Micro L3GD20H 3-axis 16-bit gyroscope
- ST Micro LSM303D 3-axis 14-bit accelerometer / magnetometer
- Invensense MPU 6000 3-axis accelerometer/gyroscope
- MEAS MS5611 barometer

Power System:
- Ideal diode controller with automatic failover
- Servo rail high-power (7 V) and high-current ready
- All peripheral outputs over-current protected, all inputs ESD protected

Ground station uses APMPlanner2 program, also open source

[4.a.2] Mission Sensors

[4.a.2.1] Target Identification

[4.a.2.1.a] Machine Vision

First iteration – Single CMUCam5 Pixy with fisheye lens. We initially chose a single camera configuration because it would simplify the image processing demands. A wide angle (“fish eye”) lens could be used to view the entire arena. As stated in our PDR: “The CMUcam5 is the best choice. It has been built to run specifically on the Raspberry Pi and other similar open source microcontrollers. The tracking capabilities are also already built into the CMUcam5.

Second iteration – Multiple CMUCam5 Pixy cameras. After further investigation we concluded that the optimal design architecture involves multiple cameras. This will help combat the serious limitations presented with the Pixy's small sensor size and the distortion inherent to wide field of view lenses. The pixy camera will only detect objects of a single color that cover at least 4 pixels on its detector.

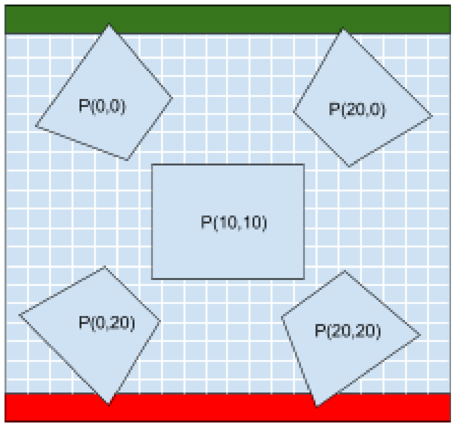


Figure 4. Viewing region assignment per camera

Final iteration – Single ELP camera with fisheye lens. The final design for our ground robot tracking system has continued to evolve to meet the requirements set earlier in this document. Due to the lack of image quality in the Pixy CMUcam5 and their lack of contribution to the tracking of the system, the four camera array with one down facing camera system has been replaced with a single ELP 180° down facing camera. This is mated to a single Raspberry Pi that will take the input data and transform it into useful information for the autopilot. The specifics of this code are outlined in the software section below. These elements are together inside of a 3D printed cage that was custom designed for this application. This cage will suspend the camera below the UAS as to meet the requirement for seeing the grid. The UAS used for the competition will be a 3DR Iris.

Subsystem/Sub-assembly and Interface Design (Hardware)

Subsystem A: ELP 180° Camera

Figure 5. ELP camera at left. At right is the camera mounted in 3D printed cage with Raspberry Pi computer

The ELP fisheye lens camera has the ability, when pointed straight down to see a large majority of the grid. It communicates through a USB interface to the Raspberry Pi.

The arena floor may or may not have problems with specular reflection. Vertical polarizers can reduce problems with glare (below).

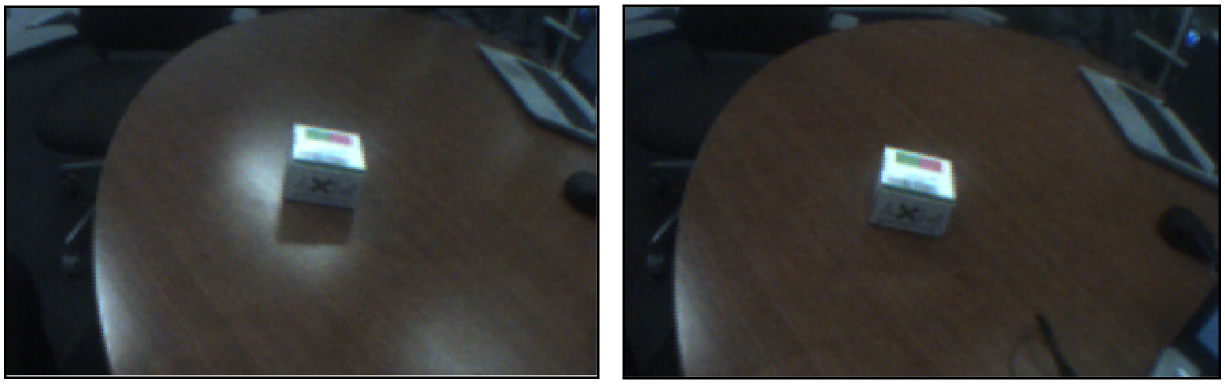


Figure 6. Glare reduction comparison: without polarizer vs with

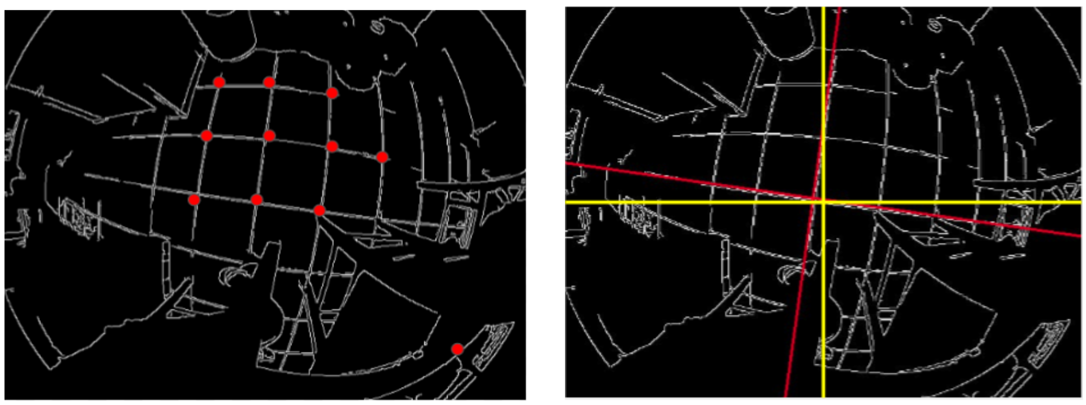


Figure 7. Grid detection using corner detection (left) and Hough transforms (right)

The left image shows corners detected by Shi-Tomasi corner detection. These are used to map the corners of the grid tiles to positions in space by exploiting knowledge of their known dimensions.

The right image shows the center axis of the ELP camera's center FOV in yellow, and the detected lines by the Hough Transform. By taking the angle between the two sets of lines, a relative angle to the grid can be estimated.

[4.a.2.1.a] Lidar tracking

We are investigating the use of a lidar scanner (Hokuyo URG-04LX) in order to augment the camera-based machine vision system for detecting and tracking mission robots. The idea is that lidar and camera systems have different strengths and weaknesses and can complement each other.

The air vehicle is required to spend most of its time out of ground effect, which means the lidar scan plane is necessarily inclined at some nonzero angle relative to the floor. This angle complicates the tracking algorithm due to ambiguities in measuring radial velocity.

The problem is as follows – one basic function in tracking is to measure the relative velocity of an object being tracked. One way of doing this is to compare successive frames of tracking data and take the difference in position. Lidar data consist of an array of range points expressed in polar coordinates (r, θ). It's easy to determine tangential velocity dθ/dt, but radial velocity dr/dt is much more problematic. The system is getting echoes mostly from an irregular surface on top of the robot, resulting in data that's noisy to the point that it's hard to know how to move the lidar to prevent the robot from moving out of the scan plane and vanishing altogether.

One solution to this problem is the use of a nodding lidar, but we'd like to avoid the mechanical complexity and mass of a nodding mechanism. Doppler lidar is another possibility.

A simpler alternative might be to to have the ability to discriminate between the near and far edge of the robot. For example, if the robot disappears, and if the last known echo was of the near edge, then it is more likely the robot was moving further away when it vanished. The reverse applies to the far edge. Sensing the difference makes it easier to move the lidar to re-acquire the robot or to prevent its disappearance in the first place.

The question is whether we can get this information by analyzing the shape of the range data curve. Can we correlate shape as a function of the distance between the scan plane and center of the robot? In Figure 8 (below) lidar data was recorded as a function of scan plane location. The scan plane is inclined approximately 20° relative to the floor. The lidar-to-floor slant range varied from about 1.11 m to 1.45 m. In the range plots, small squares are units of cm, larger squares are 10 cm. Note that in the "mid level" plot the 340 mm diameter of the robot is apparent in the data.

In the case where the lidar scan plane intersects the far edge of the robot, two things are true at the same time – the apparent width of the robot is narrow, and there are 2 pronounced stairstep discontinuities where the range data jumps to the floor.

It may be possible to simply use these 2 parameters – apparent width and dual floor jumps – to discriminate the far edge from other echoes. More work is needed. We assume here that both temporal and spacial sampling rates are high enough to avoid aliasing problems. We also assume tracking of lone robots – if 2 or more robots are close together, a different approach would be needed.

Also note that the discussion in this section is specific to mission robots. Obstacle robots might actually be easier to track because they're much taller than mission robots.

Measurements were taken with an iRobot Create robot without the superstructure modification for the IARC competition. Results would change with the newer Create 2 with the added superstructure, especially with the top switch pressure plate. In this case the large discontinuities at the far edge would be even more pronounced than without the plate.

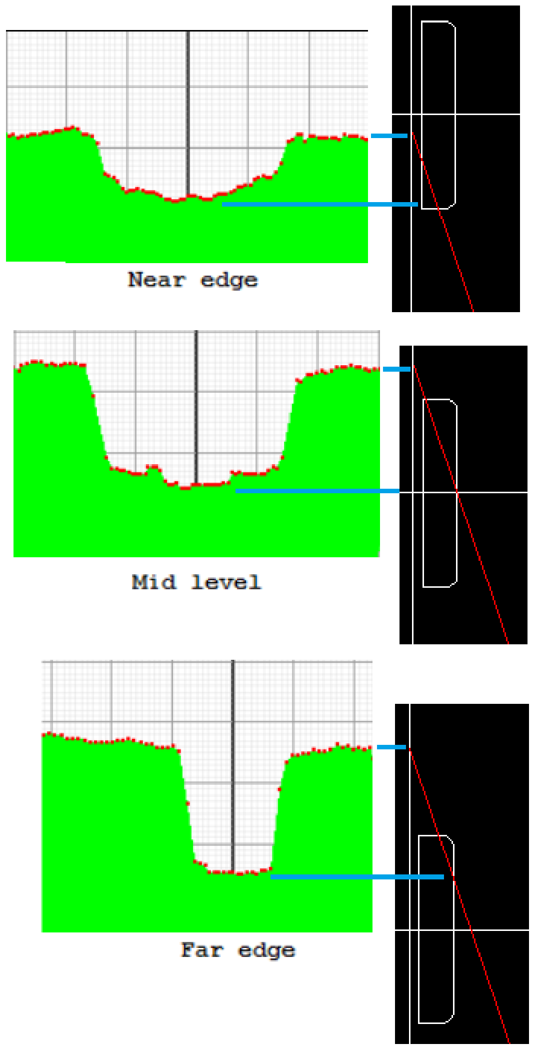


Figure 8. Range shape as a function of scan plane location.
Side view of mission robot on right, red line is lidar scan plane.

[4.a.2.2] Threat Avoidance

Protective Cage and EPP Structure
A protective cage surrounds the 4 main rotors. The cage is built primarily of carbon fiber rods and tubes, and is supported by a central EPP structure on the fuselage. A pair of small EDFs are positioned outside the cage, but are protected by a combination of small size, shrouds and a block of EPP in the tail. Sensors in the nose are also protected by an EPP structure.

Recovery from Inverted Landing
The vehicle is designed to be able to recover from an inverted landing. This is a side benefit of the ability to tilt the main lift propellers. In order to perform an inverted takeoff, the tail rotors would elevate the tail as much as possible. Combined with tilting the main rotors back as far as possible, the main thrust vector would be oriented vertically, ready for takeoff (Figure 9).

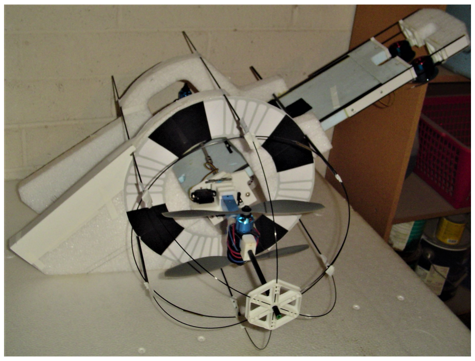


Figure 9. Vehicle in inverted takeoff orientation.

[4.b] Communications

A radio modem allows 2-way communications between the ground station and air vehicle. The radio is a 3DR Radio Telemetry V2 (915 MHz), based on HopeRF HM-TRP module

[4.c] Power Management System

A 11.1 VDC lithium-polymer battery powers the propulsion and lift systems, as well as all electronics on the vehicle. A power distribution board routes power to individual ESCs that drive all 6 rotors. The board also contains voltage and current sensors for the main battery, as well connections to BEC voltage regulators built into the 6 ESCs. The BECs supply power to the rest of the system.

[5] OPERATIONS

[5.a] Flight Preparations

[5.a.1] Checklists

Checklists are used for preflight inspections. The airframe is checked for damage, rotors are checked for integrity, communications are checked for data link integrity and controls are checked for proper operation.

[5.b] Man/Machine Interface

Since the vehicle spends most of its time hovering or flying at relatively low airspeeds, it's not required to have a low drag coefficient. Therefore the structure of the vehicle is open, with equipment easily accessible for operation, maintenance and replacement.

[6] RISK REDUCTION

[6.a] Vehicle Status

A large number of parameters are streamed in real time to the ground station from the air vehicle. Parameters include Euler angles, angular rates, voltage, current and altitude.

[6.a.1] Shock/Vibration Isolation

The sensors are mounted on soft foam to isolate the internal gyros and accelerometers. In addition, all propeller blades are balanced in order to reduce vibration.

[6.a.2] EMI/RFI Solutions

In the flight controller, all peripheral outputs over-current protected, all inputs ESD protected.

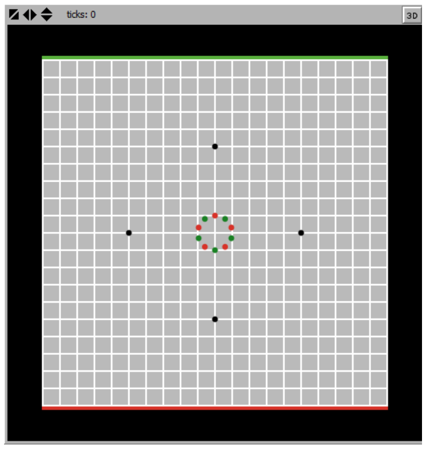
[6.b] Safety

The cage structure reduces somewhat the chances of injury due to spinning propellers.

[6.c] Modeling and Simulation

In addition to the software on board the tracking system, a simulation was created to analyze the movements of the mission robots. The mission robots' movement is chaotic without the addition of a UAS to the system. Having knowledge of the probabilities of deviation from the current heading will provide insight to the locations of robots in the situation a robot is out of the FOV. Using the provided code to program the mission and obstacle robots, the movements were replicated in the simulation. The positions and headings are then tracked via output to a text file. The robot position data tracked is used to generate probabilities of the mission robots in the case that the mission robots are lost or out of the FOV.

The simulation is an agent-based simulation built in Netlogo 5.1. Netlogo has two main types of agents - "patches" and "turtles". The patches were used to generate the grid in a scaled model of the arena. The turtles are used as the robots, circles of the color of the plate on the back. Obstacle robots are colored black for recognition. Figure 10 shows displays from this simulation. At left are the initial starting locations as per the IARC Mission 7a rules and at right are the locations of the robots after a minute.



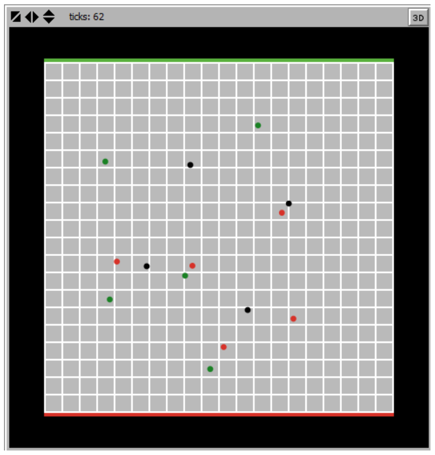


Figure 10. Simulated ground robot locations at T = 0 s (left) and T = 60 s (right)

Given calculated probabilities for the heading deviations, the tracking algorithm implemented can have a predictive quality to determine the next location of the robot in a given time period. The ground robots move at a speed of 0.33 meters per second. Collision and rotation probabilities, or the probability the robot would stay in the same location as the second before, are insignificant with a probability of less than 5%. The expectation of a forward path is to be assumed when the robot is considered standalone. As more robots enter the field of view, conditional probabilities can be calculated to determine the likelihood of collision.

[6.d] Testing

Flight testing lends itself to an academic lab environment, since testing can occur indoors in cluttered environments. Large outdoor flight test areas are not required.

[7] CONCLUSION

The Pima Community College UAV Club has designed an air vehicle system to herd a group of 10 ground-based robots in a predefined arena, while simultaneously avoiding a second group of 4 obstacle robots. The vehicle uses machine vision as well as lidar and sonar scans to sense its environment.

Acknowledgements – the team wishes to thank Chien-Wei Han for his support of the project as faculty advisor. We're also grateful to Frank Oliver for mentoring our University of Arizona student contingent. Thanks also to Jerry Petrey for his invaluable help in getting our ARM board up and running with a bare-metal Ada 2012 compiler.

[8] REFERENCES

[1] Official Rules for the International Aerial Robotics Competition MISSION 7, Version 12.1, May 2016, http://www.aerialroboticscompetition.org/downloads/mission7rules_051016.pdf

[2] Guzman, Stephanie, et. al., PCC's Autonomous Air Vehicle System for IARC 2016, 1 June 2016. Most of this paper was copied verbatim except for incremental changes due to improved system design.

[3] https://github.com/AdaCore/Certyflie

[4] Guzman, Stephanie, et. al., Autonomous Aerial Tracking of a Herd of Ground Robots, 4 May 2016, Team #15041, Pima Community College UAV Club. Sections of this paper were copied verbatim.

[5] O. Songhwai, et. al., Markov Chain Monte Carlo Data Association for Multiple-Target Tracking, IEEE Trans. Autom. Control, 54(3): 481-497, 2009.