AIAA Guidance, Navigation, and Control Conference
10 - 13 August 2009, Chicago, Illinois

AIAA 2009-5658

Indoor Navigation for Unmanned Aerial Vehicles

D. Michael Sobers, Jr.[*][†] Girish Chowdhary[‡] and Eric N. Johnson[§]

Georgia Institute of Technology, Atlanta, GA, 30332-0152, USA

The ability for vehicles to navigate unknown environments is critical for autonomous operation. Mapping of a vehicle's environment and self-localization within that environment are especially difficult for an Unmanned Aerial Vehicle (UAV) due to the complexity of UAV attitude and motion dynamics, as well as interference from external influences such as wind. By using a stable vehicle platform and taking advantage of the geometric structure typical of most indoor environments, the complexity of the localization and mapping problem can be reduced. Interior wall and obstacle location can be measured using low-cost range sensors. Relative vehicle location within the mapped environment can then be determined. By alternating between mapping and localization, a vehicle can explore its environment autonomously. This paper examines available low-cost range sensors for suitability in solving the mapping and localization problem. A control system and navigation algorithm are developed to perform mapping of indoor environments and localization. Simulation and experimental results are provided to determine feasibility of the proposed approach to indoor navigation.

I. Introduction

Autonomous mobile robots that can effectively navigate unknown environments could be utilized for a wide range of applications, including search and rescue, disaster assessment, reconnaissance, or other tasks that would be risky or impossible for a human to perform. The problem of localization and mapping unknown environments is typically solved by using a scanning laser or other type of rangefinder to measure the environment.[1] This approach has seen success for large- to medium-sized ground robots. For UAVs, traditional localization and mapping often Global Positioning System (GPS) measurements to estimate vehicle position. Recent advances have also been made in using vision sensors for target tracking and obstacle avoidance[2] or to estimate vehicle pose.[3] These techniques have been tested using large outdoor platforms, such as the GT Max, which is based on the Yamaha R-Max UAV helicopter (see Figure 1). For sub-kilogram indoor flying vehicles, however, these sensor suites are less effective. For example, GPS signals will likely be unavailable, and laser scanners are relatively heavy and expensive. By utilizing a stable flying platform and relying on the structured nature of most indoor environments, it is possible to simplify the localization and mapping problem to the point where lightweight, inexpensive range sensors can accomplish the task. In addition, the implementation of simple altitude-hold control reduces the complexity of mapping and localization for UAVs to essentially a two-dimensional problem.

II. Vehicle and Sensors

A. Aerial Platform

The choice of platform for indoor flight is influenced by several factors. Airspace is typically limited, so traditional fixed-wing platforms have a disadvantage. Most rotorcraft are unstable in flight and must have an inertial measurement unit (IMU) and some additional method to estimate velocity or attitude in order

[*] Major, USAF; Graduate Research Assistant, School of Aerospace Engineering, AIAA Student Member.

[†] The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

[‡] Graduate Research Assistant, School of Aerospace Engineering, AIAA Student Member.

[§] Lockheed Martin Associate Professor of Avionics Integration, School of Aerospace Engineering, AIAA Member.

Figure 1: The GT Max UAV performing vision-based target tracking and obstacle avoidance using a weather balloon.

to be effectively stabilized. Single-rotor helicopters and quad-rotor designs fall into this category, where maintaining stable flight without GPS or an external attitude estimation system is itself a challenge. For this reason, only stable platforms were considered in this research. Lighter-than-air craft were not considered feasible due to their limited payload capacity and poor maneuverability. The desire for a stable, commercially available vehicle with payload capacity resulted in the choice of a coaxial helicopter for the sensor platform. The coaxial helicopter has a pair of counter-rotating blades, making the vehicle more compact since no tail rotor is required for yaw control. The bottom set of blades has cyclic control for maneuvering, while the upper set of blades has a Bell stabilizer (sometimes called a flybar) to counteract vehicle pitch and roll, providing some attitude stability. Several manufacturers make radio-controlled coaxial helicopters of various sizes. The helicopter selected for this research was the E020 Big Lama,[5] made by E-Sky® (see Figure 2). It has a rotor diameter of 46cm and weighs approximately 410g in its stock configuration. Initial flight tests indicated that the stock vehicle has a useful payload capacity of approximately 50g. Removal of the canopy and motor upgrades significantly increased the available payload capacity. The final flight configuration, with avionics, a larger battery, brushless motors, and a protective shroud weighed 605g.

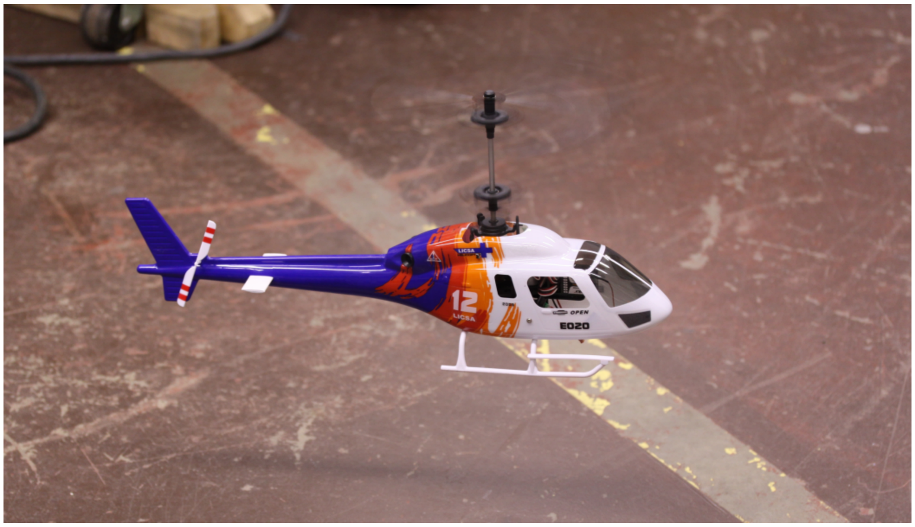
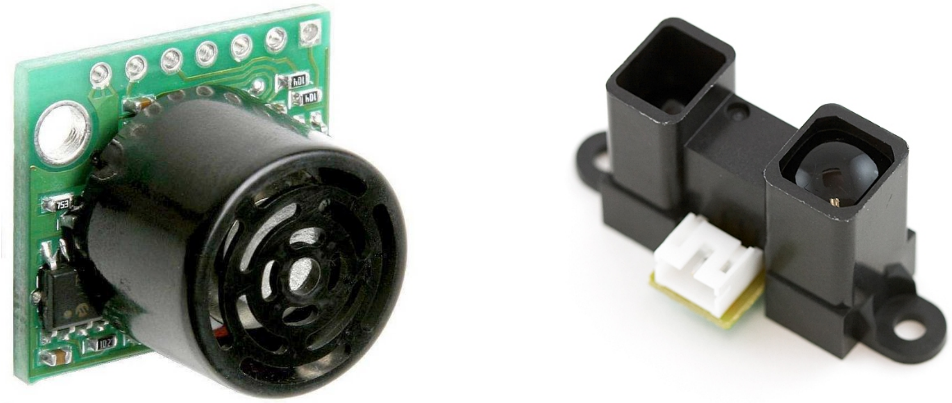


Figure 2: The E-Sky® Big Lama. Photo courtesy Allen Wu. *Note: the tail rotor on this aircraft is neither functional nor required.*

B. Sensor Selection

Several range sensors were selected based on their range and resolution, with an emphasis on low-cost commercially available sensors. The MaxBotix® LV-MaxSonar® was chosen for the primary mapping and localization tasks (see Figure 3a). It has a narrow beam width compared to typical sonar range finders, and its maximum range characteristics make it a good choice for measuring overall room dimensions and locating obstacles and doorways. The SHARP GP2Y0A02YK0F infrared sensor was selected for close obstacle avoidance during forward flight and for wall-following navigation modes. It has a shorter range than the MaxSonar® , but it has a better resolution and a narrow beam width. In addition, multiple IR sensors can be operated at the same time with less interference than sonar. See Table 1 for detailed information on the sensors. A gyro is also included in the sensor package to measure yaw rate, which is integrated to provide relative angular displacement between range measurements. The selected gyro has a range of ± 300 degrees per second and weighs approximately 2g.



(a) MaxBotix® LV-MaxSonar®

(b) SHARP GP2Y0A02YK0F Infrared Sensor

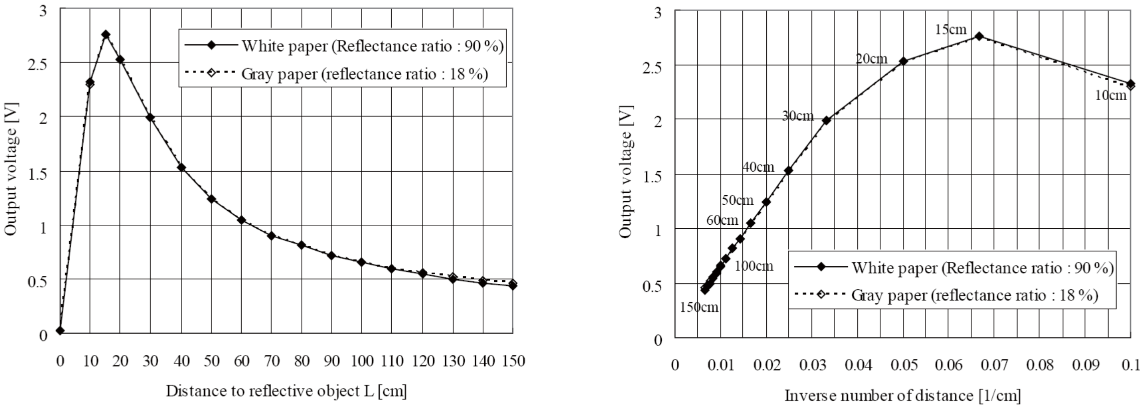
Figure 3: Lightweight, low-cost range sensors suitable for indoor navigation. Photos courtesy Sparkfun™ Electronics.[6]

Table 1: Range Sensor Manufacturer Specifications

Sensor	MaxSonar®[7]	SHARP IR[8]
Range	0.15-6.45 m	0.2-1.5 m
Resolution	2.54 cm	1 cm
Weight	4.3 g	4.8 g

C. Range Sensor Characterization

Each sensor was tested to determine the useful range and error characteristics. The SHARP infrared sensor has an analog output, and the correlation between output voltage and range is nonlinear. However, the reciprocal relationship is linear over most of the useful range of the sensor. Figure 4 shows the calibration curves provided by the manufacturer.[8] Actual voltage/range relationships were measured for two IR sensors and were comparable to the specification. In addition to determining the calibration curves, the sensors were each sampled for 10 seconds at different fixed ranges to determine the standard deviation. The analog output voltage is updated approximately every 38ms. The MaxBotix® sonar range finders output data via serial RS232 protocol, a Pulse Width (PW) signal, or analog voltage output. For initial testing, the serial output was used. During flight tests, the PW signal was also used. For the PW signal, the sonar outputs a logical high signal that has a width of 147 microseconds per inch. Each measurement cycle for the sonar takes approximately 50ms. Operating multiple sonar at the same time can cause interference in some circumstances. However, the sonar can be operated in a chain, with each sensor triggering the next one automatically. Alternatively, the sonar can be triggered individually at specified times to avoid interference. The next sections provide further insight into the sensor characterization and data collection process for the IR and Sonar range sensors.



(a) SHARP IR range/voltage response curve.

(b) SHARP IR inverse range response curve.

Figure 4: SHARP GP2Y0A02YK0F infrared range sensor response curves.[8]

1. *IR Sensor Testing and Calibration*

Two identical IR sensors were tested to determine whether each sensor would require its own calibration curve. Initial testing consisted of collecting data from the range sensors at a fixed distance for approximately ten seconds. The analog output voltage was measured using a 10-bit analog-to-digital (AD) converter. The output voltage was then converted to range using the manufacturer's calibration curve. Figure 5 shows the result for a white paper target at a distance of 36cm. The data show about 90% of the measurements have a mean near the actual range, while approximately 10% are 1cm short of the actual range, with some sparse outliers reading as much as 4-5cm short. The sensor performed well, with a mean value of 35.91cm and a standard deviation of 0.41cm. However, the pattern observed in 10% of the data prompted a more in-depth analysis of the voltage output from the sensor.

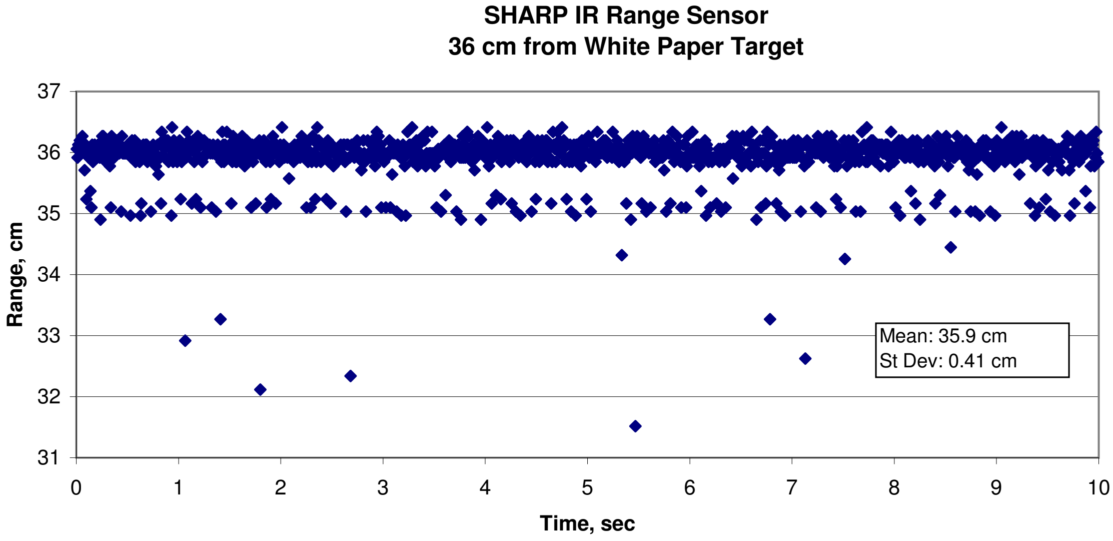


Figure 5: Initial test of the SHARP IR Range Sensor. The test was conducted at 36 cm from the target (white paper) for 10 seconds.

The IR sensor was connected to an oscilloscope to determine the characteristics of the output voltage. As shown in Figure 6, the noise is periodic, consisting of a region approximately 0.13ms long with a period of 1.1ms. In this region, the output voltage is higher than the rest of the data, resulting in a shorter reported range. The time scale on the oscilloscope was reduced to record the output more accurately, and a voltage spike was observed at the beginning and end of the elevated reading (see Figure 7). It was determined that the characteristics of the noise fully account for the observed range measurement data. The 10% grouping of short range readings correlate to the elevated voltage output, while the outliers correlate to voltage recorded during one of the spikes seen at the beginning and end of the elevated region. As a result, a median technique was used to filter the data, whereby three data points were collected with the middle value of the three reported as the range. Using this filtering technique, the range measurement was greatly improved as shown in Figure 8. After filtering the data, the mean range reading was 36.00cm, with a standard deviation of 0.11cm. The three-point median filter described above was used on all subsequent IR measurements during initial testing as well as during the flight tests.

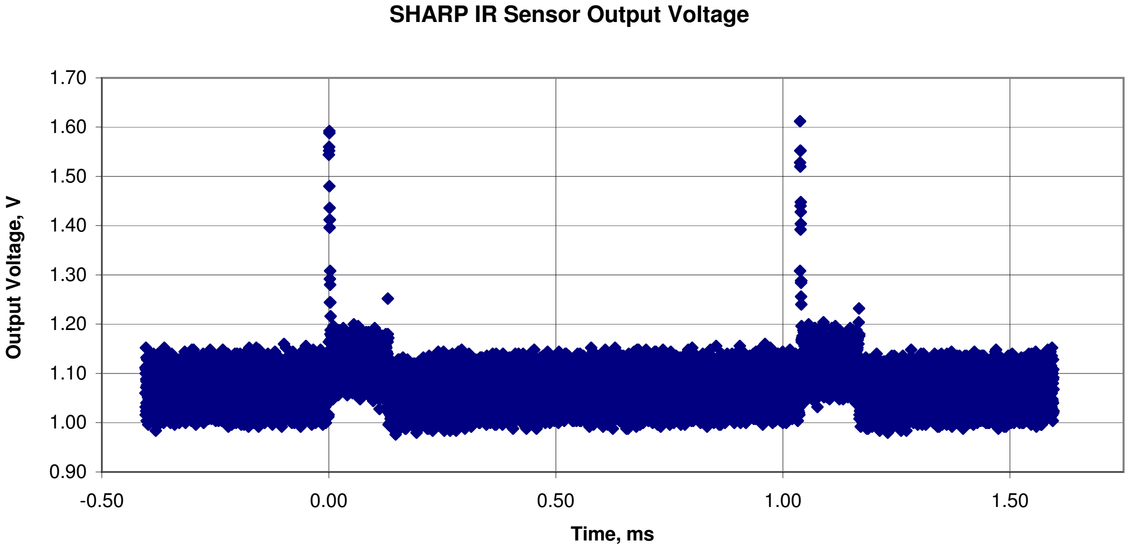


Figure 6: SHARP IR sensor output voltage. Note the periodic nature of the sensor noise.

Next, the two IR sensors were tested at fixed intervals from a white paper target to determine the voltage-to-range relationship. The sensors were mounted to a tripod, and each sensor was operated independently to avoid interference. Then, output voltage was recorded for ten seconds each at 10 cm intervals from 20cm to 190cm. The output voltage was filtered before being recorded using the filtering algorithm described above. The data were averaged for the ten second sample, and the voltage for each sensor was plotted versus target distance. Figure 9 shows the voltage/range curve for the sensors tested. Although the sensor response is nonlinear, a graph of the voltage versus inverse distance shows some linear regions. Three linear regions were identified (see Figure 10) and linear approximations were used for these regions to calibrate the sensors. Using linear approximations for the relationship over these three regions produced an error of less than 6% over the manufacturer's advertised range of 20-150cm. The sensors may be used up to 190 cm under some circumstances, with calibration error of less than 10% (see Figure 11). The standard deviation was also calculated for the two sensors using data collected over the 10-second period at each distance interval (see Figure 12). Subsequent testing of a different batch of IR sensors showed that a single linear approximation can be used for the entire advertised range of the sensor.

After calibrating the SHARP IR sensors, they were both mounted to the test vehicle for initial flight testing. Due to mounting constraints, the initial test was done with the sensors mounted side-by-side. In practice, the sensors should be placed so that their illumination patterns are at least 20 cm apart to avoid interference (the beam width of each sensor is approximately 10cm at maximum range). During the initial flight test, the vehicle was flown at varying altitude while data from the two sensors was recorded. Figure 13 shows the range measurements recorded from both range sensors. Note that due to the nonlinear response of the sensor, distances below the minimum range appear higher than normal. For this test, range measurements after about eight seconds into the flight are valid. The two sensors performed nearly identically over most

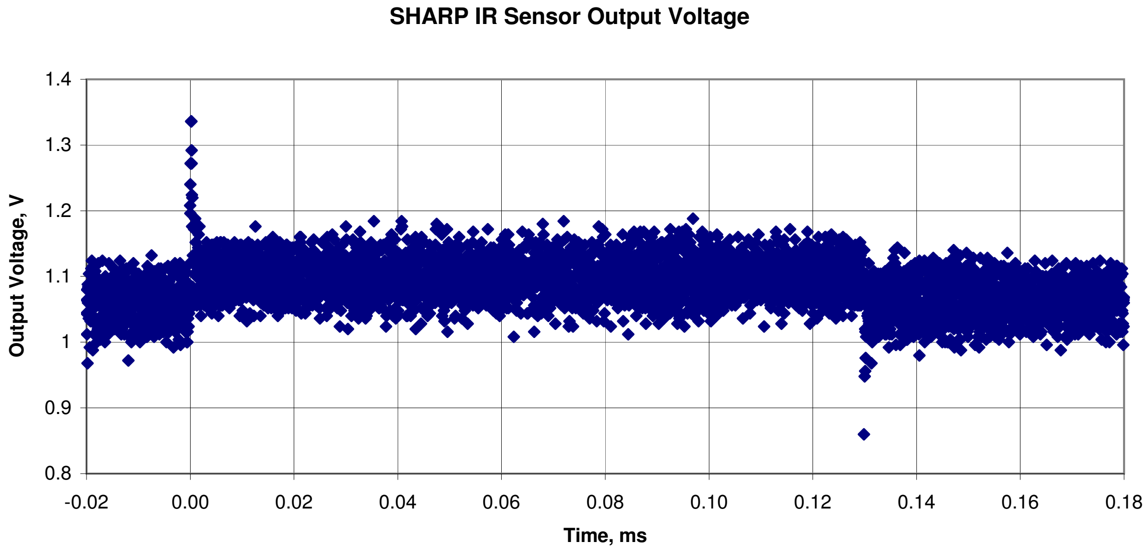


Figure 7: SHARP IR sensor output voltage, higher resolution capture. Note the spike before and after the elevated voltage region.

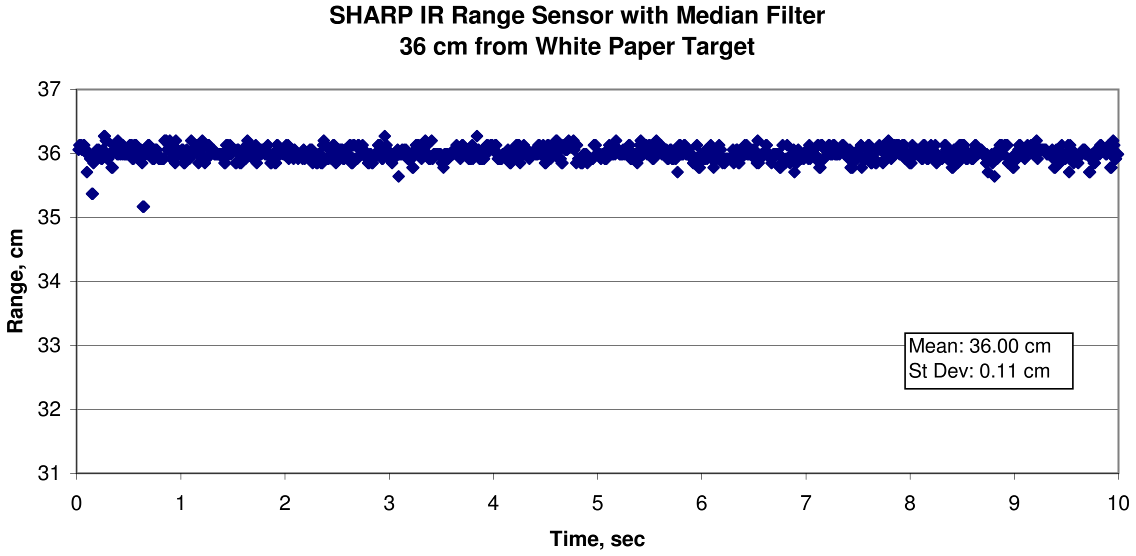


Figure 8: SHARP IR sensor output voltage using three-point median filter. Note the improvement in the mean and standard deviation.

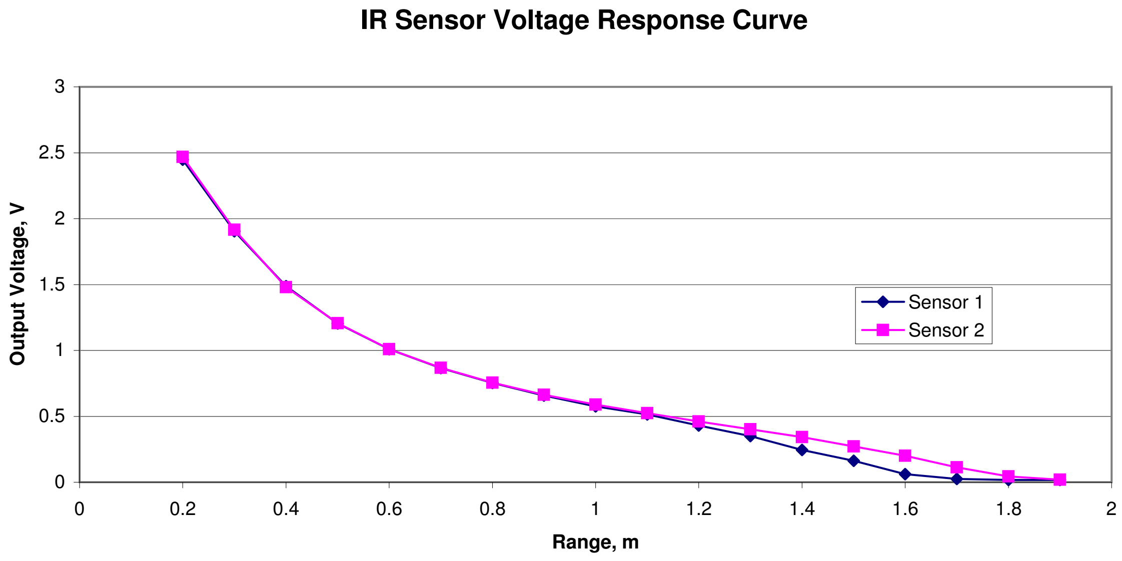


Figure 9: SHARP IR sensor voltage response curve. Each data point represents the average over a ten-second period, filtered as described above.

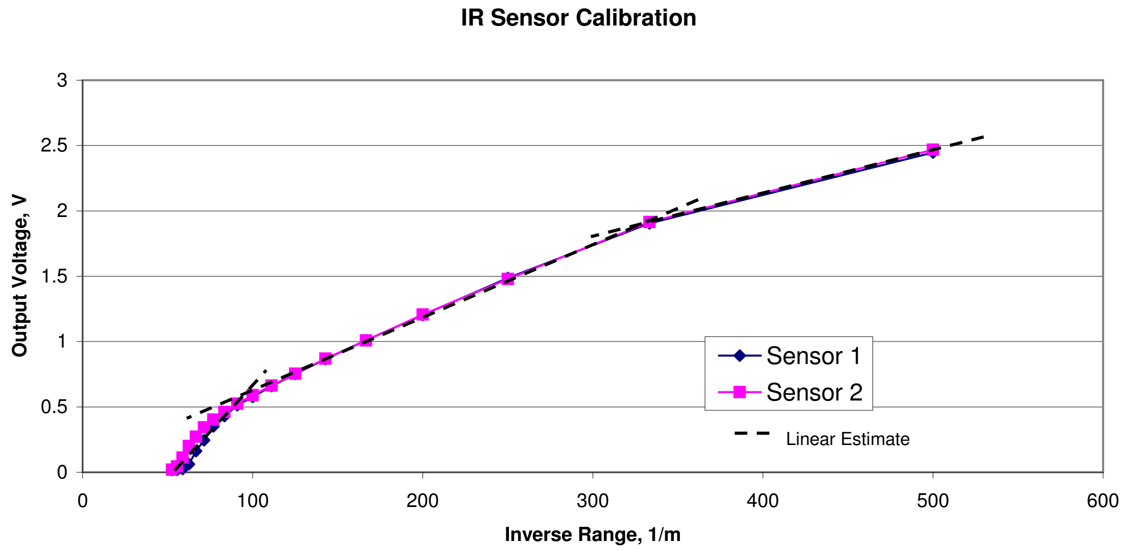


Figure 10: SHARP IR sensor calibration curves are nearly linear for the inverse range. The data were divided into three discrete linear regions.

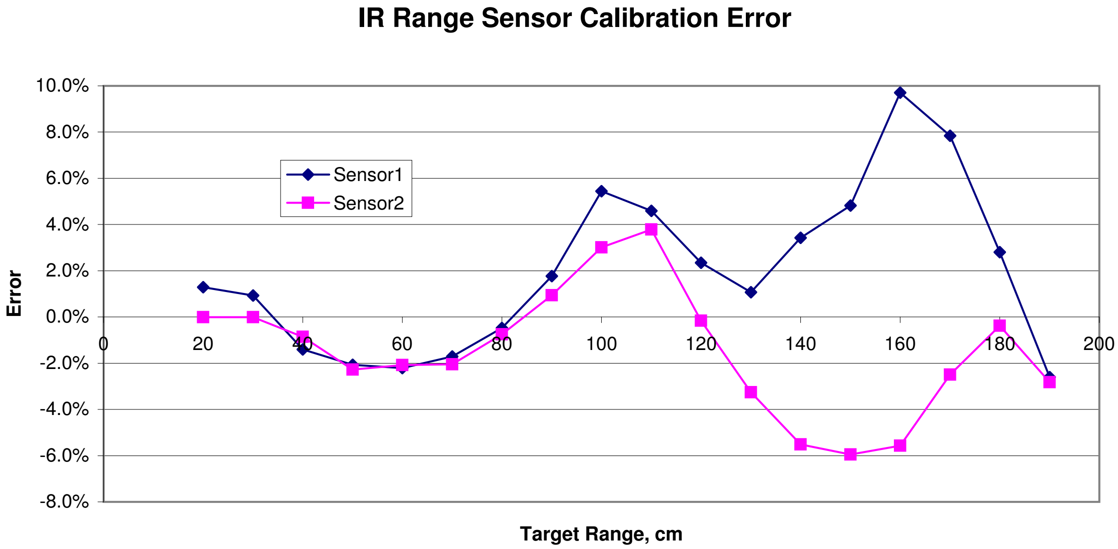


Figure 11: SHARP IR sensor calibration error. Three linear approximations are used for the voltage/inverse range relationship. Note: The manufacturer's advertised sensor measurement range is 20 - 150 cm.

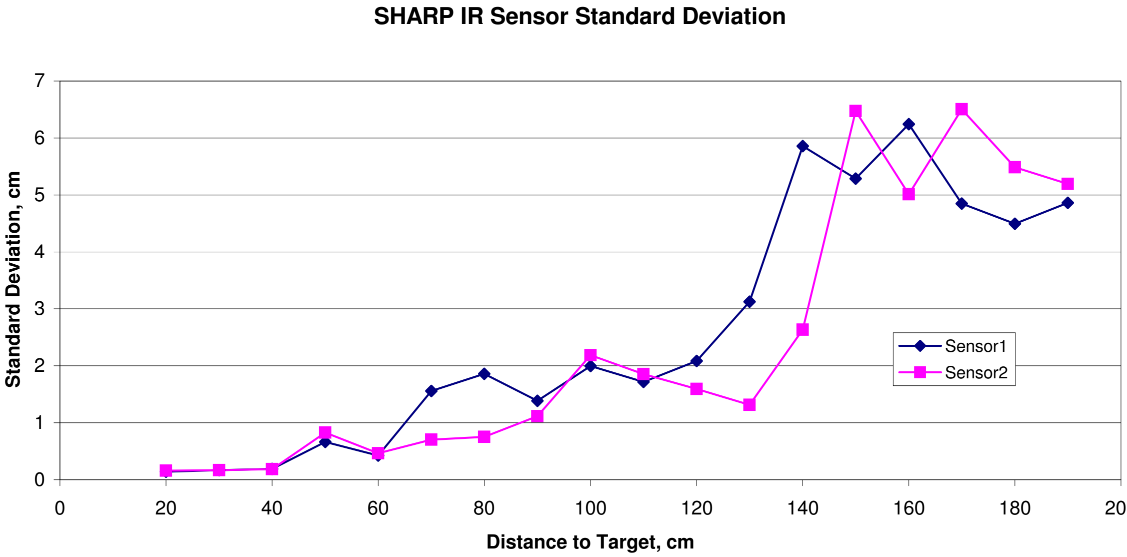


Figure 12: SHARP IR sensor standard deviation over measurement range. Two sensors were measured independently for ten seconds at each distance interval.

of their design range, however the measurements were quite noisy above approximately 120cm. This may have been due to interference between the sensors, which were operating simultaneously, since the standard deviation at this range was measured to be approximately 2cm for independent sensor operation.

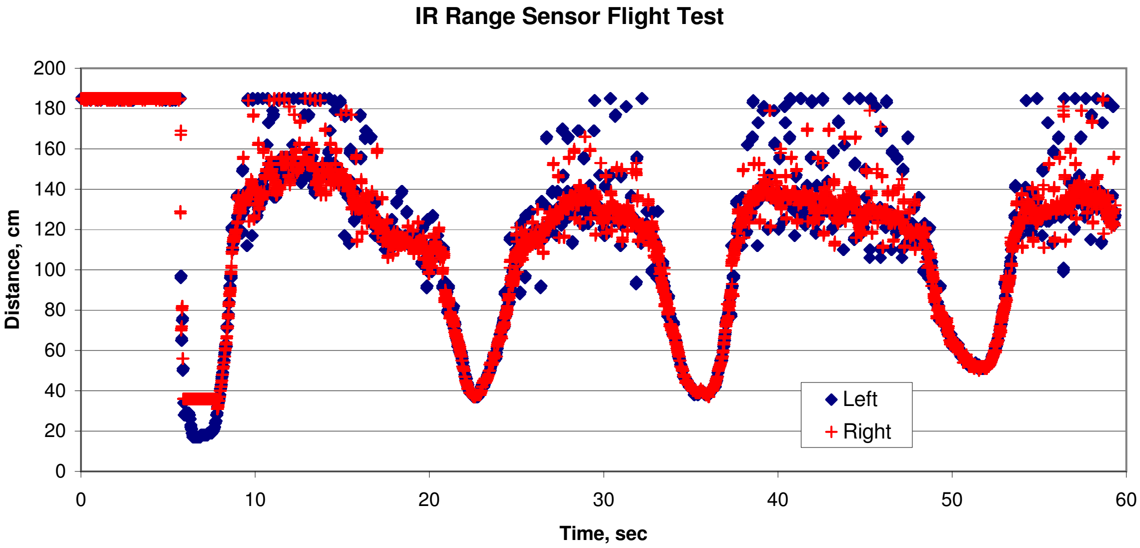


Figure 13: SHARP IR sensor flight test. Two sensors were placed side-by-side and aimed at the ground. During the flight, the vehicle was flown to different altitudes while recording data.

2. Sonar Testing

Two ultrasonic range sensors were tested to determine which is better suited to indoor navigation. The MaxBotix® LV-MaxSonar® -EZ1™ and LV-MaxSonar® -EZ4™ sensors are both inexpensive and lightweight, making them possible candidates for small UAVs. The two sensors are nearly identical, but the EZ4™ has a narrower beam width than the EZ1™ . Initial comparisons consisted of powering the sensors and reading the data via the RS-232 serial connection. The sensors were then aimed at targets at various ranges to determine qualitative difference between the two. Both sensors produce a digital output, and showed no noise when pointed at a fixed, flat target. In a cluttered environment, however, small movements of the sensors produced "noisy" range readings due to varying strength of the sonar return from different objects. The sensors are designed to return the range to the first object detected. This can produce varying results in a cluttered, moving environment. The EZ4™ showed more noise in cluttered environments than the EZ1™ , possibly due to its narrower beam hitting various objects of differing ranges when the sensor is in motion.

Once qualitative analysis was complete, the sensors were mounted to a tripod and tested in a more controlled environment. The tripod was set in the middle of a small room with the sensor platform 122cm above the ground, 173cm from the left and right walls, and 198cm from the front wall. The sensors were turned through a range of 180° beginning facing the right wall, turning counterclockwise to face the front wall, and finishing with the left wall. Each test was completed independently to prevent possible interference between the sensors. The sensors were turned by hand, though an attempt was made to ensure a fairly constant angular rate. Next, the sensors were operated simultaneously with similar results.

The range measurements of the initial test is shown in Figure 14. The results, similar for both sensors, were somewhat unexpected. When facing a flat wall, the sensors read the perpendicular distance to the wall, regardless of sensor orientation. The result is a reading which remains constant, until the sensor is turned past an angle where the wall is no longer detected. As shown in the figure below, a graph of the range readings over time (at an approximately constant angular rate) show a series of straight lines, with some features such as corners and doorways observed. The horizontal lines correspond to the perpendicular distance from the sensor to major feature points in the room. The first feature detected during the sensor sweep was the right wall at 173cm. As the angle increased, the sensors began to lose range measurement to the right wall and pick up the front right corner at 264cm. The EZ4™ sensor picked up the corner first, while the EZ1™ sensor with the wider beam maintained a reading to the right wall longer before picking up the corner. As the sensor angle was increased, both sensors eventually picked up the front wall, which was at a distance of 198cm. The reading remained fairly constant until the sensors were turned far enough to pick up an open doorway in the front left corner. As the sensors were rotated further, they both eventually picked up the left wall at 173cm.

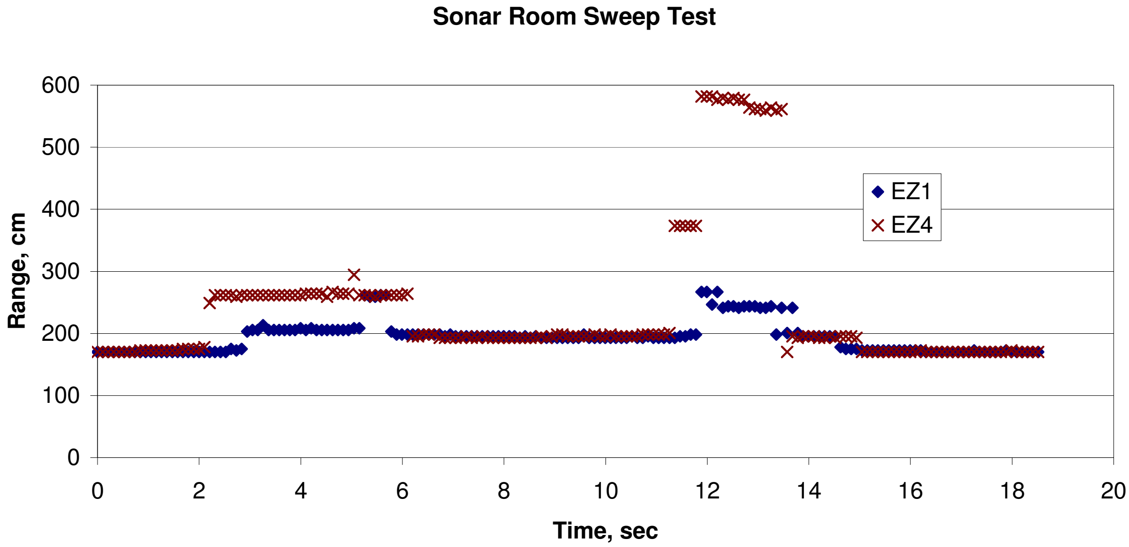


Figure 14: Distance measurements recorded using the EZ1™ and EZ4™ in simultaneous operation during a 180° sweep of a small room. Note the straight lines observed that correlate to the distance to the right, front, and left walls at 173cm, 198cm, and 173cm respectively. Some features such as corners and an open doorway are also observed.

Although the sensor orientation was not measured directly, the sensor platform was turned at a reasonably constant angular rate, and the time for each data point was recorded. As a result, the range data can be plotted on a 2-D map using the estimated angle information (see Figure 15). Although the sonar returns do not map to the wall locations as expected, some features can be identified from the 2-D mapping. The arc that is visible from major geographic features represents the sonar wave pulse as it expands from the sensor. The sensors report the distance to the first return, which corresponds to the perpendicular distance to the walls and corners. Thus, the center of each visible arc in the 2-D plot represents the location of the sensor within the room. Given room dimensions and accurate heading information, it should still be possible to localize a sonar sensor platform in a given room.

In order to determine room dimensions from observed range data, it is useful to consider the data shown in Figure 14. The straight lines representing distance to major geographical features are easily detected using a histogram approach. Figure 16 shows a histogram of the range data recorded during the test. The data bins are 10cm wide, so data in these bins represent an average value halfway between the bin label and the next lower bin. Looking at the histogram, the prominent features appear at the 180cm and 200cm bins. These represent measurements of approximately 175cm and 195cm, which correspond to the distance to the walls (173cm and 198cm respectively). An even more accurate estimate can be done by averaging the actual range measurements for the data that falls into the highest bins. As long as the data bin size is chosen wisely, however, the proposed method of using the middle of the bin for the measurement estimate is accurate enough for room size estimation.

Further testing has shown that the sonar range sensors return the shortest perpendicular distance to any object or wall detected, regardless of the angle between the sensor and the object. As a result, small changes in sensor angle do not affect the range returned. For this reason, the sonar range sensor is an excellent choice for measuring altitude for the chosen vehicle. Small changes in the pitch or roll angle would cause a downward-pointing line-of-sight sensor (like the IR sensor) to read long, whereas the sonar is unaffected. For this reason, the MaxBotix® LV-MaxSonar®-EZ1™ was chosen for altitude measurement during the flight tests. These initial findings indicate that the sonar have uniquely different properties than the IR sensors. The combined properties of the IR and Sonar range sensors work together to improve overall mapping and localization.

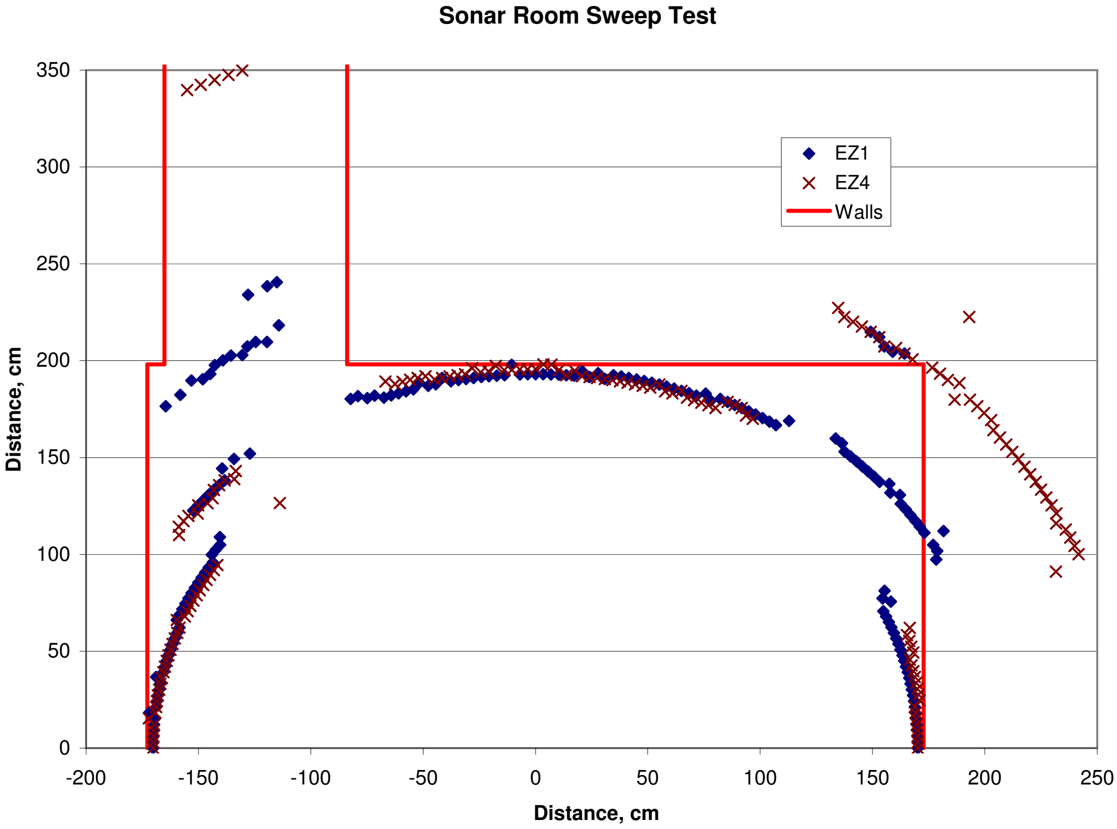


Figure 15: Distance measurements plotted using estimated sensor angle and measured range. Actual wall locations are show for reference. Note the EZ4™ more easily detects corners and open doorways than the EZ1™.

D. Sensor Integration

An ATMega128 onboard microprocessor is used to read the sensors and process the data for navigation. The data from these sensors is processed and filtered appropriately according to their error characteristics for simple obstacle avoidance or wall-following behavior, or more complex mapping and localization tasks. The IR range sensors have analog voltage outputs, which are read by onboard analog to digital input channels. One sonar, used for altitude, is read via serial port. The ATMega128 has one additional serial port, which is used for a data link. Four sonar range sensors are arranged looking to the front, back, left and right of the aircraft as depicted in Figure 17. These sensors are read using the pulse-width method mentioned above via an interrupt pin on the ATMega128. They are used to measure the room dimensions and provide relative position information. In addition to the sonar, two infrared sensors are placed 45cm apart looking forward for heading control, improved obstacle avoidance during forward motion, for detection of openings such as windows or doors, and for wall-following behavior. Figure 18 shows the sensors mounted to the test vehicle.

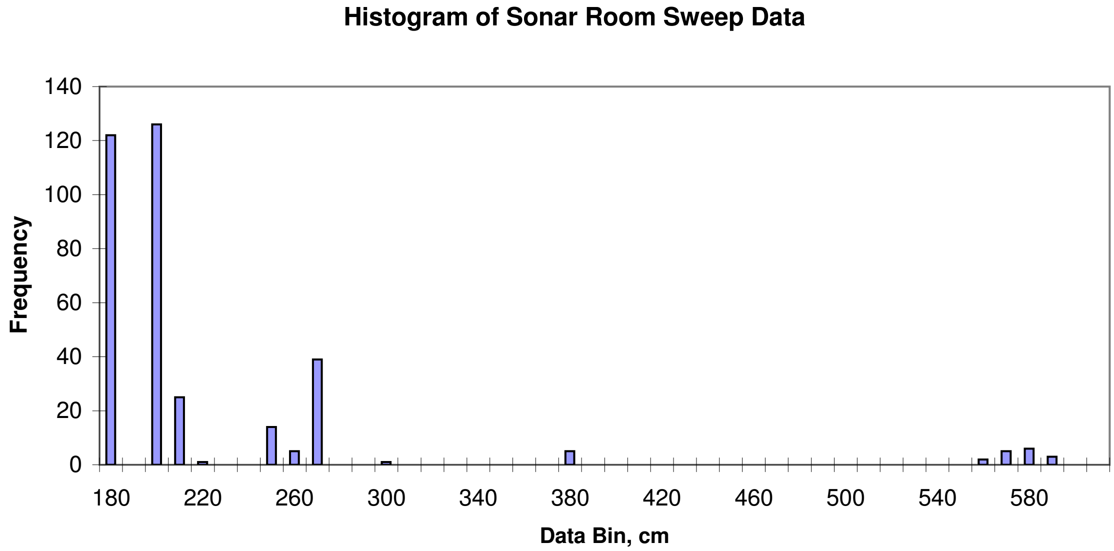


Figure 16: This histogram shows the frequency of measurements in each 10cm bin. The highest peaks represent room measurements of 175cm (average of 170-180cm bin) and 195cm (average of 190-200cm bin). Actual room measurements are 173cm and 198cm, respectively.

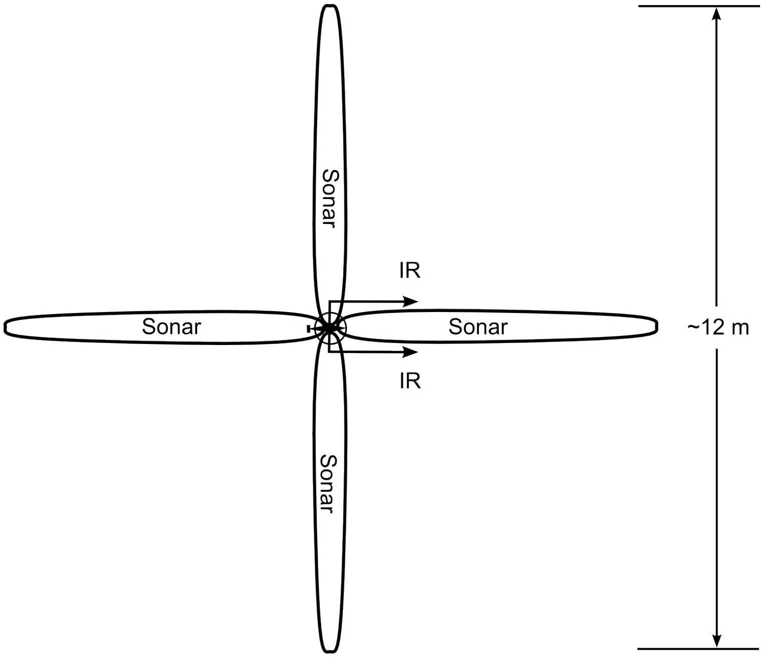


Figure 17: Range sensor layout.

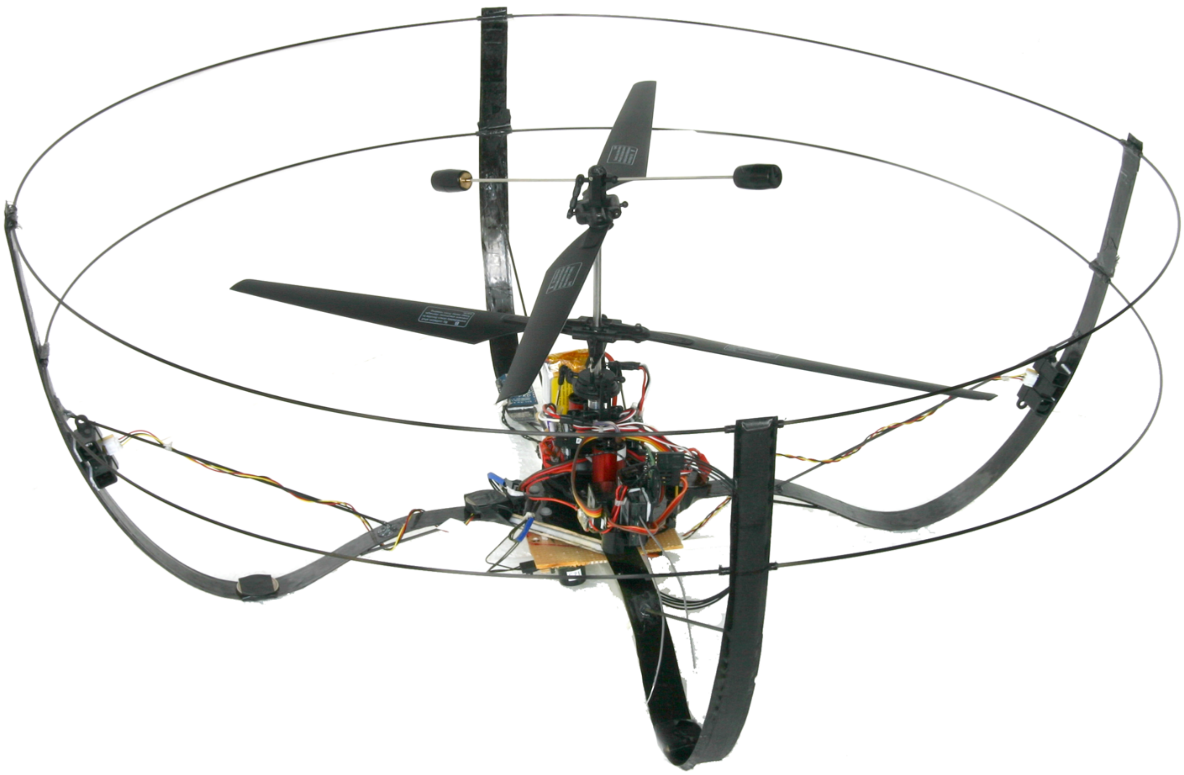


Figure 18: Flight vehicle with sensors and protective shroud installed.

III. Navigation Algorithm

Mapping and localization in complex unknown environments can be difficult, even with sophisticated range sensors. This research focuses on the use of low-cost commercially available solutions for range sensing, so additional care must be taken to ensure the data recorded are useful for navigation. Both the IR and sonar range sensors occasionally reported readings near maximum range when the sensor failed to detect a wall or obstacle at a particular instant. These and other outliers can cause catastrophic results for a feedback control system that expects smooth error measurements. For this reason, a smart filtering routine was developed to prevent large step or impulse inputs from adversely affecting the controller.[9] A Kalman filter is used to estimate the range, and the covariance of the residuals is used to detect and ignore outliers beyond three standard deviations. Occasionally an actual discontinuity in range occurs, such as flight over an obstacle on the ground. The smart filtering routine recognizes such events and adjusts the range estimate to match the new measurements without changing the velocity estimate. Figure 19 shows an example of the smart filter in operation.

Once a reasonable range estimate is calculated, mapping and localization becomes possible. To use the range data collected by these low-cost IR and sonar sensors, additional assumptions are made to simplify the problem. By assuming that rooms are rectangular in shape and aligned to some Cartesian reference frame, the mapping problem is significantly reduced. The traditional method of measuring, recording, remeasuring, and adjusting map boundaries produces tens of thousands of data points, requiring extensive processing power and memory to be carried onboard the aircraft. By taking advantage of the regular structure exhibited by the majority of indoor environments, rooms can be mapped and stored using a 2-D parametric representation by determining characteristics such as dimension, location of room center, location of navigable openings, and location and radius of obstacles detected. Once initial mapping of a room has been completed, the map is stored and assumed correct while the vehicle remains in the room. The range sensors are then used to provide vehicle position relative to the stored map of the room being explored. Once the initial room has been explored to satisfaction, the vehicle flies through a detected opening into the next room and the process begins again. In this way, multiple rooms are explored with stored coordinates linking their mutual

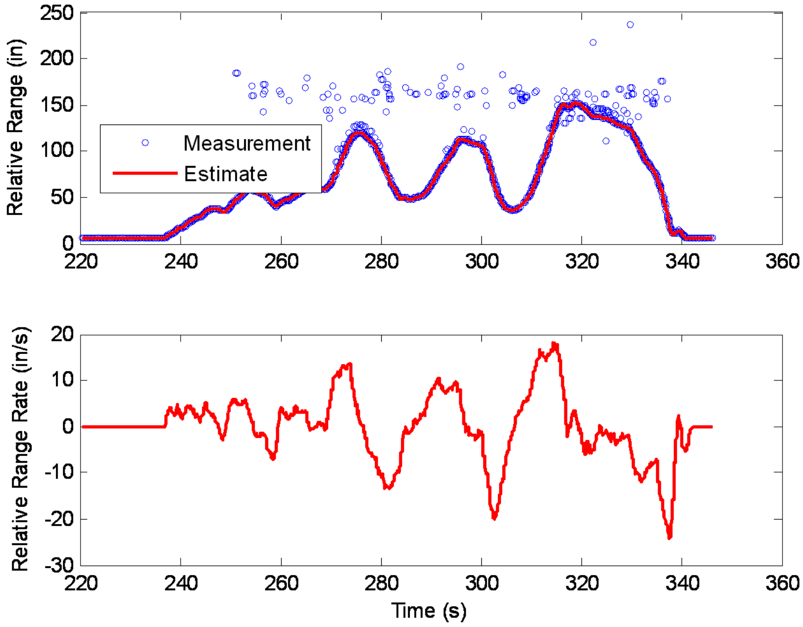


Figure 19: Range measurements recorded by an EZ1™ sonar pointed downward during flight. Note that the smart filtering routine is successfully able to maintain the range estimate in the presence of outliers. The estimated range data was used to calculate the range rate estimate, so no outliers are present in the rate data.[9]

openings, and relative localization within each room is plotted on a global coordinate system based on the current room coordinates. The overall navigation scheme, described in further detail in sections that follow, is summarized in Figure 20.

A. Mapping

The first step in the process of mapping a room begins with the vehicle finding its way to the middle of the room. While this step is not strictly necessary, it is likely to provide the best view of the entire room. In addition, range measurements taken at regular angular increments are more spread out when the measurements are taken from the middle of a room. To find the middle of the room, the vehicle flies to a position where the "front" and "back" sensors read approximately the same range, while the "left" and "right" sensors also measure approximately the same range. The vehicle orientation is not important at this time, so if the above range criteria cannot be suitably met, the vehicle undergoes a small yaw maneuver and tries again. Figure 21 shows how the vehicle uses its range sensors together to find the middle of a room.

Once the vehicle is satisfied that it is near the middle of the room, it performs a 360° yaw maneuver. During the maneuver, the rate gyro data is integrated to provide an angle measurement (relative to the original unknown orientation) for each range datum. The vehicle adds the range measured from the "front" and "back" sensors together to get a total longitudinal measurement, and it adds the range from the "left" and "right" sensors to get a total lateral range measurement. As a result, a full room sweep produces data similar to that shown in Figure 22. After the initial yaw sweep is accomplished, the data is analyzed and room dimensions are determined. For a rectangular room, a regular pattern of local minimum lateral and longitudinal range is clearly visible, repeated every 90°. Obstacles within the room produce short range readings as well, but these can typically be identified by comparing the two paired sensor ranges. In other words, if the two lateral or longitudinal sensors do not read approximately the same range, and obstacle is likely present at the shorter range and the reading at that angle should not be used for room dimensioning. If the vehicle drifts during the yaw maneuver, the error can be detected and corrected by comparing the range measurements identified at 90° intervals. The histogram for this data set is shown in Figure 23.

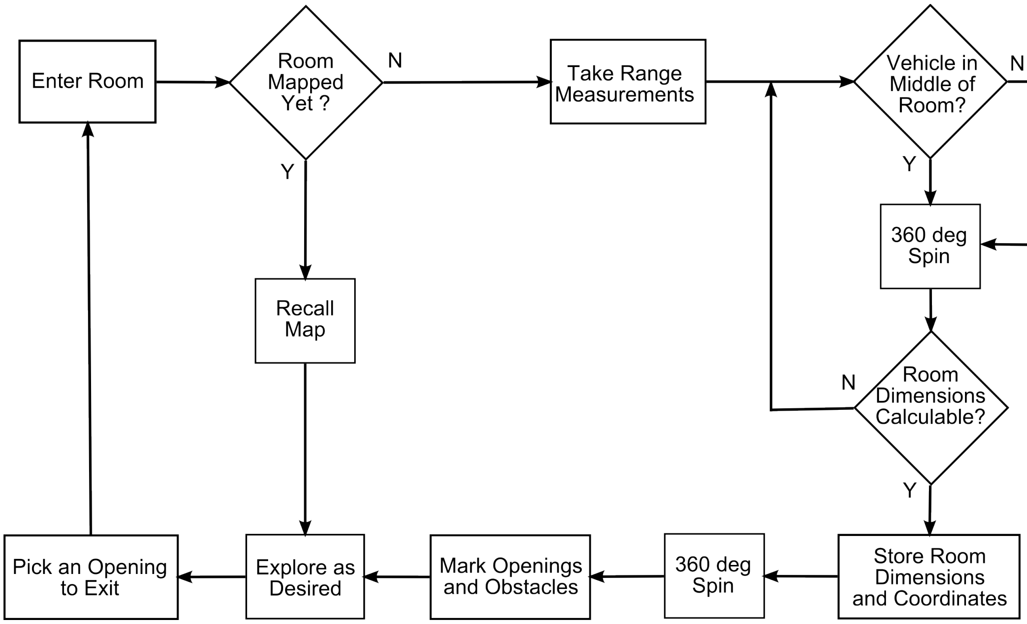
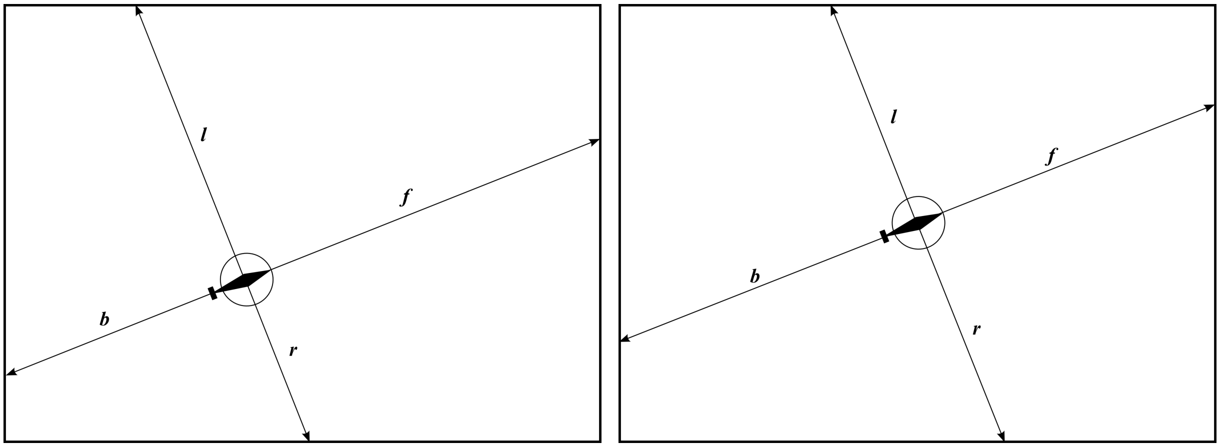


Figure 20: Navigation Scheme Flowchart. If the range information collected from the sensors is inconsistent with the the estimated vehicle position and orientation, the navigation sequence may be repeated as desired.



(a) Initial position

(b) Final position

Figure 21: Moving to the center of the room by attempting to set l=r and f=b.

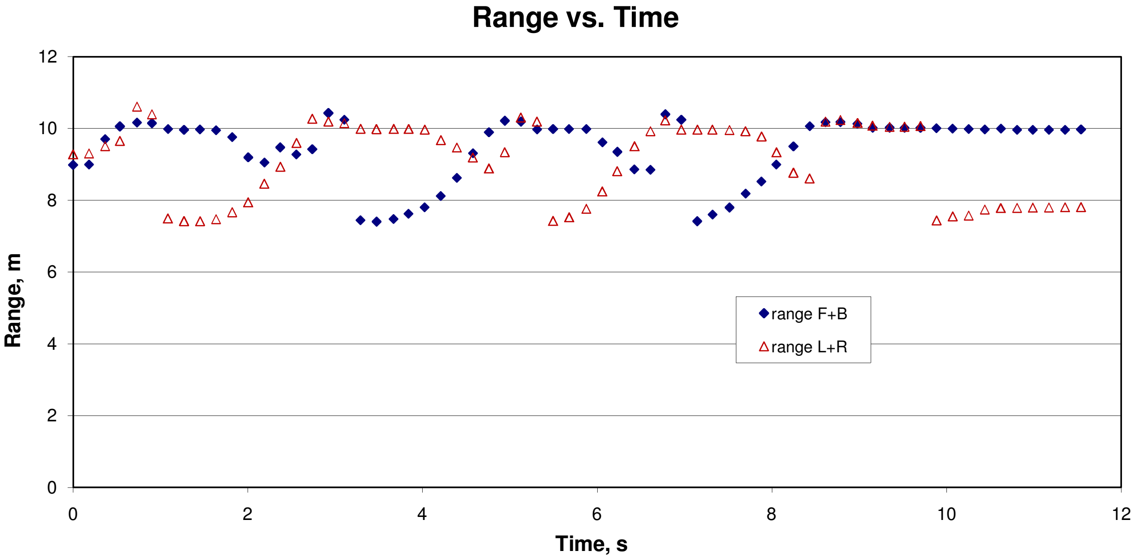


Figure 22: Simulated range measurement data during a room mapping 360° yaw sweep.

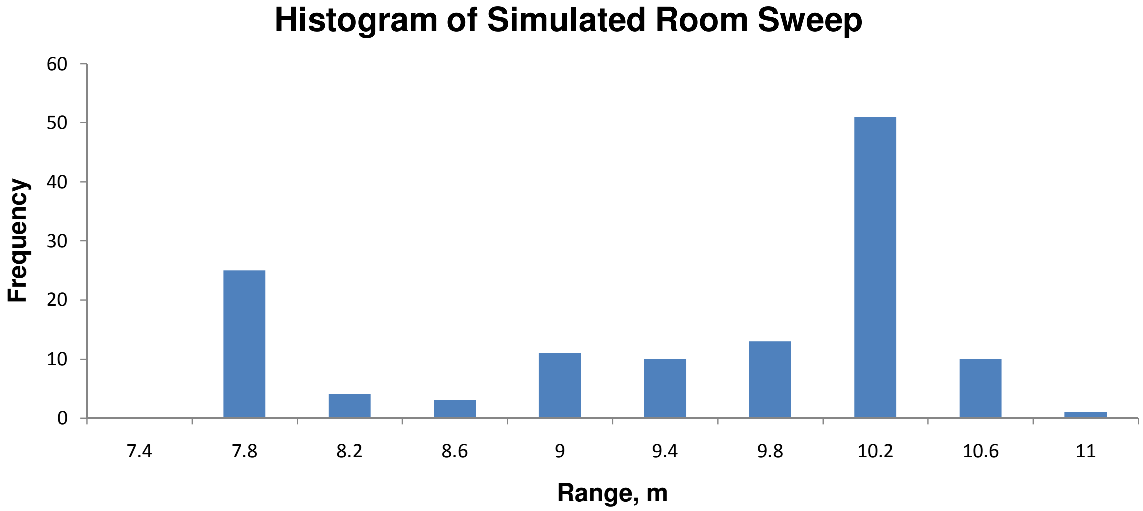


Figure 23: In this histogram of the simulated room measurement, the room dimensions are determined by observing the two data bins with the highest frequencies. In this test, bins 7.4m-7.8m and 9.8m-10.2m have the highest frequencies, corresponding to average measurements of 7.6m and 10.0m. Actual room dimensions were 7.3m and 9.8m.

After the first room has been measured, the origin of a Cartesian reference frame is set at the center of the room. All future measurements can then be referenced to this coordinate system. A data structure is established to store the parameterized room data, which saves significant memory over storing all of the range data recorded (see Table 2). Parameters for the openings and obstacles are stored in substructures with their identifiers incremented based on the number of openings and obstacles detected (see Table 3). In this way, upon re-entering a room, the map can be fully recalled from the simple stored parameters. For example, `Room4.Opening2.Coords` would store the location of the second opening in fourth room explored. Likewise, `Opening2.ConnectsTo` would store the ID of the room adjoined to Room 4 via Opening 2. Obstacle location and size can be recalled in a similar manner.

Once the dimensions for a given room are calculated, another yaw sweep is performed, this time relative to the established coordinate system. The room dimensions are confirmed, and obstacles and potential doorways are located by looking for range readings that are closer or further than expected based on the room dimensions. A sweep of an empty room with two exits was performed in simulation. The data presented in Figure 22 can be plotted using the heading estimate from simulated gyro data. Once the room dimensions are calculated, the room data can be corrected for drift that occurs while the vehicle is performing the yaw sweep maneuver. Figures 24 and 25 show the room mapping plot before and after correcting for vehicle drift. Some rooms encountered may not be strictly rectangular. Rooms shaped similar to the one shown in Figure 26 are identified as two separate rooms, with an opening adjoining them.

Table 2: Data structure for storing room parameters

Label	Data Type
Room ID	string
Room Coordinates (x,y)	(integer,integer)
Room Dimensions (x,y)	(integer,integer)
Number of Obstacles	integer
Number of Openings	integer

Table 3: Data substructures for storing opening and obstacle data

(a) Substructure for openings

Label	Data Type
Opening ID	string
Opening Coordinates (x,y)	(integer,integer)
Opening Width	integer
Room Connected To	string (Room ID)

(b) Substructure for obstacles

Label	Data Type
Obstacle ID	string
Obstacle Coordinates (x,y)	(integer,integer)
Obstacle Size (x,y)	(integer,integer)

B. Localization

Using the room data developed during the mapping stage, the vehicle can use its range sensors to determine its location within a given room. Theoretically, a given set of four range measurements does not determine a unique position within a rectangular room. However, if the vehicle maintains even a crude estimate of its position over time and the vehicle heading is correctly estimated to within ±45°, the possible positions for a given set of range measurements will be narrowed from four down to one. Location of obstacles and openings in the room give further clues to actual vehicle position. In addition to the sonar range finders, the two forward-facing IR sensors can be used to align the vehicle perpendicular to a wall, which allows "scanning" of a room while maintaining a known orientation by traversing from one end of the room to the other and measuring range to the opposite wall. In addition, using the two IR range sensors to maintain heading enables a less computationally expensive exploration routine whereby the vehicle follows a wall from room to room without actually knowing where it is. Employing this technique allows the vehicle to "escape" from a room if it cannot accurately determine its position or the position of openings in the room.

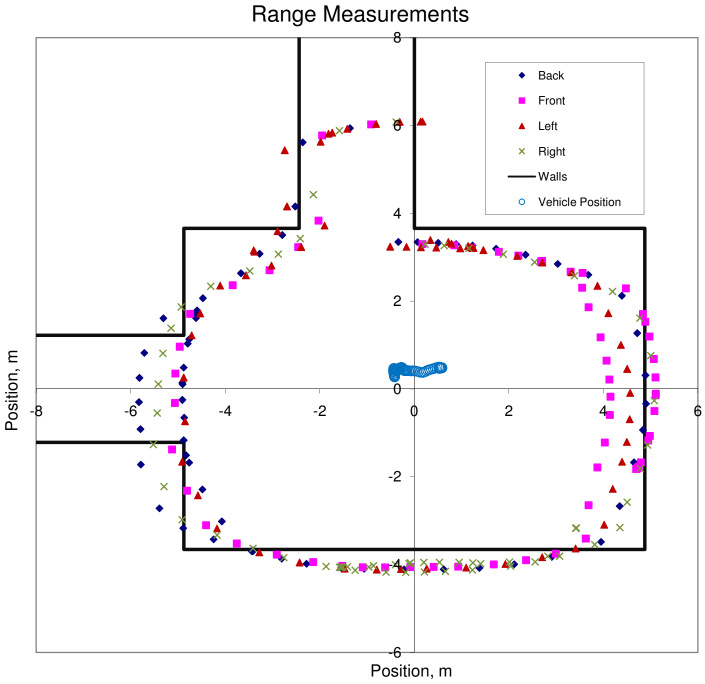


Figure 24: Simulated room sweep. The vehicle is rotated through a heading change of 360° while measuring range to the front, back, left, and right.

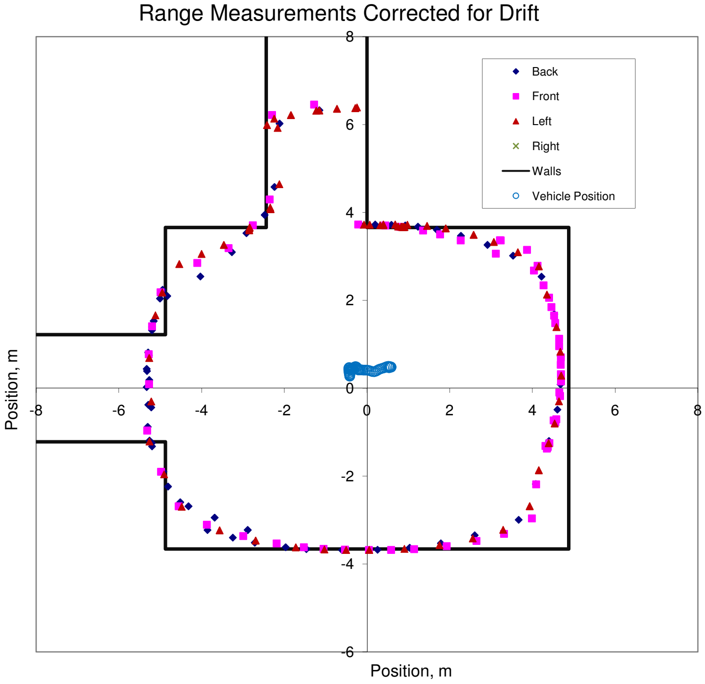


Figure 25: Simulated room sweep. The vehicle is rotated through a heading change of 360° while measuring range to the front, back, left, and right.

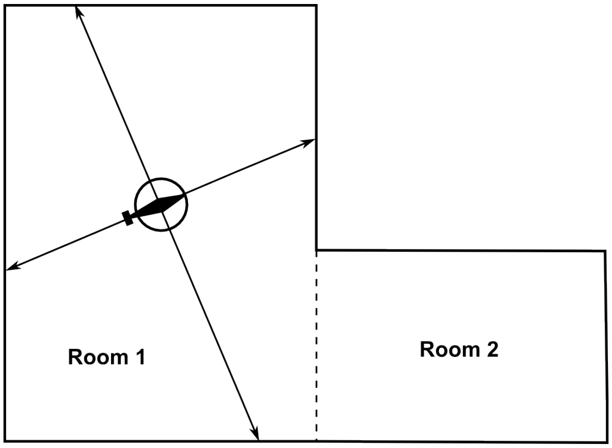


Figure 26: A room that is not rectangular would be broken up logically into two rooms with an adjoining opening between them.

IV. Guidance Algorithm

Several different guidance algorithms are possible using the sensor configuration described above. Possible behaviors range from simple random flight to thorough room-by-room exploration of the entire mapped environment. This research was conducted in preparation for the 2009 International Aerial Robotics Competition (IARC).[10] This competition, held in July 2009, required the autonomous entry into a building mock-up and successful navigation within the building in search of a specified target. Thus, the guidance algorithm flight tested and presented below is tailored to the simplest logic required to achieve mission requirements. Using this simple guidance algorithm allows a simpler navigation scheme as well, whereby the vehicle only needs to determine its altitude, distance to one wall, heading with respect to that wall, and detection of obstacles in the vehicle's flight path. Thus, an altitude-hold controller is used to maintain a fixed altitude throughout the flight, and a wall-following routine is used to explore the competition arena. Once in autonomous mode, the vehicle follows the guidance algorithm described below in Figure 27.

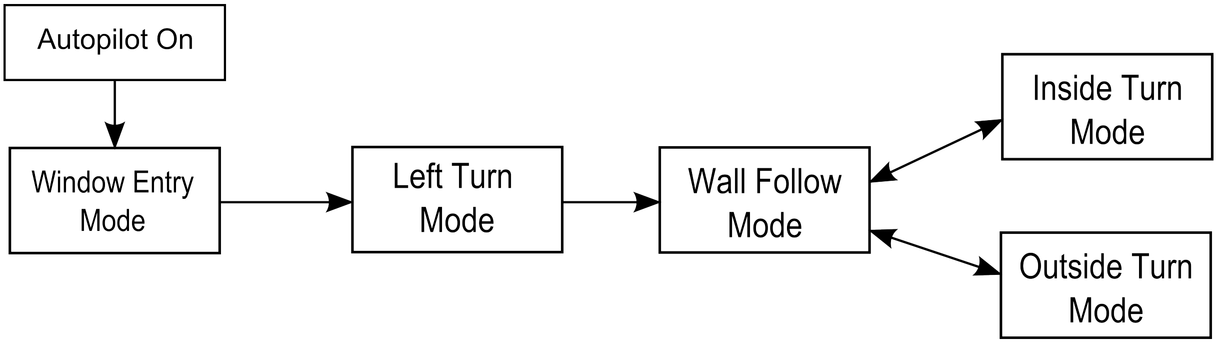


Figure 27: This flow chart shows the logic followed by the wall-following guidance algorithm developed for Georgia Tech's entry in the 2009 International Aerial Robotics Competition. Specific sensor inputs will cause the algorithm to progress to successive logic blocks.

When the autopilot is enabled, the vehicle begins in "Window Entry" mode. In this mode, it flies forward searching for the arena entry portal. If an object is detected on the forward-looking left or right IR sensors, the lateral controller adjusts the flight path so that the vehicle is in the center of the window. Once the vehicle enters the competition arena, walls are detected by the left IR sensor and the vehicle enters "Left Turn" mode. In this mode, it turns to the left until the forward-looking IR sensors detect the wall. Once the forward-looking sensors detect the wall, "Wall Follow" mode begins. In this mode, the longitudinal controller maintains a commanded distance from the wall, while the heading controller maintains the desired heading with respect to the wall. The vehicle then flies along the wall to the right, using the right facing sensor to detect walls and obstacles in the flight path. During lateral flight, the lateral controller does not try to maintain a fixed position, rather it monitors the side-looking sensors and tries to prevent the vehicle from getting within a specified distance of obstacles. The different corner-turning modes are entered upon different conditions. If a wall or obstacle is detected in the direction of flight, the vehicle enters "Inside Turn" mode, whereby it changes its heading to either turn the corner (for concave corners) or fly around the obstacle. This is achieved by giving an open-loop yaw command until no obstacle is seen by the right IR sensor. If one of the forward-looking IR sensors detects a step change to max range while the other sensor still reads near the estimated wall distance, a convex, or outside, corners has been detected. The vehicle then enters "Outside Turn" mode and it yaws to the left in order to continue around the corner. Once an inside or outside corner has been turned and valid range measurements are seen on the two front IR sensors, the vehicle returns to "Wall Follow" mode and continues flight.

V. Control Algorithm[9]

Leveraging the inherent attitude stabilization afforded by the coaxial rotorcraft platform used, the task of the controller is reduced to providing servo deflections such that the vehicle is able to track a commanded position. Although the controller has access only to the local range information, by exploiting the fact that indoor structures have walls the controller can be designed to follow the walls. This circumvents the requirement for a global position fix and increases the reliability of the local range measurements. The control architecture used here is a Proportional/Integral/Derivative (PID) design with gain scheduling applied such that the controller uses different gain values depending on where the vehicle is with respect to the wall. The rotorcraft dynamics around a valid trim value can be approximated by a linear model of the form:

$$\dot{x} = Ax + B(u + u_{trim}) \tag{1}$$

where x is the state of the system including the position, velocity, and the angular rates. The control input is given by u. Note that this control input is intended to provide a correction around the trim of the rotorcraft, given by u_{trim}. Let the commanded position be given by x_c which can be a constant command or the output of a reference model:

$$\dot{x}_c = A_{rm}x_c + B_{rm}r \tag{2}$$

where $r(t)$ is a reference input to the reference model. The error between the state and the command is:

$$e = x - x_c \tag{3}$$

This results in the error system:

$$\dot{e} = Ae + (A - A_{rm})x_c + Bu - B_{rm}r \tag{4}$$

The PID control action can be summarized by the following equation:

$$u = K_p e + K_d \dot{e} + K_i \int_{t_0}^{t} e(t)dt \tag{5}$$

Closed loop stability can now be ensured by choosing the PID gains K_p,K_d,K_i such that the error system (Equation 4) is rendered stable.

The key to successful and reliable implementation of PID control lies in the estimation of the derivative and the integral term. It is preferable to use direct measurements to form the derivative of the error $\dot{e}$, and the integral term $\int_{t_o}^{t} e(t)dt$. This requires measurement of the state derivative. In the case presented in this paper, position is to be controlled, hence the measurement of the system velocity and integral of position is required. However, these measurements are not always available, and hence an approximation must be used. Traditionally, the error is directly differentiated to obtain $\dot{e}$, however this method is highly susceptible to measurement error and noise. For this research, a Kalman filter based local velocity estimator was used in lieu of measured data.

In the traditional method for rotorcraft control, a position loop commands a velocity command, the velocity loop commands an attitude command, and the attitude loop achieves attitude stabilization by using the actuators to control the angular rate. In this case, due to the lack of accurate angular rate and attitude information, the traditional method of nested control loops cannot be utilized. This problem is solved by selecting a platform with inherent stability in roll and pitch, with a heading lock gyro to control yaw. The position command is then directly linked to the actuator deflection using a PID control logic as described earlier. The control action can now be achieved by using four independent control loops:

1. Altitude Hold: The function of this control loop is to use the filtered measurements from the downward pointed sonar for altitude control. A PID architecture is used, where the derivative of the position is calculated using the Kalman filter architecture as described previously. During vehicle operation, varying battery voltage level affects the throttle trim value. To counter this effect, an integral part is required in the controller. Instead of integrating the position as is traditionally done, the servo commands output by the controller is integrated. In this way the system can inherently handle actuator saturation and integration windup. Furthermore, servo commands are easier to measure since they are assigned by the controller. Figure 28 shows the schematic of the altitude control loop. The lateral and the longitudinal control loops have a similar architecture.

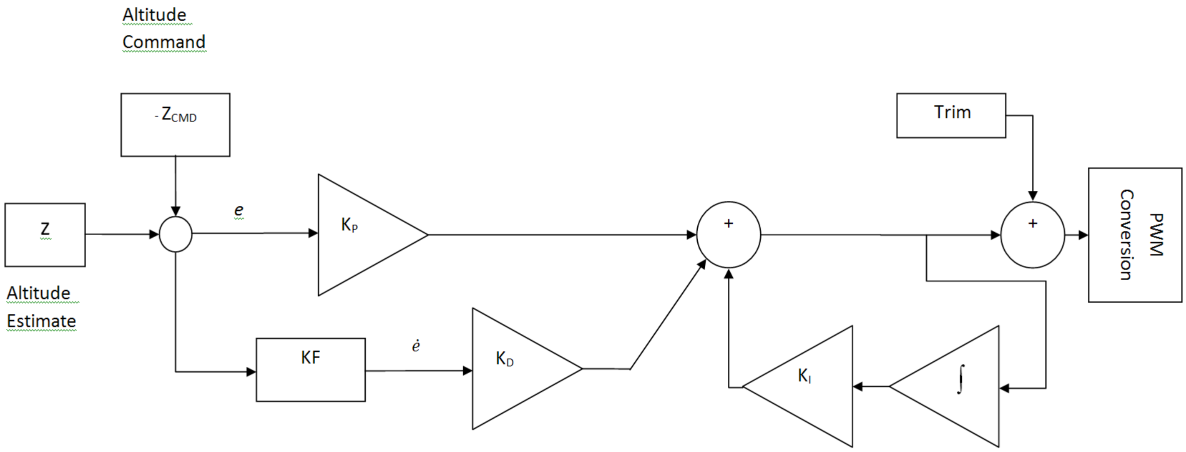


Figure 28: Architecture for the altitude hold controller.

2. Heading Hold: The function of this control loop is to control the heading of the vehicle. The heading information is formulated by using dual IR range sensors mounted with a mutual offset. The purpose of the heading hold is to maintain a relative heading with respect to a local reference, such as a wall. Let x_L denote the range estimate from the left IR sensor, x_R denote the range estimate from the right IR sensor, Ψ denote the relative heading of the vehicle, and let L denote the horizontal offset between the two sensors, then the heading estimate can be formulated as in Equation 6. The IR sensors are accurate only within a limited range of an obstacle, hence this heading estimate is highly susceptible to noise near maximum range. A computationally efficient way of handling this issue is to filter the IR range measurements instead using an Extended Kalman Filter to filter the heading measurements.

$$\Psi = \arctan\left(\frac{x_R - x_L}{L}\right) \tag{6}$$

3. Longitudinal Position Control: The function of the longitudinal control loop is to ensure that the vehicle maintains a fixed relative distance from a wall or obstacle. Longitudinal control is achieved by using forward mounted sonar measurements and forward mounted IR measurements. IR measurements should only be used while in close range of the obstacle. The architecture of the longitudinal position control is similar to that of the altitude hold.

4. Lateral Position Control: The function of lateral position control is to detect and avoid obstacles in the lateral path of the vehicle. The control loop architecture is similar to the altitude hold controller.

VI. Simulation Results

A simulation was initially developed using the open-source software Blender.[11] In the simulation, coaxial helicopter vehicle dynamics are simulated, as well as the sonar and IR range sensors and gyro mentioned above. The simulation allowed for easy configuration of different test environments, with variable room size and obstacle placement. In addition, the guidance and navigation algorithms were easily tested and refined before attempting to fly the actual hardware. In addition, mapping and localization algorithms were tested using the simulation. Figure 29 shows a screen capture from the Blender simulation, and the data presented in Figures 24 and 25 were generated using this simulation.

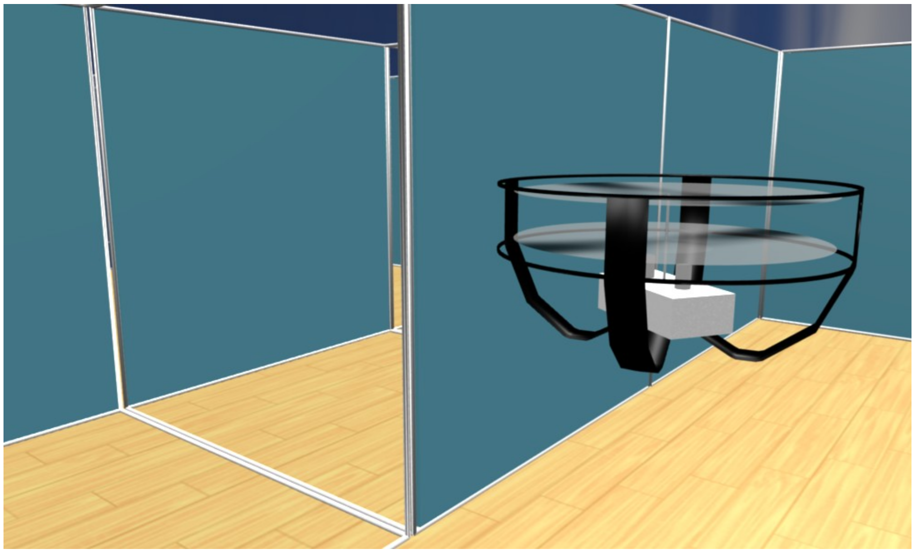


Figure 29: Screen capture from Blender simulation.

Once the vehicle was built and flight testing had begun, a higher fidelity simulation was developed using in-house software. In addition to simulating the vehicle dynamics, sensors, and test environment, this simulation actually ran the same control software that was loaded onboard the aircraft. Thus, guidance algorithms, as well as control system gain values and scheduling were further refined during the flight test program. Figures 30 and 31 show the flight vehicle simulation during successful autonomous navigation of a simulated arena for the 2009 IARC.

VII. Experimental Results

Flight testing for this research was completed in two phases. In the first phase, the initial avionics system consisted of four EZ4™ sonar for longitudinal and lateral ranging, one EZ1™ sonar for altitude ranging, and two IR sensors for heading estimation (see Figure 17). The sensor data was transmitted to a ground computer for processing, and servo commands were sent via a 2.4GHz hobby radio control transmitter. This setup facilitated testing of the sensors in flight under manual control before the onboard sensor processing and control algorithms were completed. During this phase of flight testing, characterization of the sonar and IR sensors was refined and altitude control and longitudinal obstacle avoidance were demonstrated. It was determined that in order to avoid interference between the sonar, each should be operated independently. With five sonar operating at 50ms per measurement, it takes 250ms to sample all of the sensors. However, a 4Hz update rate is not ideal for operating the altitude control loop, which is more sensitive to data rate. It is recommended to interleave the altitude measurement with the mapping sensors in the following manner: A-S1-A-S2-A-S3-A-S4, where "A" represents an altitude measurement, and "S1-S4" represent the four mapping sonar. Using this sampling order provides an altitude measurement at a rate of 10Hz, while mapping occurs at a slower rate of 2Hz. For longitudinal control in close proximity to a wall or obstacle, the IR sensors can be used since they operate at a rate of 25Hz. Unfortunately, due to problems with the sonar (possibly due to wiring or vibration), two of the sonar did not operate correctly using the first avionics package built and flown. As a result, it was determined to focus on building and testing another set of avionics specifically

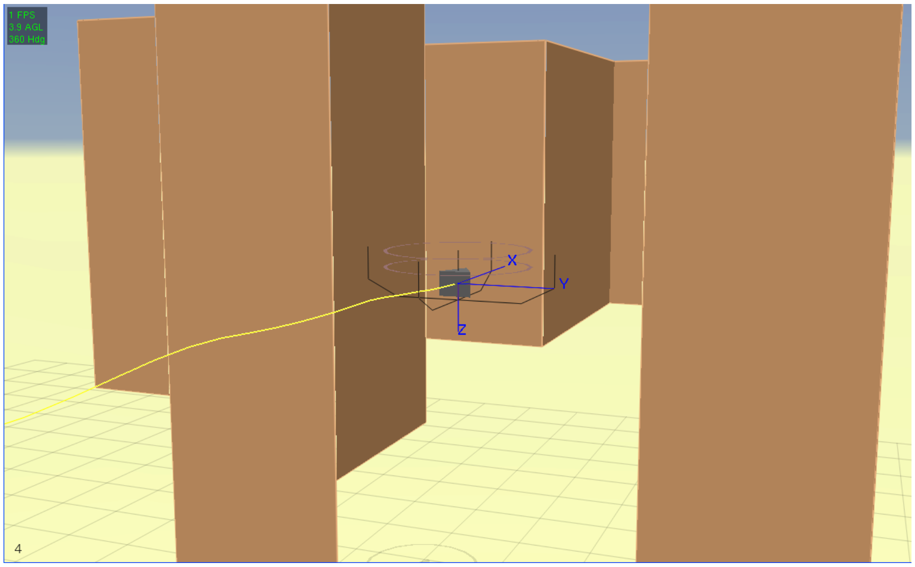


Figure 30: Screen capture from flight vehicle simulation.

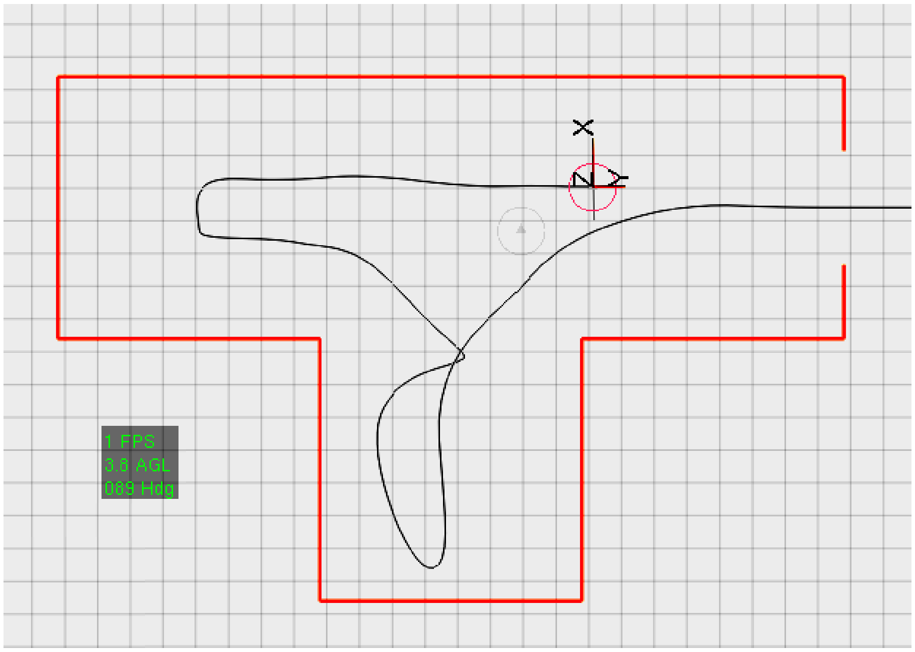


Figure 31: Screen capture from flight vehicle simulation showing navigation solution.

designed to test the wall-following guidance algorithm in preparation for the upcoming IARC competition.

Phase two of the flight test program involved a slight redesign of the sensor layout, as well as an upgrade from the first microprocessor to the ATMega128 described above. Due to the unique properties detailed the section on Sonar Testing, one EZ1™ sonar was used for altitude measurement only. The pair of forward-looking IR sensors were retained from the original design to enable heading and longitudinal control. Additional IR sensors of the same model were added to measure range in the lateral directions. A rear-facing IR sensor was not installed due to a limited number of analog sensing channels on the microprocessor. Fortunately, development of a new and improved safety shroud enabled the vehicle to survive mild contact with obstacles, so a rear-facing IR sensor was not required. The second sensor package was configured as shown in Figure 32 and is pictured in Figure 18.

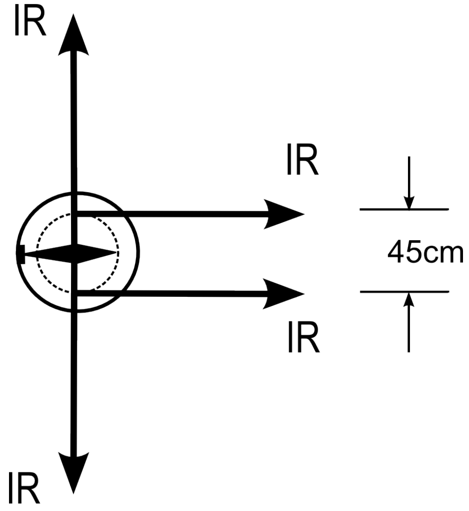
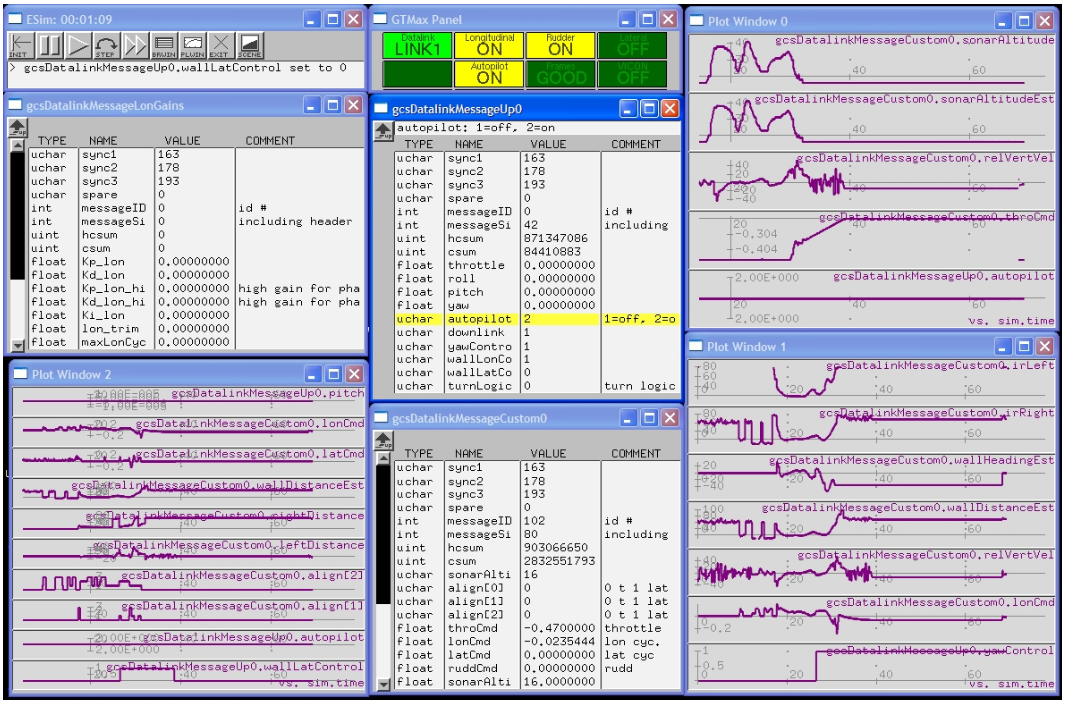


Figure 32: Range sensor layout.

During the second phase of flight testing, all navigation, guidance, and control algorithms were moved onboard the ATMega128 microprocessor. Thus, the controllers operated at a the same rate as the sensor measurements, while the range data and vehicle status was transmitted to the ground at a rate of 10Hz for monitoring and post-flight analysis. A Ground Control Station (GCS) was developed to enable real-time monitoring of all sensor data, adjustment of control loop gains, and as a pilot interface during manual flight. In addition to real-time monitoring, all sensor data and vehicle status information can be recorded for future use. A screenshot of the GCS is shown in Figure 33. A separate computer was also used to view and process imagery collected via an onboard camera, which was required to complete the IARC mission. Although it is possible to use the camera for measuring velocity (via optical flow) or possibly position,[3] such methods were not employed by this system.

A typical flight test began with manual takeoff and flight to a desired altitude. Next, the altitude control loop was enabled, bringing the vehicle to a commanded altitude while all lateral and longitudinal controls were still managed by a remote pilot. The altitude loop gains were tuned very early in the flight test program, and its performance was superb, so altitude control was generally enabled during testing of all the other control loops. After altitude control was proven, yaw control was tuned by adjusting the PID gains and schedules. Next, longitudinal and lateral control loops were tuned until satisfactory performance was achieved. Finally, guidance algorithms were enabled and adjusted such that the vehicle could maintain a desired range from a wall while flying laterally along the wall. Guidance logic was adjusted to enable the vehicle to make inside and outside turns within the building structure while exploring its environment. Flight times lasted approximately seven minutes, with all range sensors, microcontroller, datalink, and a camera with dedicated video link operating simultaneously.

Once the control loops and guidance logic had been tested, the vehicle was flown in the IARC arena. The competition allowed four attempts to navigate a maze constructed in a basketball arena. Each attempt was initiated by enabling completely autonomous control from a distance of 3m outside the competition arena. This distance was outside the range of the IR sensors used for heading and longitudinal control, so the vehicle flew forward at a predetermined speed until the arena structure was detected. The forward-looking


ESim: 00:01:09
INIT
STEP
BRWIN
PLWIN
EXIT
SCENE
GTMax Panel
Datalink LINK1
Longitudinal ON
Rudder ON
Lateral OFF
Autopilot ON
Frames GOOD
VICON OFF
Plot Window 0
gcsDatalinkMessageCustom0.sonarAltitude
gcsDatalinkMessageCustom0.sonarAltitudeEst
gcsDatalinkMessageCustom0.relVertVel
gcsDatalinkMessageCustom0.throCmd
gcsDatalinkMessageUp0.autopilot
vs. sim.time
Plot Window 1
gcsDatalinkMessageCustom0.irLeft
gcsDatalinkMessageCustom0.irRight
gcsDatalinkMessageCustom0.wallHeadingEst
gcsDatalinkMessageCustom0.wallDistanceEst
gcsDatalinkMessageCustom0.relVertVel
gcsDatalinkMessageCustom0.lonCmd
gcsDatalinkMessageUp0.yawControl
vs. sim.time
Plot Window 2
gcsDatalinkMessageUp0.pitch
gcsDatalinkMessageCustom0.lonCmd
gcsDatalinkMessageCustom0.latCmd
gcsDatalinkMessageCustom0.wallDistanceEst
gcsDatalinkMessageCustom0.rightDistance
gcsDatalinkMessageCustom0.leftDistance
gcsDatalinkMessageCustom0.align[2]
gcsDatalinkMessageCustom0.align[1]
gcsDatalinkMessageUp0.autopilot
gcsDatalinkMessageUp0.wallLatControl
vs. sim.time
20
40
60

```
> gcsDatalinkMessageUp0.wallLatControl set to 0
```

gcsDatalinkMessageLonGains

TYPE	NAME	VALUE	COMMENT
uchar	sync1	163	
uchar	sync2	178	
uchar	sync3	193	
uchar	spare	0	
int	messageID	0	id #
int	messageSi	0	including header
uint	hcsum	0	
uint	csum	0	
float	Kp_lon	0.00000000	
float	Kd_lon	0.00000000	
float	Kp_lon_hi	0.00000000	high gain for pha
float	Kd_lon_hi	0.00000000	high gain for pha
float	Ki_lon	0.00000000	
float	lon_trim	0.00000000	
float	maxLonCyc	0.00000000	

gcsDatalinkMessageUp0

autopilot: 1=off, 2=on

TYPE	NAME	VALUE	COMMENT
uchar	sync1	163	
uchar	sync2	178	
uchar	sync3	193	
uchar	spare	0	
int	messageID	0	id #
int	messageSi	42	including
uint	hcsum	871347086	
uint	csum	84410883	
float	throttle	0.00000000	
float	roll	0.00000000	
float	pitch	0.00000000	
float	yaw	0.00000000	
uchar	autopilot	2	1=off, 2=o
uchar	downlink	1	
uchar	yawContro	1	
uchar	wallLonCo	1	
uchar	wallLatCo	0	
uchar	turnLogic	0	turn logic

gcsDatalinkMessageCustom0

TYPE	NAME	VALUE	COMMENT
uchar	sync1	163	
uchar	sync2	178	
uchar	sync3	193	
uchar	spare	0	
int	messageID	102	id #
int	messageSi	80	including
uint	hcsum	903066650	
uint	csum	2832551793	
uchar	sonarAlti	16	
uchar	align[0]	0	0 t 1 lat
uchar	align[1]	0	0 t 1 lat
uchar	align[2]	0	0 t 1 lat
float	throCmd	-0.4700000	throttle
float	lonCmd	-0.0235444	lon cyc.
float	latCmd	0.00000000	lat cyc
float	ruddCmd	0.00000000	rudd
float	sonarAlti	16.0000000	

Figure 33: Screen shot of ground control station.

pair of IR sensors were used to adjust lateral position of the vehicle with respect to the entry portal, while the side-looking IR sensors were used to determine when the vehicle had entered the arena. Once the vehicle was inside, the wall-following guidance algorithm was initiated.

On the first attempt, the vehicle began autonomous flight with a slight yaw rate, which was due to improper capture of the yaw trim value during the handoff from manual to autonomous control. This caused the vehicle to miss the arena entry portal. The second attempt ended when the vehicle again missed the portal, but this time it detected the arena walls. The guidance algorithm was initiated and the vehicle began its wall-following routine on the outside of the arena structure. On the third attempt, the vehicle successfully entered the arena and began the wall-following routine. After traversing approximately four meters along the wall, an obstacle on the floor was detected by the altitude sonar and the altitude control loop caused the vehicle to ascend. IARC judges ended the attempt at their discretion when it appeared that the vehicle would exceed the arena maximum altitude limit of 2.4m. The outlier detection algorithm was then adjusted such that obstacles detected below the vehicle would have less effect on the altitude controller. On the fourth and final attempt, the vehicle successfully entered the arena and began its wall-following routine. It successfully negotiated the obstacles on the floor, and flew down a hallway for approximately 12m. Unfortunately, an integral gain introduced to improve heading control had the incorrect sign, and after a few minutes the vehicle began to fly in a circular motion. This prevented the wall-following algorithm from maintaining proper heading and wall distance, and the attempt ended. Altitude data for a practice run preceding the fourth attempt is shown in Figure 34. Lateral and Longitudinal data, as well as controller commands, are shown in Figure 35 .

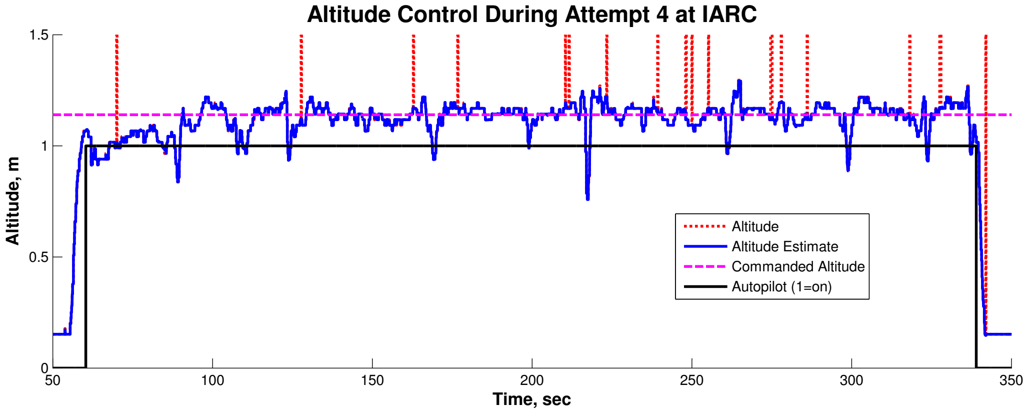


Figure 34: Altitude measurements and estimate. The performance of the outlier detection routine is visible here. The several "short" range measurements correspond to an object being placed and withdrawn beneath the vehicle during the flight. Note that the outlier filter correctly eliminates bad range measurements while incorporating real step changes in range. The autopilot command represents the enabling and disabling of the altitude-hold controller.

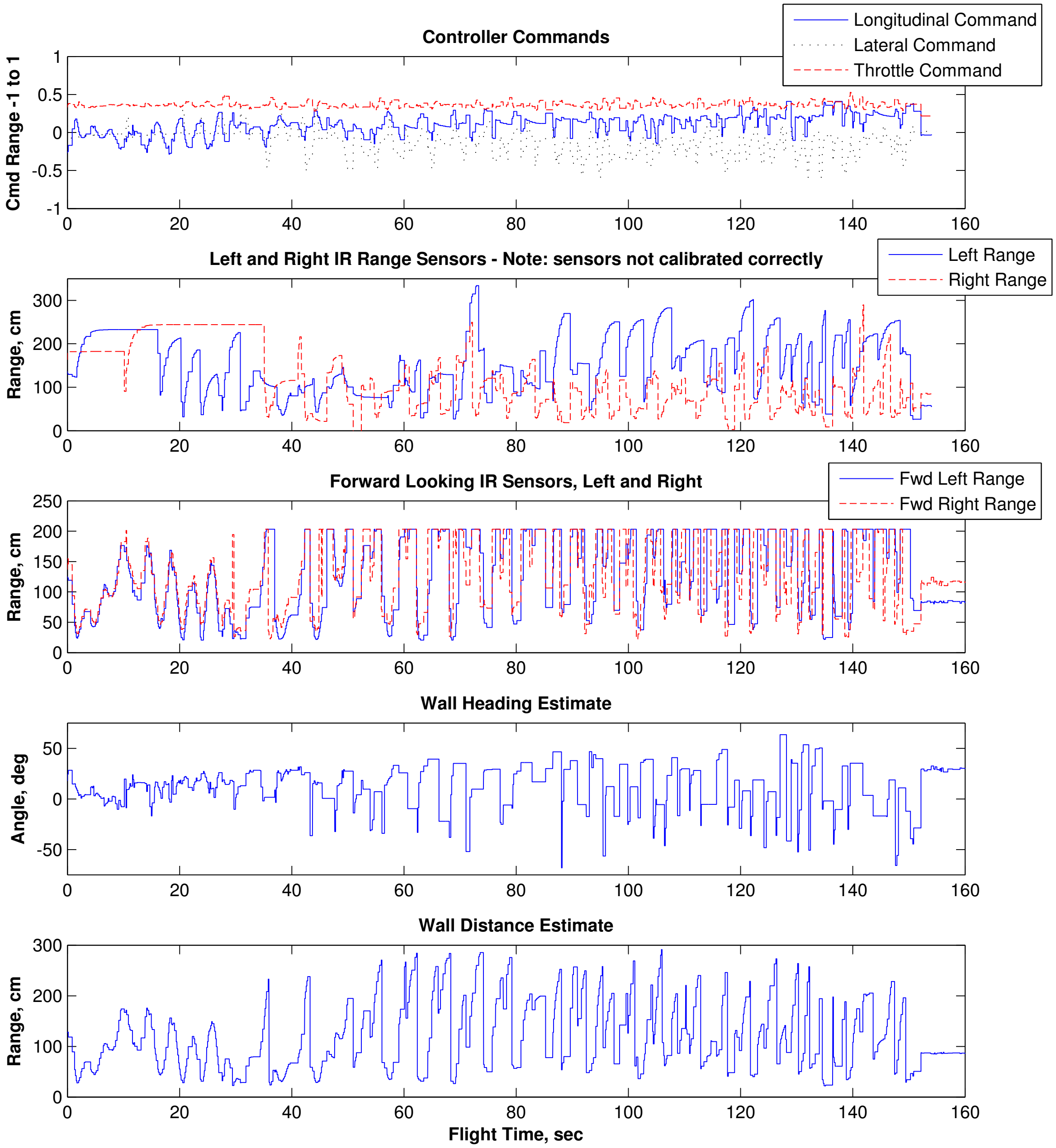


Figure 35: Flight data recorded during the fourth attempt of the GTAR vehicle at the International Aerial Robotics Competition, 2009.

VIII. Conclusion

This research proposed the use of inexpensive, lightweight range sensors for indoor UAV navigation. Two potential range sensors were tested for suitability and error characteristics. The SHARP infra-red range sensors provide a narrow beam and a higher resolution distance measurement, at the expense of decreased range (approximately 150-180cm maximum). The MaxBotix® EZ1™ and EZ4™ ultrasonic range sensors had a longer range (up to 6.45m) and a wider beam. In addition, the sonar has unique measurement properties due to the physics of the metrology technique used for ranging. The sonar sends out a sound pulse that expands on a spherical front. When this wave front encounters an object, it is reflected back toward the sensor. When the first return is detected, a time-of-flight calculation is used to determine range. As a result, the perpendicular distance to a wall will always have the shortest return. Hence, if one of the sonar can detect a wall at all, it will always return the perpendicular distance to that wall. Similar effects were observed when the sensors were aimed at the corner of a room. Thus, the sonar range sensors return the same distance regardless of moderate changes in angle to the target. This makes them useful for altitude ranging, as the vehicle roll and pitch do not greatly affect the distance measurement to the ground. The sonar are less useful for scanning a room to make a 2-D plot of the room, although by using a histogram analysis, room dimensions were determined from experimental sonar data. In addition, simulations were performed using sonar and the histogram analysis to determine room dimensions in real time. The addition of a yaw rate gyro was simulated, which along with wall-following behavior to correct for drift, could allows a heading estimate to be maintained during the flight as well. With a heading estimate and the combination of sonar and IR sensors for ranging, a small passively stable aerial platform could be used for basic mapping and localization.

Once the range sensors were analyzed and their measurement and error characteristics were determined, simulations were completed to develop and refine mapping and localization techniques using the low-cost range sensors. Finally, a two-phase flight test program was initiated to further develop navigation, guidance and control algorithms. During the first phase of the flight test, altitude control and longitudinal control were demonstrated. During the second phase of the flight test program, heading control and lateral control were developed, as well as navigation and guidance algorithms as required to complete mission requirements for the International Aerial Robotics Competition. During the competition, the vehicle was successfully able to enter and negotiate portions of a maze via a wall-following guidance algorithm. Thus, the completion of the flight test program demonstrated the capability to navigate unknown indoor environments utilizing only simple, ultra low-cost range sensors mounted to a passively stable coaxial rotorcraft vehicle.

IX. Acknowledgments

This research would not have been possible without the help of the Georgia Tech Aerial Robotics (GTAR) team. The GTAR members are all volunteers and spent numerous hours during their “free” time working on the project in preparation for the IARC. The team members include (in alphabetical order): Girish Chowdhary, Claus Christmann, Dr. Eric Johnson (academic advisor), Scott Kimbrell, John Ottander, Dr. Erwan Salaün, Syed Shah, Michael Sobers (team lead), Allen Wu, and Shusaku Yamaura. Thanks also to Jeong Hur for help and advice during construction of the vehicle.

References

[1] Zunino, G., *Simultaneous Localization and Mapping for Navigation in Realistic Environments*, Licentiate thesis, Royal Institue of Technology, Sweden, 2002.

[2] Watanabe, Y., *Stochastically Optimized Monocular Vision-Based Navigation and Guidance*, Ph.D. thesis, Georgia Institute of Technology, Georgia, Nov 2007.

[3] Wu, A. D., and Johnson, E. N., "Methods for Localization and Mapping Using Vision and Inertial Sensors", AIAA Paper 2008-7441, Aug. 2008.

[4] Ivey, G. F., and Johnson, E. N., "Investigation of Methods for Simultaneous Localization and Mapping Using Vision Sensors", AIAA Paper 2006-6578, Aug. 2006.

[5] E-Sky E020 "Big Lama" product website: http://www.twf-sz.com/english/products.asp?prodid=0291

[6] Sparkfun Electronics website: http://www.sparkfun.com/

[7] *LV-MaxSonar® -EZ4™ Data Sheet*, MaxBotix® Inc., 2007.

[8] *SHARP GP2Y0A02YK0F Distance Measuring Sensor Unit Data Sheet*, Sheet No.: E4-A00101EN, SHARP Corporation, 2006.

[9] Chowdhary, G., Ottander, J., Salaün, E., and Johnson, E., *Low Cost Guidance, Navigation, and Control Solutions for Miniature Air Vehicle in GPS Denied Environments*, First Symposium on Indoor Flight, International Aerial Robotics Competition, Jul. 2009.

[10] International Aerial Robotics Competition website: http://iarc.angel-strike.com/

[11] Blender Foundation website: http://www.blender.org/