Unmanned Aerial Vehicle of BITS Pilani, Dubai Campus for the International Aerial Robotics Competition 2011

Saurabh Ladha, Deepan Kishore Kumar, Robin Singh ,Pavitra Bhalla, Anant Mittal, Aditya Jain, Anshul Upreti, Pratik Raju Joshi, Prof. Dr. R.K. Mittal, Dr. Anand Kumar

Birla Institute of Technology and Science, Pilani-Dubai Campus, Dubai, UAE

ABSTRACT

The Intelligent Flying Object for Reconnaissance (IFOR) is an autonomous aerial vehicle that has been developed by BITS Pilani, Dubai Campus students. The vehicle is capable of localizing itself using the SLAM algorithm, stabilize its attitude (pitch, roll and yaw) and altitude using PID controllers, plan paths around obstacles and navigate an unknown indoor environment with wall following guidance. In addition, it has been designed to be capable of pattern recognition which would enable it to recognize images and signs. These features enable the IFOR to execute the 6th mission of the International Aerial Robotics Competition, which involves scanning an unknown indoor arena protected by laser barriers and cameras, bestrewn with obstacles, in search of a flash drive.

1. INTRODUCTION

The field of robotics is witnessing a paradigm shift in the operation and use of robots. With various robots becoming autonomous and intelligent day by day, their application and use has increased tremendously. *Aerial robots* specifically have an edge over other autonomous vehicles due to their higher degrees of freedom and agile maneuverability. To tap these features and advance the applications and versatility of the vehicle is the aim of the team.

1.1 Problem Statement

The 6th mission of the IARC requires teams to construct a fully autonomous aerial robot capable of self controlled flight within a confined environment. The vehicle will first be required to enter a 'military' facility through a one square meter or larger opening from a designated launch area 3m away. The vehicle will have to search for a target area ('Office of the Chief of Security') while avoiding unknown obstacles such as walls, columns and furniture as well as visible security systems like a video camera and a laser barrier in a hallway.

The building will contain several signs indicating the route to the target area as well as appropriate indicators allowing avoidance of the security systems. Once the target area is found the vehicle is expected to locate a target object (Black colored pen drive), pick it up and drop a decoy in its place before proceeding to fly back out of the window which it had entered through.

1.2 Conceptual Solution

We have designed an autonomous vehicle which leverages the use of an off the shelf quadrotor as the base platform. This has allowed us to focus on the autonomy of the vehicle instead of the intricacies of flight dynamics. We have used the Ascending Technologies' (Asctec) Pelican as the quadrotor of choice for its payload capacity and high structural integrity. A quadrotor, by nature, is aerodynamically unstable. This demands the use of a Stability Augmentation System (SAS) which the Pelican is already equipped with. The SAS relies on data from the Inertial Measurement Unit (**IMU**) to keep the vehicle stable in flight. We have equipped the quadrotor with a scanning laser range finder to enable exteroceptive sensing of the surroundings. Simultaneous Localization and Mapping (SLAM) is used to calculate the coordinates of the vehicle in a global frame which are used to correct drift. A wall following algorithm is used to guide the vehicle through the “maze” of rooms and a path planning algorithm is used for calculating the optimum path around obstacles. Data from three cameras are processed by Optical character recognition (OCR) techniques to identify signboards and recognize the pen drive. Blob detection is used to recognize the ON/OFF state of the Blue LED and trigger the start of the mission. All the modules are monitored by a Command Module which serves as the link between all systems and also allows for changes to be made dynamically to each of them. All processing is done on board by an Intel Atom 1.6 GHz processor.

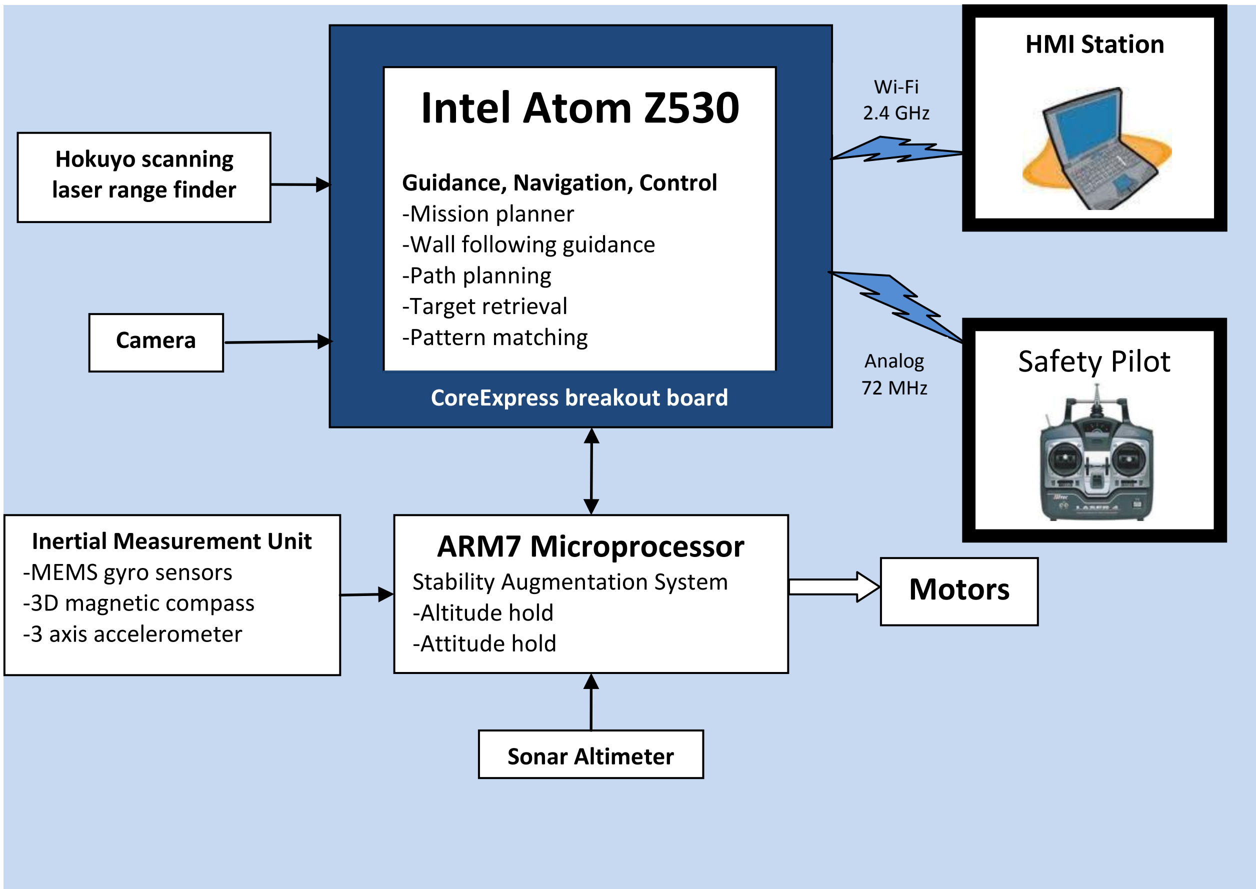


Figure1: System architecture

1.3 Yearly Milestones

In our first attempt at the 6th Mission, we intend to design a vehicle that is capable of flying autonomously(i.e. without pre-programming the flight) by performing drift control, intelligent navigation through the arena, path planning and image identification (limited to recognition of the pen drive and signboards). Real time tracking of the target object and its subsequent retrieval will be implemented in the following year.

2. Air vehicle

The Pelican weighs about 980 g and has a payload capacity of 500 g, ideally suited for covert missions.

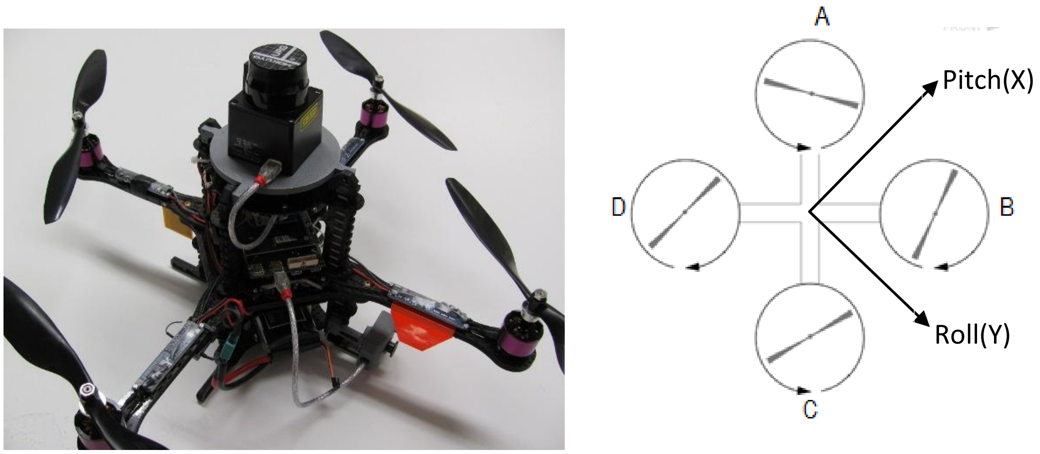


Figure 2: The Asctec Pelican quadrotor and the propeller action.

2.1 Propulsion and lift system

The quadrotor is a Vertical Takeoff and Landing (VTOL) rotorcraft which is propelled by four rotors. Unlike a helicopter, a quadrotor does not require a separate propeller to negate the angular momentum generated by the movement of the driving propeller/s; this is due to the fact that, in a quadrotor, a set of two propellers is made to rotate clockwise while the other set is made to rotate anticlockwise. Hence the net angular momentum about the centre is zero.

The vehicle can be flown in square or diamond configuration, in diamond configuration the pitch/roll axes are the diagonals between the propellers. In such a configuration only two propellers are involved in causing pitch/roll at any point of time. In a square configuration the directions of forward pitch and forward roll are as marked in the diagram above. Pitch (positive) is achieved by reducing the angular velocity of two propellers (A and B) and increasing the velocity of the other two propellers (C and D), whereas, Roll (positive) is achieved by increasing velocities of propellers A and D and reducing the velocities of B and C.

2.2 Stability Augmentation systems

The quadrotor, by nature, is an aerodynamically unstable system. It hence demands to be one that is mechatronic in nature with its dynamics being controlled by PID generated signals. The input to the PID loops are the readings from the IMU which consists of three gyrometers and a triple axis accelerometer. The raw readings from the sensors are mapped onto tilt angles which are then compared to the desired setpoint. The preset PID constants are then used to calculate the required output to the motor at a frequency of 1 kHz. The control loops ensure that the vehicle achieves a desired attitude with minimal error. The loops for Pitch, Roll and Yaw are implemented on the ARM7 processor of the Pelican and are tuned off-the-shelf.

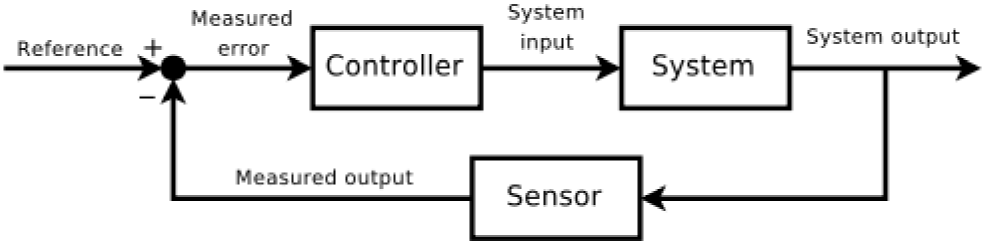


Figure 3: PID control system architecture

We have designed and implemented an altitude PID control loop on the Intel Atom processor, which can be described as below:

$$Command_{Throttle} = K_c\{\Delta z + \frac{1}{Ti} \int \Delta z \ dt + \tau \frac{d}{dt} \Delta z\}$$

where Δz is the difference between the desired altitude and actual altitude.

The readings of a SONAR altimeter are used as the input to the loop.

2.3 Guidance Navigation and Control (GNC)

The GNC System we have built consists of *SLAM, Drift Control, Orientation module, Wall follower, Path planner and Command module*

SLAM

SLAM is an acronym for the Simultaneous Localization and Mapping algorithm. It consists of multiple parts; landmark extraction, data association, state estimation, state update and landmark update. The algorithm estimates the apparent change in position of landmarks (unique regions) between the current scan and the map assuming the surroundings to be stationary. The vector opposite to this one represents the change in position of the vehicle. “Global” coordinates hence calculated are used to generate a map of the environment. Therefore, the map and coordinates are simultaneously estimated from each other.

We have used the CoreSLAM algorithm from *http://www.openslam.org* as a base platform for our custom SLAM algorithm due to its computationally light nature. SLAM is not meant to be used for real time applications since map generation requires substantial processing power, we hence modified the algorithm to perform a scan-scan comparison instead of a scan-map comparison. Therefore, at any point there are only two scans being studied by the system and the map is not generated during the mission which allows for the algorithm to be used solely for the purpose of localization in real time.

Drift Control

Drift control is most important to the performance of the vehicle in the mission since milligram imbalances in weight, apart from other factors, cause a quadrotor to drift unintentionally. The coordinates which are output by SLAM are used by the Drift control algorithm to estimate the difference between the desired position and the actual position. The error in meters, for pitch and roll respectively, is then used in the following PID loops which are used to command the pitch and roll of the vehicle:

$$Command_{Pitch} = K_c\{\Delta p + \frac{1}{Ti} \int \Delta p \; dt + \tau \frac{d}{dt} \Delta p \}$$

$$Command_{Roll} = K_c\{\Delta r + \frac{1}{Ti} \int \Delta r \; dt + \tau \frac{d}{dt} (\Delta r)\}$$

$\Delta \boldsymbol{p}$ is the X-axis-component of position error

$\Delta \boldsymbol{r}$ is the Y-axis-component of position error

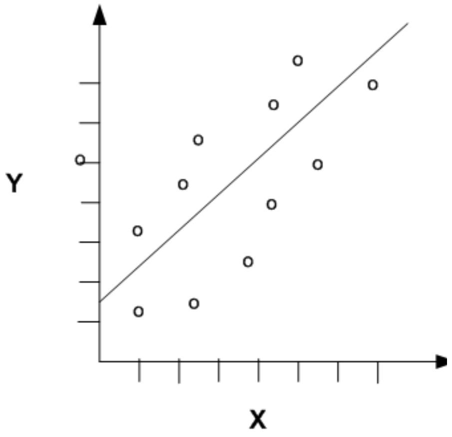
K_c is the controller parameter, from which proportional K_p (=K_c X 1), integral K_i (=K_c / T_i) and derivative K_d (=K_c X τ) constants are obtained as per the Zeigler-Nicholls tuning rule.

The loops are implemented on the Intel Atom processor.

Orientation Module

The purpose of this module is to ensure that the vehicle is oriented such that it is parallel to the walls of a corridor. This is necessary for the *Wall follower* to work correctly and also for the OCR algorithms. In all real world scenarios the wall of any indoor environment is never perfectly smooth or flat, especially to a Light Detection and Ranging device. This factor is to be considered to align the quadrotor either parallel to the wall or to determine its current orientation with respect to any particular wall. To align the quadrotor with the wall using only LIDAR data, we perform statistical analysis to solve for the slope of the wall with respect to the current orientation of the quadrotor. This is accomplished through the method of least-squares linear regression.

If each point on the wall is represented as (x, y) and we take n such points on the wall, we obtain the slope of the wall with respect to the present orientation of the quadrotor as **b**, given by:



$$b = \frac{n \sum (xy) - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

Figure 4: A wall in reality

By increasing the scanning range under consideration we get more points on the wall, which gives a statistically better estimate of the slope.

Wall follower

Once the vehicle is oriented the quadrotor is ready to begin navigating through the maze, this is accomplished by the use of a wall following guidance algorithm which maintains the right wall as a reference. The pitch of the vehicle is controlled for varying its linear velocity.

The wall follower is based on the detection of spikes i.e. a drastic change in a continuous set of values as predefined by the programmer. The right wall follower implemented by us can be described as below:

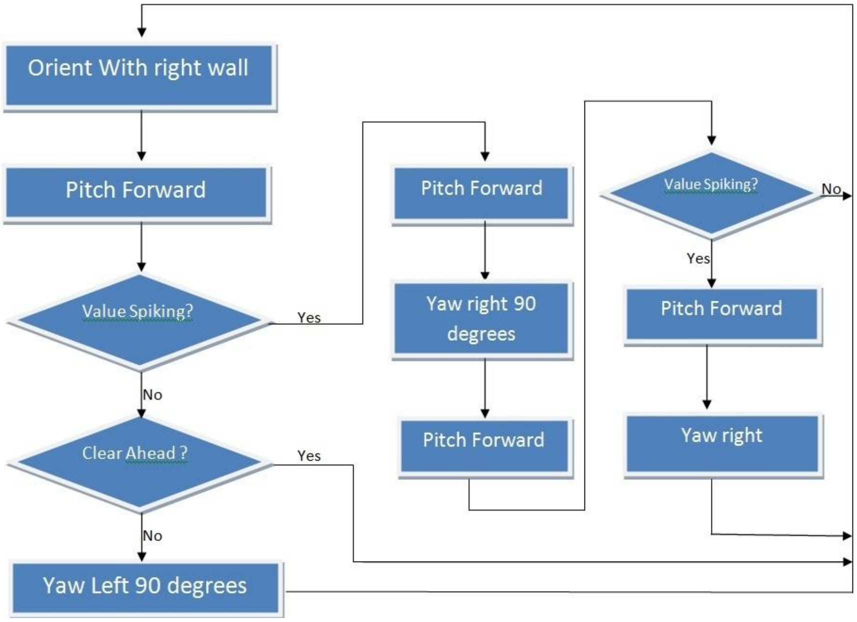


Figure 5: Wall follower flow chart

Path planner

This module identifies objects and classifies them as *obstacles* if they are in the intended path of travel.

On finding an *obstacle* in the scan, the algorithm proceeds to suitably find a path that would avoid the *obstacle* with the minimum possible deviation from the original path. We define a *safe distance*(which is contingent on the aerial vehicle's kinematics) from an *obstacle* and a minimum clearance width based on which a scanning range of θ degrees in front of the quadrotor is calculated to be sufficient to identify the existence of an *obstacle*. If in this angular range, a point is found to be within the "clear zone" (arc sweep of width θ degrees and radius equal to *safe distance)* in front of the vehicle it is declared to be an *obstacle*. This is the point at which the "growth algorithm" takes over the navigation of the quadrotor.

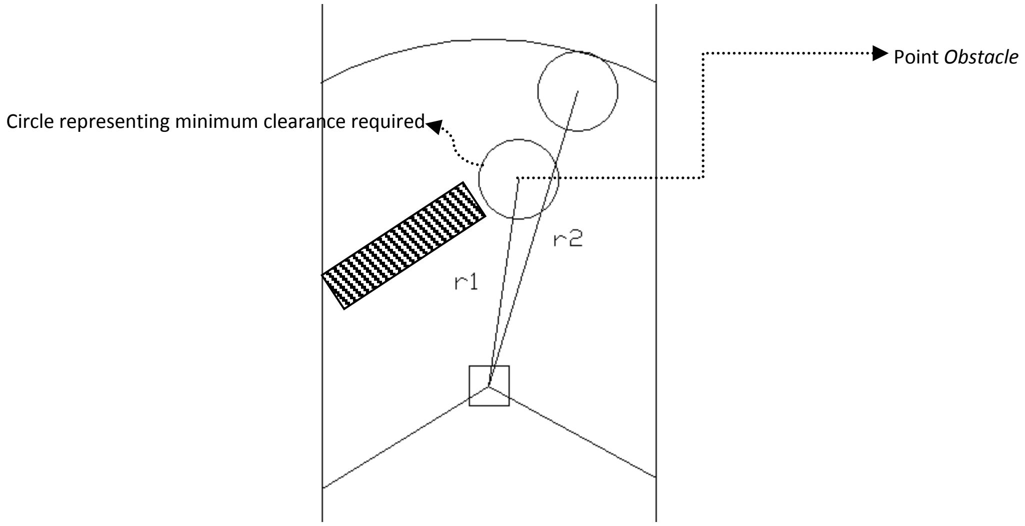


Figure 6: Obstacle detection and growth algorithm

The simplest method to calculating the optimum path is to find the longest free path and move the quadrotor in the direction thus found. This could, more often than not, cause a collision as shown in the above figure. In the figure two point *obstacles* of interest are highlighted, both of which have a distance r1 and r2 from the vehicle respectively, as can be seen, the longest free path will be judged to be r2 which would lead to a collision with point *obstacle* 1. Hence the growth algorithm is invoked, in the algorithm, every point in the desired scanning range is considered to be an obstacle and is "grown" by the half the clearance width of the vehicle. This allows for the vehicle to be considered to be a point object since its dimensions are superimposed on its entire environment.

To "grow" the environment, a circle(of radius equal to half the clearance width) is drawn around every point and all LIDAR range readings are shortened such that no part of any of these line segments fall within a “growth” circle. This simulates the “growth” of all *obstacles* as shown below.

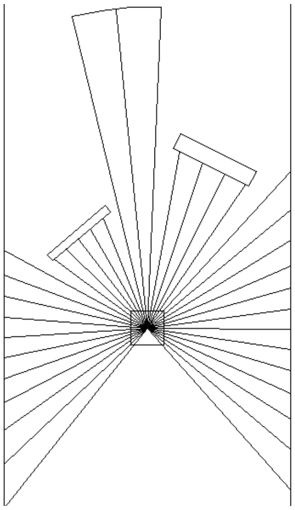


Figure 7: Actual environment

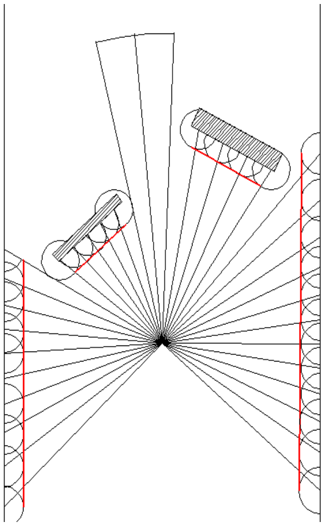


Figure 8: “Grown” environment

Figure 7 and 8 show range readings in the actual environment and then in the new “grown” environment (shown in red) in which the vehicle is considered to be a point object. On performing this growth procedure the algorithm proceeds to find the optimum path.

Although the optimum path is to be selected within the 120 degree span ahead of the vehicle, it is necessary to study the entire scanning range of 240 degrees to ensure that no *obstacle’s* growth circle is entering the 120 degree zone marked. If an *obstacle* outside the zone does, indeed, cause for a growth circle to enter into the zone it shall be treated as any other circle and the range line segments within the 120 degree zone shall be reduced accordingly before the algorithm proceeds to find the optimum path. The optimum path is selected based on a *priority factor* assigned to each free path which factors in the length of the path and its angular deviation from the intended path.

Command Module

This module is in charge of forming the link between all the above modules. The *Image processing (IP)* module relays news of positive/negative identification of signboards to the *Command module* which in turn can override/permit the *Wall follower* to enter a particular room. In addition the *IP* module also informs the *Command module* if a laser barrier is detected, in which case a message is relayed back to the base station through the HMI which is also controlled, solely, by the *Command module*. The module is also responsible for entry through the window for which it uses inputs from the *IP* module to find out when the Blue LED is ON/OFF and triggers the start of the mission accordingly.

The command module also ensures that the vehicle does not yaw more than 90 degrees from its original path when it is being controlled by the *Path planner*. This, in turn, ensures that that the quadrotor does not go back into an already swept area.

2.6 Flight Termination System

The quadrotor though autonomous in its flight can also be manually commanded to abort. A safety kill switch mechanism has been developed in order to attain this feature. The Pelican has a built in termination system due to which, the vehicle lands the moment the transmitter falls out of range. This feature is used for flight termination via a kill switch which shuts the transmitter off and hence causes the vehicle to descend.

3. Payload

3.1 Sensor suite

GNC Sensors

- *Asctec Pelican Inertial Measurement Unit (IMU):* The IMU consists of MEMS gyro sensors, a triple axis accelerometer and triple axis magnetic compass. This allows for the calculation of tilt angles which can be polled up to 100 times per second.

- *Hokuyo URG04LX-UG01 Light Detection and Ranging Device:* A scanning laser range finder of weight 160 gm, angular resolution 0.36 degree, range 5.6 m and a frequency of 10 Hz

- *Point Grey Firefly MV cameras*: The cameras provide data to the *IP* module which allow for detection of the Blue LED and laser barriers in addition to recognition of signboards.

- *MaxSonar LV altimeter*: The SONAR altimeter is connected to the Intel Atom processor through a serial port link which can be polled up to 20 times per second

3.2 Mission sensor

Image Processing (IP) module

We have equipped the vehicle with *Point Grey Firefly MV cameras* that provide the input for the *image processing* module. The cameras have been placed on the right, left and front of the quadrotor. The camera on the right and left side will be used as inputs to *Optical Character Recognition (OCR)* algorithm in order to identify the signboard of 'Office of the Chief of Security'. The competition involves recognizing a given set of Arabic letters from a set of words; the target room can be identified by this particular word. Live images are broken down into ASCII characters which are then compared to the predefined set. The ASCII characters need not necessarily represent actual Arabic letters and could arbitrarily be assigned to unique patterns by the programmer. A *Match Closeness Parameter* is then calculated to check if the similarity between the two is within acceptable limits, if so, the *Command module* is notified of the same.

All three of the cameras are used to find and identify the pen drive and laser barriers. Blob detection is used to detect the laser barrier as well as the Blue LED at the opening of the arena.

3.3 Communications

The vehicle communicates with a base computer via an Xbee module over the Zigbee protocol in order to deliver telemetry data. The other communication links include a Wi-Fi link to deliver real time video to the base station. Both of these links operate at 2.4 GHz. Finally, a safety pilot can take control over the vehicle at any time using a Futaba radio controller operating at 72 MHz

3.4 Power system

The quadrotor is powered by an 11.1V Lithium Polymer Battery. A power board is used to distribute power and communication lines to all motor controllers and other systems on board. The power board comprises of a switching power regulator to generate a stable 6V supply for the Auto Pilot board and a high power MosFET to switch current ON and OFF.

4. Operation

4.1 Flight preparation

Each flight test is performed with utmost precaution following the mentioned safety procedure which ensures a safe and smooth flight of the quadrotor.

4.2 Checklist

1. Double Check LiPo battery voltage using voltmeter
2. Examine the propellers, safety mounts, nuts and screws for any damage
3. Test communication link between the quadrotor and the Ground station
4. Enable safety pilot and check kill switch action before flight
5. Check status LED’s

4.3 Man/Machine interface

The vehicle communicates via Wi-Fi link to two base computers in order to relay real time video feeds from the onboard cameras, another monitor is used to display the status of the vehicle ex: “Laser barrier detected” and the Xbee link is used to display telemetry data in real time on the fourth monitor in the base computer cluster. This forms the live HMI of the system which allows for constant monitoring of the vehicle by the judges.

5. Risk reduction

5.1 Vehicle status

Two status LEDs allow for a check on certain critical vehicle states. The blinking of a Red LED on the processor board of the Pelican indicates the initialization and calibration of the sensors. Once the sensors are calibrated a green LED blinks rapidly indicating the flight control software is running. In case the battery voltage drops under 9.8 volts a loud tone is emitted, with the beeping becoming faster as the voltage drops.

5.2 Shock / Vibration isolation

The Asctec Pelican is built on a carbon fiber frame which has a very high Ultimate Tensile Strength (UTS); thus is it can withstand a heavy impact without necking. It is also fitted with soft pads below the arms to cushion impacts.

5.3 EMI/RFI Solutions

The magnetometer is very sensitive to EMI, hence it is mounted above all electronics such as IMU and the processors.

Safety

The IFOR is equipped with landing gear designed in a manner to deflect shock from the electronic system. The propellers are also covered, which ensures safety to both bystanders and the vehicle in case of a mishap.

The ON/OFF switch on board is designed active low, so if for some reason the mechanical switch breaks or looses connection the vehicle will remain ON. However this mechanism is overridden by the safety kill switch.

Modeling and Simulation

The intelligence modules were extensively tested for robustness before and after burning the codes onto the on board processor. Image processing was developed from scratch to meet the requirements of the competition. The PID controllers were tested on Lab View.

Testing

Device / Routine	Testing
HOKUYO LIDAR	**Tested while running the intelligence software.**
SLAM	**Real time execution and experimental determination of localization accuracy.**
Cameras	**Pre flight and in flight testing of image processing**
Drift Control	**Onboard as well off board testing for suitability to tuning**

Conclusion

We have designed IFOR to be a fully autonomous quadrotor that would be able to successfully accomplish the tasks of the IARC. The IFOR system comprises of Simultaneous Localization and Mapping which has been optimized for real-time localization, drift control using simple PID controllers and a detailed set of Intelligence algorithms. The navigation of the quadrotor is handled by the wall following algorithm while path planning ensures that indoor environments can be explored regardless of the presence of unknown obstacles. Finally, image processing enables the quadrotor to scan for patterns, edges and symbols and make corresponding control outputs through a master program in order to guide the vehicle to its target.

Acknowledgements

We would like to thank Oussama El Hamzaoui, author of “CoreSLAM : a SLAM Algorithm in less than 200 lines of C code” who guided us in the customization of CoreSLAM.

References

1. Johnson, E. N. and Schrage, D. P., “System Integration and Operation of a Research Unmanned Aerial Vehicle,” *AIAA Journal of Aerospace Computing, Information and Communication, Vol. 1, No. 1, Jan 2004, pp. 5–18.*

2. Durrant-Whyte, H.; Bailey, T. (2006). "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms". *Robotics and Automation Magazine* 13 (2): 99–110.

3. Bruno Steux, Oussama El Hamzaoui (2010), “CoreSLAM : a SLAM Algorithm in less than 200 lines of C code”. *International Conference on Control, Automation, Robotics and Vision (ICARCV), 2010*

4. Andrew J. Davison. Real-time simultaneous localization and mapping with a single camera. *Ninth IEEE International Conference on Computer Vision (ICCV’03) - Volume 2*, 2003.

5. Chowdhary, G. V. and Johnson, E. N., “Theory and Flight Test Validation of Long Term Learning Adaptive Flight Controller”, *Proceedings of the AIAA Guidance Navigation and Control Conference, Honolulu, HI, 2008.*

6. Michelson, R., Rules for the International Aerial Robotics Competition 6th Mission, http://iarc.angel-strike.com/IARC_6th_Mission_Rules.pdf

7. Hoffmann, G. M., Huang, H., Waslander, S. L., & Tomlin, C. J. (2007). Quadrotor Helicopter Flight Dynamics and Control Theory and Experiment. *AIAA Guidance, Navigation and Control Conference and Exhibit*, *20*(23), 1-20.

8. Point Grey - Imaging - Firefly MV CMOS USB 2.0 / Point Grey Firefly MV Camera. (n.d.). *Point Grey CCD and CMOS Digital Cameras for Industrial, Machine, and Computer Vision*. Retrieved June 1, 2010, from http://www.ptgrey.com/products/fireflymv/index.asp