Northwestern Polytechnical University Team Entry for the 2012 AUVSI International Aerial Robotics Competition

1 June 2012

Fei Liu, Yinan Sang, Jie He, Jie Chen, Jie Fan, Ruichao Li, Haoyu Li
Northwestern Polytechnical University, Xi'an P.R. China

ABSTRACT

The Icarus-UAV project stresses on building an autonomous air vehicle in confined environment with relatively low cost and simple structure. The vehicle is based on a 4-rotor flight, with an FPGA as the central processing unit and transfers all necessary data to the remote workstation. The vehicle plays a role as a data collector and command executor thus avoiding extra computation unit for the vehicle. A SLAM algorithm is used for navigation with the help of Laser-3D environment remodeling unit. A smart coding pattern enables the Icarus-UAV team to program at a more abstract level for specific mission. These features enable the Icarus-UAV to execute the 6th mission of the International Aerial Robotics Competition.

1 INTRODUCTION

1.1 Problem Statement

The 6th mission of the International Aerial Robotics Competition set a background story and with the purpose of promoting the indoor navigation technology using small autonomous air vehicle. The story and mission requirements are descried as follows:

“A highly confidential prospectus, the paper which closely related to destroying the global benefit, is hidden in the safety department located on a remote village!” said by Mole, the agent who has been lurked in the Republic of Nali. In order to ruin Nali’s conspiracy, we have a plan to make a small independent aircraft to get the specific information from enemies’ security gap which has been explored in advance.

Requirement 1: In 10 minutes, slip into the gap; change the flash disk and retreat secretly. Take the flash disk to the judge. Missions complete.
Requirement 2: Come to light: Set off the alarm and fail in the slipping; change the flash and retreat while take it to the judge. Mission failed.
Requirement 3: Abort the mission. Mission failed.
Requirement 4: Failed either requirement 1 or 2; detonate the aircraft and destroy the flash before the detection limit.

The mission should be divided into 3 levels:

1. The recognition of the environment and stability of the aircraft.

2. The design of the optimal route.
3. Mechanical design for grabbing and changing the flash device.

1.2 Conceptual Solution

The Icarus-UAV team uses a commercial quad-rotor frame and a NAZA autopilot (both are products of *DJI-Innovations* Company) as air vehicle in order to focus our mind into building a set of simple and cheap devices that can turn a regular quad-rotor into a fully autonomous UAV. We use a self-made laser rang finder to detect the environment in order to perform Simultaneous Localization and Mapping (SLAM) algorithm. We also use additional altimeters including barometer and sonar to determine the height of vehicle. ZigBee protocol is the way we use to achieve real time data exchange wirelessly. Inside the air vehicle, an FPGA (Field Programmable Gate Array) chip is used to act as a central processing unit in order to gain high parallel computation and real time response performance and cut down cost as well as weight and power consumption. A remote ground workstation (PC) is used to perform imagine processing, SLAM algorithm and other task logic computation.

1.3 System Architecture

As we can see from the system architecture diagram below, the FPGA chip almost connects every other part on the air vehicle. In consideration of high performance in concurrent processing and real time data exchanging, we chose FPGA instead of DSP (Digital Signal Processor) or ARM (Advanced RISC Machines) solutions. The FPGA acts as a middle ware between the NAZA Autopilot and the safety pilot command receiver. In manual control mode, the FPGA acts as wires, i.e. let the signal form receiver directly goes into NAZA autopilot; in auto navigation mode, it breaks down the signal sent to the NAZA, uses the commands form ZigBee wireless link and mixes them with current sensors' signal and sends revised commands into NAZA. Sensors connected to the FPGA include a self-made Laser rang finder (laser scanner) and two altimeters.

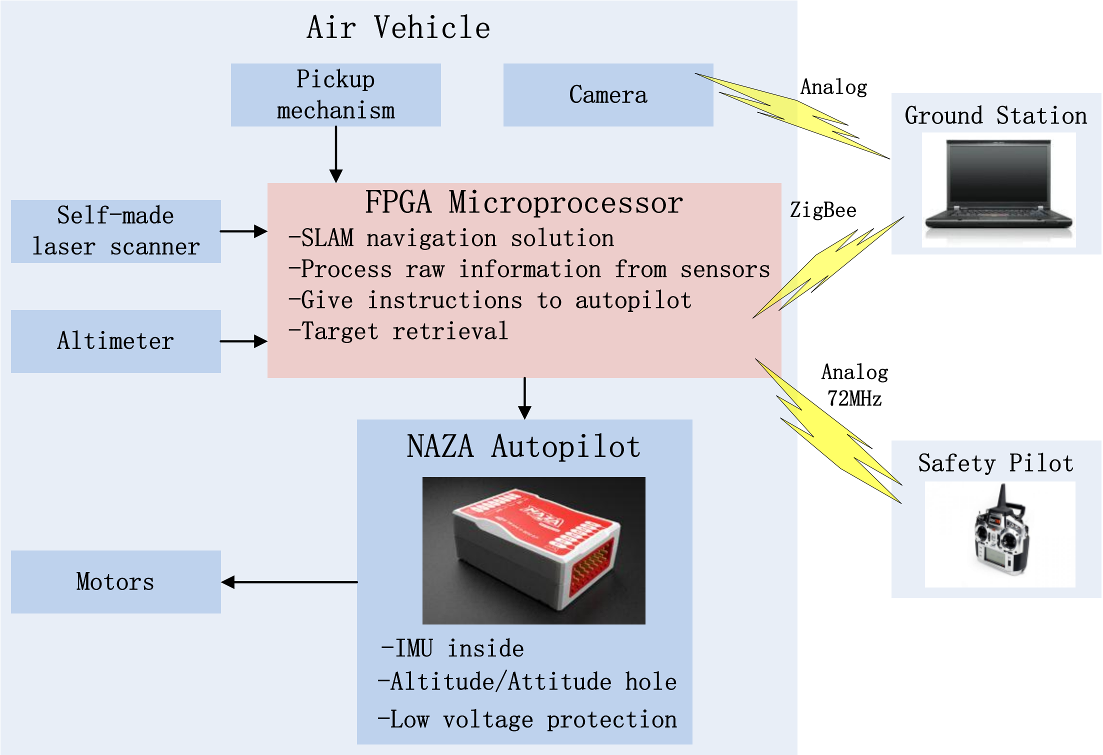


Figure 1. System Architecture

1.4 Yearly Milestones

Air vehicle development is scheduled for 2012/03. 3D environment rebuild based on infrared and FPGA is scheduled for 2012/04; Laser-2D environment rebuild unit is scheduled for 2012/05; SLAM tests are scheduled for 2012/05.

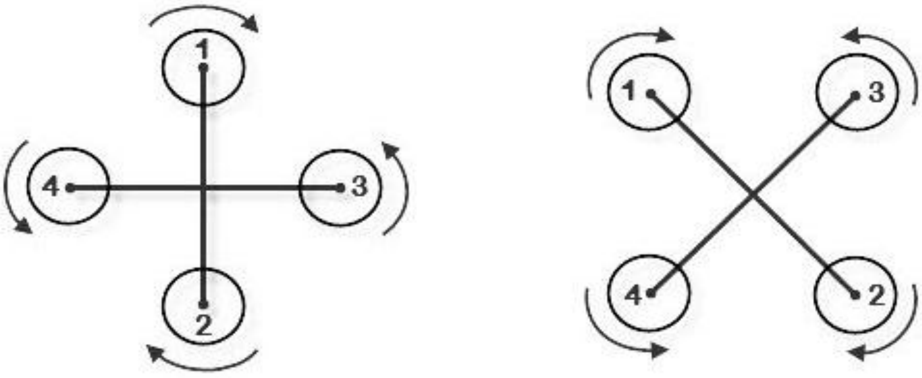
2 AIR VEHICLE

The Icarus-UAV team uses a commercial quad-rotor as the air vehicle in order to stress our attention into its payload, SLAM algorithm and task programming. The air vehicle weights about 850g and has good performance even when total weight is added to about 1500g with payload. The frame model we used is called *Flame Wheel 330*, provided by *DJI-Innovations*. The picture below shows a top view of the air vehicle.

Figure 2. Air Vehicle

2.1 Propulsion and Lift System

Quad-rotors commonly exists two flying mode, cross mode and X mode, the rotation direction of rotors is shown in Figure 3:



Cross Mode X Mode

Figure 3. Quad-rotor Flying Mode

To control cross mode is relatively easier than X mode, but considering with the fact that to have obstacle as less as possible in the forward looking direction, we take the latter as our flying strategy. In the X flying mode, increase or decrease the speed of all rotors will lift or lower the UAV. Changes done to any of two adjacent rotors can make it leans to the corresponding direction while increasing the speed of rotors on the diagonal line will make it rotate.

2.2 Guidance, Navigation and Control (GNC)

We use a commercial lightweight multi-axis control platform produced by *DJI-Innovations* (Figure 4), named NAZA, to stabilize the UAV. We use this control platform for reasons as follows:

It makes UAV have high quality of stable performance and good mobility at the same time.
It’s a small all-in-one module includes controller, gyroscope, accelerometer and barometer; it saves room and reduce the weight; it supports multiple flying modes and can be upgraded easily through Internet. In addition, it cost fewer compared with developing one alone and our development cycle is shortened.

Figure 4. NAZA Autopilot

Stability Augmentation System and Attitude stabilize methods

NAZA applies a gyroscope to sense the attitude of UAV, which has high sensitivity, low drift rate and is very suitable for sustained steady flight. Accelerometer measures accelerator to help control moving speed. Barometer for flight with fixed height.

Figure of control system architecture

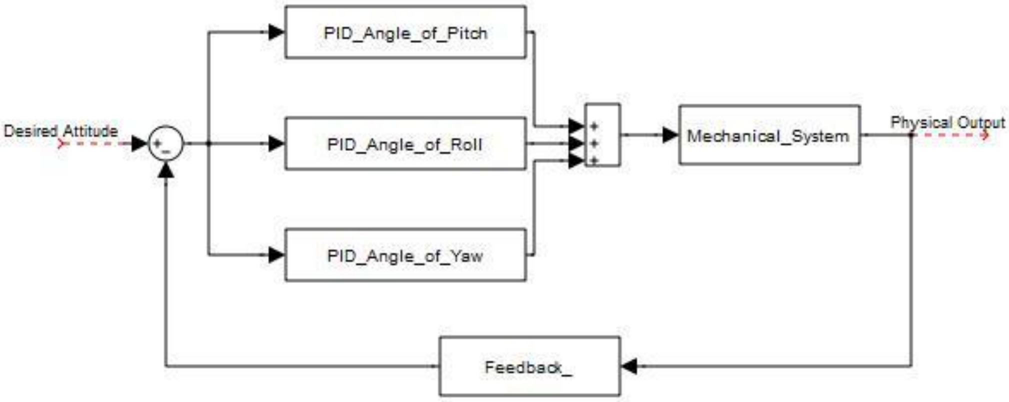


Figure 5 Control System Architecture

Navigation – SLAM Algorithm

SLAM is an acronym for the Simultaneous Localization and Mapping algorithm. SLAM addresses the problem of simultaneously recovering a map and the air vehicle pose from sensor data. The map contains N features (landmarks) and shall be denoted Θ=$\theta_1, \ldots, \theta_N$. The path of air vehicle will be denoted $s^t = (s_1, s_2, \ldots, s_t)$, where t is a time index and S_t is the pose of the air vehicle at time t.

The SLAM algorithms calculate variants of the following posterior probability distribution:

$$p(\Theta, s^t | z^t, u^t, n^t)$$

Where $z^t = z_1, z_2, \ldots, z_t$ is a sequence of measurements (e.g., range and bearing to nearby landmarks), and $u^t = u_1, u_2, \ldots, u_t$ is a sequence of the air vehicle controls. The variables $n^t = n_1, n_2, \ldots, n_t$ are *data association variables* --- each n_t specifies the identity of the landmark observed at time t. Initially, we assume n^t is known; we relax this assumption below.

A variety of SLAM algorithm implementations are available for free use at the web site *OpenSLAM.org*[1]. We will use the *GMapping* algorithm for our MAV. Using the algorithm required the following works:

- Build the motion model of the air vehicle, in the form of the conditional probability distribution $p(s_t | u_t, s_{t-1})$. This distribution describes how a controlu_t, asserted in the time interval $[t-1, t)$, affects the resulting pose.
$$p(s_t | u_t, s_{t-1}) = g(s_t, \theta_{n_t}) + \varepsilon_t$$
- Additionally, the air vehicle is given a probabilistic measurement model, denoted $p(z_t | s_t, \Theta, n_t)$, describing how measurements evolve from state.
$$p(z_t | s_t, \Theta, n_t) = h(u_t, s_{t-1}) + \delta_t$$

Here g and h are nonlinear functions, and ε_t and δ_t are Gaussian noise variables with covariance R_t and P_t, respectively.

Navigation – A(A-Star) Path Planning Algorithm*

A*(A-Star) arithmetic is an effective method to find the solution of shortest path in static network. Formula:

$$f\ (n) = g\ (n) + h\ (n)$$

f (n) means the evaluation function started from the initial point, passed by node n to the target; g (n) is the actual cost from the initial point to node n; h (n) is the estimated cost of the shortest path from node n to the target.

The chosen of evaluation function, h(n), is the key to get the shortest path, in other words, the best solution.

To avoid the aircraft's track rules, take the Euclidean distance between the nodes based on the grid as the estimated cost. So once g(n) is unchanged, f (n) will limited by h(n), ensure the aircraft will force on the shortest path (node is close to the target(n) is decreasing with h(n)), the plan which is much efficient and effective than Dijstra arithmetic.

2.3 Flight Termination Unit

In order to guarantee that flight mission can be aborted, a separated kill switch has been applied. Kill switch is defined as the fifth channel of the safety pilot. When the kill switch triggered, the Icarus-UAV will cut off power supply after it descend safely.

A manual takeover switch is available so that a human safety pilot can take over the control at emergency situation. Also, the Icarus-UAV has a built in fail-safe that kills the motors and takes an automatic descent if the pitch/roll angles reach a certain threshold which is defined at 70 degree.

3 PAYLOAD

3.1 Sensor suite

1. The NAZA autopilot provides the basic GNC sensors as well as auto stabilizing control signal. The build-in GNC sensors include a triple axis accelerometer and a triple axis angular speed sensor.
2. A self-made scanning laser range finder. It consists of a Sony ¼ inch 420 lines analog CCD Camera and an 808nm infrared laser beam generator. An additional 810-1100nm filter lens is fixed to the camera in order to avoid extra visible light.
3. Another ¼ inch 420 lines Sony CCD camera is used for image processing.
4. A Barometer is used for absolute altitude measurement.
5. A Sonar is used for relative altitude measurement.

3.2 GNC Sensors

Altimeter and NAZA autopilot

The NAZA autopilot provides the basic stabilizing and controlling support, which has been introduced in the previous sections. The attitude of Icarus-UAV is controlled by NAZA autopilot. It is a small all-in-one module includes controller, gyroscope, accelerometer and barometer. The sensors in NAZA autopilot guarantee the Icarus-UAV's attitude and altitude stability. Because the barometer in NAZA can’t be read out, so an ultrasonic height sensor located in the bottom of the Icarus-UAV is needed for estimate relative altitude. In addition, an extra barometer is used to get absolute altitude so as to measure the height of the object below the air vehicle.

Laser range finder

In order to get range information for SLAM, a self-made laser range finder is set up in the front of the Icarus-UAV. The unit consists of a ¼ inch 420 lines Sony CCD camera and an 808nm infrared linear laser beam generator.

The fundamental principle [4] of the self-made laser range finder is based on geometry method. Consider a 1-dimension case for simplicity which is illustrated in Figure 6. The system consists of an imager and a laser generator. The imager can be any kind of sensor matrix sensitive to the laser beam. In this case, we use an infrared sensitive camera because it is easy to get form regular shops. The target whose distance is to be measured is marked as “object”.

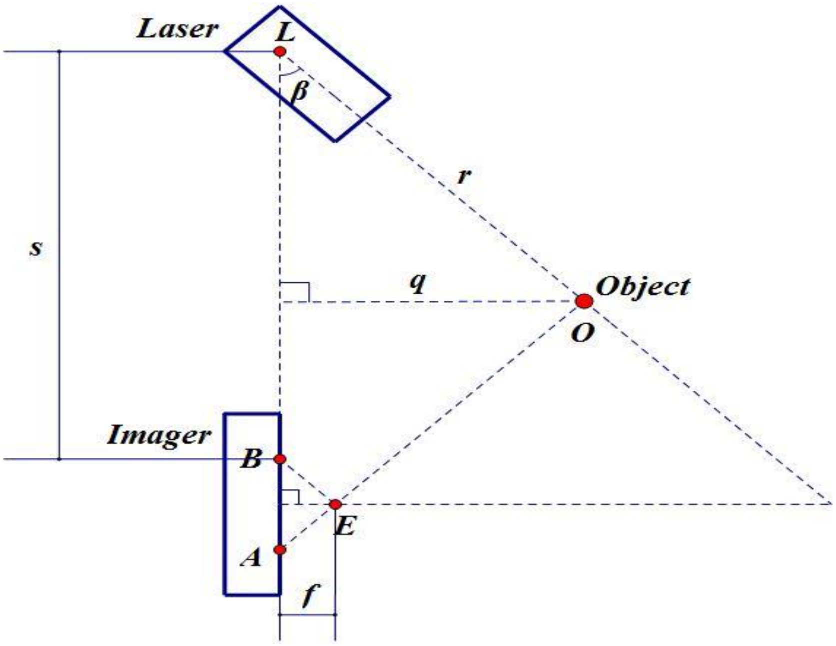


Figure 7. 1-Dimension Laser Distance Measurement

The laser generator and the imager are secured on a base board (line ABL) and the distance "s" is a known constant. The Imager is parallel fixed with the base board while the laser generator has an angle of "β". "f" indicates the focal distance of the imager and is a constant. Variable "x" indicates the distance between point "A" and "B". "A" is the projection of object lighted by the laser beam; we get "B" when we draw an imaginary line parallel to the laser beam "LO" and goes through point "F"(The focus of the imager). "r" indicates the distance we want to know between the scanner and object. "r" can be calculated via the following equation using the law of similar triangular ($\triangle \mathrm{FAB} \sim \triangle \mathrm{OAL}$) :

$$\mathrm{r} = \frac{1}{\sin\beta} \cdot \frac{\mathrm{f} \cdot \mathrm{s}}{\mathrm{x}}$$

The variable "x" is in proportion to the distance of the object which can be calculated using the following equation:

$$\mathrm{x} = \mathrm{PixelSize} \cdot \mathrm{PX} + \mathrm{Offset}$$

PixelSize indicates the CCD pixel size of the imager which can be known via factory manual or be calibrated. "PX " refers to the coordinate of x-axis of the imager. As the coordinate is a discrete value, hence an offset is needed for denser preciseness. This can be achieved by statistic methods [5] because the laser point in the image usually covers many consecutive pixels.

The above method can be easily extended to 3D measurement, as long as we turn the laser point into a line, and add a rotation pivot at the center of the base board, as is shown in figure 6. Note that each "PY" (coordinate of y-axis of the imager) reflects a flat surface similar to that in figure 5. "f' " is a fixed "f" using the Pythagorean theorem:

$$\mathrm{f}' = \sqrt{(\mathrm{PY} \cdot \mathrm{PixelSize})^2 + \mathrm{f}^2}$$

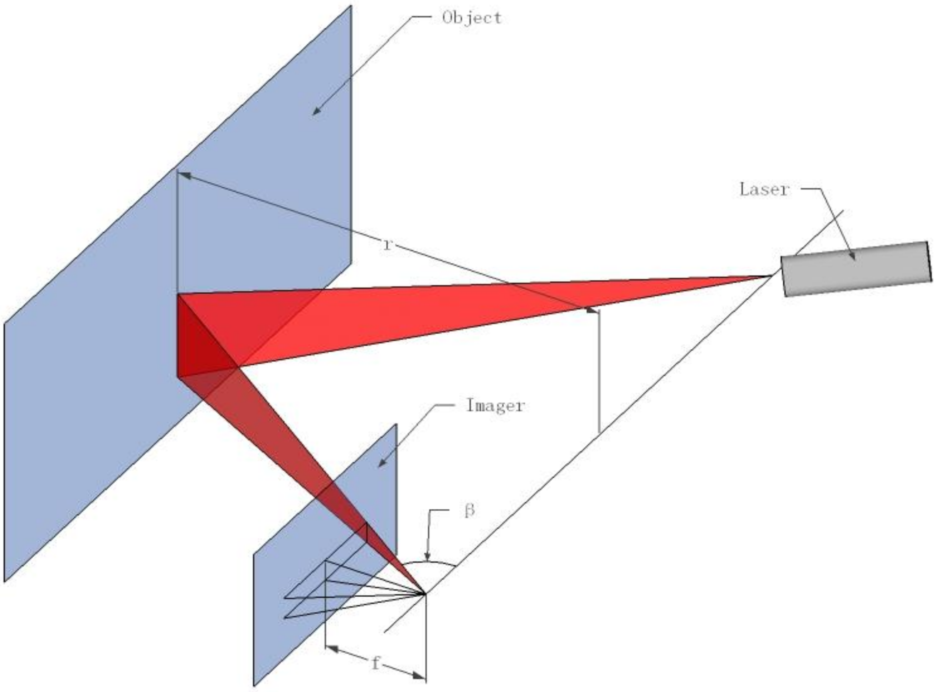


Figure 8. 3-Dimension Laser Distance Measurement

The following picture shows a whole view of the self-made laser range finder.

Figure 9. Self-made Laser Range Finder

The following picture shows a snapshot from the camera when we are testing and trying to measure the distance of a stack of books.

Figure 10. Testing Laser Range Finder (The Purple Line Is Laser Beam)

The angle “β” we chose is 83° and the “s” is 150mm. Some adjustment and calibration methods are needed when programming software in order to counterstrike the impreciseness due to mechanical assembling.

The preciseness of the whole laser scanner can be estimated as follows. We can rewrite the equation $r = \frac{1}{\sin\beta}\cdot\frac{f\cdot s}{x}$ as follows and turn it into differential format:

$$x = \frac{f\cdot s}{\sin\beta}\cdot\frac{1}{r} \xrightarrow{\text{differential}} dx = \frac{-f\cdot s}{\sin\beta}\cdot\frac{1}{r^2}\cdot dr \ \ (f, s, \beta \text{ are constants})$$

It’s getting clear that when “r” gets larger, |dx| gets smaller. However, the resolution of x is determined by PixelSize and is fixed, which means the preciseness of the scanner is lowered as the distance between scanner and objects is enlarged. After some experiments, we can ensure our scanner have acceptable accuracy at range of 3 meters, enough to perform SLAM afterwards.

3.3 Mission Sensors

Target Identification – SIFT

In our system, the image matching will works for the recognition of the flash drive, the 3D reconstruction, the stereo matching and the target tracking.

The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. We can use SIFT to solve the problem of target matching, there are the following steps[6]:

a. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

b. Key point localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

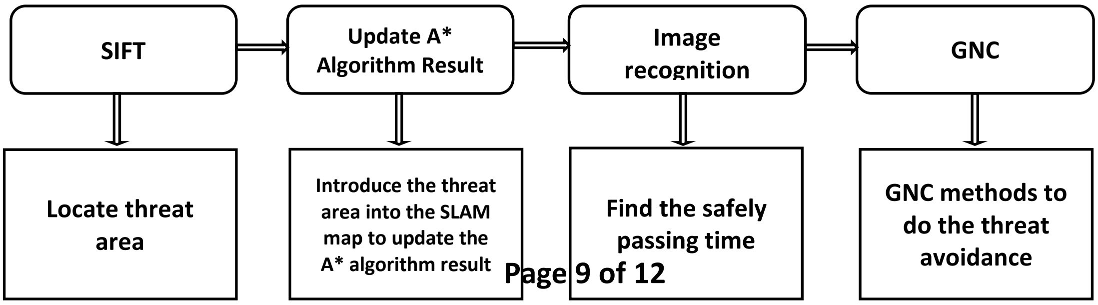
c. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image ata that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

d. Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

e. Geometric correction: Delete the exterior point.

Threat Avoidance

We can use the method described in the following flow chart to do the threat avoidance:



Using the method above, we can find and locate the threat area in the real scenes. Then we can introduce the threat time and area into the SLAM map to update the $A*$ algorithm result. The safely passing time can be obtained by the image recognition algorithm. At last, the Icarus-UAV can do the threat avoidance by the GNC methods.

3.4 Communications

The FPGA unit exchanges data with the ground workstation via Zigbee wireless link at 2.4 GHz. The interface between FPGA and Zigbee module is serial port. A Hitec Optic 6 safety pilot is used for manual control and safety purpose which operates under 72Mhz. Data of cameras is sent directly to the ground workstation via analog video signal.

3.5 Power Management System

With 2500MAH lithium polymer 3 cell 11.1V batteries, the Icarus-UAV can get approximately 15 minutes of battery life. The system has a two level low voltage protection. The first level of protection voltage is set to 11.1V. A red LED on the Icarus-UAV would alarm when the voltage goes below 11.1V. As the descending voltage reaches to 9.6V, it means the battery would be damaged for further usage and the second level of protection is activated. In order to protect battery and avoid the Icarus-UAV out of control, the Icarus-UAV will land automatically. Battery voltage can also be measured and transmitted to ground workstation for the sake of voltage monitoring.

4 OPERATIONS

4.1 Flight preparation

Before each flight, a check list is required to ensure a safety and smooth flight. Each team member has learned the following checklist. In addition, each teammate has a specific item for him/her to perform under other members' monitoring for utmost precaution.

Checklist

TABLE 1. CHECK LIST FOR FLIGHT PREPARATION

Step 1. Check Appearance	Ensure no screw is loosen Ensure all plugs is correctly connected Ensure no damage has done to propellers Ensure battery and controller are secured stably
Step 2. Check electronic devices	Check the voltage of battery Check electronic plugs are all correctly connected
Step 3. Check communication	Ensure no other 72Mhz wireless device is working around Ensure monitoring LED works properly Ensure safety pilot works well under manual control Ensure the mission termination switch works well Ensure Zigbee data link works well Ensure The ground work station can get proper images
Step 4. Test flight	Let the Icarus-UAV fly a pre-programmed initial program at a low height to ensure it operates well

4.2 Man/Machine interface

The state of air vehicle is sent to the ground workstation via ZigBee data link, which makes it possible for us to build software to monitor these variables and perform analyzing methods. Also, as the data link is a bidirectional one, control commands can be sent to the air vehicle by pressing buttons via the software.

4.3 Risk reduction

4.3.1 Vehicle Status Monitoring

High luminance LEDs are used to show current vehicle statues. In normal cases, the yellow LED will flash at a low frequency; when problem occurs, for example, the battery voltage is too low for the vehicle to operate, red LED will flash. This is a simple but highly robust way for monitoring, even when the wireless data link has broken. In addition, this information will also be transferred to remote workstation (if the data link is working) for further processing.

4.3.2 Shock/Vibration Isolation and Safety

The vibration of Icarus-UAV is mainly generated by 4 motors as well as the propellers. In normal cases, the overall vibration is not major threats to the device inside the air vehicle because the mechanism is properly balanced. Even though the threats are small, the Icarus-UAV team still uses several ways to isolation vibration and protects autopilot with its accelerometers, decreasing the chance of electronic failure. This is done by the following methods:

1. Low density sponge is used as gap when securing devices like 72 MHz receiver, ZigBee module and FPGA unit.
2. 3M double face adhesive tape is used to secure the NAZA autopilot. Sponge is not used because the accelerometer inside NAZA needs to be sensitive to any tiny change of the air vehicle.
3. The air vehicle frame is properly designed. We have been showed with a video showing the air vehicle frame under factory testing condition, which indicates highly robust of the frame itself against shock and damage.

4.3.3 EMI/RFI Solutions for safety

EMI/RFI Solutions for controller

One consideration for us to choose FPGA as the centurial processing unit is for its robust feature under complicated EMI/RFI environment compared with high speed DSP. This is because the Crystal Oscillator we use is only 50 MHz, and the PLL inside the chip will upgrade the frequency to 150 MHz. In other words, high frequency operation (the most vulnerable part) only occurs inside a single FPGA chip, ideal for isolation methods like Cooper Pour.

EMI/RFI Solutions for communication

ZigBee itself is a highly robust indusial standard protocol, capable of frequency hopping when RFI occurs. Furthermore, in order to lower the chance of losing control even more, we have chosen a module with transmitted power of 200mw.

We have also developed robust software capable of navigating and safely terminating the air vehicle when the ground workstation lose signal of raw camera image.

4.4 Modeling and Simulation

We are using QT SDK to build our simulation environment including following segments:

Use ODE (Open Dynamic Engine) to build a virtual physical environment for testing the auto navigation algorithm. We've chosen ODE because it is simpler and open source. In the simulation environment, the Icarus-UAV physical module will be built as well as the surroundings. The engine can generate gravity, detect impact, which is ideal for algorithm modeling and simulation.

We have chosen QT as development tool because we can both run our code under Windows and Linux.

4.5 Testing

The Icarus-UAV will be both tested under simulation and real condition. The simulation testing is a precondition of the real flight testing and this reduce the risk of air vehicle crash due to improper coding.

5 CONCLUSION

The Icarus-UAV team has built a fully autonomous quad-rotor, capable of challenging the 6th mission of the International Aerial Robotics Competition. The air vehicle consists of a commercial affordable frame and a NAZA autopilot. The Icarus-UAV team mainly stress on building a set of devices to turn an ordinary quad-rotor into an autonomous UAV. Unlike other solutions, the Icarus-UAV uses an FPGA as the central processor, thus gaining high performance in parallel computing, real time data exchanging and lower battery consumption. The Icarus-UAV team also build an infrared laser rang finder to help do the SLAM algorithm. Finally, imagine processing including blue LED detection, flash drive recognizing and laser barrier detection is achieved by wireless camera and the ground workstation.

6 ACKNOWLEDGMENTS

We would like to thank *DJI-Innovations*, the sponsor who provide us with the basic air vehicle frame as well as enhanced gyro-sensor based stabilizing control unit called NAZA.

7 REFERENCES

[1] http://openslam.org/
[2] Montemerlo M, Thrun S, Koller D, et al. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges[C]. Proceedings of the International Conference on Artificial Intelligence. Acapulco, Mexico: IJCAI, 2003: 1151-1156.
[3] Montemerlo M, Thrun S, Koller D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem[C]. Proceedings of the National Conference on Artificial Intelligence. Cambridge, MA, USA: MIT Press, 2002: 593-598.
[4] K. Konolige, J. Augenbraun, N. Donaldson, C. Fiebig, and P. Shah. A low-cost laser distance sensor. In Int. Conference on Robotics and Automation (ICRA), 2008.
[5] Mertz, C., J. Kozar, J.R. Miller, and C. Thorpe. Eye-safe Laser Line Striper for Outside Use. Intelligent Vehicle Symposium, 2002.
[6] David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.