Autonomous Air Vehicle Implementation Into The 2004 International Aerial Robotics Competition

Adam Felske, David Muncaster
DeVry Institute of Technology
Calgary, Alberta
http://numun.ath.cx/

Abstract
DeVry Institute of Technology, Calgary's Autonomous Air Vehicle (AAV) team is to compete in the 2004 International Aerial Robotics Competition (IARC). The team is to make use of a multipurpose combination helicopter and ground vehicle platform to compete in all levels of the IARC competition. This document contains information pertaining to AAVs plan of attack, the components used, and the systems implemented.

INTRODUCTION

IARC in itself is 4 separate competitions which embodies an overall goal. The first stage is comprised of a 3 km ingress along predetermined waypoints. The second stage is the identification of a specific building, all of the 'viable' openings (viable being a portal greater or equal to 1 x 1 meters) and communication of this fact back to the base station. The third stage is to then venture into the building and identify a target within the building based on the predetermined mission specifications (outlined in the IARC handout). The fourth, and possibly most difficult, stage is a time limit of 15 minutes for the first three stages. The stages must be completed in order for the next stage to be available. Thus enters our plan to complete all four stages to end this phase of IARC.

AAV (Autonomous Air Vehicle) is an autonomously flying helicopter with sub-vehicle being built and designed to be flexible enough to complete IARC with relative ease. The team is comprised of 2 EET students, 2 CIS students, and 2 BSOP students. The EET students are: David Muncaster: team leader; Adam Felske: navigations, GPS mechanic and co-leader. The CIS students are: Nabeel Khan: Helicopter AI, and Mapping Software; Thomas LaFreniere: Base Station Software, Drivers. The BSOP students are: Jana Hall, and Lily Tumax. The BSOP students are in charge of the entire business aspect of the project. The following is an account of our methodologies for the completion of all four stages.

Initial Overview

With a look at the rules governing IARC, our first concern was initial size of the primary vehicle. Our first approach was to create an "all-in-one" solution that would be able to do all of the stages. However, when it came to traveling, identification, traveling inside and then more identification there would simply not be enough time to have an "all-in-one" solution. Thus we decided to use a primary vehicle and a sub-vehicle, released from the 1x1 meter constraints for the main vehicle, we concentrated on the physical and digital architecture and ‘thought process’ of a robot that has very limited intelligence. Our only concerns for the physical aspect of the vehicle were that it could sense, see, and feel its movements and surroundings. By using this critical type of thinking we were not restricting ourselves to a specific size, in which we would have to make things fit. Our first priority was the first stage of IARC, the deceptively easy 3 km ingress. For this, we laid out a list of items that the vehicle would need to keep track of:

- Position in 3d space
- Orientation
- Stability (outside forces i.e. wind)
- Height above ground
- Speed
- Next waypoint
- Stage status

For the building identification (second stage) we identified more items:

- Vision
- Hovering ability

And finally for the tapestry identification inside the building (third stage) we identified:

- Entry via a 1x1 m portal
- Positional location inside the building
- Terrain navigation.

Taking a look at all the specifications of the three stages, compounded with the 15 minute time limit of stage 4, we decided the parts we would need are:

- Helicopter
- Processor capable of nearly 10,000 MIPS (Million Instructions Per Second)
- IMU (Inertial Measurement Unit)
- GPS (Global Positioning System)
- Gimbaled Camera with a large field of movement
- MRA (Miniature Radar Altimeter)
- Data link with 10 km + range
- Back-up Data link with 50km + range
- Tracked ground vehicle
- Laptop computer, web-cam, 54G wireless card for ground vehicle
- 54G access point
- Servo Controller

With the identification of the parts and the basic outline of our system completed shown below in Figure 1.1. We began outlining the steps that would be taken to complete IARC.

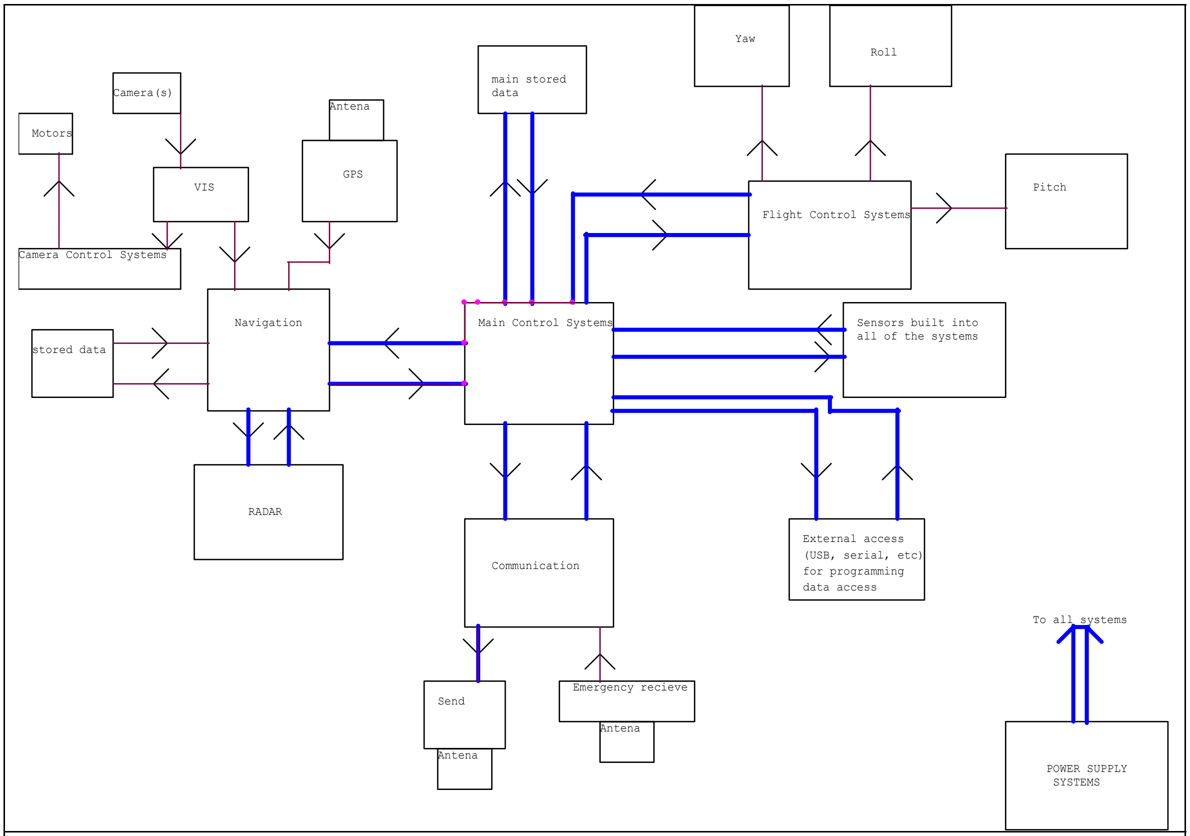

Figure 1.1 Outline of our process handle and control diagram flow

For implementation into IARC we had to stay along a fully autonomous approach with only the initial ignition sequence being human controlled. Therefore, we began to allocate time to each of the stages in IARC. For simplicity and a healthy time margin, we decided that 5 minutes per stage would be an attainable goal. At that pace, the helicopter would need to fly at a minimum speed of 36 km/hour. For stage 2, a payload of 45 pounds would be needed to carry the sub-vehicle to the target building. And for stage three, the helicopter would need a hover time of 5 minutes while the sub-vehicle was deployed and worked inside the building. With all of the specifications in mind, the air vehicle, sub-vehicle and all of the components necessary were found and the implementation was begun.

Air Vehicle: Helicopter

While a helicopter is difficult to control, it is the only available air vehicle that can fill all of our needs. The helicopter that was chosen, was a prototype target drone developed by Boeing Canada, and donated by Schreiner Target Services. The helicopter weighs 150 pounds without drone, has a rotor diameter of 11 feet, a 30 hp engine, and lifting capabilities of 100 pounds. The flight time is currently unknown, but is estimated at over 2 hours based on the 10 liter fuel tank. The sensors are mounted in varying positions around the helicopter, but the main processing center is located at the nose of the aircraft along with the power systems. The computer and power systems are mounted in a custom built case that is designed to survive a 25 foot fall with minimal damage (1/8 inch aluminum was used). Power is provided to all systems (averaging 500 watts) by a

24 volt alternator and back-up batteries. By balancing most of the weight of the helicopter and its sub systems, the helicopters will be able to make the 3 km high speed dash with the sub-vehicle without many balancing issues. However, when the drone is deployed, the helicopter will have a harder time keeping balanced, but this is irrelevant because the helicopter does not need to move very fast, it only needs to stay in orbit above the target building. With this in mind, the next section will briefly go over the sub-vehicle.

Sub-Vehicle: Treaded vehicle

A two track, treaded vehicle was chosen to be our ground vehicle. The tracks will be positioned similar to a tank; the treads will stretch from the front to the back of the vehicle. A camera will be mounted to the vehicle, giving it object recognition and navigation information, which will be fed back to the helicopter via 802.11 wireless network protocols. The total weight of the vehicle will be less than 35 lbs. This will allow it to be carried easily by the helicopter, and allow it to travel through the building at high speeds to cover the highest area in the shortest possible time. The helicopter will be able to take control of the vehicle at any time if needed; however, the vehicle will be able to run autonomously, with no control required from the outside.

Vision System

While other parts of the AAV design and implementation are essential to the success of the competition, they are all relatively easy goals to accomplish (with the possible exception of the 'intelligence'). However, one of the most crucial and most difficult to implement aspects of both stages 2 and 3, is the use of an intelligent vision system that can distinguish between different objects. The AAV team has decided upon a multi-segment vision system with multiple layers which will help filter out not only entire pictures that do not meet the required specifications, but also help cut down on the amount of calculations. We estimate that at a rate of 30 frames per second, the amount of processing power needed will be roughly 30-45% of our available processing power. This value will be greatly dependant on how sensitive the vision system is set to, and how many similar "hits" are found in single frame. Shown below is a block diagram (Figure 1.2) showing a rough outline of the vision systems process of manipulating the data sent from the camera.

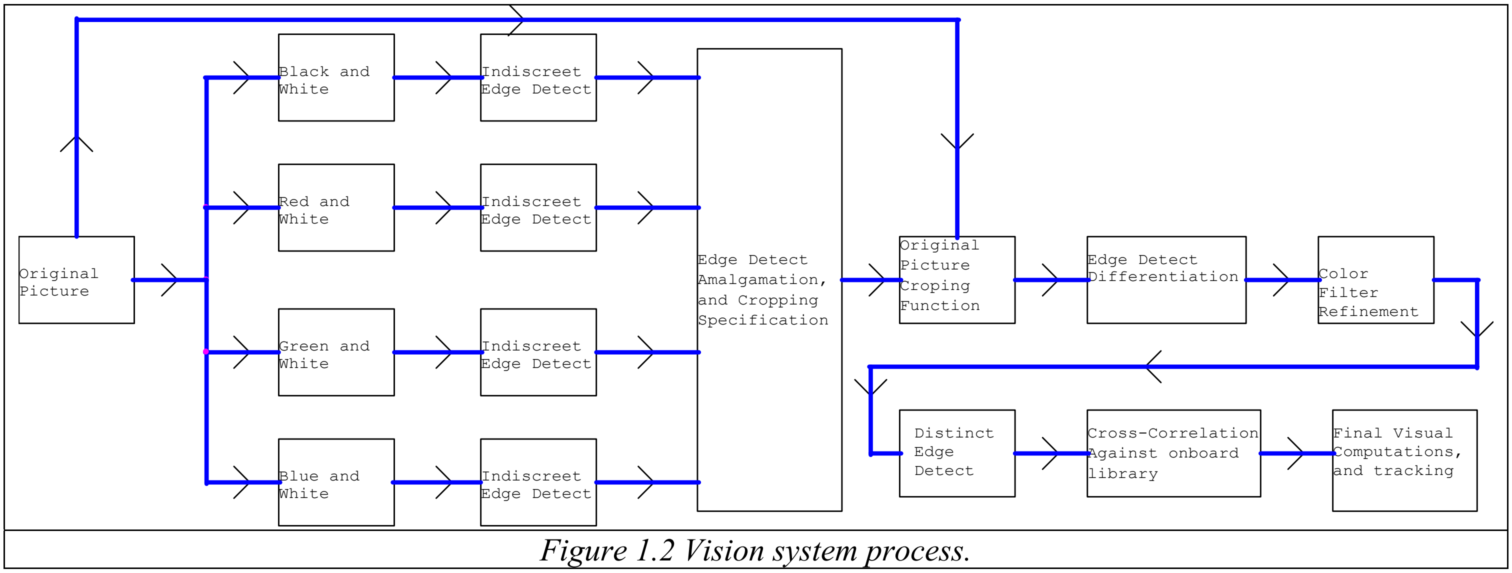

Figure 1.2 Vision system process.

The block diagram shows how the pictures will be processed, although at first glance it may seem that there is a lot of redundancy, the different levels of edge detect are in place so as to help reduce the overall number of calculations needed. The block diagram can be separated into two sections, each with there own specific purpose.

•First section: Original and color conversion into indiscrete edge detect.

-This section is to help minimize the number of complex calculations that will be done later. What happens is the original picture is multiplied into 4 separate pictures, these pictures are then reduced to the base colors and then each pixel is reduced to either one extreme or another. One example would be for the top filter (black and white). In this filter, if the pixel information is ‘123, 213, 12’ the result would be ‘0, 0, 0’. This is because the majority of the RGB information is below a value of 127. A similar process is also done for the other 3 filters. Once the filtering is completed, each picture is sent into its own indiscrete edge detect. The goal of the indiscrete edge detect is to locate all instances of a specific shape (circle for the IARC symbol) with very loose parameters. This will help defend against a shape being rejected because the point of view is at a large angle. Also, another spin off from locating a shape with generic inaccuracy is that the number of calculations needed is less then if high accuracy edge detect was implemented. Once a shape or shapes are found the next step will be to combine all four of the edge detects to find out if the shapes located are of particular interest. If the combination of all 4 edge detects comes with up with a positive ‘hit’ on 2 or more of the filtered pictures in the same spot (give or take 10 pixels or so), then that spot, or shape, is considered to be a positive hit. Once a positive hit is found (can be more then one in a single picture) the pixel locations are then noted and a log file which will accompany the original picture for the rest of its journey through all of the filters is created. This log file contains all of the data pertaining to that original picture, as well as all of the data that has been discovered about the picture from each step. This log file will be used as a history file, and will not only aid in the debugging stage but it will also enable the vision system to interact with other helicopter systems by simply sending log file data to them.

- Second Section: Original Picture cropping and multiple filtering algorithms

  -The second section can be broken down into sub-systems that are all intended to aid in the identification and classification of the particular 'cropped' image. The first step of the second section will be to take the original picture and copy the outlined areas that were identified in the first section. The copies are then appended to the end of the same picture file so as to keep all pertinent data in the same place. With the regions separated, the next steps will be:

  1) Edge Detect Differentiation: This will be intended to filter out, or differentiate, shapes that were picked up that were somewhat similar to the intended shape of the intended target, but are obviously different (a square with rounded corners). As with all steps, the EDD will append its findings (if any) to the log file.
  2) Color Filter Refinement: This step will use a percentage calculation of the colors present in the identified target and compare them to that of the image the vision system is looking for. Again, the information that is found by the CFR will be appended.
  3) Distinct Edge Detect: The DED will use a much more precise edge detect routine combined with the previous edge detects to determine with greater accuracy not only the general shape of the object (circle for the IARC logo) but also the contents of the shape. This will be accomplished by a combination of knowledge of the camera angle/orientation, distance from ground, GPS values and pixel information. Thus the IARC 'X' can be found with the lettering. Again, the information that is discovered by the DED will be appended.
  4) Cross-Correlation: A DSP (Digital Signal Processing) cross-correlation algorithm will be used to determine:
     i. What angle the targeted image is being viewed at.
     ii. What percentage likelihood is the image viewed the needed image.
     iii. What orientation the target image is being viewed at.

     All of this information will also be appended to the log file.
  5) Final Visual Computation: What this final step will do is take all of the logged information, and use it as a report to the main processing core. The information received will contain:
     i. If the picture viewed was completely in the frame, or if part of it was missing (i.e. only half of the image was found) and which direction the camera will need to turn.
     ii. Approximate distance to target
     iii. Using past pictures, as well as current picture, whether or not the picture is becoming more visible or less visible, and suggested flight corrections (i.e. go back to 'X' GPS coordinate, we had a better picture there).
     iv. Current state of the visual mapping that is being generated (whether or not all sides of building 'X' has been seen).

v. Simulated GPS coordinates for a hover/landing spot.
vi. Information relating to Ground Drone deployment.

Navigation

The AAV navigation systems are comprised of:

- IMU
- GPS
- MRA
- Camera

The IMU gives the navigational system the information it needs to be aware of its location in a 3D world. The X, Y and Z coordinates are all obtained by the IMU and processed within the computer. This device supplies the navigational system with acceleration, the information needed for flight stability and with GPS, an idea of the crafts current heading.

Not only does the GPS enable to craft to get its current heading information, it allows the AAV platform to ascertain its location within a 2D top-down world. Since, the crafts mission is to fly autonomously, the GPS coordinates have to be accurate. This accuracy is essential to proper flight path determination and collision avoidance with objects in front, behind or to the sides of the craft. In order to prevent ground collision the MRA had to be implemented.

The MRA provides the helicopter with a perfectly accurate height level in relation to the terrain it traverses over. Knowing the current height of the craft is essential in preventing a crash with the underlying terrain.

Information regarding the immediate port/starboard and aft/fore barriers such as buildings, trees and walls is provided by the turret camera. The camera in combination with the MRA, allows our systems to develop a map of our immediate area. With this information, the helicopter is capable of “knowing” were it is in relation to its close range 3D maps, thus, allowing for navigation to occur in tight areas.

These four devices when combined with software, form the basis of a complete navigation system and give our platform the ability to be autonomous. All information regarding stability, heading, location in a 2D world, and a 3D map of its immediate area, provide AAV with every piece of required information to allow for a computer to take full control over the craft. Therefore, these devices are the ones that will allow us to compete in all levels of IARC.

Flight Stabilization

Flight stabilization is maintained by the IMU. The information this device provides in conjunction with PID (proportional integral differential) programming, allows for completely stable flight. Whether the helicopter is in flight, hovering, taking off or landing, the computed information provided to the system allows the craft to maintain a steady position. The IMU passes information to the control system that contains its X, Y and Z position over time. Using these constantly changing values, the PID programming prevents minor or major flight alterations by obtaining the proportional, rate and sum changes over time. The reasons all three of these types of changes need to be analyzed, is to create high accuracy corrections. If only the proportion of change over time was to be considered, then the craft would have to over correct itself to see a positive change in its last misalignment. A proportional controller allows for a quick response, but inversely it also prevents fast accurate corrections. This controller creates the affect of overshoot, and steady-state error. Thus, proper stability is not obtainable by only using proportional logic. The addition of a derivative or rate controller to the proportional controller allows for accuracy to be obtained faster. It does this by preventing the overshoot characteristics presented by the proportional controller. However, the initial steady-state error is still present in stability. In order to prevent the steady-state error in stability corrections, a summing or integral method can be applied. This method smoothes out the entire change implemented into the system. When all three of these methods are combined, PID based controller programming allows for acceptable flight stability and stable autonomous flight.

Conclusion

DeVry Institute of Technology, Calgary's Autonomous Air Vehicle (AAV) team is to compete in the 2004 International Aerial Robotics Competition (IARC). AAV (Autonomous Air Vehicle) is an autonomously flying helicopter with sub-vehicle being built and designed to be flexible enough to complete IARC.

Whether the helicopter is in flight, hovering, taking off or landing, the computed information provided to the system allows the craft to maintain a steady position. With the implementation of our sub vehicle, vision systems, navigation systems, and flight stabilization software, the AAV platform will be able to compete in all levels of IARC, and win with relative ease.