COLLISION AVOIDANCE FOR UNMANNED AERIAL VEHICLES

TAN HAN YONG
B.Eng.(Hons), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2008

ACKNOWLEDGEMENT

The author wishes to express sincere appreciation for the assistances given by the following persons in carrying out the work successfully:

Associate Professor Gerard Leng (National University of Singapore), supervisor of the project, for his patience, advice and guidance in the project.

Mr. Oi Tze Liang (fellow researcher and team mate on the project) for his patience, effort and contribution to this project.

Mr. Teoh Weilit (fellow researcher and team mate on the project) for his contribution to this project.

Mr. Kam Mun Loong (fellow researcher in Cosy Lab) for his assistance and contribution to this project.

Mr. Asfar (fellow researcher in Cosy Lab) for his assistance and contribution to this project.

The technical staff of the Dynamics Laboratory namely: Ms Amy, Ms Priscilla, Mr. Ahmad and Mr. Cheng. For their equipment and technical support in the project.

TABLE OF CONTENT

TITLE PAGE……………………………………………………………………..

ACKNOWLEDGEMENTS.. I

TABLE OF CONTENT...……................. II

SUMMARY…………………………………………………………………. V

LIST OF FIGURES ……………………………………………….............. VII

LIST OF TABLES…………………………………………………………….. IX

CHAPTER 1: LITERATURE SURVEY ……………………………………. 1

1.1 Chapter Summary………………………………………………………... 9

CHAPTER 2: DESIGN AND ANALYSIS OF COLLISION AVOIDANCE ALGORITHM 11

2.1 Proportional Navigation law in Collision Avoidance……………...…….. 11

2.2 Analysis of Collision Avoidance Algorithm……………………………... 19

2.2.1 Line of Sight Rate Sensor…………………………………………….. 20

2.2.2 Range and Heading Sensors ………………………………………….. 22

2.3 Chapter Summary………………………………………………………... 26

CHAPTER 3: COLLISION AVOIDANCE SIMULATION VALIDATION 27

3.1 Measurable Variables…………………………………………………….. 27

3.1.1 Range Reading, R…………………………………………………….. 28

3.1.2 Obstacle Heading Angle, ζ …………………………………………... 28

3.2 Simulation Models……………………………………………………….. 29

3.2.1 Simulated UAV Dynamics Model……………………………………. 31

3.2.2 Simulated Collision Avoidance Model ………………………………. 34

3.3 Simulation Results for Collision Avoidance Performance………………… 36

3.4 Chapter Summary………………………………………………………... 38

CHAPTER 4: CAS ARCHITECTURE & IMPLEMENTATION………… 40

4.1 Flight Platform Information……………………………………….............. 40

4.2 UAV Flight Computer ……………………………………………………….. 42

4.2.1 PC104 3-stack Configuration ………………………………………… 43

4.2.2 Micropilot Autopilot ………………………………………………… 44

4.3 CAS Sensor Selection: Sonar Sensor ………………………………………… 44

4.3.1 Sonar Sensor Selection……………………………………………... 44

4.3.2 Sonar Sensor with PC104 Setup……………………………………… 45

4.3.3 Sonar Sensor Array Design …………………………………………... 45

4.3.3.1 Single Sonar Sensor Max Range & Field of View Experiment 45

4.3.3.2 Sonar Array Design …………………………………………... 47

4.3.4 Sonar Array CAS Performance Verification………………………….. 49

4.3.4.1 Ground Test Stage 1: Sonar Ranging with Engine running at idle 49

4.3.4.2 Ground Test Stage 2: Sonar Ranging with Engine running at just before takeoff 51

4.3.4.3 Flight Test Stage 1: Sonar Ranging with UAV at hover ……... 52

4.3.4.4 Preliminary CAS Sonar Array Conclusion…………………... 53

4.3.5 Sonar Array Interference Investigation ………………………………… 54

4.3.5.1 Sonar on Ground Test ………………………………………... 55

4.3.6 Sonar Array Interference Conclusion ………………………………… 56

4.4 CAS Sensor Selection: Laser Range Finder……………………………….... 57

4.4.1 Laser Range Finder Specifications…………………………………… 57

4.4.2 Laser Range Finder Flight Test Performance Validation…………….. 58

4.4.2.1 Calibration: LRF Calibration Experiment …………………… 58

4.4.2.2 Flight Test: LRF Functional Experiment…………………….. 59

4.5 CAS Architecture………………………………………………………… 61

4.5.1 Collision Avoidance Algorithm System Design……………………… 61

4.5.2 Servo Sweeping Mechanism……………………………………………….. 63

4.5.3 CAS Controller Selection………………………………………………….. 66

4.5.4 Gyro Controlled Pitch Stabilised Mount……………………………… 70

4.6 Chapter Summary…………………………………………………………….... 74

CHAPTER 5: FIELD VALIDATION ………………………………………… 76

5.1 Field Validation Setup………………………………………………………… 76

5.1.1 Video Positioning – Wireless Camera……………………………….... 76

5.1.2 Video Positioning – Grid Design…………………………………….... 77

5.1.3 Grid Design Setup…………………………………………………….... 79

5.2 Field Validation Experiments………………………………………………. 80

5.2.1 Collision Avoidance Algorithm Functionality Test………………….. 80

5.2.2 Varying gain, k value experiments…………………………………… 83

5.2.3 Image Analysis……………………………………………………….... 84

5.2.4 Field and Simulation Results…………………………………………. 87

5.2.4.1 Gain, K value = 1…………………………………………….... 87

5.2.4.1 Gain, K value = 2…………………………………………….... 89

5.2.4.1 Gain, K value = 3…………………………………………….... 91

5.2.4.4 Results comparison…………………………………………… 92

5.3 Chapter Summary……………………………………………………….... 93

CHAPTER 6: CONCLUSION……………………………………………….. 94

REFERENCES ………………………………………………………………... 96

APPENDIX…. ……………………………………………………………… 100

SUMMARY

Collision avoidance for Unmanned Air Vehicles (UAV) is a necessity if the UAV is to fly in an area whereby the terrain is unknown. Collision avoidance is a field widely researched on especially amongst the robotics community. But most of the existing collision avoidance algorithms require knowledge of terrain and even the location of the obstacles with respect to the robot.

This project seeks to verify a collision avoidance algorithm implemented onto an actual hardware system for the UAV. The terrain and obstacles are unknown to the UAV before flight and collision avoidance is expected of the UAV using information relayed only through onboard sensors. Literature survey of existing works on collision avoidance and collision avoidance in UAVs, revealed a particular study that approached collision avoidance from a missile guidance point of view and hence is especially applicable to flight platforms maneuvering in an unknown terrain. The result is a modified version of the Proportional Navigation (PN) guidance law that serves as a collision avoidance algorithm.

A thorough theoretical study of the collision avoidance algorithm based on PN guidance was conducted, detailing how the information required for the collision avoidance can be obtained from currently commercially available sensors that can be mounted onboard and to put the information to good use in a collision avoidance system (CAS). This is then followed by a simulation of the selected UAV flight platform together with the designed CAS system in place. Simulation helped determine the optimum collision avoidance gain, k value of 2 that should be used for the actual flight test. The simulation also provided results of simulated flight paths that are to be verified with actual field testing.

A flight platform with the sensory and control equipment necessary for implementing the PN collision avoidance algorithm is put together to realize the algorithm in actual hardware. A section is dedicated to detailing the specifications of the hardware and the sensors that are used to put the CAS system together. This is followed by a write up of the actual field test carried out. The results of the field testing was then collected and compiled for a comparative study between the simulated flight path and the actual flight path of a collision avoidance run is similar. This is to determine if the actual hardware system with an implemented CAS system performs as well as the simulated results.

Eventually, the comparative study shows that the field results that are collected have errors that are within a 4% range for k values 1 and 2 and a maximum error of 7.6% was recorded for k = 3. Considering the complexity of the outfield experiments, the outfield results error are within a small range and considered to agree with the simulation results as obtained, verifying a working CAS system in hardware.

LIST OF FIGURES

Figure 2.1: Collision Avoidance Scenario…………………………………….. 12

Figure 2.2: Graph of ρ_{min} vs (180^o - ψ_o)…………………………………….. 24

Figure 2.3: Plot of Possible Flight Paths…………………………………….. 25

Figure 3.1: Simulation Window Layout…………………………………….. 30

Figure 3.2: Helicopter PID Closed Loop……………………………………. 34

Figure 3.3: Variables Measured During Simulation…………………………… 35

Figure 3.4: Simulated Collision Avoidance Flight Path with Gain, k value = 1 36

Figure 3.5: Graph of Percentage Error vs k value……………………………… 37

Figure 4.1: UAV Flight Computer Mounted………………………………... 41

Figure 4.2: Equipment Layout in Flight Computer Box……………………….. 42

Figure 4.3: Sonar Array Design……………………………………………... 47

Figure 4.4: Sonar sensor array range and field of view……………………….. 48

Figure 4.5: Sonar Sensor Circuitry Layout………………………………...... 48

Figure 4.6: Optilogic RS100 Laser Range Finder……………………………… 57

Figure 4.7: RS100 Calibration Curve………………………………………….. 59

Figure 4.8: 1st RS100 LRF Flight Test……………………………………….. 60

Figure 4.9: Collision Avoidance Algorithm Flowchart………………………... 62

Figure 4.10: Servo Specification……………………………………………….. 63

Figure 4.11: CAS Minimum Obstacle Size……………………………………. 64

Figure 4.12: RS-100 on Servo………………………………………………….. 64

Figure 4.13: Actual Lab Layout………………………………………………... 65

Figure 4.14: RS100 Room Profile Result……………………………………… 66

Figure 4.15: Basic Stamp BS2X Controller……………………………………. 67

Figure 4.16: CAS System design with BS2X Controller………………………. 68

Figure 4.17: New CAS Hardware Setup…………………………………….. 69

Figure 4.18: Gyro Specification……………………………………………... 71

Figure 4.19: Gyro linked stablised mount…………………………………… 72

Figure 4.20: Complete CAS System Design………………………………… 73

Figure 5.1: Wireless Camera Attached onto UAV Skid……………………….. 77

Figure 5.2: Grid Design……………………………………………………... 78

Figure 5.3: Theodolite……………………………………………………….. 79

Figure 5.4: Actual Grid Setup Outfield……………………………………… 80

Figure 5.5: Collision Avoidance Maneuver to the Right of Obstacle………….. 81

Figure 5.6: Collision Avoidance Maneuver to the Left of Obstacle…………… 82

Figure 5.7: Screen Capture of Flight Video…………………………………. 84

Figure 5.8: Measurements made on Screen Capture of Flight Video………….. 85

Figure 5.9: UAV Position Correction Calculation……………………………... 86

Figure 5.10: Corrected Calculation on Screen Capture……………………...... 86

Figure 5.11: Simulated and Actual Flight Path Comparison for Gain, k = 1…... 87

Figure 5.12: Simulated and Actual Flight Path Comparison for Gain, k = 2…... 89

Figure 5.13: Simulated and Actual Flight Path Comparison for Gain, k = 3…... 91

LIST OF TABLES

Table 3.1: Gains obtained via outfield experimentations…………………………. 33

Table 4.1: Function of Each Component in Flight Computer…………………….. 43

Table 4.2: Flight Computer Controller Details…………………………………… 43

Table 4.3: Sonar Sensor Comparison Chart………………………………………. 44

Table 4.4: Max range values of sonar sensor…………………………………….... 46

Table 4.5: Sonar sensor field of view…………………………………………….. 46

Table 4.6: Results for Sonar Ranging with Engine Running at Idle……………… 50

Table 4.7: Results for Sonar Ranging with Engine Running at Just Before Take-off 52

Table 4.8: Experiment Comparison Chart………………………………………... 54

Table 4.9: Sonar Interference Summary Table…………………………………… 56

Table 4.10: RS100 LRF Specifications…………………………………………… 57

Table 5.1: Varying Gain, k Experiments…………………………………………. 83

Table 5.2: Comparison of Field and Simulation Data for Gain, k =1…………….. 88

Table 5.3: Comparison of Field and Simulation Data for Gain, k =2…………….. 89

Table 5.4: Comparison of Field and Simulation Data for Gain, k =3…………….. 90

CHAPTER 1 - LITERATURE SURVEY

The main objective of this project is to develop a collision avoidance algorithm for an autonomous Unmanned Aerial Vehicle (UAV). The algorithm should be able to steer the UAV away from stationary obstacles that are unknown to the UAV initially. Thus, it should have onboard sensors that allow the UAV to sense and avoid the obstacle. In this chapter, a review of some of the related work will be discussed.

Before moving to collision avoidance in aerial vehicles, a study of the collision avoidance algorithms that have been developed for ground robots was conducted to gain further understanding of the subject matter of collision avoidance. One of the earlier works on collision avoidance was conducted by [1]. [1] showed that a low level control system coupled with sensory information was sufficient to perform basic collision avoidance in real-time. The algorithm that was proposed in [1] is termed the potential field method and was first implemented into robotics in this research. This is not as competent a system but required much less computational power than works whose approach to collision avoidance has been a higher level path planning algorithm.

In a later work, [2] presented a reactive collision avoidance algorithm which takes into account the dynamics of the hardware system that the algorithm was implemented on. [2] implemented 2 different collision avoidance navigation algorithms, namely the Nearness Diagram Navigation and the Potential Field Method on a Nomadic XR4000 robot. The 2 different algorithms were algorithms that originally did not take into account the dynamics of the system that it was implemented on. [2] incorporated reactive navigation into these 2 algorithms and showed in hardware demonstration with the XR4000 that it was possible to achieve (1) collision free navigation during execution run and (2) was able to give the guarantee of stopping the robot safely with an emergency stop policy.

Sensor inaccuracy is addressed in [3] which incorporated 2 different algorithms. [3] incorporated the Certainty Grid for obstacle representation and Potential Field for navigation, resulting in the approach entitled Virtual Force Field. Although the sensor that was used was the sonar sensor and it was incorporated for a ground robot, the concept of virtual force field is a concept worth considering if the sensor used for the UAV has a high inaccuracy rate.

In [4], a laser scanner was used to provide sensory information of the obstacle that is in the robot's path. The laser scanner information provides information of the obstacle that helps the robot to plot an optimal path around the obstacle and achieving the final destination but with a slightly altered path from the original intended path. [4] provides a good insight to the usage of the laser scanner as a sensor for detecting obstacles onboard a platform.

Having gathered some concepts in collision avoidance work previously done on the ground robots and manipulators, the literature survey moves into research on the work done for collision avoidance algorithms and sensors that are implemented on flight platforms on unmanned aerial vehicles. There have been a substantial amount of research on collision avoidance path planning using GPS signals to relate the position of the UAV to avoid known obstacles. As this project's aim is navigating in an unknown terrain, the use of onboard sensor and reactive algorithms would be the focus of the remaining section of the literature survey.

A framework to approach collision avoidance is introduced in [10]. The research done here in [10] explores the technical requirements of an unmanned aerial vehicle and interestingly divides the space around the UAV into 2 zones. The first and the larger zone is the temporal sphere of deconfliction and the second inner and closer sphere is the temporal sphere of collision avoidance. Deconfliction according to [10] describes a manoeuvre that eliminates the threat of a potential collision, but not requiring the UAV to make drastic manoeuvre to avoid the obstacle so as to replan the initial flight path. Collision avoidance is the opposite, requiring drastic actions and changes to the flight paths. This is an interesting concept and could be worthwhile to explore if a sensor of long enough range can be used. [10] also explores creating a set of laws for the UAV which are an adaptation of the 3 laws of robotics. [10] also moves on to describe certain technical requirements like data link between UAV and the grounds and various sensors that could be mounted onboard to provide the necessary sensory information to achieve deconfliction and collision avoidance.

An approach to collision avoidance in UAVs is to make use of cameras mounted to the front of the UAV to monitor obstacles that may appear in the flight path of the UAV. [5] explores this by having a single camera to detect obstacles in front of the UAV. As only stereo vision can provide range to the visually detected obstacle, [5] makes use of a sequence of images and optic flow line calculations to determine the range of the obstacle from the UAV. This is a rather novel approach as it is particularly suited to flight platforms that do not have the adequate payload to operate a 2 camera stereo vision system onboard.

And in [6], 2 more image processing algorithms were used to process the image obtained from onboard a UAV to determine the presence of obstacles. [6] proposes that these systems can be installed onto current UAV to achieve the “sense and avoid” awareness using these computer vision and image processing algorithms. [6] claims that the tests done on such vision systems coupled with an appropriate image processing algorithm is able to achieve first detection at up to 6.5km, which is a much higher range that what an alert human observer can achieve.

[7] approached the collision avoidance problems using the same idea of image processing. The research focuses on the meeting the FAA regulations on unmanned aircraft to be able to sense and avoid local air traffic sufficiently so as to be comparable to the see and avoid requirements of manned aircraft. [7] uses optical sensors that operate in the infrared band to detect obstacles. This system is mounted onboard the Global Hawk and Predator UAVs and achieve results that had the potential to meet the FAA requirements.

[8] acknowledges that high computational requirement remains the biggest hurdle for using vision based systems to provide sensory information on obstacles. In turn, [8] proposes a similar concept to [5] and uses monocular images to provide sensory information on obstacles, reducing the computational power that is required onboard the UAV. [8] achieves this through another algorithm which is the feature density distribution analysis. With this algorithm, the UAV can recognise some of the features

as obstacles and perform collision avoidance. The novelty in this approach is that it does not require accurate feature extraction of the images and thus is able to drastically reduce the complexity of the image processing algorithm. For a small scale UAV with limited payload which translates to less sensitive optical sensors and less computational power, it is possible to achieve collision avoidance.

The complexity of the obstacles that vision based systems can differentiate depends very much on the algorithms that process the images. Very often, near invisible obstacles escape the processing and register as no obstacles. An example would be power lines that are hardly visible to the human from the air, not to mention vision systems. [9] proposes a image processing algorithm that is designed to detect thin obstacles such as power lines and wires.

Another approach to collision avoidance without the use of the vision based sensory information is the use of radar technology. [11] explores the use of a radar sensor to detect obstacles in the path of the UAV. A conceptual radar design is used in simulations of collision avoidance with a stationary obstacle. The radar simulation was able to detect obstacles in it's path with a 90% probability, subjected to the designed radar and specifications. [11] shows that radars could be a viable solution to the complex vision based systems.

In [12], data fusion of multiple sensors was investigated and used as a means to provide collision avoidance. The sensors that were used consists of pulsed radar, 2 infrared cameras and 2 normal video cameras. With this amount of sensory information, a data fusion algorithm was devised which is an adaptation of Kalman filter. [12] did a comparative study of 3 different algorithms, namely, Conventional Filter in Rectangular Coordinates, Conventional Filter in Spherical Coordinates, and Extended Filter in Rectangular Coordinates. The extended filter in rectangular coordinates proved to be the best algorithm to use and the other 2 algorithms also had satisfactory performances. But the extensive range of sensors that were made available was only possible as the RMAX radio helicopter converted UAV was used which had a payload of more than 10kg. This luxury of payload is not share by many of the smaller and more accessible radio helicopters and hence might not prove very feasible if a small UAV with collision avoidance was to be developed.

As with the sensory information provided by ground robots, sensors mounted on the UAV face the same problem of sensor uncertainty. These uncertainties will have more dire implications as most fixed wing UAVs do not have the ability so just stop moving like the ground robots. This issued was explored by [18] . [18] suggests that most collision avoidance problems are often divided into sub problems, e.g. detection, estimation and planning and are studied independently. [18] proposes to study all facets of collision avoidance as a single problem, and also taking into account the aircraft dynamics and computer vision sensor limitations into an integrated framework. It also extends the research area into formation flying and obstacle avoidance as a formation. Eventually, simulations of a formation flight of 3 UAVs were able to decide the optimum path of flight to take when manoeuvring around an obstacle, taking into account flight dynamics and sensor limitations.

In a similar research to [10], [20] adopted the approach to collision avoidance in separation of the air space around the aircraft. The onboard collision avoidance algorithm, OCAS, used in [20] separated the airspace into 3 different layers , the first and closest to the UAV is the collision avoidance layer. The second layer is middle horizontal deconfliction layer and the outer most layer is the vertical deconfliction layer. [20] focuses on the research of the middle horizontal deconfliction layer which makes use of the Vector Field Histogram (VFH) method which is commonly used in ground robots for collision avoidance against static obstacles.

The OCAS algorithm proposed by [20] was an adaptation of the VFH which took into account the trajectory of the UAV and also the moving obstacles that came into the flight path of the UAV. Simulation results showed the OCAS to be successful in avoiding both static and moving obstacles. This is an interesting approach to the collision avoidance problem but did not show how the OCAS could be implemented on UAVs with sensors that exist today.

In [22], the collision avoidance algorithm was approached from a different direction, one that required the line of sight angle (LOS) between the UAV and the obstacle it is trying to avoid. The LOS angle usage has been commonplace in the field of missile guidance but has seen little application in the field of collision avoidance. [22] makes use of the LOS angle between the obstacle and the UAV and calculated the relative coordinates between the 2 objects. In doing do, [22] successfully calculates the distance between the UAV and the obstacles with requiring knowledge of the position of either the UAV or the obstacle. This is most unlike the other research that has been done on collision avoidance which usually requires the knowledge of the position of the UAV either through GPS or some other sensors to put the UAV into a position on a reference frame. Eventually, the algorithm is proved successful through the use of simulation and can be extended to multiple static and dynamic objects. This research proves to be very insightful and could be considered for implementation as it requires relatively low amount of sensory information to execute a collision avoidance manoeuvre.

Another approach to the collision avoidance is the modification of guidance systems that already exists. [23], a modification of the proportional navigation, a technique commonly used in missile guidance was used for collision avoidance. The proportional navigation-based collision avoidance guidance (PNCAG), termed by [23] required the positional knowledge of both the UAV and the obstacle that it was to avoid. The velocity vector of the UAV and that of the obstacle, if it was a dynamic obstacle, was also required.

[23] explained that these information could be easily obtained through radar mounted onboard and also though GPS sensors that could be mounted on the UAV. The algorithm was simulated for inter aerial vehicle collision avoidance and showed to execute the collision avoidance manoeuvre successfully. Although [23] did not provide any successful hardware verification of the algorithm, it remains, however, as a very insightful review of the proportional navigation for collision avoidance.

In [24], another collision avoidance algorithm was developed. It was similar to [23] that it was an adaptation of the proportional navigation guidance in missile theory but took on a completely different approach to the problem of collision avoidance. The novelty of the approach in [24] is that it makes use of the range of between the UAV and the obstacle as main sensory information. This is highly possible with sensors that are available commercially and can be mounted onboard the UAV. Another piece of information that is required is the sign line of sight angle. Combining the sign of the line of sight angle and the range between the UAV and the obstacle, and through the algorithm provided by [24], the UAV was able to execute a collision avoidance manoeuvre past the obstacle.

In addition to that, the algorithm in [24] showed a directly proportional relationship between the range of the obstacle and the UAV to the minimum clearance distance that the UAV will come within the obstacle, showing great potential for adjustment of the algorithm when implemented on actual hardware. Eventually, the algorithm was implemented onto a simulation and showed to work, even for multiple UAVs executing a formation flight and manoeuvring past obstacles. The research done in [24] shows great potential for implementation onto hardware as it makes use of sensory information that could be easily provided with sensors that are commercially available today.

1.1 Chapter Summary

In this chapter, literature survey begins with work done in the field of collision avoidance on ground robots. This is to provide a good grasps of the existing collision avoidance algorithms that could be modified to fit into collision avoidance in UAVs.

This is followed by survey on research in the field of collision avoidance in aircrafts and UAVs. The survey showed much research went into the use of vision based systems to provide sensory information for the use of collision avoidance. Many of these works also branched research on image processing of the images obtained from the vision systems. They ranged from monocular images to stereo images to even to infrared images.

Eventually, the survey showed works that made use of sensor data fusion, by having many different sensors onboard and using all the sensory information available. Research was also carried out in a novel adaptation of the proportional navigation in missile guidance to be used in collision avoidance which related the range and line of sight angle to the performance of the collision avoidance manoeuvre.

CHAPTER 2 - DESIGN AND ANALYSIS OF COLLISION AVOIDANCE ALGORITHM

2.1 Proportional Navigation law in Collision Avoidance

Proportional Navigation (PN) guidance law is an algorithm used in missile guidance systems. This segment is based on the work done from reference [24] and goes to show the proportional navigation law in detail and up to the point where it is modified for use as a collision avoidance system.

With reference to the diagram below, consider a scenario where a flight platform has the following attributes.

1.	Vertical Axis	:	Y
2.	Horizontal Axis	:	X
3.	Flight platform	:	F
4.	Position of Flight Platform	:	(x_f, y_f)
5.	Velocity vector of flight platform	:	V
6.	Flight platform heading angle	:	χ
7.	Applied acceleration of flight platform	:	a
8.	Position of obstacle	:	Origin
9.	Diameter of no-fly zone around obstacle	:	d
10.	Range of flight platform to obstacle	:	R
11.	Line of Sight (LOS) angle	:	θ
12.	Obstacle Heading	:	ζ

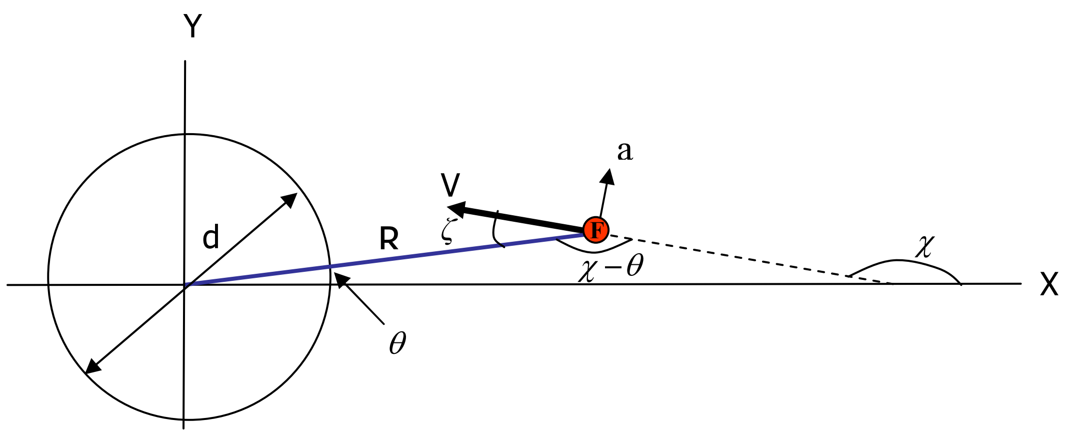


Figure 2.1: Collision Avoidance Scenario

With the above definition, it can be deduced that:

$$R^2 = x_f{}^2 + y_f{}^2 \qquad (1)$$

$$x_f = R\cos\theta \qquad (2)$$

$$y_f = R\sin\theta \qquad (3)$$

This determines the equations of motion for the flight platform's position.

And

$$V = \sqrt{u^2 + v^2} \qquad (4)$$

$$\dot{x}_f = \frac{dx_f}{dt} = u = V\cos\chi \qquad (5)$$

$$\dot{y}_f = \frac{dy_f}{dt} = v = V\sin\chi \qquad (6)$$

$$\ddot{x}_f = \frac{du}{dt} = a\left(\frac{-v}{V}\right) \qquad (7)$$

$$\ddot{y}_f = \frac{dv}{dt} = a\left(\frac{u}{V}\right) \qquad (8)$$

which in turn determines the equation of motion for the flight platform's velocity. As shown in equation (7) and (8), it is assumed that there is only an applied acceleration orthogonal to the flight platform's velocity as aerodynamic forces on flight surfaces are always orthogonal to the velocity.

A practical approach to the algorithm will be to base it on variables that will be measurable from the flight platform. One such variable is the range, R, between the obstacle and the flight platform. This can be easily obtained by having range sensors onboard the flight platform.

Another one such variable is the flight platform bearing, χ, where the angles are measured positive anticlockwise from the X axis.

Thus, in order to obtain rates such as $\dot{R}$ and $\dot{\theta}$ in terms of measurable variables range R, and flight platform heading χ, take equation (1):

$$R^2 = {x_f}^2 + {y_f}^2$$

differentiate with respect to time, t:

$$\begin{aligned}
2R\frac{dR}{dt} &= 2x_f\,\dot{x}_f + 2y_f\,\dot{y}_f \\
\frac{dR}{dt} &= \frac{x_f}{R}\dot{x}_f + \frac{y_f}{R}\dot{y}_f \\
\frac{dR}{dt} &= V\cos\theta\cos\chi + V\sin\theta\sin\chi \\
&\Downarrow \\
\frac{dR}{dt} &= \dot{R} = V\cos(\chi - \theta) \qquad (9)
\end{aligned}$$

Also:

$$\tan\theta = \frac{y_f}{x_f}$$

differentiating with respect to time, t:

$$\sec^2\theta \ \frac{d\theta}{dt} = \frac{\dot{y}_f \, x_f - \dot{x}_f \ y_f}{x^2{}_f}$$

$$\begin{aligned}
\sec^2\theta \ \frac{d\theta}{dt} &= \frac{\dot{y}_f \, x_f - \dot{x}_f \ y_f}{x^2{}_f} \\
&= \frac{x_f \ V\sin\chi \ - y_f \, V\cos\chi}{\left(R\cos\theta\right)^2} \\
&= \frac{x_f \ V\sin\chi \ - y_f \, V\cos\chi}{R^2} \quad \sec^2\theta \\
&\Downarrow
\end{aligned}$$

$$\frac{d\theta}{dt} = \frac{V}{R}\left(\frac{x_f}{R}\sin\chi - \frac{y_f}{R}\cos\chi\right)$$

$$\frac{d\theta}{dt} = \dot{\theta} = \frac{V}{R}\ \sin(\chi-\theta) \qquad (10)$$

Also, to obtain rates $\dot{V}$ and $\dot{\chi}$ in relation to the applied acceleration a, take equation (4):

$$V = \sqrt{u^2+v^2}$$
$$V^2 = u^2+v^2$$

differentiate with respect to time, t:

$$2V\frac{dV}{dt} = 2u\frac{du}{dt} + 2v\frac{dv}{dt}$$

Substituting in equations (5) – (8)

$$V\frac{dV}{dt} = V\cos\chi \,.\, a\left(\frac{-v}{V}\right) + V\sin\chi \,.\, a\left(\frac{u}{V}\right)$$

$$\frac{dV}{dt} = \cos\chi \,.\, a\left(\frac{-V\sin\chi}{V}\right) + \sin\chi \,.\, a\left(\frac{V\cos\chi}{V}\right)$$

$$= a\,(-\cos\chi\sin\chi + \cos\chi\,\sin\chi)$$

$$\Downarrow$$

$$\frac{dV}{dt} = \dot{V} = 0 \qquad (11)$$

Also

$$\underset{\sim}{V} = V\begin{Bmatrix}\cos\chi\\ \sin\chi\end{Bmatrix} \qquad \& \qquad \underset{\sim}{a} = a\begin{Bmatrix}-\sin\chi\\ \cos\chi\end{Bmatrix}$$

Thus,

$$\dot{\underset{\sim}{V}} = \underset{\sim}{a}$$

$$\Downarrow$$

$$\frac{dV}{dt}\begin{Bmatrix}\cos\chi\\ \sin\chi\end{Bmatrix} + V\begin{Bmatrix}-\sin\chi\\ \cos\chi\end{Bmatrix}\frac{d\chi}{dt} = a\begin{Bmatrix}-\sin\chi\\ \cos\chi\end{Bmatrix}$$

$$\Downarrow$$

$$V\begin{Bmatrix}-\sin\chi\\ \cos\chi\end{Bmatrix}\frac{d\chi}{dt} = a\begin{Bmatrix}-\sin\chi\\ \cos\chi\end{Bmatrix}$$

Substituting in equation (11)

$$V\frac{d\chi}{dt} = a$$

$$\frac{d\chi}{dt} = \dot{\chi} = \frac{a}{V} \qquad (12)$$

Thus, we obtain the following rates:

$$\frac{dR}{dt} = \dot{R} = V\cos(\chi - \theta) \qquad (9)$$

$$\frac{d\theta}{dt} = \dot{\theta} = \frac{V}{R}\sin(\chi - \theta) \qquad (10)$$

$$\frac{dV}{dt} = \dot{V} = 0 \qquad (11)$$

$$\frac{d\chi}{dt} = \dot{\chi} = \frac{a}{V} \qquad (12)$$

For the convenience of analysis, the next segment proceeds to express the rate equations (9) – (12) in non-dimensional equations. In order to do so, we define the relative heading of the flight platform as follows:

$$\boxed{\psi = \chi - \theta \qquad (13)}$$

Expressing range as a non-dimensional term, define

$$\boxed{\rho = \frac{R}{R_o} \qquad (14)} \quad , \text{ such that when t = 0 , } R(t) = R(0) = R_o$$

Expressing time as a non-dimensional term, define

$$\tau = \frac{t}{\frac{R_o}{V}} \quad \text{such that} \quad d\tau = \frac{V}{R_o}dt + t\frac{\dot{V}}{R_o} \quad \Rightarrow \quad \boxed{\frac{d\tau}{dt} = \frac{V}{R_o} \qquad (15)}$$

Thus, from equation (9):

$$\dot{R} = V\cos(\chi - \theta)$$

$$\frac{dR}{dt} = V\cos\psi$$

$$\frac{dR}{d\tau}\cdot\frac{d\tau}{dt} = V\cos\psi$$

$$\frac{d}{d\tau}\left(\frac{R}{R_o}\right)\cdot\frac{d\tau}{dt} = \frac{V}{R_o}\cos\psi$$

$$\Downarrow$$

$$\frac{d\rho}{d\tau} = \cos\psi, \qquad \rho(0) = 1 \qquad (16)$$

From equation (10):

$$\dot{\theta} = \frac{V}{R}\sin(\chi - \theta)$$

$$R\frac{d\theta}{dt} = V\sin\psi$$

$$\frac{R}{R_o}\frac{d\theta}{d\tau}\frac{d\tau}{dt} = \frac{V}{R_o}\sin\psi$$

$$\Downarrow$$

$$\rho\frac{d\theta}{d\tau} = \sin\psi, \qquad \psi(0) = \psi_o \qquad (17)$$

From equation (12)

$$\dot{\chi} = \frac{a}{V}$$

$$\frac{d\chi}{d\tau}\cdot\frac{d\tau}{dt} = \frac{a}{V}$$

$$\frac{d\chi}{d\tau} = \frac{a}{V}\cdot\frac{dt}{d\tau}$$

$$= \frac{a}{V}\cdot\frac{R_o}{V}$$

$$\Downarrow$$

$$\frac{d\chi}{d\tau} = \frac{a}{\frac{V^2}{R_o}} = \alpha \qquad (18)$$

Also, from equation (13),

$$\psi = \chi - \theta$$
$$\Downarrow$$
$$\frac{d\psi}{d\tau} = \frac{d\chi}{d\tau} - \frac{d\theta}{d\tau}$$

Substituting equation (17) & (18)

$$\frac{d\psi}{d\tau} = \alpha - \frac{1}{\rho}\sin\psi \qquad (19)$$

Thus, the non-dimensional rate equations are as follows:

$$\boxed{\begin{array}{lll} \dfrac{d\rho}{d\tau} = \cos\psi, & \rho(0) = 1 & (16) \\[2ex] \rho\dfrac{d\theta}{d\tau} = \sin\psi, & \psi(0) = \psi_o & (17) \\[2ex] \dfrac{d\chi}{d\tau} = \dfrac{a}{\dfrac{V^2}{R_o}} = \alpha & & (18) \\[2ex] \dfrac{d\psi}{d\tau} = \alpha - \dfrac{1}{\rho}\sin\psi & & (19) \end{array}}$$

The following are points to note for using the non-dimensional rate equations,

- Collision avoidance occurs only if $-\pi \le \psi_o < -\pi/2$ or $\pi/2 < \psi_o \le \pi$
- R_o is the initial detection distance between the flight platform and the obstacle.

Analysing the non-dimensional rate equations, it can be observed that if equations (16) and equations (19) are arranged in the form as follows:

Equation (16) / equation (19):

$$\frac{\frac{d\rho}{d\tau}}{\frac{d\psi}{d\tau}} = \frac{\cos\psi}{\alpha - \frac{1}{\rho}\sin\psi}$$

$$\Downarrow$$

$$\frac{d\rho}{d\tau}.\frac{d\tau}{d\psi} = \frac{\cos\psi}{\alpha - \frac{1}{\rho}\sin\psi}$$

$$\Downarrow$$

$$\boxed{\frac{d\rho}{d\psi} = \frac{\cos\psi}{\alpha - \frac{1}{\rho}\sin\psi} \qquad (20)}$$

A relationship between ρ and ψ is obtained in equation (20). This suggests that with a suitable α algorithm, a flight path to avoid an obstacle can be obtained based on measurable variables obtainable onboard a flight platform with the appropriate sensors. This is the objective of collision avoidance. As such, the collision avoidance algorithm lies in determining an appropriate expression for α, such that the flight platform will avoid the obstacle. It should do so with a minimum distance away from the center of the obstacle, determined by the value of ρ_{min}.

2.2 Analysis of Collision Avoidance Algorithm

As shown in equation (20), the key to the collision avoidance algorithm is to determine a relationship for α such that the collision avoidance maneuver can be performed effectively by the flight platform.

The analysis of the different collision avoidance algorithm will be based on sensors that are currently available. This practical approach ensures that the developed algorithm can actually be implemented and it's performance verifiable.

2.2.1 Line of Sight Rate Sensor

One of the available sensors that can be used is the LOS (line of sight) rate sensor. The LOS rate sensor signal can be incorporated into the collision avoidance algorithm by using the signal as the value of α in equation.

The LOS rate can be represented by a modified equation (17). Through the use of such an equation, analysis on the effectiveness of the LOS rate as a collision avoidance algorithm can be carried out.

Using

$$\alpha = -k\frac{d\theta}{d\tau} = -\frac{k}{\rho}\sin\psi$$

Equation (20):

$$\frac{d\rho}{d\psi} = \frac{\cos\psi}{\alpha - \frac{1}{\rho}\sin\psi}$$

after substituting new α term from above working becomes:

$$\frac{d\rho}{d\psi} = \frac{\cos\psi}{-\frac{k}{\rho}\sin\psi - \frac{1}{\rho}\sin\psi}$$

$$= \frac{-\rho\cos\psi}{(k+1)\sin\psi}$$

$$\Downarrow$$

$$\frac{d\rho}{\rho} = \frac{-\cos\psi}{(k+1)\sin\psi}d\psi$$

On integration:

$$\int_{\rho_o}^{\rho_f} \frac{1}{\rho} d\rho = \int_{\psi o}^{\psi_f} \frac{-\cos\psi}{(k+1)\sin\psi} d\psi$$

$$\left[\text{In}\,\rho\right]_{\rho o}^{\rho_f} = \frac{1}{(k+1)} \left[\text{-In}(\sin\psi)\right]_{\psi_o}^{\psi_f}$$

An effective measure of the collision avoidance algorithm will be the minimum distance the flight path is away from the obstacle, ρ_{min}. From equation (20), it can be determined that ρ_{min} occurs when $\frac{d\rho}{d\psi} = 0$ and this occurs only when

$\psi = 0$ or $\psi = \pm\frac{\pi}{2}$

Thus when

$$\psi_f = \frac{\pi}{2} \Rightarrow \rho_f = \rho_{\min}$$

Thus, using conditions: $\psi_f = \frac{\pi}{2}$, $\rho_f = \rho_{\min}$ and $\rho_o = 1$

$$\Downarrow$$

$$\text{In}\,\rho_f = \text{In}\,\rho_{\min} = \frac{1}{(k+1)} \left[\text{-In}(\sin\frac{\pi}{2}) + \text{In}(\sin\psi_o) \right]$$

$$\text{In}\,\rho_{\min} = \text{In}\left(\sin\psi_o\right)^{\frac{1}{(k+1)}}$$

$$\rho_{\min} = \left(\sin\psi_o\right)^{\frac{1}{(k+1)}} \qquad (21)$$

As can be seen from equation (21), there is a major problem with the use of the LOS rate signal as the collision avoidance algorithm. Should the initial relative heading angle $\psi_o = \pi$, then the ρ_{min} value becomes 0 irrespective of the gain, k value. This will mean that the flight platform will not perform any collision avoidance when it is on a head on collision course with the obstacle.

This is so because the PN theory was originally developed for missile guidance. The basic principle for missile guidance is for the missile to zoom in for a head on collision. Thus, the PN serves missile guidance very well but will not perform satisfactorily for collision avoidance. Hence, another algorithm for α will have to be developed.

2.2.2 Range and Heading Sensors

Another algorithm that would be possible is to make use of range reading that can be obtained through range sensors. Also, a proportional gain, k, can be added to control the collision avoidance algorithm. The following equation shows the algorithm:

$$\alpha = -\frac{k}{\rho}$$

With that, Equation (20) becomes:

$$\frac{d\rho}{d\psi} = \frac{\cos\psi}{-\frac{k}{\rho} - \frac{1}{\rho}\sin\psi}$$

$$\Downarrow$$

$$\frac{d\rho}{\rho} = \frac{-\cos\psi}{k + \sin\psi}\, d\psi$$

On integration:

$$\int_{\rho_o}^{\rho_f} \frac{1}{\rho}\, d\rho = \int_{\psi_o}^{\psi_f} \frac{-\cos\psi}{k + \sin\psi}\, d\psi$$

$$\left[\text{In}\,\rho\right]_{\rho_o}^{\rho_f} = \left[\text{-In}(k + \sin\psi)\right]_{\psi_o}^{\psi_f}$$

Using the same conditions as above: $\psi_f = \dfrac{\pi}{2}$, $\rho_f = \rho_{\min}$ and $\rho_o = 1$

$$\text{In}\,\rho_f = \text{In}\,\rho_{\min} = \text{-In}(k + \sin\frac{\pi}{2}) + \text{In}(k + \sin\psi_o)$$

$$\text{In}\,\rho_{\min} = \text{In}\left(\frac{k + \sin\psi_o}{k+1}\right)$$

$$\rho_{\min} = \frac{k + \sin\psi_o}{k+1} \qquad (22)$$

Conducting the same test on the second algorithm, it can be seen that should the initial relative heading angle $\psi_o = \pi$, i.e a head on collision, the $\rho_{\min}$ value reduces to $\dfrac{k}{k+1}$. Thus, if the value of k = 1, and on a head on collision course, then equation (22) becomes:

$$\rho_{\min} = \frac{k}{k+1} \quad \Rightarrow \quad \frac{R_{\min}}{R_o} = \frac{1}{2}$$

$$\Rightarrow \quad R_{\min} = \frac{R_o}{2}$$

To investigate further, k values of 1, 2, 3 and 4 were used and a plot of ρ_{min} vs ($180^o - \psi_o$) is shown in Figure 2.2 below.

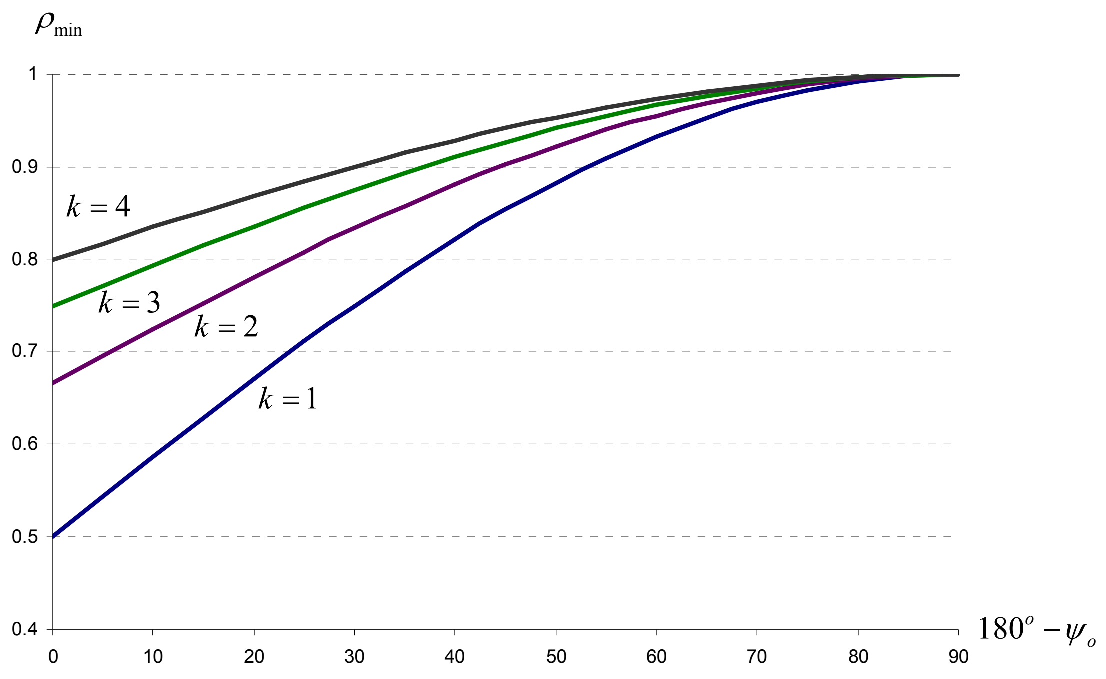


Figure 2.2: Graph of ρ_{min} vs ($180^o - \psi_o$)

The algorithm should be able to provide the right signal no matter which orientation the flight platform approaches the obstacle from. The calculations above are exhaustive if the flight platform approaches the obstacle from the 1st quadrant of the coordinate system. Thus a maneuver direction function should be built into the collision avoidance algorithm.

The function can make use of the obstacle heading, ζ. The resultant sign from the function will be able to provide the correct maneuver direction as indicated by the following:

$$\beta(\zeta) = \begin{matrix} 1 & \text{if } \zeta \geq 0 \\ -1 & \text{otherwise} \end{matrix}$$

Hence the complete collision algorithm would be as follows:

$$\boxed{\alpha = -\beta(\zeta)\frac{k}{\rho}}$$

Given the above complete collision avoidance algorithm, the aim is then to obtain results that can be plotted out as shown in the sample plot in Figure 2.3 below.

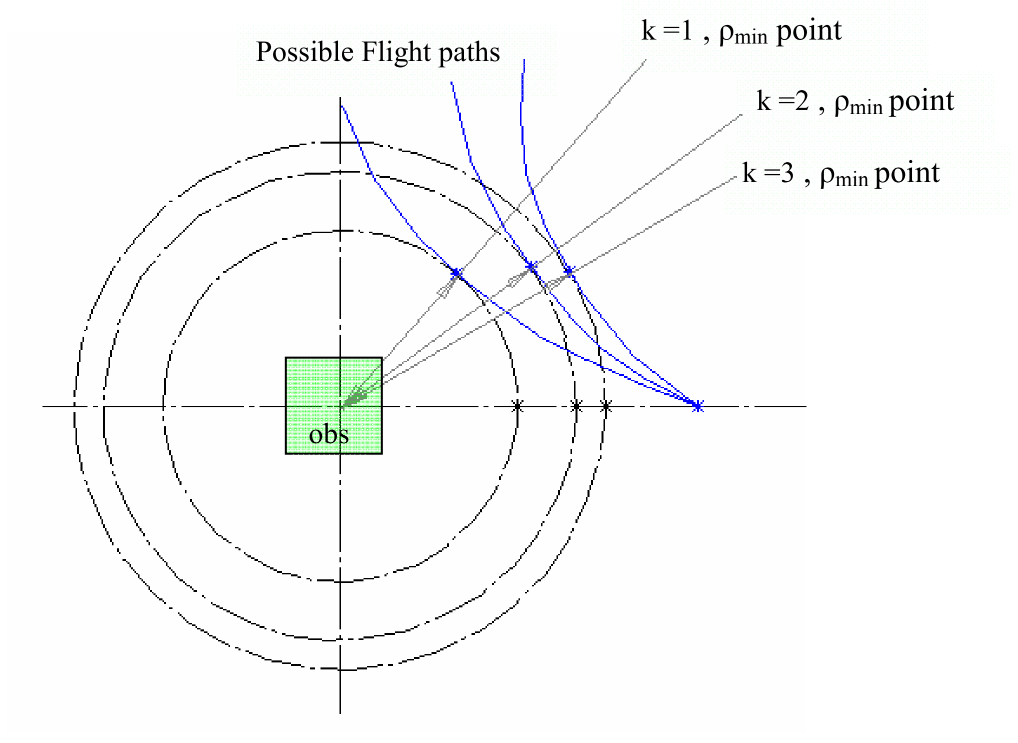


Figure 2.3: Plot of Possible Flight Paths

This will be carried out firstly through simulation of the collision avoidance and secondly through actual hardware tests outfield. The results between the simulation and hardware tests will then be collected. Eventually, the results will be compared for verification of successful implementation of the collision avoidance algorithm.

2.3 Chapter Summary

In this chapter, the proportional navigation law was modified to suit a collision avoidance situation, creating a collision avoidance algorithm. This algorithm was analyzed and found to be viable with measurable variables whose values can be obtained with commercially available sensors. This chapter lays the ground work upon which a collision avoidance algorithm is simulated in software and later demonstrated in hardware in the following chapters.

CHAPTER 3 - COLLISION AVOIDANCE SIMULATION VALIDATION

The previous chapter explains in detail the theoretical aspects of the collision avoidance algorithm. Thus, the next stage of progression will see the collision avoidance algorithm simulated in a computer simulation. The results of the collision avoidance simulation will also be detailed in this chapter and will serve as a comparison with the actual data collected from field experiments which will be detailed in chapter 5. This chapter details not only the results but also the variables used in the simulation. These variables are carefully selected, evident from chapter 2, so that implementation of the collision avoidance algorithm in actual hardware is feasible

3.1 Measurable Variables

From the previous chapter, the collision avoidance algorithm is follows:

$$\boxed{\alpha = -\beta(\zeta)\frac{k}{\rho}}$$

This algorithm is developed based on 2 variables that are measurable with sensors that are currently commercially available. The 2 variables are as listed below:

1. Range of UAV to obstacle , R
2. Obstacle Heading , ζ

The following working will show how the Range, R and the obstacle heading, ζ can be used in the collision avoidance algorithm.

3.1.1 Range Reading, R

Recall that from chapter 2, $\rho = \frac{R}{R_o}$ whereby R_o is the initial detection distance. Hence, whether it is in simulation or in actual field data recording, the very first reading or value that is obtained by the ranging sensor R_o that registers detection of obstacle should be recorded and stored in memory.

Range readings, R, will then be consistently recorded as the UAV moves closer to the obstacle and thus providing a constant update of the variable ρ. This will provide one of the variable readings in the calculation of α.

3.1.2 Obstacle Heading Angle, ζ

Recall that from chapter 2, $\beta(\zeta) = \begin{matrix} 1 & \text{if } \zeta \geq 0 \\ -1 & \text{otherwise} \end{matrix}$

Hence, during the simulation or actual field data recording, the obstacle heading should be consistently monitored. The function β will only result in a either +1 or -1 value, providing the direction signal for the collision avoidance algorithm.

Thus, from the above measurable variables calculation, the collision avoidance algorithm can be implemented through the calculation of the following equation:

$$\alpha = -\beta(\zeta)\frac{k}{\rho}$$

$$\Downarrow$$

$$\alpha = \pm\frac{k\,R_o}{R}$$

Whereby k is the adjustable gain of the collision avoidance algorithm, R is the continually updated range reading of the obstacle to the UAV, R_o the initial detection distance and the polarity of the value of α provided through the measurement of the obstacle heading. With this mathematical representation of the collision avoidance algorithm simplified to the terms of the measurable variables, these variables will be used as the collision avoidance parameters in the simulation.

3.2 Simulation Models

The collision avoidance simulation is based on the framework that has been developed by CoSy Laboratories. The framework consists of a model of the UAV set into a simulated, to scale area of 60m x 40m. The figure below shows the layout of the simulated window, the figure below has been resized slightly to show the simulated UAV and obstacle more clearly and hence is not to scale as per the simulation.

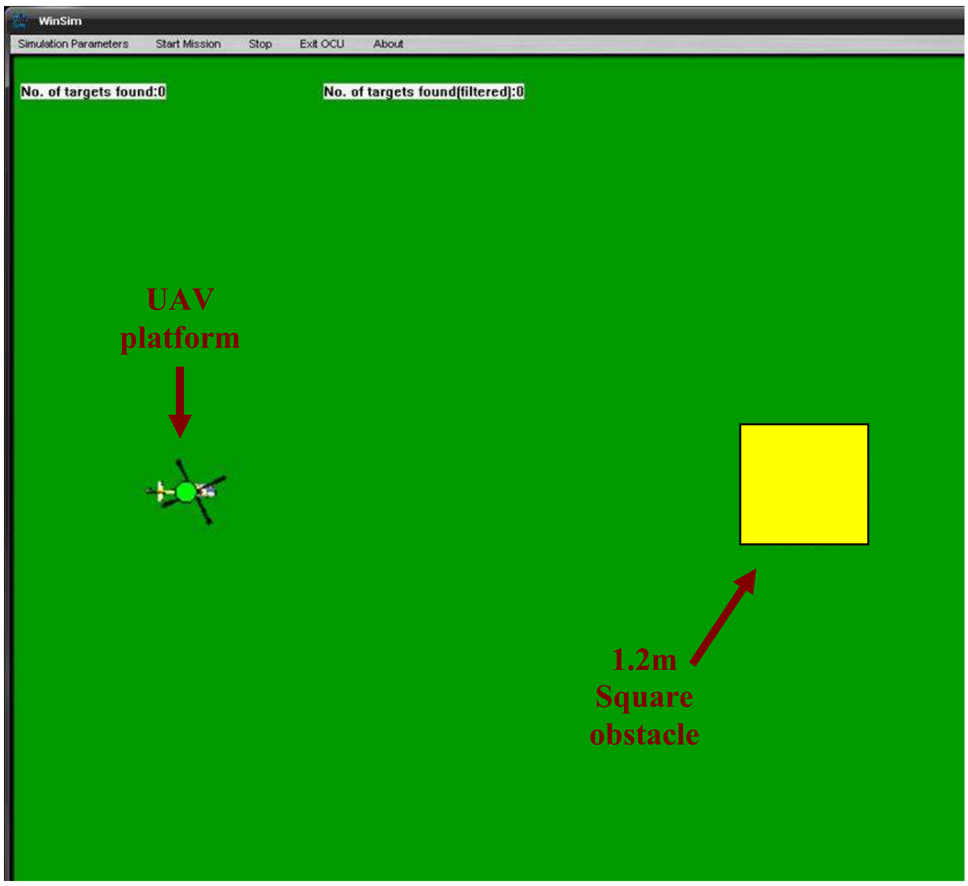


Figure 3.1: Simulation Window Layout

The simulation platform is able to simulate a UAV platform as well as an obstacle in the flight path of the simulated UAV. The obstacle is a 1.2m square block and will simulate being in a position that will form a head on collision with the UAV. As the performance of the collision avoidance is closely related to the flight dynamics of the UAV, the following section will detail the flight dynamics model of the simulated UAV.

3.2.1 Simulated UAV Dynamics Model

The UAV simulated dynamic model is defined with a 1st order transfer function with a delay. However, the gains in the dynamics model are tuned through actual hardware tests out in the field. Each of the 4 degrees of freedom for cyclic control of roll, pitch, yaw and altitude is tested with the same procedure of measuring the human pilot's input to the remote control, and measuring the output of the actual UAV through data logging onboard the UAV itself through numerous flight tests. The results are then analysed and a model fitting the data collected was employed in the simulation. The simulated UAV dynamics model with 4 degrees of freedom has been verified in simulation to be stable based on the report [20] and hence will be used in the simulation without further verification on the part of this report. As the main focus is the collision avoidance which controls the yaw, following workings will be for the yaw control loop of the simulated UAV. The simulated UAV dynamics model for yaw employs a 1st order transfer function with a delay. The transfer function is shown below:

$$\frac{Y(s)}{U(s)} = \frac{K_0 e^{-\lambda t}}{s+\tau}$$

Whereby U(s) is the input (yaw servo input for tail rotor cyclic control) and Y(s) is the output (actual yaw cyclic angle output). Also, K_o is the output gain, τ being the time constant and λ being 1/ τ. The equation correctly models the rates of changes yaw. However, the simulation model should control the heading angle itself rather than the yaw rates.

In order to do so, the equation is rewritten in the time domain by inverse Laplace transform to the following:

$$\frac{dy}{dt} + \tau y = K_o u(t - t_d)\,,$$

Where t_d represents time delay of the transfer function and τ having units of y/s.

And a second state variable is introduced, where

$$\frac{dx}{dt} = y$$

Finally a final state variable is added, which is to be used as the input into the yaw PID controller, where

$$w = cmd - y$$

Therefore, the variables can be rewritten as such where

$y = x_1$, $x = x_2$, and finally $w = x_3$,

which gives the basic simulation equation below. The roll set of equations is used as the example:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \tau & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + K_O \begin{pmatrix} u_{delayed} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ cmd \end{pmatrix}$$

x_1:	Rate of change of heading
x_2:	Actual heading
x_3:	Integral of error between commanded and actual heading
$u_{delayed}$:	Actuator command output by PID, but factored with a delayed of a certain time due to the actual reaction time in an actual hardware system.
cmd:	Desired Heading Guidance Algorithm

The other 3 Degrees of Freedom follow the same structure. To enforce as much commonality with the results obtained from field testing, the PID controller type was used. This will allow the gains found from tuning the PID loops on the field to be used for the system. Currently, the equations governing the PID controller for a single degree of freedom is as shown below.

$$u_{delayed} = \begin{pmatrix} K_p & K_i & K_d \end{pmatrix} \left[\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} cmd \\ 0 \\ 0 \end{pmatrix} \right]$$

Where

K_p:	Derivative Gain
K_i:	Integral Gain
K_d:	Derivative Gain
U_{delyed}:	Actuator command output by PID autopilot
cmd:	Desired Attitude / Height / Forward Velocity set by Guidance Algorithm

The tuned gain values of the simulated flight control are as listed in the following table:

UAV degree of freedom	**Proportional Gain (K_p)**	**Integral Gain (K_i)**	**Derivative Gain (K_d)**
Roll	0.1824	0.0001	0.1123
Pitch	0.0460	0.0002	0.0470
Yaw / Heading	0.0300	0.0002	0.0272
Altitude	0.2092	0.0009	0.3176

Table 3.1: Gains obtained via outfield experimentations

Standard methods such as Nichols Ziegler of obtaining the gains are not applicable as the gains are obtained through actual hardware trial and error testing, which provides the best fit for the UAV model that is developed.

The PID loop diagram is as shown below:

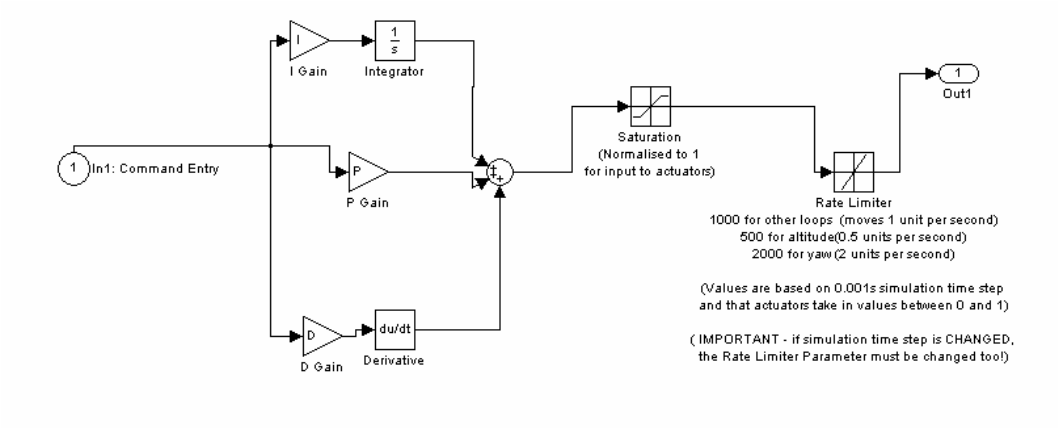

I Gain
1/s
Integrator
1 In1: Command Entry
P Gain
D Gain
du/dt
Derivative
Saturation
(Normalised to 1
for input to actuators)
Rate Limiter
1000 for other loops (moves 1 unit per second)
500 for altitude(0.5 units per second)
2000 for yaw (2 units per second)
(Values are based on 0.001s simulation time step
and that actuators take in values between 0 and 1)
(IMPORTANT - if simulation time step is CHANGED,
the Rate Limiter Parameter must be changed too!)
1 Out1

Figure 3.2: Helicopter PID Closed Loop

3.2.2 Simulated Collision Avoidance Model

The collision avoidance model in the simulation comprises of 2 main sections. They are as listed in the following:

1. Collision Avoidance Sensor Model
2. Collision Avoidance Algorithm
3. Simulated UAV Position measurement

The Collision Avoidance Sensor Model in the simulation simulates the sensor in actual hardware. Data comprising of the measurable variables range, R and obstacle heading, ζ is simulated and passed to the simulated UAV. This is done by consistently calculating the coordinate distance between the UAV and the obstacle. The moment the calculated range falls within a set detection range of 10m, the collision avoidance algorithm is activated.

Incorporated into the UAV simulated code is the Collision Avoidance Algorithm. The Collision Avoidance Algorithm acts upon the data simulated by the sensor model and calculates the $\alpha = \pm \frac{k R_o}{R}$ value. This value is then fed back into the UAV simulation model, which will calculate the appropriate yaw rate, to direct the simulated UAV to manoeuvre away from the obstacle.

The position of the simulated UAV will be monitored in the simulation as this will determine the collision avoidance effectiveness. The following figure will show the variables that are measure during the simulation.

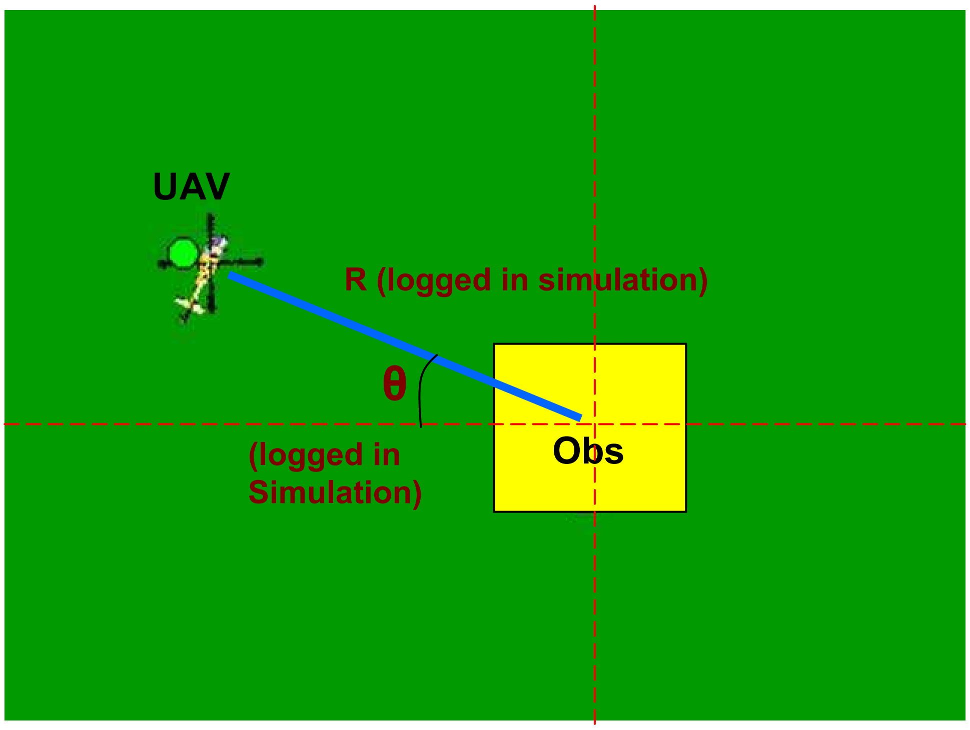


Figure 3.3: Variables Measured During Simulation

The variables R and θ are logged for every simulation time step. This will give a clear indication of the flight path that the simulated UAV makes while executing the collision avoidance manoeuvre.

3.3 Simulation Results for Collision Avoidance Performance

The section details the results that were obtained from the simulation of the collision avoidance performed by the simulated UAV. In order to assess the capability of the collision avoidance algorithm, a worst case scenario of a head-on collision situation is simulated. The R and θ readings were than taken. The following diagram shows the flight path of one of the simulation runs of the UAV executing a collision avoidance algorithm.

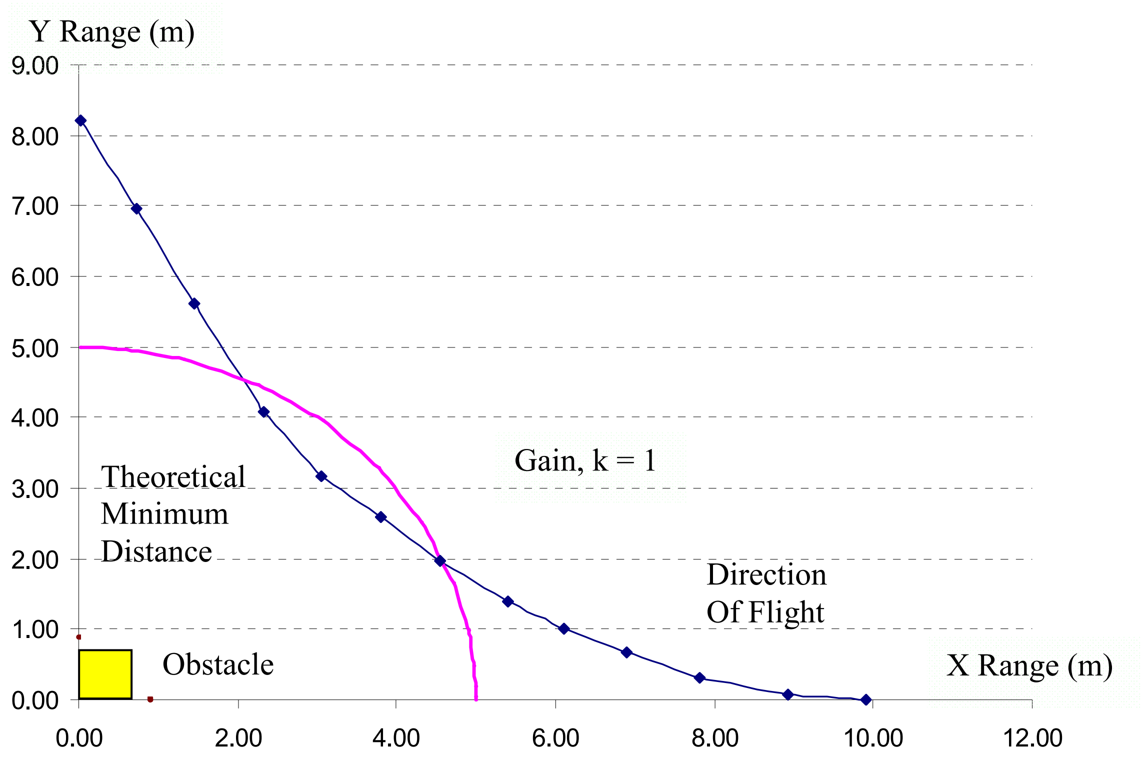


Figure 3.4: Simulated Collision Avoidance Flight Path with Gain, k value = 1

As seen from the above Figure 3.4, the simulated flight path breached the theoretical minium distance. It is 11.9% nearer to the obstacle than the theoretical minimum distance as calculated in chapter 2. The theoretical minimum distance calculation in chapter 2 is based purely on the flight kinematics of the UAV. This calculation does not take into account the flight dynamics and delay of the UAV in actual hardware. This dynamics is reflected in the simulation and hence results in the breaching of the theoretical minimum distance as shown in the Figure 3.4 above. Thus, the objective now is to determine the collision avoidance gain, k, value that results in the lowest percentage error of the minimum distance from the theoretically calculated minimum distance.

Simulation runs were then conducted for k values ranging from k = 1, to k = 4 in steps of 0.01 increments. For each simulation run, the simulated UAV is made to proceed on a path in a head-on collision course with the obstacle. The range readings for each simulated collision avoidance run were then tabulated and the minimum distance of each of the flight paths were than obtained from the readings.

The percentage error was than calculated and tabulated against the k values that were used in the simulation as shown in the Figure below.

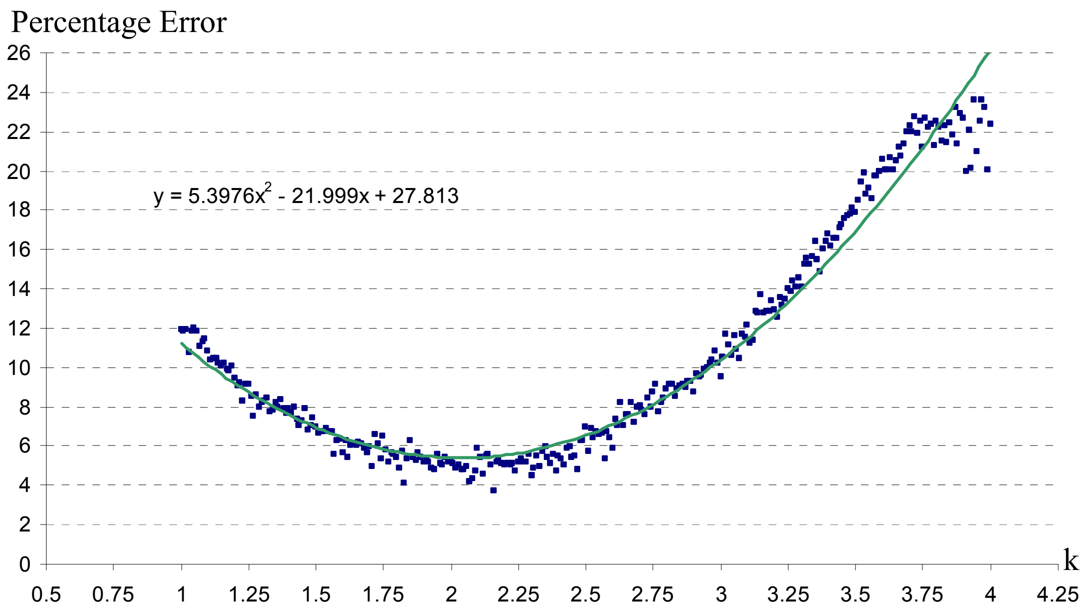


Figure 3.5: Graph of Percentage Error vs K value

Both the sensor model as well as the simulation parameters is modelled with errors as in real hardware situations. Thus, the simulation does not result in a smooth curve as expected. Hence, a curve was drawn to represent the trend of the percentage errors.

As seen from the chart above, the lowest percentage error occurs in the k value region of between 2 to 2.25. The curve that is obtained from the tabulated results is as shown in the chart as well and from the curve, the k value with the minimum percentage error is 2.10. This value according to the curve is a 4.0% error of breaching the theoretical minimum distance within range of the obstacle.

The lower the percentage error, the less the breeching of the flight path minimum safety distance from the obstacle. Hence, k=2.10, which gives the lowest breeching of the flight path minimum safety distance should be used.

The errors increase for other values of k as these values affect yaw control, which is tuned via actual hardware experiments. There are hardware limitations to the value of k that can be used, and thru this simulation, it has indicated that the actual hardware performance will deteriorate with higher levels of gain k. It also indicates that there is a hardware limitation, where k values higher than 3.75 shows a levelling of percentage error with no significant improvement or deterioration of performance

Thus, it can be concluded from the simulation that the optimum value to set the collision avoidance gain, k to is 2.10, which results in the lowest percentage error and thus, the lowest breeching of the minimum distance away from the obstacle.

3.4 Chapter Summary

In this chapter, the simulation of the collision avoidance algorithm was conducted. The simulated collision avoidance algorithm was incorporated into the current simulated UAV code. The collision avoidance simulation makes use of 2 variables that can be measured in reality with commercially available sensors, they are the range, R between the UAV and the obstacle and ζ, obstacle heading angle between the UAV and the obstacle.

The simulation was then run on the worst case scenario of a head on collision with an obstacle, and was tested with different gain, k values ranging from k = 1 to k = 3 in increments of 0.01. The minimum distances for each of these flight paths were obtained and the percentage error of the minimum distance from the theoretically calculated minimum distance was calculated. This is then plotted out in a chart and the curve equation was obtained. It is found from the equation that the optimum k value is 2.10, resulting in a 5.24% error of breaching the theoretical minimum distance within range of the obstacle. The next chapter will describe in detail the sensors that are tested and selected to be mounted onto the UAV to obtain the above mentioned measurable variables R and ζ. Chapter 5 will then compare the results of the simulation with results collected from outfield testing using real hardware.

CHAPTER 4 - CAS ARCHITECTURE AND IMPLEMENTATION

In the previous chapter, the collision avoidance algorithm was tested in simulation. The collision avoidance algorithm simulation model relies on 2 measurable variables. These are the range, R between the UAV and the obstacle and ζ, obstacle heading angle between the UAV and the obstacle. The collision avoidance system (CAS) is designed precisely to provide the abovementioned required sensory information. In this chapter, the actual hardware of the UAV is explained to facilitate the understanding of the incorporation of the sensors onto the UAV. The sensor selection process is also detailed to elaborate on the capability of the sensors and the how the range, R and the obstacle heading ζ are obtained.

4.1 Flight Platform Information

The CAS is part of a collaborative project where the flight platform is required to be fully or partially autonomous, carry out a search mission and during this search, avoid all stationary obstacles in its way.

Hence, details of the selection criteria of the flight platform will not be covered in full detail, and only the relevant information will be included in this report. The flight platform is a remote controlled hobby helicopter, Raptor 90. The detailed specifications are as follows

Engine Size	:	0.90 cu inches
Full Length of fuselage	:	1410mm
Full width of fuselage	:	190mm
Total height	:	*465mm* *
Main rotor diameter	:	1640mm
Tail rotor diameter	:	260mm
Full equipped weight	:	4.8 kg
Payload	:	5.0kg
Estimated flight endurance	:	10 – 15 mins

* - *not including flight computer enclosure*

The raptor 90 is a suitable candidate for our flight platform given its payload and flight endurance and hence it was selected as the flight platform for the Unmanned Aerial Vehicle (UAV) for the project. The Figure 4.1 below shows the UAV mounted with the flight computer enclosure mounted underneath the landing skids.

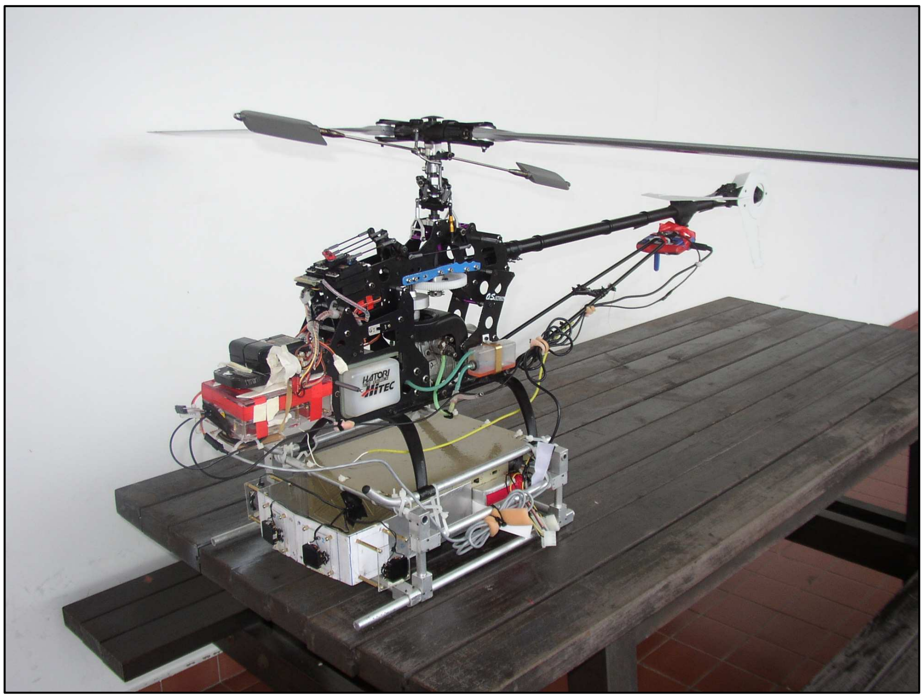


Figure 4.1: UAV Flight Computer Mounted

The next section of the report will detail the computer systems that make up the UAV flight computer which allows integration of the eventual CAS system.

4.2 UAV Flight Computer

The UAV flight computer is housed in an enclosure that is attached to the landing skid of the UAV. It provides the computing capability to the UAV so that it can carry out its mission or testing in the field. This section will reveal the components of the flight computer. The Figure 4.2 below shows the layout of the equipment inside the flight computer enclosure.

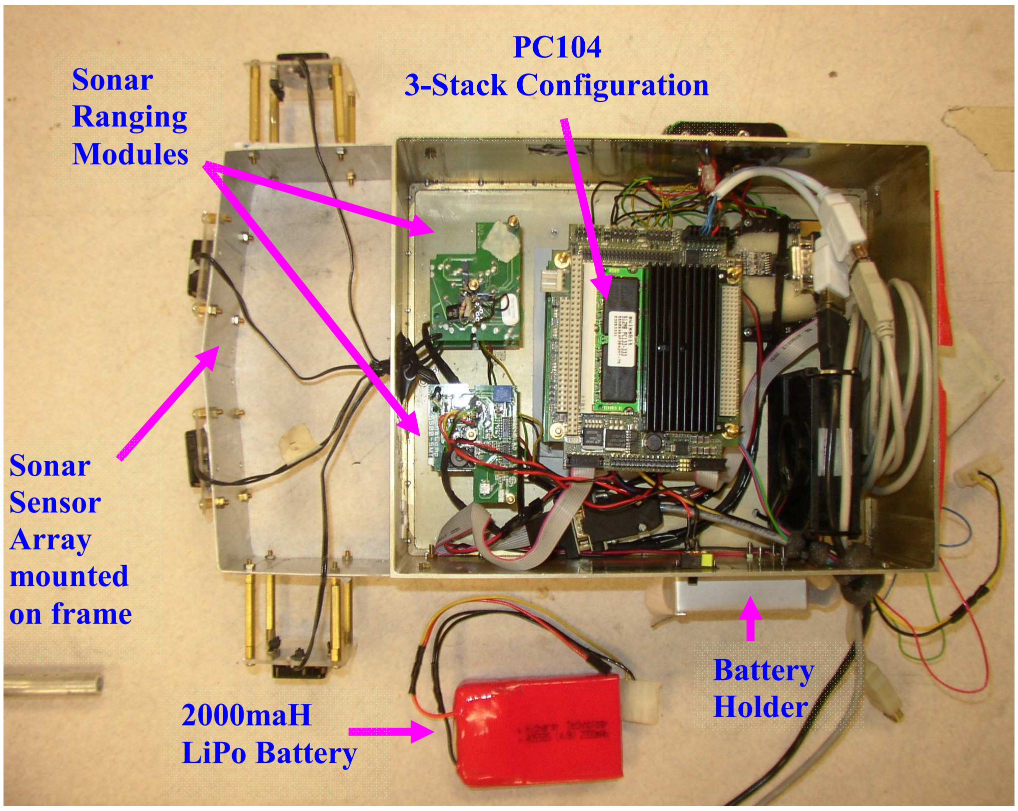


Figure 4.2: Equipment Layout in Flight Computer Box

The function of each of the component that is installed in the flight computer enclosure is as listed in the Table 4.1 below.

Flight Computer Subsystem	**Function**
PC/104 3-Stack Configuration	Main Flight Computer
Sonar Array (4 x Senscomp 7000)	Collision Avoidance System sensor
Micropilot MP2028	Integrated Autopilot package

Table 4.1: Function of Each Component in Flight Computer

4.2.1 PC104 3-stack Configuration

The PC104 3-stack configuration flight computer is the main flight computer that is onboard the UAV. The PC104 not only stores the control program for the sonar sensor array, which is responsible for collecting and processing the data obtained from the sensors, it also uses the data through a collision avoidance algorithm program written and residing on the PC104 itself. The collision avoidance algorithm program then controls that UAV through the PC104 and performs the collision avoidance. The details of this control will be discussed in later sections of this report. The details of the process power of the PC104 are listed in the following table:

Stack #	**Board Type**	**Functions**
1	Main Board – 650MHz ULV Celeron, 512MB RAM, 2GB CF Card storage	USB interface with WiFi adapter, and external I/O devices RS-232 interface with Micropilot VGA interface with external monitor.
2	Digital I/O Board	Interfaces with CAS Sonar Array
3	5V / 12V Regulated Power Supply	Supplies regulated 5V & 12V DC to relevant equipment

Table 4.2: Flight Computer Controller Details

4.2.2 Micropilot Autopilot

The Micropilot is autopilot controller complete with the roll, pitch, yaw attitude sensors fully embedded into a small and compact circuit board. This autopilot will be responsible for the autonomous control of the remote piloted hobby flight platform. The sensor suite onboard the micropilot is also capable of logging the UAV's attitude and altitude which is essential in the verification of the performance of the CAS system.

4.3 CAS Sensor Selection and Design: Sonar Sensor

4.3.1 Sonar Sensor Selection

The first sensor that was selected for the purpose as a range sensor to detect any obstacle was a sonar sensor. A few different sonar sensors were compared to select the appropriate sensor.

The table below shows the different sensors considered.

S/N	Model	Field of View (°)	Range (m)	Voltage (*V*) / Current (*mA*)	Weight (*g*)
1.	Devanteach SRF08	45	0.03 – 6	5 / 15	13
2.	SensComp 7000 Package	22.5	0.15 – 11	6 / 2 000	48
3.	Sharp GP2Y3A003K0F	25	0.04 – 3	5 / 33	20

Table 4.3: Sonar Sensor Comparison Chart

Although the SensComp 7000 is much heavier, requires more voltage, more current and has a smaller field of view, the longer detection range makes the sensor far more advantageous. Thus, the SensComp 7000 was eventually selected as the main sensor for the CAS.

4.3.2 Sonar Sensor with PC104 Setup

The Senscomp 7000 Sonar Sensor is to be interfaced with the PC104. This interface is achieved via the PC104 Digital Input – Output (DIO) board. In order to obtain readings from the SS that is now connected to the PC104, the following had to be performed:

- Configuring the DIO board to gain control of the DIO pins.
- Using the DIO pin to send INIT signal that initiates the SS to send a pulse
- Using the DIO pin to "listen" for the return ECHO signal and obtaining the time difference between INIT and ECHO signals to obtain distance.

A program was written to obtain control of the DIO pins onboard the PC104. The program was then used to carry out experiments to measure the performance of the sonar sensors to simulate actual workings of the sensor onboard the UAV.

4.3.3 Sonar Sensor Array Design and Performance Verification

The approach to the experimentation of the SS is to determine its performance compared to the manufacturer's specification. Once established, the SS can then be arranged in the most appropriate configuration to suit the CAS. The following sections explain the experiments and results.

4.3.3.1 Single Sonar Sensor Max Range & Field of View Experiments

This experiment sets out to determine the actual maximum range and field of view (FOV) of the SS. The experiment was conducted outdoors in a wide open area with no obstacles within maximum range as per the OEM specifications. The experimental set up is detailed in Appendix.

The result for the maximum range experiment is summarized in Table 4.4. It indicates that the SS can detect obstacles up to 9m in range reliability. The SS only detects obstacle at 10m for approximately 50% of the measurements made. Thus, the maximum range for the SS will be taken as 9m due to low sensing reliability beyond that range.

Obstacle Size: 2m by 2m	
Distance (Actual) /m	Computed Distance / m
3.0	2.69
4.0	4.19
5.0	4.97
6.0	5.99
7.0	7.04
8.0	7.96
9.0	8.81
*10.0**	*9.87**
11.0	-
* - Echo detected for approx 50% of the measurements	

Table 4.4: Max range values of sonar sensor

The FOV test sets out to find out the max FOV of the SS. The results show a varying range of FOV with increasing range. The results are summarized in Table 4.5 below:

Distance (Actual) / m	Field of View (Straight in front of Sensor : 0^o)
3.0	-35^o to 30^o
5.0	-10^o to 15^o
7.0	-5^o to 5^o
9.0	-5^o to 5^o

Table 4.5: Sonar sensor field of view

Once the performance of the SS is established through the series of experiments conducted, a SS array design was then established. The main aim of the SS array was to achieve a configuration best suited for the mission objective of the flight platform

4.3.3.2 Sonar Array Design

The figure below shows the design of the sonar array. It consists of a total of 4 sonar sensors, 2 facing forward and 1 on each side of the array.

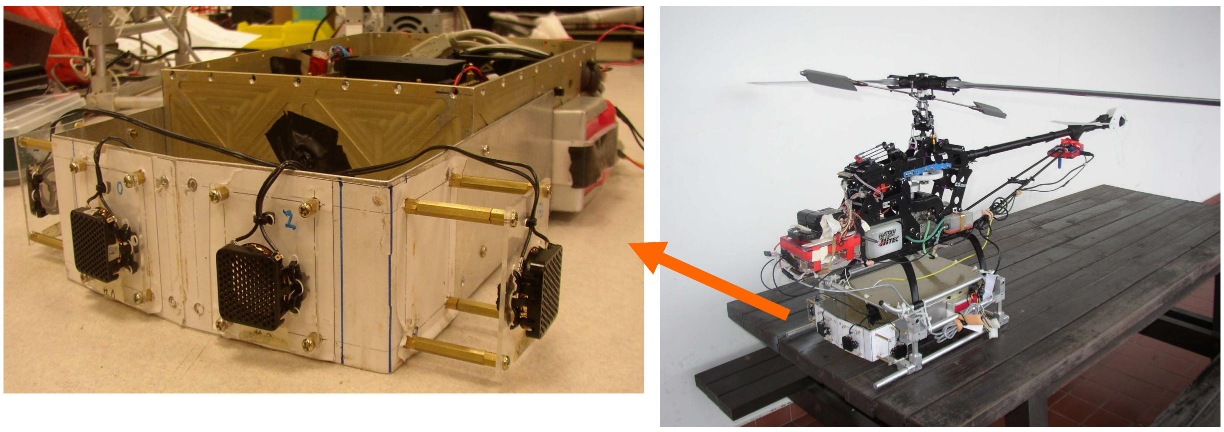


Figure 4.3: Sonar Array Design

The design focuses the range detection to the front of the UAV and accepted blind spots in the areas as shown in Figure 4.3 below. The CAD drawing of the new mounting design can be found in Appendix. This design prevents having any blind spots straight ahead of the SS array, it also has left and right SS to detect obstacles to its sides.

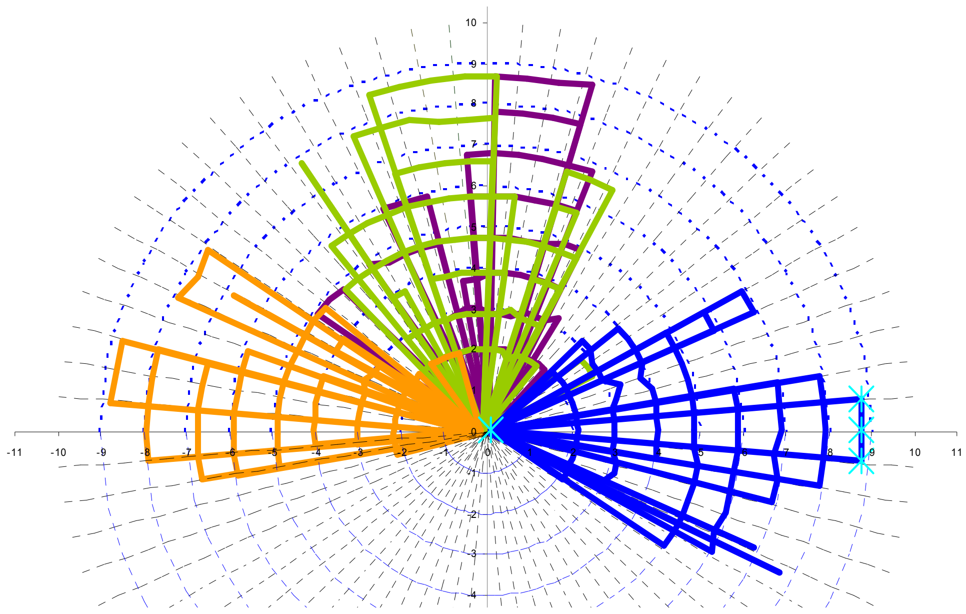


Figure 4.4: Sonar sensor array range and field of view

With the SS array design decided, the next stage of development is to fully integrate the SS array onto the Flight computer enclosure. The Figure 4.5 below shows the layout of the Senscomp 7000 circuitry arranged to stack up in the front of the PC104 inside the enclosure.

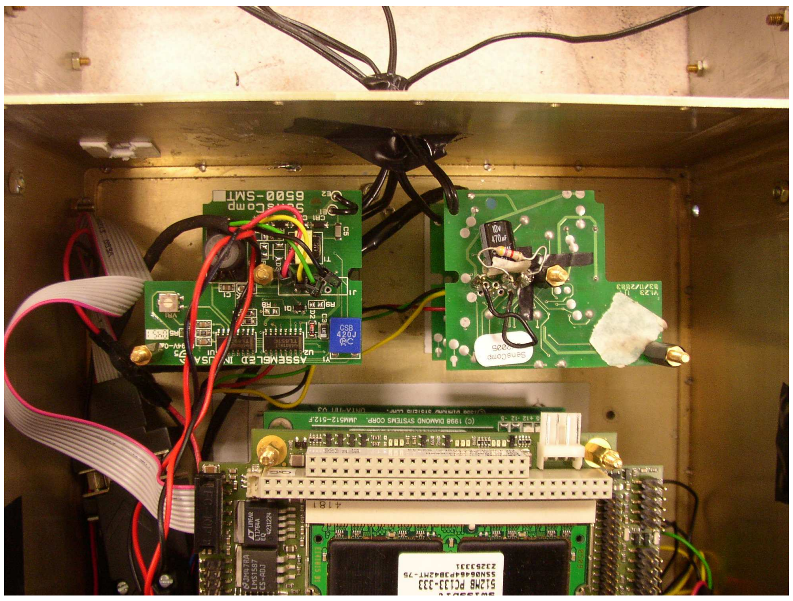


Figure 4.5: Sonar Sensor Circuitry Layout

The next step of the Sonar Sensor array is the testing and verification that the Sonar Sensors can perform in an environment which is vibration prone due to the Internal Combustion Engine that is onboard the UAV. Also, there is a need to determine whether the downwash generated by the UAV will affect the performance of the sonar sensors. This is done progressively through several ground tests and eventually manned flight tests, and will be explained in the following section.

4.3.4 Sonar Array CAS Performance Verification

In order to verify that the Sonar Array can serve effectively as the CAS system for the UAV, a series of experiments are slated for the Sonar Array. They are as listed below:

- Ground Test Stage 1: Sonar Ranging with Engine running at idle
- Ground Test Stage 2: Sonar Ranging with Engine running at just before takeoff
- Flight Test Stage 1: Sonar Ranging with UAV at hover

Each of the experiments and the results obtained will be explained in further detail in the following sections.

4.3.4.1 Ground Test Stage 1: Sonar Ranging with Engine running at idle

Objective:

This test sets out to determine if there is any interference to the sonar sensors from the running of the engine at idle and all the other vibrations that result from the running engine.

Procedure:

The engine will be started with all electronics systems related to controlling the sonar array switched on. Sonar readings will then be taken from the frontal sonar sensors when an obstacle is placed in detection range and at different ranges. The results are then tabulated and compared to previous sonar readings obtained during the design phase. The setup of the experiment can be found in Appendix.

Results:

The results show that the sonar readings are comparable to the readings obtained during the design phase; with all readings have less than 2% error from the design phase readings.

This implies that the engine at idle did interfere with the workings of the sonar sensor. The results are as shown in Table 4.6 below

Obstacle Size : 2m by 2m			
Distance (Actual) /m	Sonar Standalone	Sonar with engine idle	Error between standalone & engine idle (%)
3.0	2.69	2.65	-1.49
4.0	4.19	4.23	0.95
5.0	4.97	5.02	1.01
6.0	5.99	5.90	-1.50
7.0	7.04	7.15	1.56
8.0	7.96	8.00	0.50
9.0	8.81	8.73	-0.91
*10.0**	*9.87**	9.75*	-1.22

* - Echo detected for approx 50% of the measurements

Table 4.6: Results for Sonar Ranging with Engine Running at Idle

4.3.4.2 Ground Test Stage 2: Sonar Ranging with Engine running at just before takeoff

Objective:

This test sets out to determine if there is any interference to the sonar sensors from the running of the engine at just before take. The RPM of the engine at this throttle is higher than that of idle and thus the vibrations that result from the running engine would also be higher. As flight testing would require the expertise of a pilot to control the UAV in flight, this experiment is necessary as the results of this experiment will determine if we proceed on with hiring a test pilot to conduct the flight testing of the CAS system.

Procedure:

The engine will be started with all electronics systems related to controlling the sonar array switched on. The engine will then be throttled up to the point that slight lifting off of the UAV is witnessed. The throttle is then lowered slightly, and the engine will be considered to have achieved just before lift off RPM. Sonar readings will then be taken from the frontal sonar sensors when an obstacle is placed in detection range and at different ranges. The results are then tabulated and compared to previous sonar readings obtained during the design phase. The setup of the experiment can be found in Appendix.

Results:

The results show that the sonar readings are once again comparable to the readings obtained during the design phase, with less than 2% difference between the readings. This implies that the engine at just before take off did interfere with the workings of the sonar sensor. Neither did the vibrations from the engine that is transmitted throughout the whole UAV affect the sonar sensors. The results are as shown in Table 4.7 below

Obstacle Size : 2m by 2m			
Distance (Actual) /m	Sonar Standalone	Sonar with engine before throttle	Error between standalone & engine before throttle (%)
3.0	2.69	2.71	0.74
4.0	4.19	4.12	-1.67
5.0	4.97	4.97	0.00
6.0	5.99	5.91	-1.34
7.0	7.04	7.00	-0.57
8.0	7.96	8.1	1.76
9.0	8.81	8.9	1.02
*10.0**	*9.87**	*9.93*	0.61

* - Echo detected for approx 50% of the measurements

Table 4.7: Results for Sonar Ranging with Engine Running at Just Before Take-off

4.3.4.3 Flight Test Stage 1: Sonar Ranging with UAV at hover

Objective:

This test sets out to determine if there is any interference to the sonar sensors from the running of the engine and the vibrations it creates when the UAV is at hover. The RPM of the engine at this throttle and the mechanical vibrations when the UAV is in flight is much more different than that of all other ground tests.

Procedure:

The engine will be started with all electronics systems related to controlling the sonar array switched on. The pilot will then guide the UAV into a hover at about 10-15 meters off the ground. As the flight test is to be conducted in a wide open space, there will be no obstacles within the detectable range of 10 meters of the sonar sensors. All 4 sonar sensors in the array will be turned on and readings collected while the UAV is in stationary hover 10-15 meters above ground. The results should show that there are no obstacles within detection range. This is the simplest test that can be done before the more elaborate flight test with obstacle is setup.

Results:

The results from this test revealed that the sonar sensors are ineffective when in flight. Instead of detecting no obstacles, the sonar array indicates that there are obstacles all around it. All 4 sonar sensor in the array registered readings, and the readings for all of them were inconsistent, ranging from obstacles detected from 2m – 9m when there was clearly no such obstacle in the flight test site.

4.3.4.4 Preliminary CAS Sonar Array Conclusion

From the results gathered from the experiments conducted. It indicates the sonar sensor experiences some form of interference when the UAV is in stationary hover. Although the ground tests (with engine running) did not reveal this phenomenon, the flight test conducted in section 4.3.4.3 clearly indicates the interference problem. Thus, an investigation to isolate and negate this interference is carried out.

4.3.5 Sonar Array Interference Investigation

The aim of the next series of experiments is to try and isolate and eventually negate the effects of interference on the sonar sensors. 2 key areas were identified as potential interference sources and they are as listed below:

- Engine Vibration – Vibration affecting sonar components and affecting readings
- Rotor Downwash – Downwash affecting sonar ping propagation, affecting readings

From the previous ground tests and flight test that was conducted, the table below shows the presence of the above mentioned conditions in the tests and the corresponding results of the sonar interferences.

	Conditions		Sonar Interference?
	Engine Vibrations	Rotor Downwash	
Flight Test : Sonar with UAV @ hover	**Present**	**Present**	**Yes**
Ground test : Sonar with engine before throttle	**Present**	**Absent**	**No**
Experiment Required	**Absent**	**Present**	***?***

Table 4.8: Experiment Comparison Chart

As seen from the table above, the main remaining experiment to conduct will be one whereby there is downwash but no engine vibrations affecting the sonar sensors. This experiment is performed by mounting the sonar on a standalone ground rig, then having the UAV hover near the sonar sensor and run the sonar test.

4.3.5.1 Sonar on Ground Test

Objective:

To determine if the rotor downwash created by the rotor blades of the UAV interferes with the sonar sensors.

Procedure:

UAV made to hover near a sonar sensor which is mounted onto a rig on the ground. The purpose of this experiment is to isolate the sonar sensor from the vibrations of the UAV and ascertain the functionality of the sensor in presence of UAV rotor downwash. The sonar sensor is mounted in a test site that has no obstacles within detection range. The UAV is made to hover in such a manner that it will not register as an obstacle to the sonar sensor. Readings are then taken when the UAV is hovering near the sonar sensor. The readings should indicate that there is no obstacle when the UAV is hovering near it.

Result:

When the sonar sensor was activated with no UAV near it, it registers as no obstacle within range. But the moment the UAV started hovering near it, the readings started to become inconsistent and register obstacle within range despite having no obstacle within range. This leads to the conclusion that the interference is contributed by the downwash of the UAV.

4.3.6 Sonar Array Interference Conclusion

From the above experiments, it can be concluded that the sonar sensors are affected by the rotor downwash from the UAV. The table below shows the complete result which verifies the downwash having an effect on the sonar readings.

	Conditions		Sonar Interference?
	Engine Vibrations	Rotor Downwash	
Flight Test : Sonar with UAV @ hover	**Present**	**Present**	**Yes**
Ground test : Sonar with engine before throttle	**Present**	**Absent**	**No**
Sonar on ground rig with UAV hovering near	**Absent**	**Present**	**Yes**

Table 4.9: Sonar Interference Summary Table

As the rotor downwash affects the sensors, and the rotor downwash is most likely to be present or exacerbate the sensor performance during flight, an experiment to test the sensor performance during flight will likely yield the same results as experiment no. 1 in Table 4.9.

Also, as the range of the sonar sensor, as reported in pg 45 is at best 9.0m, the sonar sensor performance during flight experiment was thus not conducted in the interest of safety.

Short of changing a UAV for the entire project, there is no viable solution to prevent the sonar sensor from being affected by the downwash. As such, a new sensor has to be selected for the CAS system.

4.4 CAS Sensor Selection and Design: Laser Range Finders

4.4.1 Laser Range Finder Specifications

A new range sensor is proposed for the collision avoidance sensor. The primary concern is that it should not be affected by the downwash effect of the flight platform during flight. As such, the choices of commercially available sensors for this application are very limited. The most appropriate range sensor would be a laser range finder (LRF). It utilizes time of flight of the lasers emitted by the sensor and the speed of light to determine the range.

The sensor that is selected is the Optilogic Laser Range Finder, RS100. Figure 4.6 shows the LRF RS100.

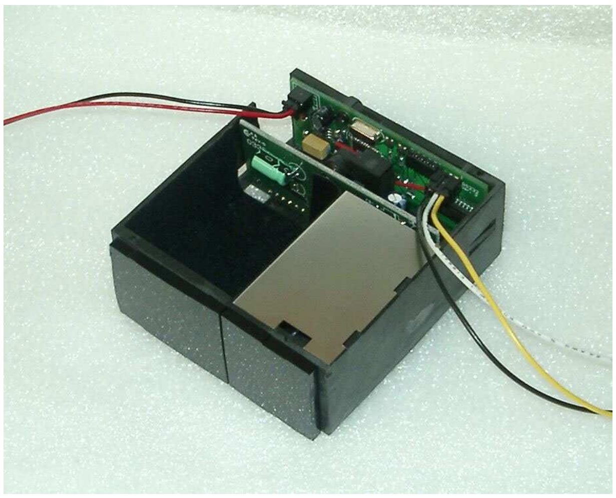


Figure 4.6: Optilogic RS100 Laser Range Finder

The LRF has the following specifications.

Accuracy	:	+/- 1 yard
Com. Protocol	:	RS232-8,N,1
Baud Rate	:	19200

Raw Data Rate	:	~200 Hz
Calibrated Data Rate	:	~10 Hz
Laser	:	Class I (eye-safe) 905nm +/- 10nm
Power	:	7-to-9 Vdc
Typical Range	:	RS100 - 100 Yards

Table 4.10: RS100 LRF Specifications

Due to previous experience with the sonar sensors, it had to be ascertained whether the RS100 LRF could work in flight before further work to use it as the CAS sensor would commence.

4.4.2 Laser Range Finder Flight Test Performance Validation

Experience from the sonar sensor testing reveals that the true validation test is a flight test. Thus a flight test is immediately scheduled before any ground testing. This is to ascertain that the laser range sensor will not register “phantom” obstacles similar to the sonar sensors where there is no obstacle in front of the flight platform. However, the LRF needs to be calibrated before it can be mounted onto the UAV for a flight test. Thus, a calibration experiment is carried to properly calibrate the LRF for ranging use.

4.4.2.1 Calibration: LRF Calibration Experiment

Objective:

To calibrate the LRF RS100 for effective ranging purposes.

Procedure:

An obstacle was set at known intervals of 1m apart. The obstacle is then moved backwards interval by interval and 3000 readings for each interval was obtained and plotted out to obtain the calibration curve. The setup of the experiment can be found in Appendix.

Result:

The result of the calibration experiment can be shown in the following diagram.

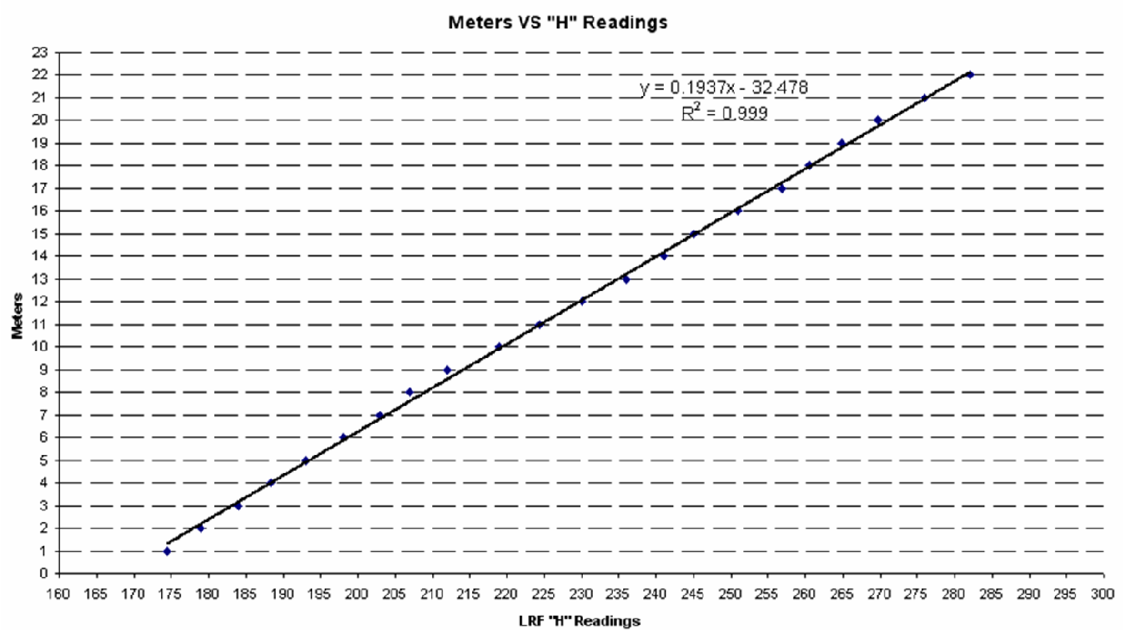


Figure 4.7: RS100 Calibration Curve

Thus, with a calibration curve in place, the LRF is verified to function properly when on ground. The next stage would be to test the LRF when in flight.

4.4.2.2 Flight Test: LRF Functional Experiment

Objective:

To determine the functionality of the LRF when mounted onboard the UAV and subjected to UAV stationary hover flight conditions.

Procedure:

The LRF is mounted onto the UAV. The LRF is then switched on and readings will be taken from the point the UAV is on the ground, till the UAV achieves stationary hover and then till the UAV lands. The readings will then be analyzed to ascertain the functionality of the LRF when subjected to flight conditions. Due to the range of the RS100 which is 90m, the test site had to be specially selected such that there will not be any obstacle or objects when the UAV performs a hover flight.

Results:

The Figure 4.8 below shows the results of the data logs from the flight test conducted. As shown from Figure 4.8, there is a region between the x-axis readings of 1055 to 4240 that shows no obstacle readings. This is when the platform is in hover flight of about 20m and that there is certainly no obstacles or objects within 100m radius of the UAV at that height. .

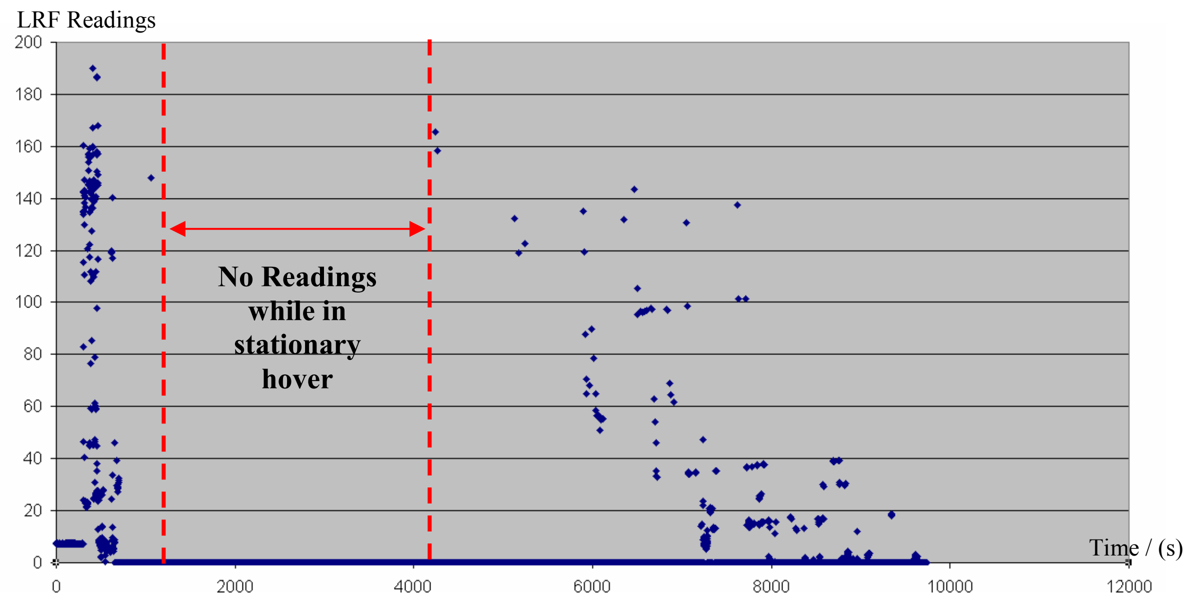


Figure 4.8: 1st RS100 LRF Flight Test

Thus, with the above experiment results, it can be shown that for a flight time of about 5 minutes, in an environment where there is no obstacle within a 100 meter range, the LRF shows that there are no readings of any obstacles at all. With this experiment, it can be shown that the LRF is not affected by the flight conditions of the UAV and thus can be used for the CAS system.

4.5 CAS Architecture

Given the validation that the LRF can work in flight conditions, a complete design of the Collision Avoidance System can developed. The next section will focus on the implementation of the collision avoidance algorithm into a system diagram flowchart

4.5.1 Collision Avoidance Algorithm System Design

From Chapter 2, the final collision avoidance equation is as follows:

$$\boxed{\alpha = -\beta(\zeta)\frac{k}{\rho}}$$

Thus, in order to obtain the appropriate yaw rate, α, of the UAV, the following readings are required. They are the range reading, R, which will provide the non-dimensional value of ρ and the bearing of the obstacle with respect to the UAV, which will provide the non-dimensional value of $\beta(\zeta)$. The readings will then be computed and translated to the appropriate yaw rate for the UAV to execute the collision avoidance maneuver.

The following flowchart shows the system implementation of the collision avoidance algorithm.

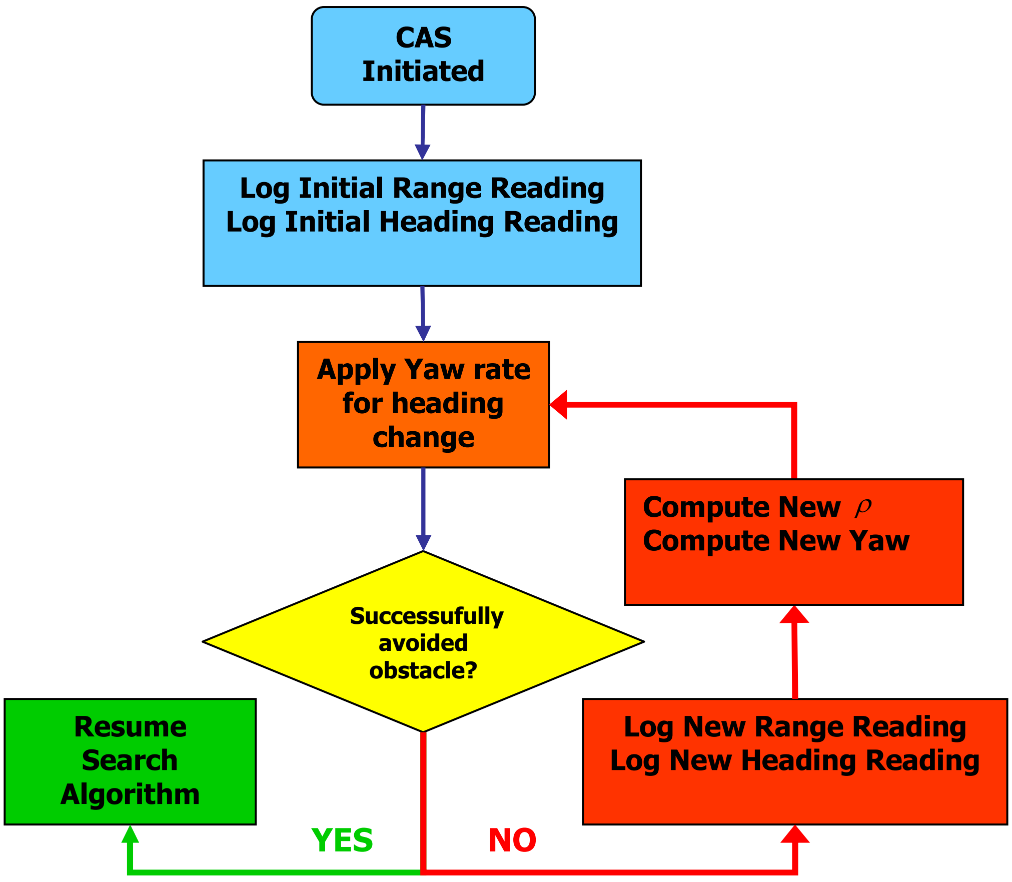


Figure 4.9: Collision Avoidance Algorithm Flowchart

In order for the collision avoidance algorithm to be successfully implemented on hardware, there a few hardware design requirements that will have to be satisfied. The following points are the main considerations for the successfully hardware implementation of the CAS.

1. Servo Sweeping Mechanism
2. Servo and LRF Controller Selection
3. Integration and Communications to PC104 flight computer
4. Pitch stabilized mount for CAS.

4.5.2 Servo Sweeping Mechanism

The RS100 LRF only gives a point range reading, mounting it stationary in front of the UAV will only be able detect obstacles straight ahead of the UAV with a negligible field of view. Hence, the CAS system will need to sweep the RS100 LRF to increase the LRF field of view.

The sweeping mechanism for the CAS is accomplished with servos. The servo is controlled by the Basic Stamp BS2x which will be further explained in the following section. The servo specifications are as listed below.

Servo	Futaba S3003	
Dimension	4.1 x 2.0 x 3.6 cm	
Weight	37.0 g	
Volt (V)	Torque (Nm)	Speed (^{o}/s)
4.8	0.313	260.9
6.0	0.402	315.8

Figure 4.10: Servo Specification

The sweeping motion is controlled by having the servo move from 45 degrees to 135 degrees in 50 separate small steps. This is the smallest resolution that the servo is capable of and is typical of most off the shelf servos. Hence, the resolution of the field of view of the CAS is at intervals of 1.8 degrees. As the project is collaboration between the CAS project and a search algorithm project, the requirement for the CAS sensing range is 8-10m. Thus, in order meet this requirement; the BS2x is configured to only relay range readings that are < 10m. With the above conditions, the Figure 4.11 shows the minimum obstacle size that the CAS with a sweeping mechanism is capable of detecting.

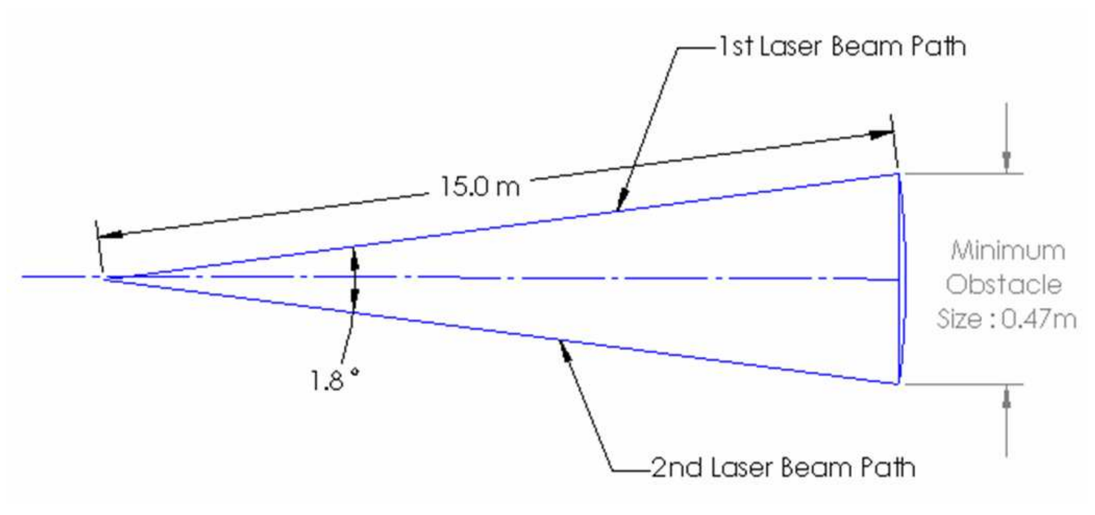


Figure 4.11: CAS Minimum Obstacle Size

The following Figure 4.12 shows the mounting of the RS100 LRF on the servo

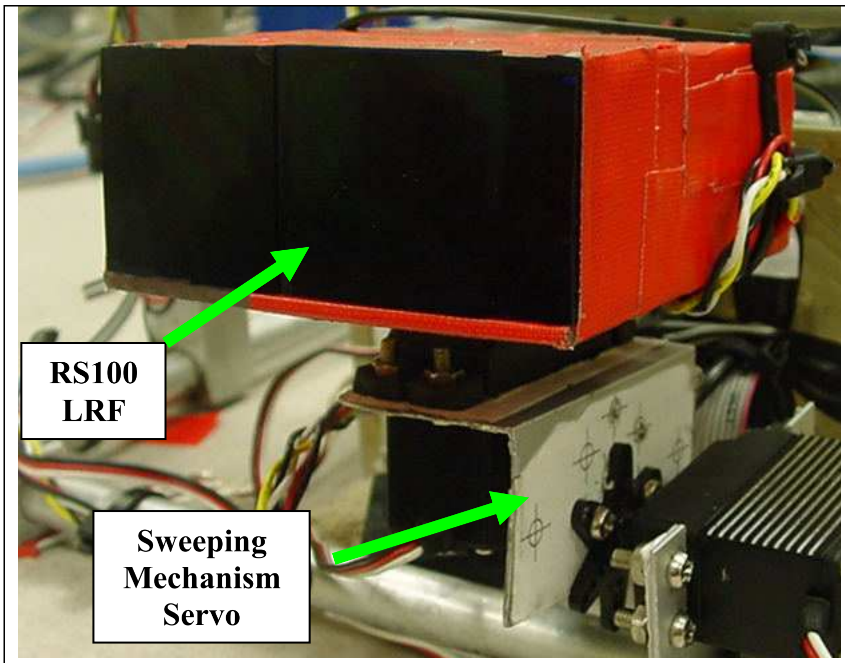


Figure 4.12: RS-100 on Servo

An experiment was carried out to test the functionality of the sweeping mechanism together with the RS100. The sweeping mechanism together with the RS100 was placed in the center of the lab. The actual layout of the room is as shown in the following Figure 4.13.

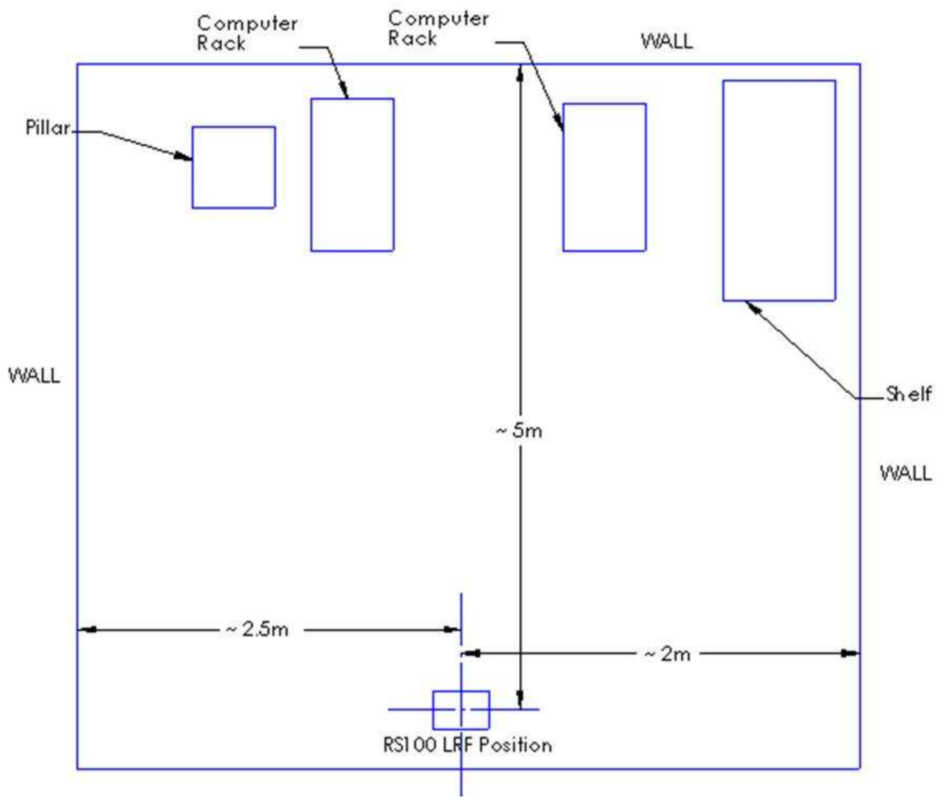


Figure 4.13: Actual Lab Layout

The RS100 with the sweeping mechanism was then started and readings were obtained. The readings were then plotted out and are shown in the Figure 4.14.

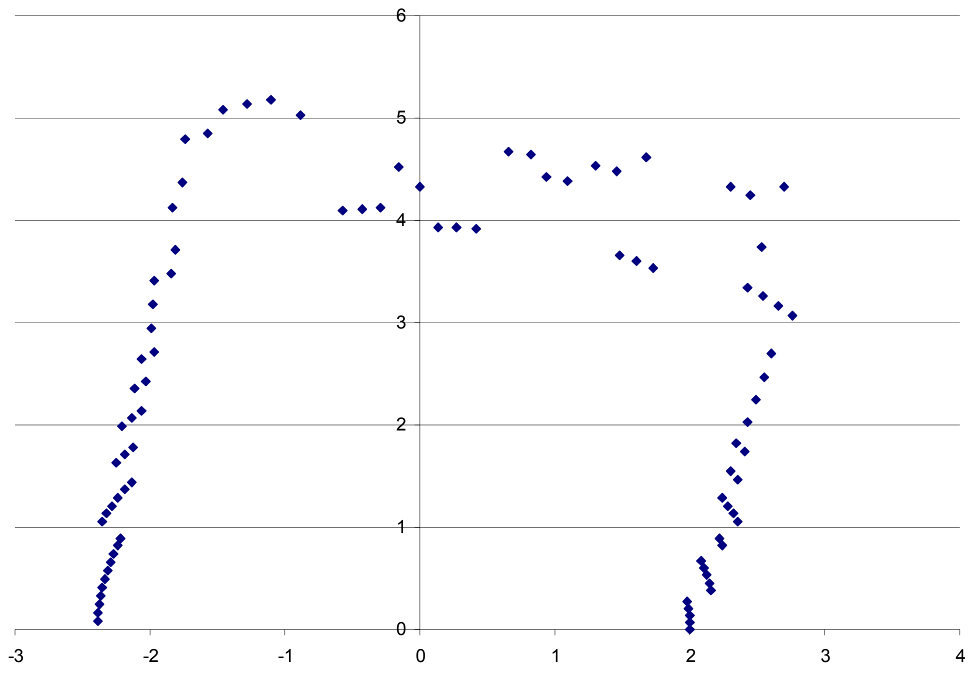


Figure 4.14: RS100 Room Profile Result

The sweeping mechanism combined with the RS100 is able to profile the lab and capture the more significant details of the lab. With the results as shown in the profiling experiment, the functionality of the LRF with the sweeping mechanism is verified.

4.5.3 CAS Controller Selection

The Controller acts as the bridge between the CAS equipment and the PC104 flight computer. The controller selected should meet the following requirements as listed:

1. Controlling the sweeping mechanism for the CAS
2. Reading range data from the RS100 LRF
3. Linking servo position with LRF input data when in that position– knowledge of range and bearing of obstacle with respect to UAV

If a controller can satisfy the above conditions, then the CAS system will be decoupled from the PC104 flight computer, and computing resources on the PC104 flight computer can be allocated to the CAS algorithm calculations and other mission objectives. With the above considerations, a controller was selected. The Basic Stamp BS2x was chosen to act as the control system for the RS100 LRF. It has the following specifications

1. 50 Mhz
2. 10,000 Instructions / sec
3. Readily available PWM servo control ports with programming library
4. Expandable Serial connection for LRF and PC104 communications

The following figure shows the Basic Stamp BS2x Controller

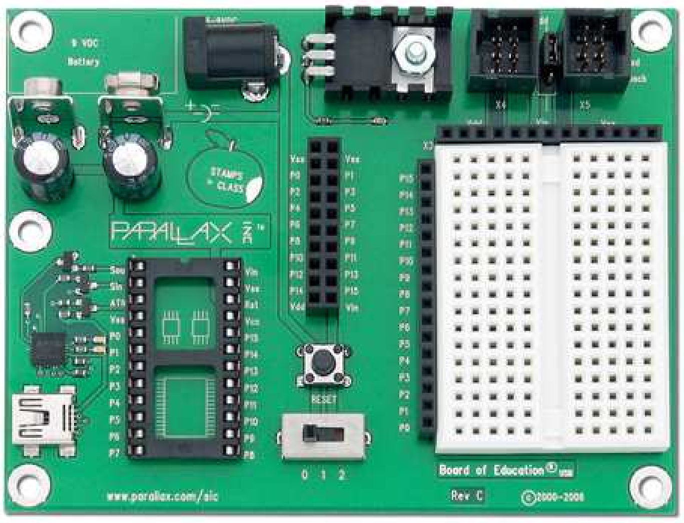


Figure 4.15: Basic Stamp BS2X Controller

Figure 4.16 below shows the system design of the CAS system with the BS2x as the controller.

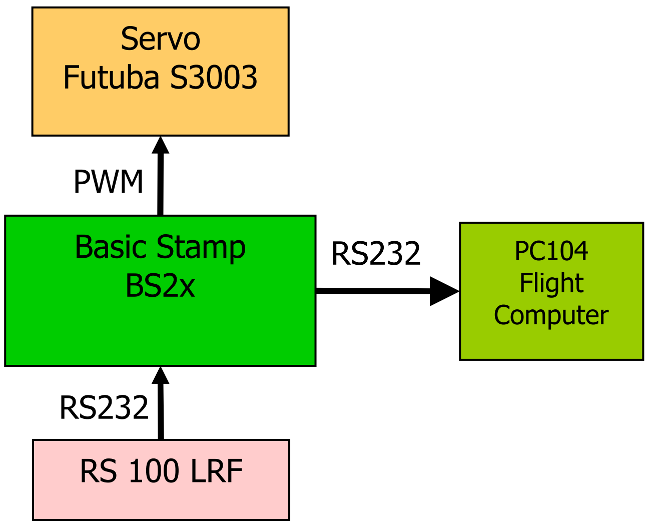


Figure 4.16: CAS System design with BS2X Controller

The RS100 requires a RS232 ANSI protocol for communications. It works well with a connection to the serial port of any PC and data can be obtained by using hyperterminal program that comes with all windows OS PCs. However, this is not an option as the aim is to decouple the RS100 from the PC104 flight computer. The BS2x comes with one built in RS232 port, this will allow communications from the RS100 to the BS2x.

The second serial port is an expansion upon the BS2x. This allows the BS2x to have an additional serial port which will allow communications to the PC104 flight computer.

Hence, the idea is to obtain readings from the RS100 through serial port 1, and then process the readings slightly before sending it out through serial port 2 to the flight computer for the CAS algorithm. All these while sending a background command to the servo to create the sweeping mechanism for increasing the field of view of the CAS.

The Basic Stamp BS2x is very well suited to perform the task as a controller as it has ready made PWM ports and at the same time allows communications directly with the RS100 and the flight computer with RS232 ports. Figure 4.17 below shows the actual BS2x setup with the Servos and the RS100 LRF connected.

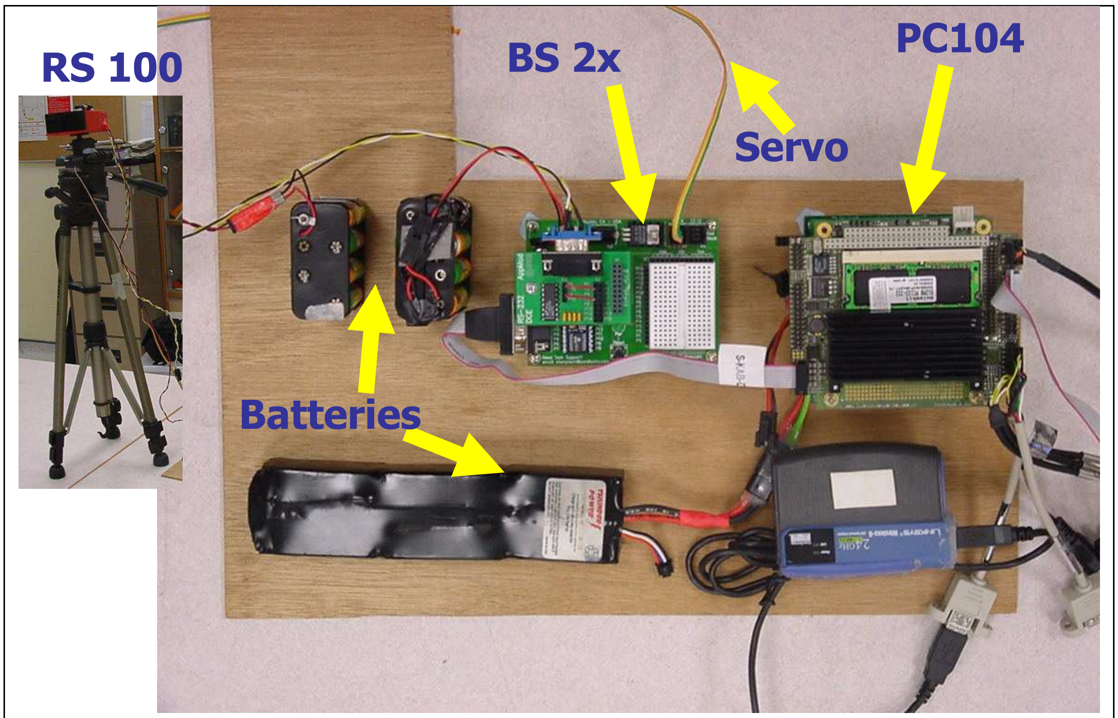


Figure 4.17: New CAS Hardware Setup

This will allow the range reading to be stamped with an angular reading, indicating the range at a particular angle with respect to the flight platform. Also, the ability to run both the servo sweep mechanism and the RS100 LRF achieves true decoupling of the CAS from the flight computer. This leaves the resources on the flight computer for the computation of the search algorithm.

4.5.4 Gyro Controlled Pitch Stabilised Mount

The pitch stabilized mount is essential in implementing a working CAS system. During forward flight, the flight platform pitches forward by a few degrees. If the CAS is mounted rigidly onto the flight platform, the CAS will be pointing towards the ground and will register the ground as obstacles. This misreading could lead to activation of the CAS evasive maneuver in a situation that does not require such maneuvers and might even crash the flight platform. As such, a pitch stabilized mount had to be designed.

The design of the pitch stabilized mount utilized a RC Helicopter Gyro sensor with head-holding features. This Gyro sensor is usually used to stabilize the helicopters in yaw direction and head-holding ensures that the helicopter maintains the same heading at all times. The gyro sensor that was selected was the Futaba GY 601 Heading Holding Gyro. The following figure shows the specifications of the Gyro with the digital servo.

Futaba GY 601 c/w Futaba Servo S9251	
• Super narrow pulse of 760μs • High Resolution 12bit A/D Conversion • SMM (Silicon Micro Machine) Gyro sensor • Coupled with S9251 high speed Futuba Digital Servo • AVCS Head Holding Capability • High Response speed of 857^{o} / sec	

Figure 4.18: Gyro Specification

As stated in the specifications of the sensor, the GY601 is capable of correcting changes in yaw directions of up to 857^{o} / sec. From the flight logs of the UAV tests, the pitch changes have never had rates of more than 100^{o} / sec.

Thus by installing the GY601 to detect attitude change in the pitch direction and having the high speed digital servo to correct that change, the GY601 would be able to provide the solution for the pitch stabilized mount.

With the AVCS Head Holding Function, the GY601 is capable of holding a position memory of the Servo Position at initialization. Thus, this head holding function is utilized for holding the CAS LRF Sensor at the correct pitch level, which is level to the ground. The GY601 will be initialized when the UAV is on the ground and level with it, and it will hold the digital servo position at this ground level no matter the perturbation of the pitch angle. Thus, once the initialization is done, the UAV can take off and the servo will hold the CAS LRF sensor at the original ground pitch level no matter the pitch attitude of the UAV.

All that is required is to mount the sensor on the flight platform, in the pitch direction and mount the RS100 with the sweeping mechanism onto a servo linked to the gyro. With such a system, the CAS will be able achieve pitch stabilized mounting. Figure 4.19 below shows the Gyro linked pitch stabilized mount.

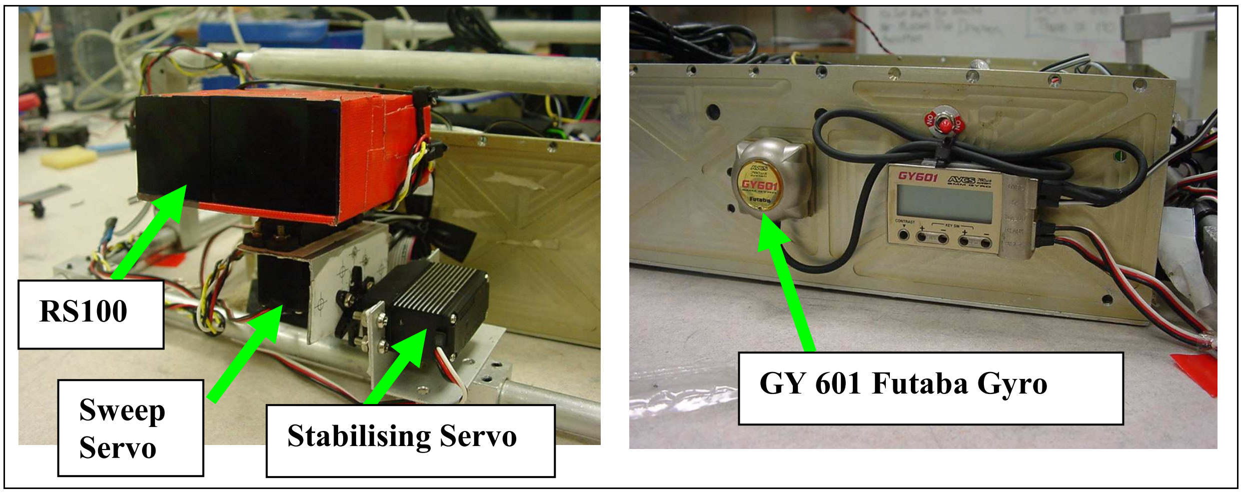


Figure 4.19: Gyro linked stablised mount

4.5.5 Complete CAS Design

With the above components of the collision avoidance hardware in place, the figure below shows a full system integration of the collision avoidance hardware and the interface with the UAV.

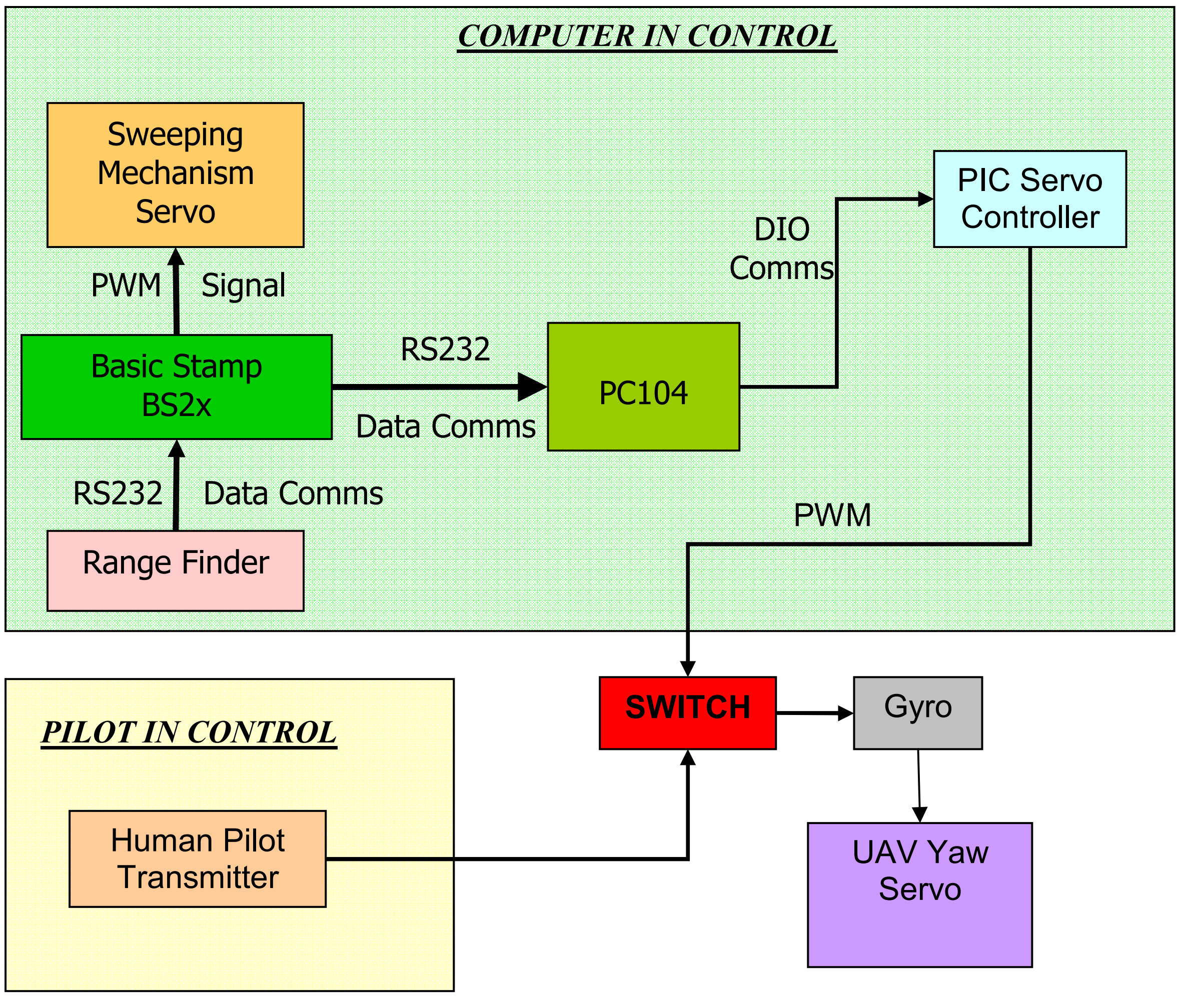


Figure 4.20: Complete CAS System Design

As seen from the figure above, there is a switch which toggles the control of the UAV yaw servo between the pilot (yellow zone) and the computer (green zone). This switch is pilot controlled so the control of the UAV yaw servo can be returned to the pilot the moment he finds the UAV going out of control. But once the pilot relinquishes control of the yaw servo to the collision avoidance system, it will be controlled by the PC104 flight computer which in turn obtains input from the CAS system. Thus, the pilot can maintain the stability of the UAV but the command to avoid the obstacle will come from the CAS system.

4.6 Chapter Summary

The CAS Design Architecture has been developed and eventually implemented in hardware. It started out with the implementation and integration of the sonar sensors with the UAV PC104 flight computer.

A whole series of tests were performed to test the CAS system with sonar sensors on ground before taking the step into flight testing with the sonar sensors. The sonar sensors showed good performances on the ground with less than 2% error with lab testing of the sonar sensors. However, upon flight testing, the sonar sensors failed to perform with sufficient accuracy, registering "phantom objects" when there is evidently no object or obstacle within the sonar sensor's range. Eventually, more tests revealed the downwash of the rotor from the UAV to be the key interference to the sonar sensors. Thus, a new sensor was selected for the CAS system.

The RS100 Laser Range Finder was selected as the new range sensor and one of the first tests was a flight test to verify the performance of the laser range finder in flight. A flight test was conducted and showed that the LRF can perform in flight conditions. Thus, the development to the CAS system could proceed and eventually completed. The summary below lists the hardware implementation and the design purpose it serves.

<u>Servo Sweeping Mechanism:</u>

Increased the LRF field of view from a single point to a 90 frontal sweep with a 2hz range reading frequency

Servo and LRF Controller Selection:

Serves as a communications hub between the LRF and the PC104 flight computer. It also controls the servo that performs the CAS sweeping mechanism. Lastly, it decouples the reading and controlling of the sweeping mechanism from the PC104 flight computer, freeing up precious computing resource onboard the PC104 flight computer for other more important tasks.

Pitch stabilized mount for CAS:

Enables the CAS LRF to constantly point forward in the direction of flight even when the UAV is pitching forward due to forward flight. This is essential for the proper functioning of the CAS system.

CHAPTER 5 - FIELD VALIDATION

In chapter 3, simulation of the collision avoidance algorithm was performed and results obtained. Chapter 4 details the work done on the hardware integration of various components onto the UAV to produce a UAV with CAS capabilities. Now in this chapter, the actual UAV with CAS capabilities is brought out to the field and tested. Results are obtained and they are compared along side the simulation results obtained in chapter 3. The following sections will detail the process of obtaining the data and the evidence of the actual flight tests conducted.

5.1 Field Validation Setup

The simulation of the collision avoidance algorithm in chapter 3 has shown that the best gain, k value to use is 2.04. A possible approach to obtain field results is to carry out the collision avoidance algorithm at various gain, k values including k = 2.0, capture the position information of the UAV and compare the results. However, GPS sensors that are available currently, even using DGPS will only provide a position with an error of about 3m which is too big an error for the requirements of this experiment. This makes accurate positioning of the UAV a problem. The approach to this problem is to use an alternate positioning solution.

5.1.1 Video Positioning – Wireless Camera

A solution to the accurate positioning data required for field validation is to use a wireless camera mounted onto the UAV and record videos of the ground. The ground will be laid with grids with number tags which radiate out from the obstacle center. While the UAV is flying and performing collision avoidance maneuver, it will be flying over these grids and the video camera recording will be able to provide accurate information on the position of the UAV with respect to the obstacle. The following diagram shows the wireless video camera attached onto the UAV.

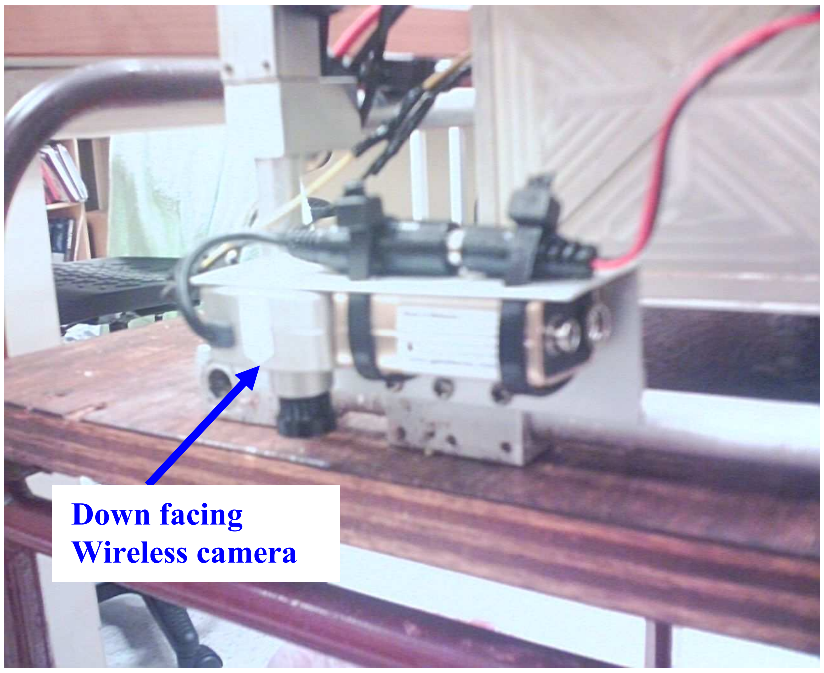


Figure 5.1: Wireless Camera Attached onto UAV Skid

This wireless video camera will record the flight of the UAV, facing down. There will be grids laid out over the flight zone so as to accurately determine the position of the UAV with respect to the obstacle.

5.1.2 Video Positioning – Grid Design

The video camera recording will be meaningless if there are no grids on the ground that is with respect to the obstacle. A grid design is shown in the following diagram.

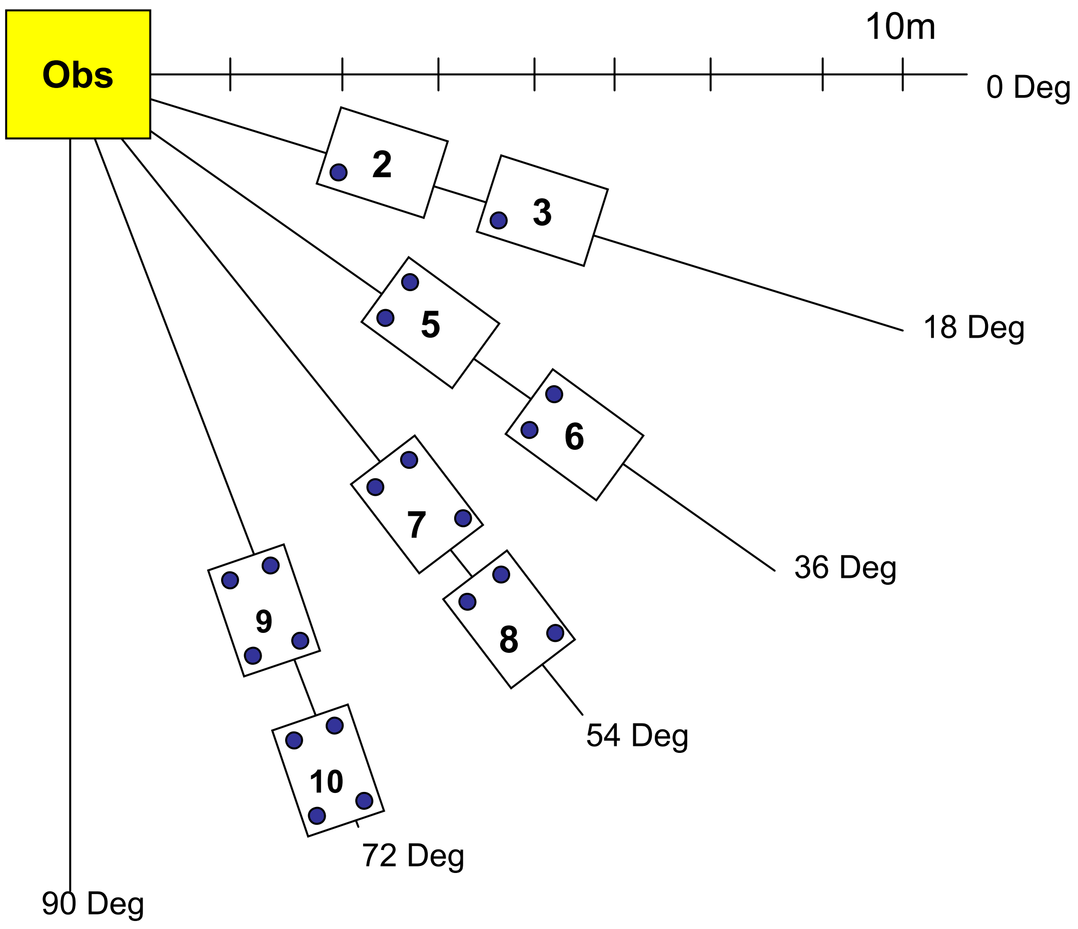


Figure 5.2: Grid Design

The design shows 4 grid lines that radiate out of the obstacle. These grid lines will have to be laid out in 18 degree intervals so that they are regularly spaced. Each grid will also be lined with number tags, each spaced out at 1m intervals up to 15m. Also, the number tags will have dots printed on the different corners. Each number tag on the same grid line will have the same number of dots printed on their corners so that it will be easy to recognize on the video which exact grid line is recorded when reviewing the video. So when the UAV performs an avoidance maneuver, the video camera that is facing downwards will be able to record grids with numbers on them. By reviewing that video, flight path data can be obtained and will be discussed in a later chapter.

5.1.3 Grid Design Setup

The setting up of the grids requires a special tool to ensure that the segments are 18 degrees apart from one another. The angles are important as they will determine the accuracy of the actual flight path information from the video feed. A theodolite is used to assist in the setting up of the grids. It is shown in the figure below.

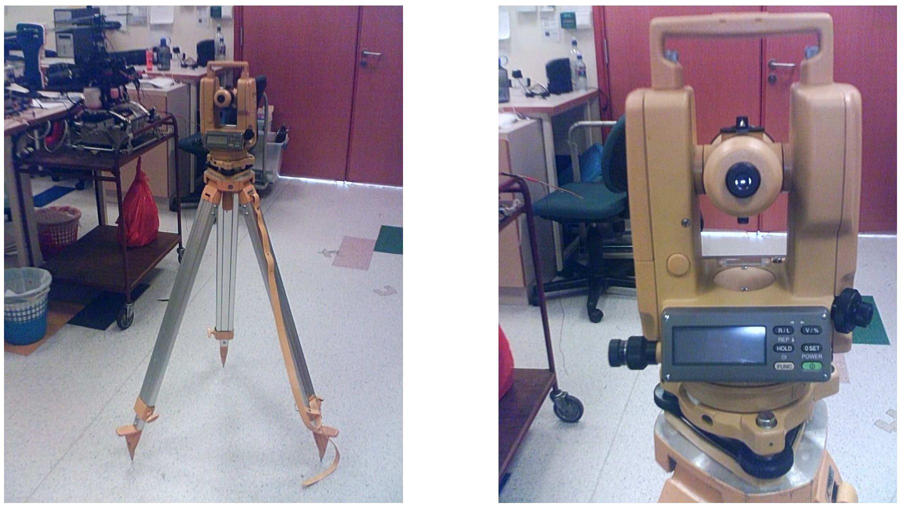


Figure 5.3: Theodolite

Theodolite is a civil engineering surveying tool able to provide accurate angle measurements in the azimuth and elevation plane. With the theodolite, the grids can be successfully set up with a high degree of accuracy. The following diagram will show the actual setup of the grids in the field.

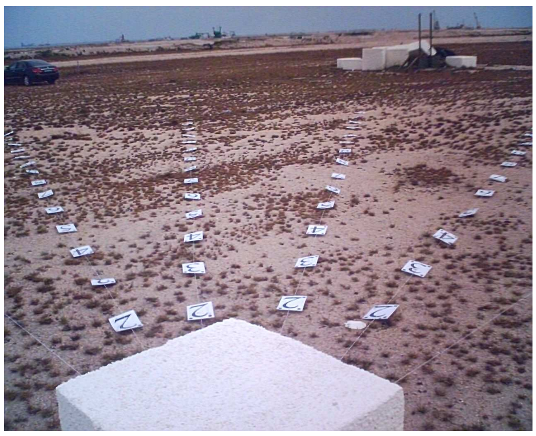
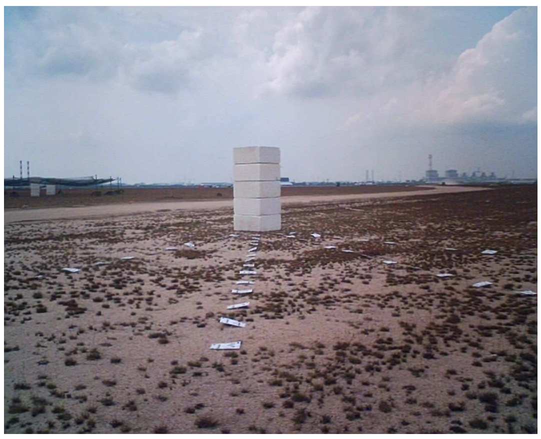



Figure 5.4: Actual Grid Setup Outfield

The figure above shows the obstacle setup as well. The obstacle is constructed out of stacked styrofoam blocks, each 1.2m x 1.2m x 1.0m in dimension. A total of 5 styrofoam blocks is stacked on top of one another forming an column 1.2m x 1.2m x 5.0m in dimension. Hence, when performing the collision avoidance algorithm, the UAV is constrained to a height limit of 5.0m.

5.2 Field Validation Experiments

Field validation experiments consist of a series of collision avoidance maneuvers performed by the UAV. The UAV is semi-pilot controlled, with the yaw control relinquished to the UAV PC104 flight computer and the pilot responsible for the stability and forward motion of the UAV. In each of the experiments, the UAV is made to perform a collision avoidance maneuver by flying head on towards the obstacle. The experiment is then repeated for different values of k.

5.2.1 Collision Avoidance Algorithm Functionality Test

Before the k value experiments were conducted, 2 basic flight tests were carried out to test the basic functionality of the collision avoidance algorithm.

The picture slide figure below shows one of the flights with the UAV made to fly in a head on collision course with the obstacle.

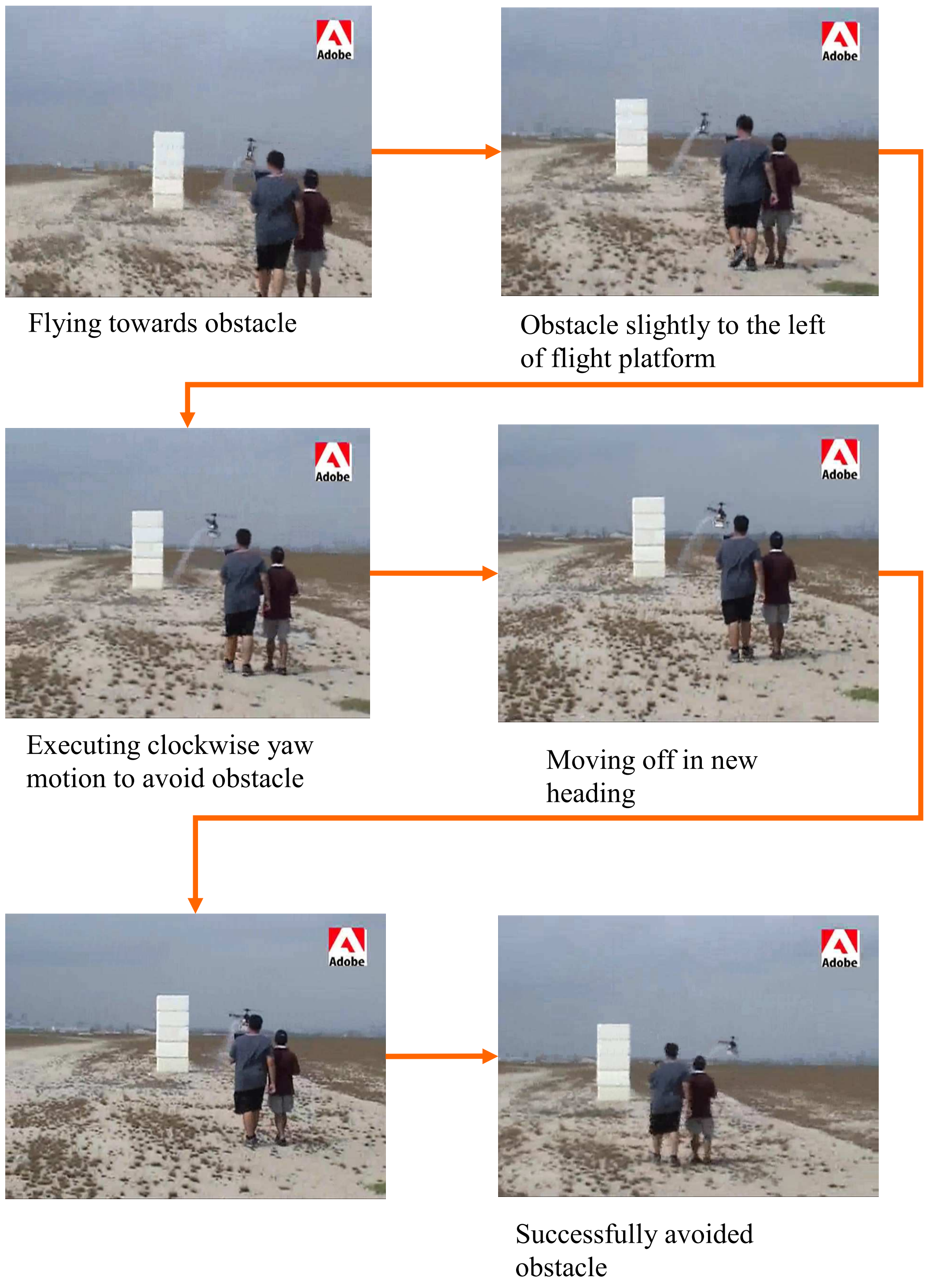

Flying towards obstacle
Obstacle slightly to the left of flight platform
Executing clockwise yaw motion to avoid obstacle
Moving off in new heading
Successfully avoided obstacle

Figure 5.5: Collision Avoidance Maneuver to the Right of Obstacle

As seen from the above picture slide show, the collision avoidance algorithm was successfully performed by the UAV flight computer. In order to test the robustness of the algorithm, another head on collision flight test was conducted as shown in the picture slide figure below.

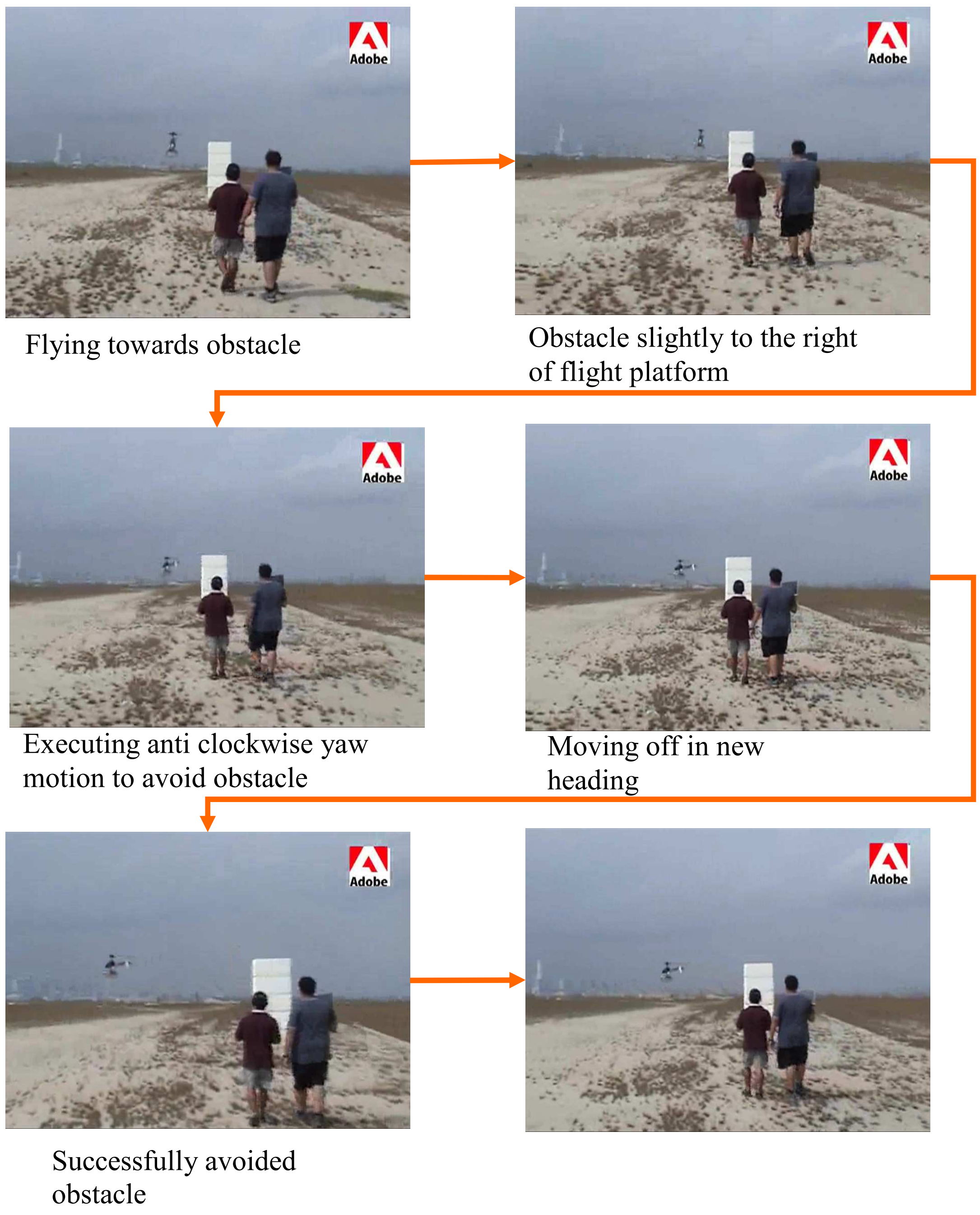

Adobe
Flying towards obstacle
Obstacle slightly to the right of flight platform
Executing anti clockwise yaw motion to avoid obstacle
Moving off in new heading
Successfully avoided obstacle

Figure 5.6: Collision Avoidance Maneuver to the Left of Obstacle

The difference in this flight test is that the obstacle is placed slightly to the right of the UAV. This is to determine if the collision avoidance was performing as per the design, to yaw anticlockwise when the obstacle is slightly to the right of the UAV. As shown in the picture slide show above, the UAV performs as designed. These 2 tests have proven the basic functionality of the CAS system.

5.2.2 Varying gain, k value experiments

As described in chapter 3, a range of values of k were simulated to test the collision avoidance algorithm. The field experiments aim to test a few k values. For each of these k values, flight paths data for a few different k values performed by the actual hardware is comparable to the flight paths generated by the simulation. By comparing the flight paths, a conclusion can be drawn as to whether the actual hardware agrees with the simulated results. The experiments conducted and conditions are shown in the table below

Gain, k	Condition	No. of Runs
1	Head on Collision	3
2	Head on Collision	3
3	Head on Collision	3

Table 5.1: Varying Gain, k Experiments

For each of the gain values, the UAV is made to perform a stationary hover with it's bearing aligned to the obstacle, but beyond detection range of the CAS system. It will then be made to fly straight ahead, on a head on collision path with the obstacle. The CAS system will then activate and perform the CAS maneuver with the results captured on video by the wireless camera.

5.2.3 Image Analysis

The videos are then analyzed and played slowly frame by frame. Whenever the frame captures the number tag on the grid lines in the center of the video frame, the frame is captured and printed out as a picture for analysis. The figure below shows a typical picture captured by the video camera which is taken for analysis.

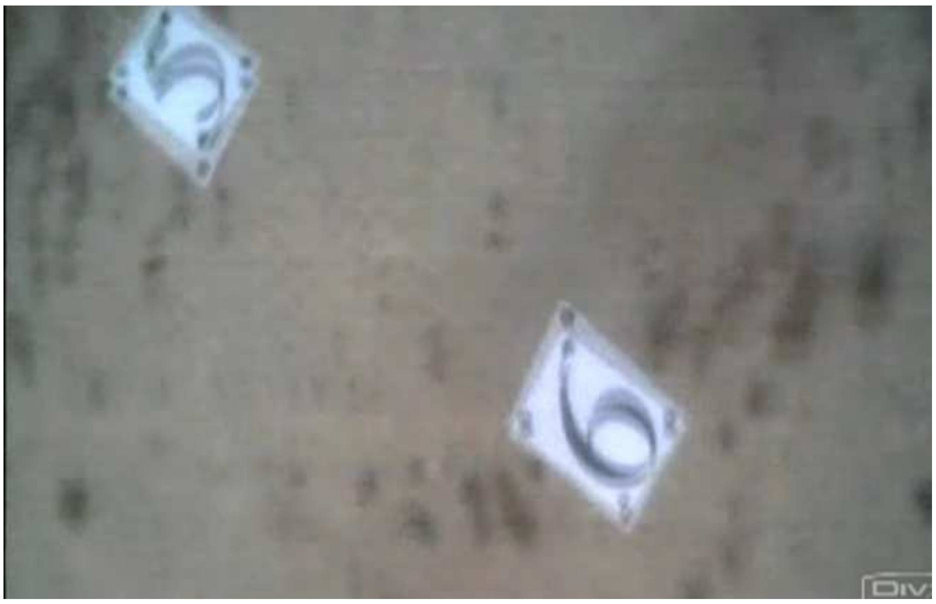


Figure 5.7: Screen Capture of Flight Video

For each of the pictures that are obtained from the flight test videos, measurements can be taken from the pictures that will provide position information of the UAV with respect to the obstacle. The figure below will show how the measurements are performed on the pictures obtained from the video.

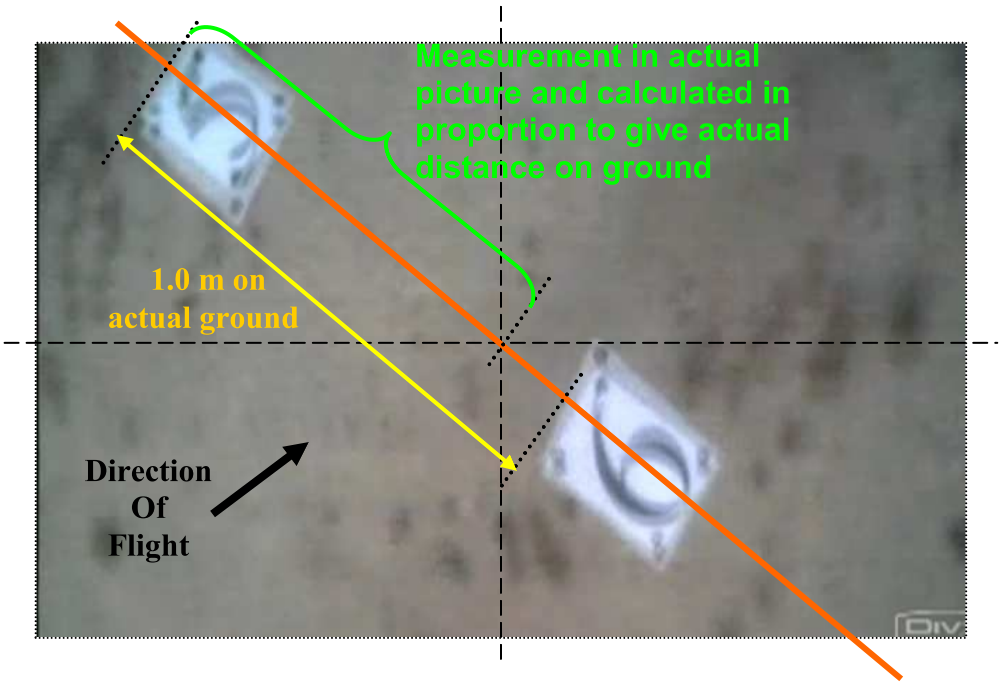

Measurement in actual picture and calculated in proportion to give actual distance on ground
1.0 m on actual ground
Direction Of Flight

Figure 5.8: Measurements made on Screen Capture of Flight Video

As shown in the picture, the measurement taken from the picture can be proportionally calculated to determine the actual distance that the UAV is away from the obstacle.

The attitude of the UAV is also taken into account for the measurement. The micropilot onboard the UAV data logs the attitudes and altitude of the UAV. The pitch angle and the altitude of the UAV at the moment the picture was taken can be obtained from the flight logs. As shown in the diagram below, further correction of the calculated values is performed.

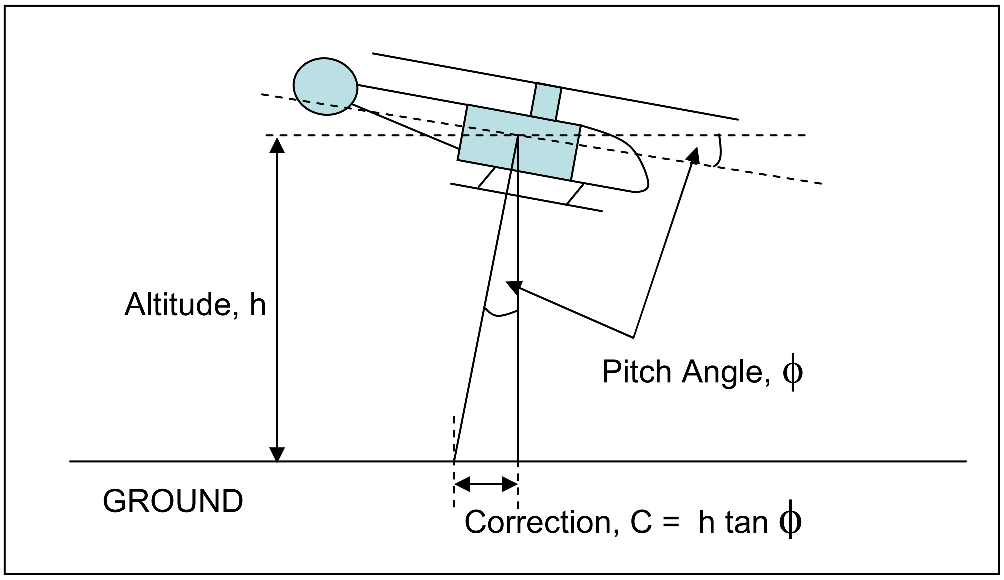


Figure 5.9: UAV Position Correction Calculation

As shown in the figure above, the correction, C is calculated based on pitch angle, ϕ and altitude, h. However, this does not directly affect the distance of UAV from the obstacle but more the grid angle that the UAV is directly on. The figure below shows how the angle translates to a correction of the grid angle.

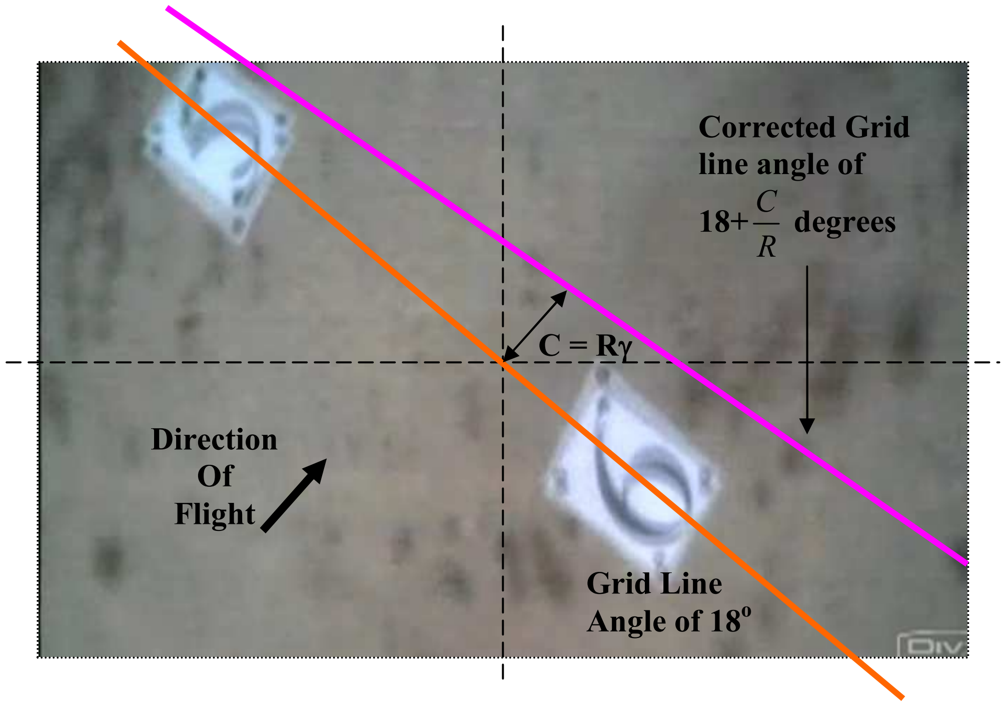


Figure 5.10: Corrected Calculation on Screen Capture

5.2.4 Field and Simulation Results

From the analysis of the images obtained from the video, the following results for each of the 3 gain values are obtained and shown below. The details of the analysis are recorded in Appendix.

5.2.4.1 Gain, K value = 1

The results for the flight test using gain k = 1 are as shown in the following diagram. It shows the simulated flight path and the field data readings plotted on the same chart to give a good basis for comparison.

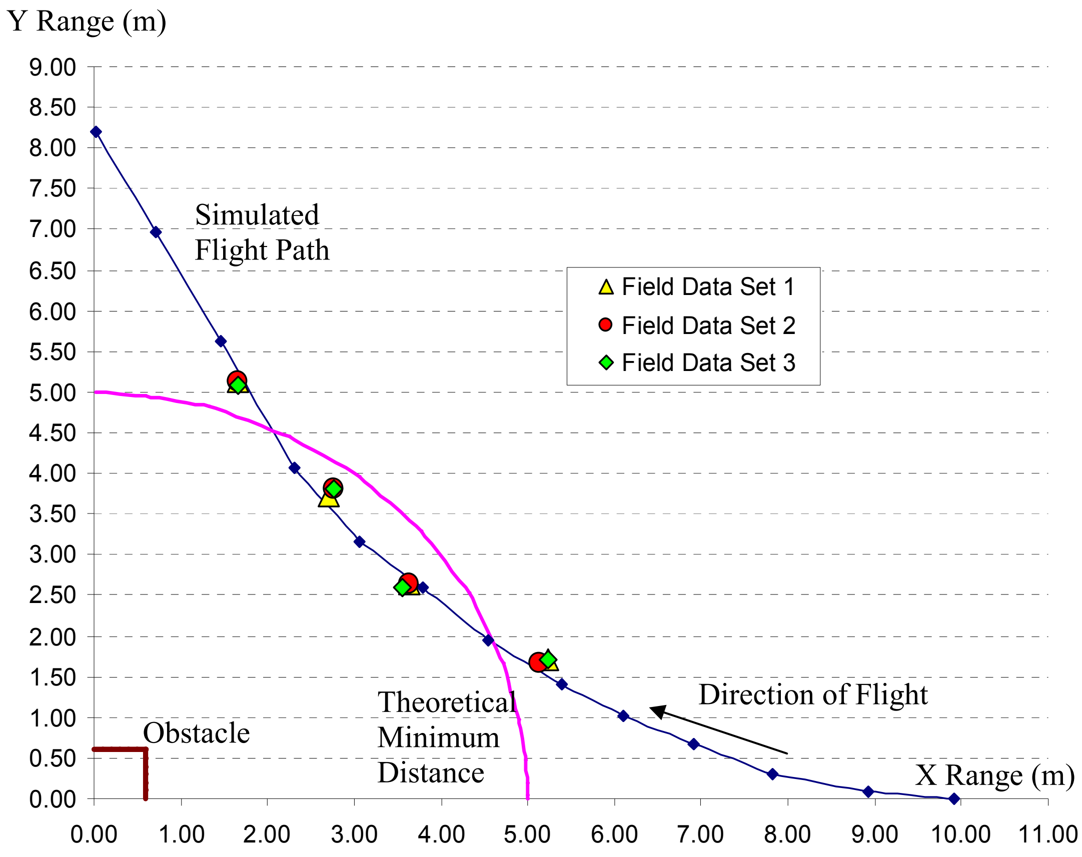


Figure 5.11: Simulated and Actual Flight Path Comparison for Gain, k = 1

The table below shows the above plotted results of the field data compared against the simulated data.

K=1	Corrected Grid Angle (^{o})	Simulation Range (m)	Field Results Range (m)	Percentage Error (%)
Field Data Set 1	18.04	5.44	5.50	-1.09
	36.05	4.52	4.50	0.49
	54.05	4.64	4.60	0.86
	72.04	5.55	5.40	2.70
Field Data Set 2	18.04	5.44	5.40	0.75
	36.04	4.52	4.50	0.49
	54.05	4.64	4.70	-1.29
	72.04	5.55	5.40	2.70
Field Data Set 3	18.03	5.44	5.50	-1.09
	36.04	4.52	4.40	2.70
	54.05	4.64	4.70	-1.29
	72.04	5.55	5.35	3.60

Table 5.2: Comparison of Field and Simulation Data for Gain, k =1

As seen from the table above, the maximum percentage error is 3.60% for the 4th reading of field data set 3, at a grid angle of 72.04^o.

5.2.4.2 Gain, K value = 2

The results for the flight test using gain k = 2 are as shown in the following diagram. It shows the simulated flight path and the field data readings plotted on the same chart to give a good basis for comparison.

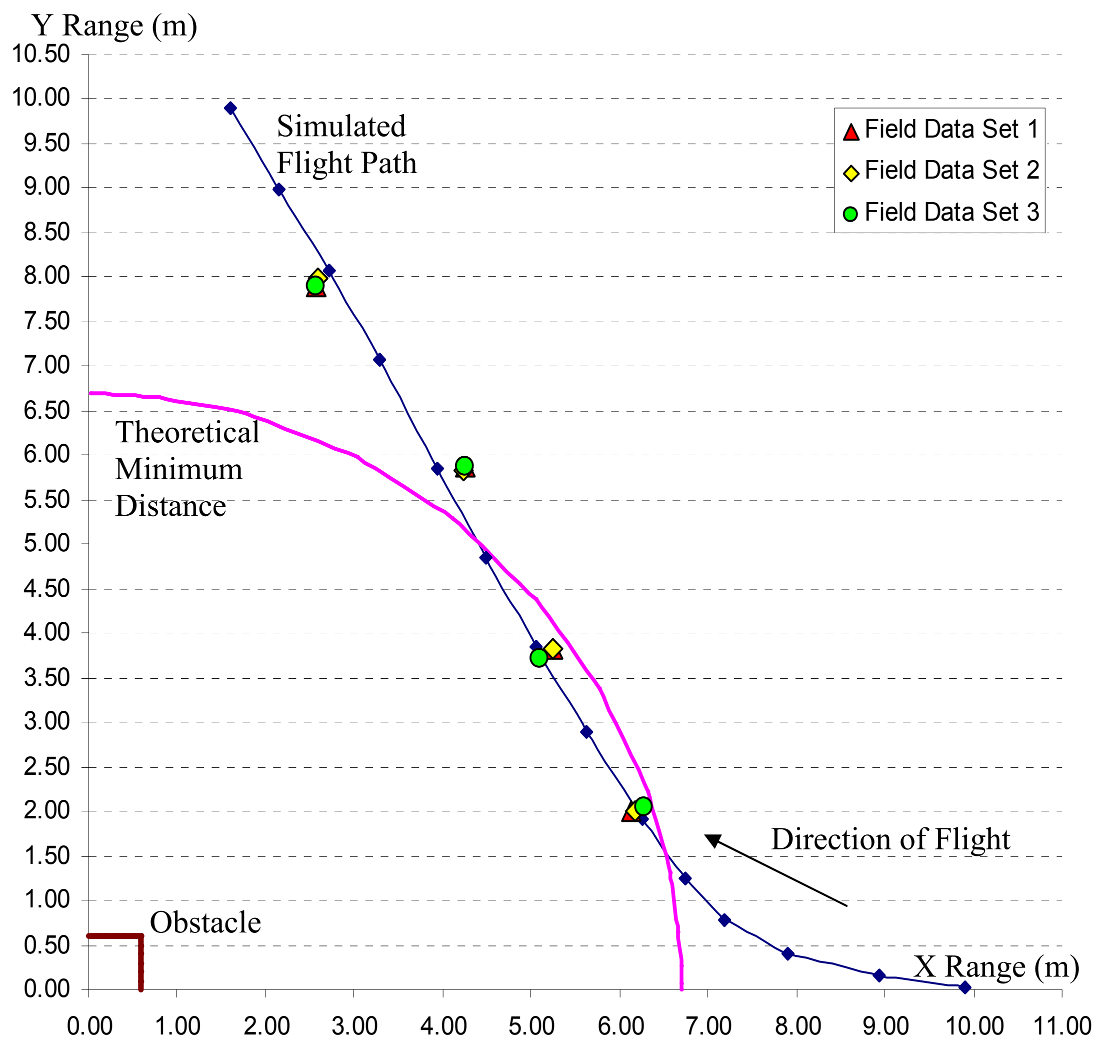


Figure 5.12: Simulated and Actual Flight Path Comparison for Gain, k = 2

The table below shows the above plotted results of the field data compared against the simulated data.

K=2	Corrected Grid Angle (^{o})	Simulation Range (m)	Field Results Range (m)	Percentage Error (%)
Field Data Set 1	18.04	6.51	6.45	0.99
	36.04	6.42	6.5	-1.20
	54.03	7.12	7.25	-1.80
	72.02	8.51	8.3	2.46
Field Data Set 2	18.03	6.51	6.5	0.22
	36.03	6.42	6.5	-1.20
	54.03	7.12	7.2	-1.10
	72.03	8.51	8.4	1.29
Field Data Set 3	18.03	6.51	6.6	-1.31
	36.04	6.42	6.3	1.91
	54.03	7.12	7.25	-1.80
	72.02	8.51	8.3	2.46

Table 5.3: Comparison of Field and Simulation Data for Gain, k =2

As seen from the table above, the maximum percentage error is 2.46% for the 4th reading of field data set 1 and 3, both at a grid angle of 72.02^o.

5.2.4.3 Gain, K value = 3

The results for the flight test using gain k = 3 are as shown in the following diagram. It shows the simulated flight path and the field data readings plotted on the same chart to give a good basis for comparison.

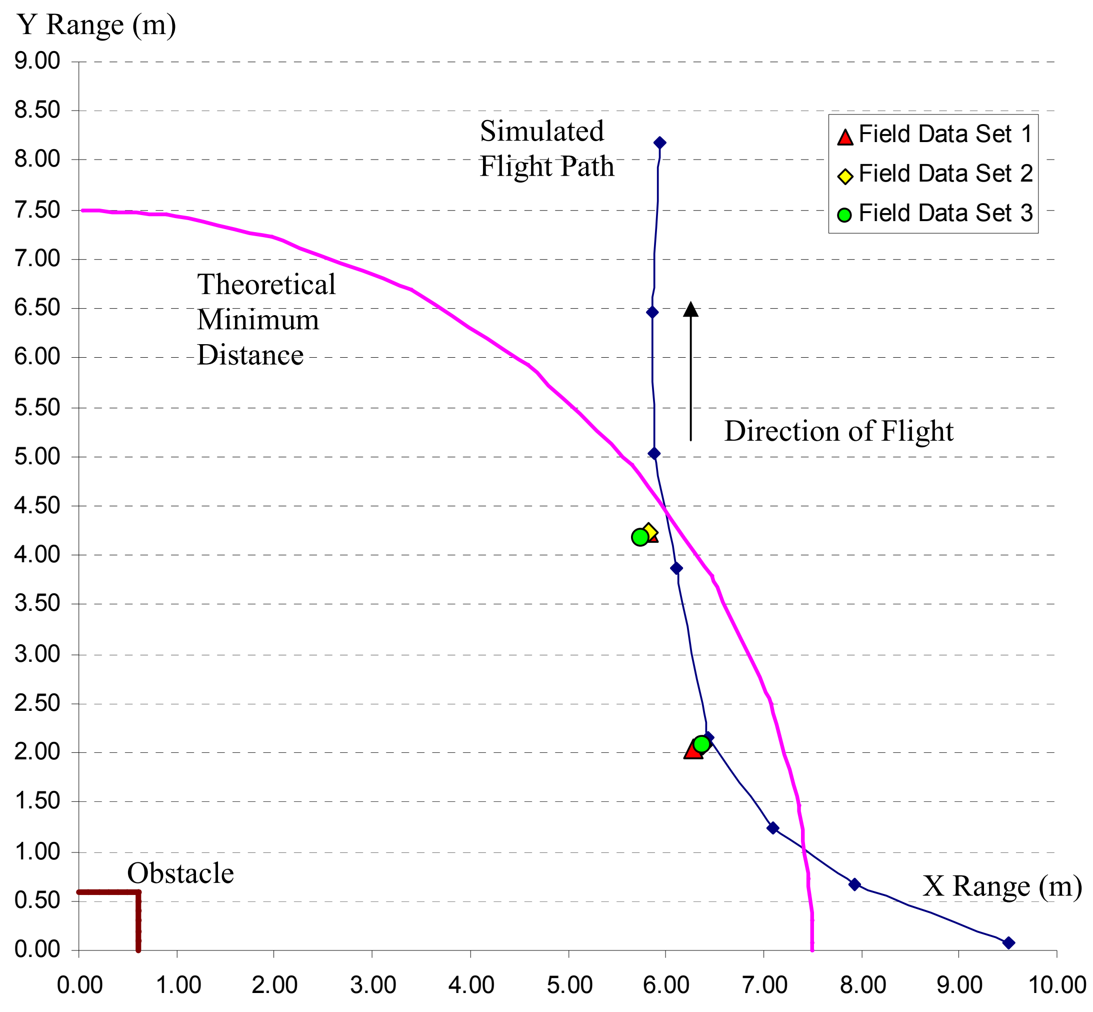


Figure 5.13: Simulated and Actual Flight Path Comparison for Gain, k = 3

There are only 2 sets of readings for each run of the field experiments. This is because the UAV has flown out of the grid's range by the time the UAV flies over the 3 grid line's angle. However, as shown in the results, by the 3rd grid angle of 54^{o}, the UAV is well out of theoretical minimum safety distance.

The table below shows the above plotted results of the field data compared against the simulated data.

K=3	Corrected Grid Angle (^{o})	Simulation Range (m)	Field Results Range (m)	Percentage Error (%)
Field Data Set 1	18.03	6.86	6.6	3.82
	36.03	7.68	7.2	6.30
Field Data Set 2	18.03	6.86	6.70	2.36
	36.03	7.68	7.20	6.30
Field Data Set 3	18.03	6.86	6.70	2.36
	36.03	7.68	7.10	7.61

Table 5.4: Comparison of Field and Simulation Data for Gain, k =3

As seen from the table above, the maximum percentage error is 7.61% for the 2nd reading of the field data reading 3, at a grid angle of 36.03^o

5.2.4.4 Results comparison

The aim of the field experiments is to determine if the performance of the actual hardware that was put together agrees with the simulation results. As seen from Figure 5.11 and Figure 5.12, the position data of the UAV fall within a 4% error for gain k = 1 and gain k =2. Also, for gain, k value of 1 and 2, the maximum percentage error was 3.60% and 2.46%. This is acceptable as the hardware is clearly performing close to the simulation results as seen from the flight path plots.

For the flight gain, k = 3, the errors are larger, with a maximum error of 7.61%. The effect of the decreasing fuel is also evident in the results of gain, k = 3, but the errors are more pronounced as compared to gain, k = 1 and 2. This could be due to the higher yaw rate that k=3 requires of the flight platform. This causes the performance to be different from those predicted in the simulation, hence the larger percentage errors for gain k = 3.

5.3 Chapter Summary

In this chapter, the setup of the field validation experiments was explained. Special attention was paid to the equipment, a theodolite, which was used in the setup to ensure the accuracy of the experimental results obtained. A wireless video camera was attached to the UAV and a video recording of the grid lines that it was flying past was taken. Using the video, accurate results of the position of the UAV with respect to the obstacle can be obtained.

These results are then compared to the simulation results that were obtained in chapter 3. The errors are within a 4% range for k values 1 and 2 and a maximum error of 7.6% was recorded for k = 3.Considering the complexity of the outfield experiments, the outfield results error are within a small range and considered to agree with the simulation results as obtained in chapter 3, verifying a working CAS system in hardware.

CHAPTER 6 - CONCLUSION

The objective of the project is to verify a collision avoidance algorithm implemented onto an actual hardware system for the UAV. The terrain and obstacles are unknown to the UAV before flight and collision avoidance is expected of the UAV using information relayed only through onboard sensors

Firstly, the collision avoidance algorithm based on the PN guidance system was adopted and put to a detailed theoretical study. This allowed for understanding and using of sensory information to achieve collision avoidance in UAVs. This is shown in a simulation study to be a successful form of collision avoidance of stationary obstacles implemented in simulation on a UAV.

Outfield testing of the actual hardware implementation of the collision avoidance system onboard a UAV, made to fly towards a stationary obstacle was conducted and results obtained. The comparative study of the outfield results and the simulation results show that the field results that are collected have errors that are within a 4% range for k values 1 and 2 and a maximum error of 7.6% was recorded for $k = 3$. Considering the complexity of the outfield experiments, the outfield results error are within a small range and considered to agree with the simulation results as obtained.

Hence, through the comparative study, it can be concluded that the collision avoidance algorithm based on PN guidance can be implemented and shown to be fully functional when mounted onboard a UAV system.

There are several areas of that can be considered for future development. The first is to take this research one step further and research into inter-UAV collision avoidance, or collision avoidance for moving obstacles. This is one area that if research is completed successfully, will create a more advanced collision avoidance system.

Another field that extensive research can be done is the swarming or formation flight of these UAVs that have the collision avoidance built into them. Currently, swarming or formation flying usually requires some form of higher coordination, requiring high level of communications and also high quality of communications to maintain a formation or swarm. The reason primarily is to prevent collisions between units. With a reactive collision avoidance system based on onboard sensors, such communications can be reduced and could possible develop more sophisticated flight formations or swarms given successful research into formation and swarm technology.

REFERENCES

[1] Khatib, O, "Real-time obstacle avoidance for manipulators and mobile robots", *Proceedings of 1985 IEEE International Conference on Robotics and Automation, Volume 2, Mar 1985 Page(s):500 – 505*

[2] Javier Minguez, Luis Montano, Khatib O, "Reactive Collision Avoidance for Navigation with Dynamic Constraints", *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems.*

[3] J.Borenstein, Y. Koren, "Real-time obstacle avoidance for fast mobile robots" *IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept./Oct. 1989, pp. 1179-1187*

[4] J.L. Martinez, A. Pozo-Ruz, S.Pedraza and R. Fernandez, "Object following and obstacle avoidance using a laser scanner in the outdoor mobile robot Auriga" *Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems Victoria, B.C., Canada 9 October 1998*

[5] Merchant, J, Pope, F "Micro UAV collision avoidance" *Unmanned Systems Technology IX 6561: K5610-K5610, 2007. Proceedings of the Society of Photo-optical Instrumentation Engineers (SPIE)*

[6] Carnie R, Walker R, Corke P, "Image processing algorithms for UAV "sense and avoid"" *2006 IEEE International Conference on Robotics and Automation (ICRA) Vols:1-10 : 2848-2853, 2006*

[7] McCalmont JF, Utt J, Deschenes M, Taylor MJ, "Sense and avoid technology for Global Hawk and Predator UAVs" *Infrared Technology and Applications XXXI Pts 1 and 2, 5783: 684-692, Part 1-2 2005 Proceedings of the Society of Photo-optical Instrumentation Engineers (SPIE)*

[8] Lee DJ, Beard RW, Merrell PC, Zhan PC, "See and avoidance behaviors for autonomous navigation" *Mobile Robots XVII 5609: 23-34, 2004 Proceedings of the Society of Photo-optical Instrumentation Engineers (SPIE)*

[9] J. Candamo, R. Kasturi, D. Goldgof, S. Sarkar, "Vision-based on-board collision avoidance system for aircraft navigation" *Mobile Robots XVI 5619: 22-36, 2005 Proceedings of the Society of Photo-optical Instrumentation Engineers (SPIE)*

[10] Barfield, F, "Autonomous collision avoidance: the technical requirement" National *Aerospace and Electronics Conference, 2000. NAECON 2000. Proceedings of the IEEE 2000 10-12 Oct. 2000 Page(s):808 - 813*

[11] Kwag, YK, Choi, MS, Jung, CH, Hwang, KY, "Collision avoidance radar for UAV", *Proceedings of 2006 CIE International Conference on Radar, Vols 1 and 2 : 488-491, 2006*

[12] Fasano, G, Accardo, D, Moccia, A, Paparone, L, "Airborne multisensor tracking for autonomous collision avoidance" *2006 9TH International Conference on Information Fusion, Vols 1-4 : 1174-1180, 2006*

[13] B. Abdul-Baki, J. Baldwin, Marc-Philippe Rudel, "Independent Validation and Verification of the TCAS I1 Collision Avoidance Subsystem" , *IEEE AES Systems Magazine, August 2000*

[14] J. Asmat, B. Rhodes, J. Umansky, C. Villavicencio, A.Yunas, "UAS Safety: Unmanned Aerial Collision Avoidance System (UCAS)" *Proceedings of the 2006 Systems and Information Engineering Design Symposium*

[15] Bengt-Göran Sundqvist, Saab AB, "Auto-ACAS - Robust Nuisance-Free Collision Avoidance" *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*

[16] Xiaohua Wang, Vivek Yadav, and S. N. Balakrishnan, "Cooperative UAV Formation Flying With Obstacle/Collision Avoidance" *IEEE Transactions on Control Systems Technology Vol. 15, No. 4, July 2007*

[17] Ran Y. Gazit, J. David Powell "Aircraft Collision avoidance based on GPS position broadcasts" *IEEE Transactions on Control Systems Technology Vol. 6, No. 9, July 2003*

[18] E.Frew, R. Sengupta, "Obstacle Avoidance with Sensor Uncertainty for Small Unmanned Aircraft" *Office of Naval Research AINS #N00014-03-C-018*

[19] Donald E. Swihart, B. BrPnnsUrom, E. Griffin', R. Rosengren4, P. Doane "A sensor integration technique for preventing collisions between air vehicles" *SICE 2002 Aug 5-7 2002 Osaka*

[20] H. Fujishima, R. Teo, Leo SK, Maj Sng WB, A. Wong, "VFH-inspired Algorithm for Multiple UAV De-confliction"

[21] Pack, D, York, G, Toussaint, G, "Localizing mobile RF targets using multiple unmanned aerial vehicles with heterogeneous sensing capabilities" *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE 19-22 March 2005 Page(s):632 - 637*

[22] Zhang Feng, Tan Dalong, Wei Yingzi, "Obstacle Avoidance for Mobile Robots Based on Relative Coordinates" *Proceedings of the 2003 IEEE Changsha, China - October 2003 International Conference on Robotics,lntelligent Systems and Signal Processing*

[23] Su-Cheol Han, Hyochoong Bang "Proportional Navigation-Based Optimal Collision Avoidance for UAVs" *2nd International Conference on Autonomous Robots and Agents December 13-15, 2004 Palmerston North, New Zealand*

[24] G. Leng "Multibody Guidance for Smart Weapons" *MINDEF NUS-NTU Joint R&D Conference Yr 2000, Singapore*

[25] Sasiadek, J.Z, Hartana, P "Sensor fusion for navigation of an autonomous unmanned aerial vehicle" *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*

[26] Stuart, D.M, "Sensor design for unmanned aerial vehicle" *Aerospace Conference, 1997. Proceedings., IEEE Volume 3, 1-8 Feb. 1997 Page(s):285 - 295 vol.3*

[27] Doherty, P, "Knowledge Representation and Unmanned Aerial Vehicles" *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on 19-22 Sept. 2005 Page(s):9 - 16*

[28] Kim, J, Sukkarieh, S, "Autonomous airborne navigation in unknown terrain environment" *Aerospace and Electronic Systems, IEEE Transactions on Volume 40, Issue 3, July 2004 Page(s):1031 - 1045*

[29] Eun Soo Jang, Seul Jung, Hsia, T.C, "Collision Avoidance of a Mobile Robot for Moving Obstacles Based on Impedance Force Control Algorithm" *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on 02-06 Aug. 2005 Page(s):277 - 282*

[30] Seul Jung, Eun Soo Jang, Hsia, T.C, "Collision Avoidance of a Mobile Robot Using Intelligent Hybrid Force Control Technique" *Robotics and Automation, 2005. Proceedings of the 2005 IEEE International Conference on 18-22 April 2005 Page(s):4418 – 4423*

[31] Zhang Feng, Tan Dalong, Wei Yingzi, "Obstacle avoidance for mobile robots based on relative coordinates" *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on Volume 1, 8-13 Oct. 2003 Page(s):616 - 621 vol.1*

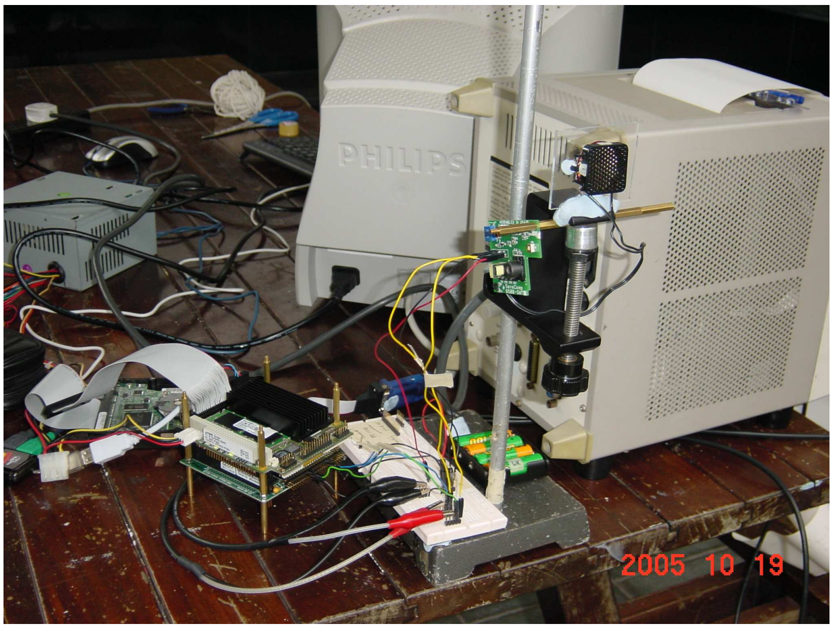
[32] Andrew D. Zeitlin, Michael P. McLaughlin, "Safety of Cooperative Collision Avoidance for Unmanned Aircraft" *25th Digital Avionics Systems Conference October 15, 2006*

[33] Rathinam, S, Sengupta, R, "Safe UAV navigation with sensor processing delays in an unknown environment" *Decision and Control, 2004. CDC. 43rd IEEE Conference on Volume 1, 14-17 Dec. 2004 Page(s):1081 - 1086 Vol.1*

APPENDIX

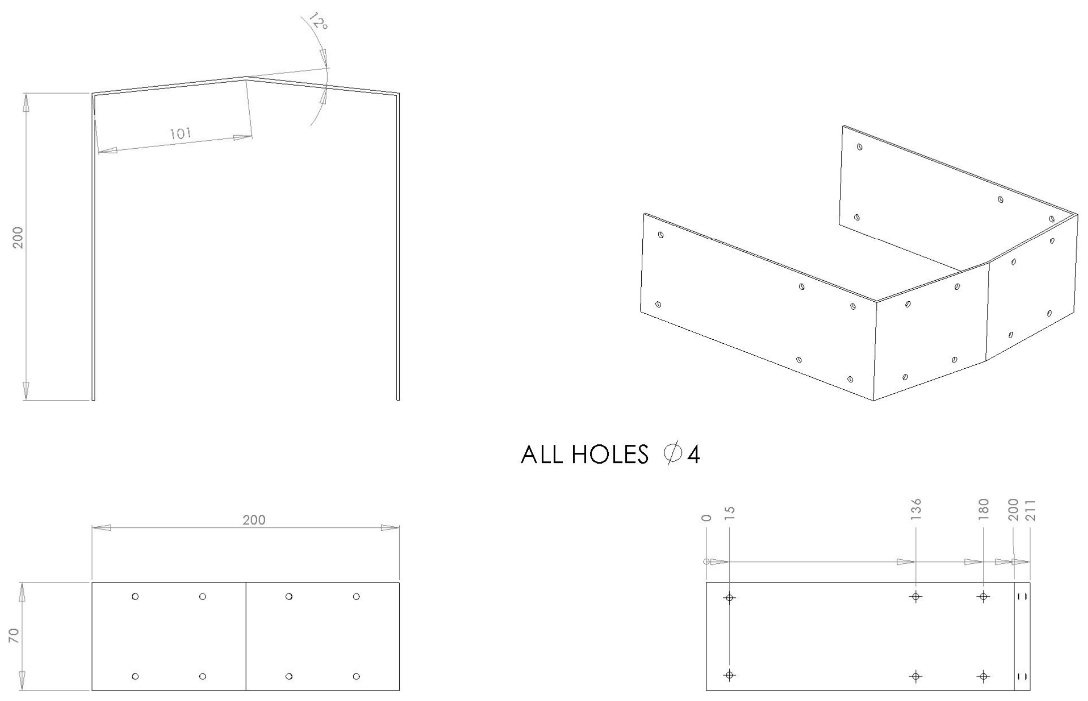
SINGLE SONAR MAX FOV EXPERIMENT SETUP (CHAPTER 4)

In order to verify the specifications of the sonar sensor, they sensor is set up with the PC104 as shown in the figure below.



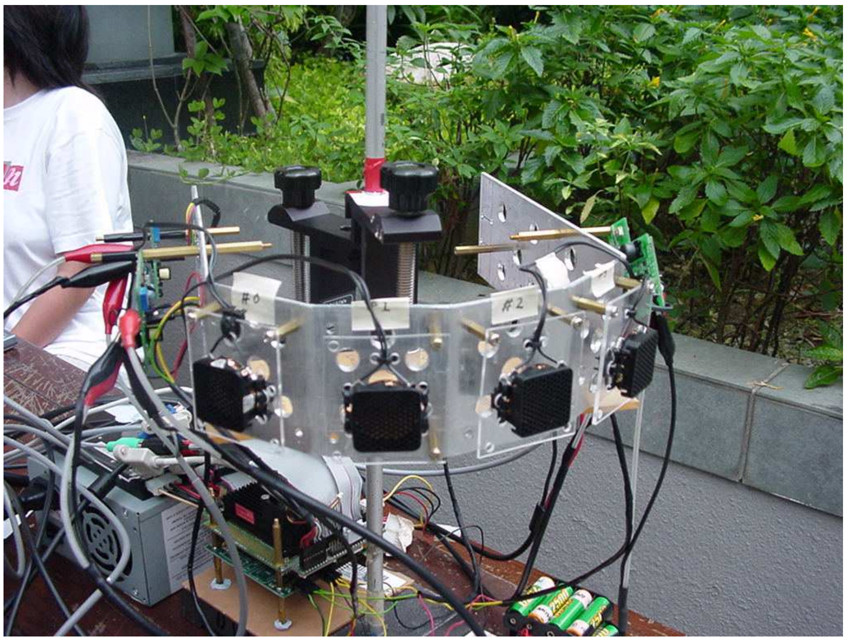
The sensor is then used to detect measured distances and then the results of the sonar readings are compared to the actual distances. This experiment also determines the Field of View of the sonar sensor at 2m intervals.

NEW SONAR SENSOR ARRAY DESIGN SCHEMATIC (CHAPTER 4)

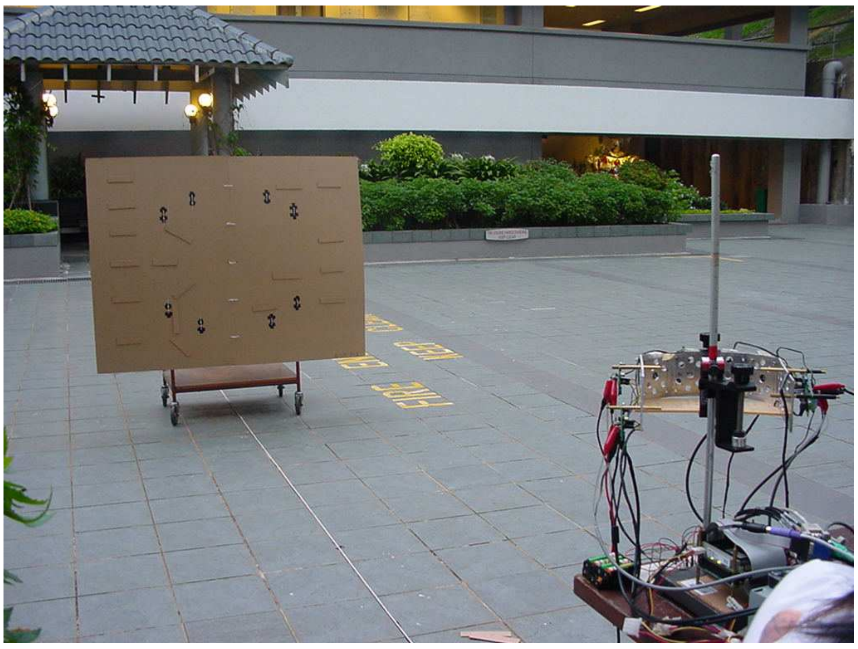


SONAR SENSOR ARRAY EXPERIMENT

After gaining an insight to the max range and field of view of the sonar sensors, 4 sonar sensors were mounted together in an array fashion and put to the test for max range and field of view. The setup is as shown below:



Similar to the single sonar setup, the sonar experiment was used to verify the max range and field of view of the entire array of sonar sensors. Most importantly, it is used to determine if more than 1 sonar sensors can be used at once. The figure below shows the setup.

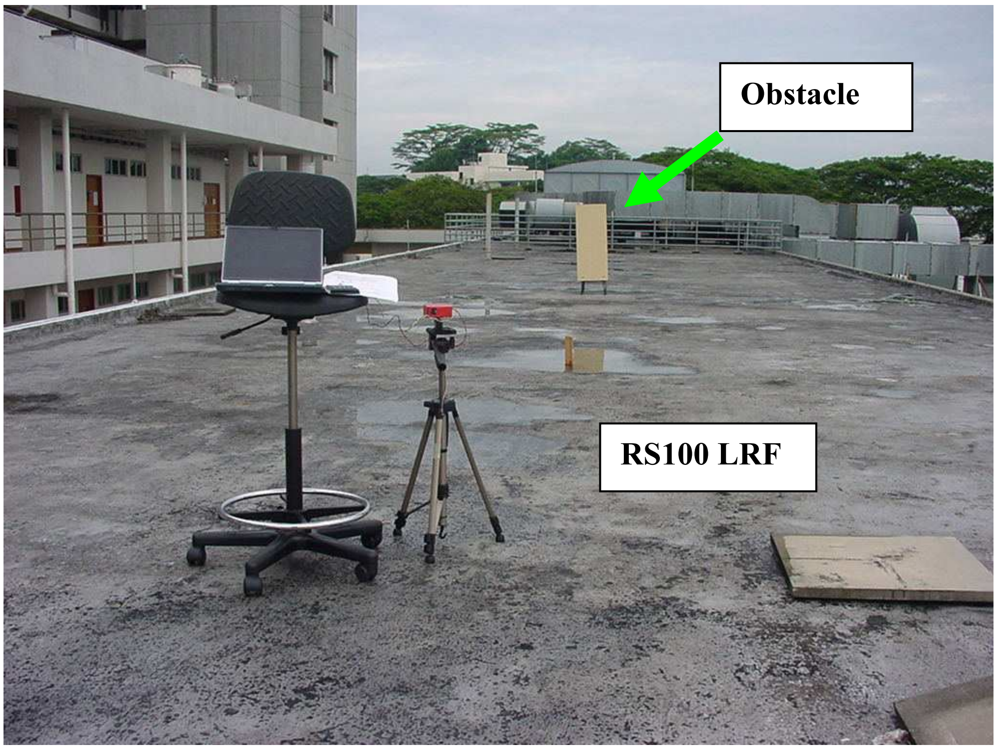


SONAR SENSOR ARRAY GROUND TEST WITH ENGINE RUNNING (CHAPTER 4)

The ground test is to determine if the sonar sensor array can work with the engine running at a low throttle. Hence, the entire flight platform mounted with the avionics box and the sonar sensor array is used to test. The figure below shows the test setup.

An obstacle is then put in front of the flight platform while the engine is off for the 1st stage testing. The second stage follows the exact same setup, but with the engine running at low throttle.

LASER RANGE FINDER CALIBRATION SETUP (CHAPTER 4)



SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 1 (CHAPTER 5)

FLIGHT 1

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 1 (CHAPTER 5)

FLIGHT 2

41
5

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 1 (CHAPTER 5)

FLIGHT 3

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 2 (CHAPTER 5)

FLIGHT 1

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 2 (CHAPTER 5)

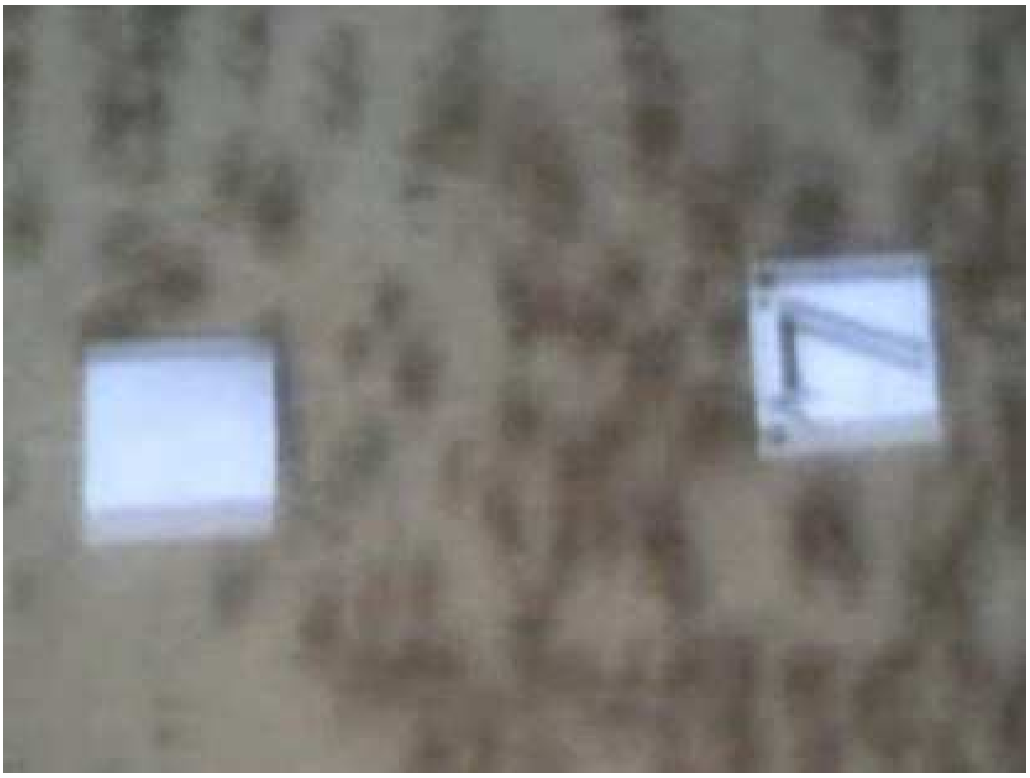
FLIGHT 2

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 2 (CHAPTER 5)

FLIGHT 3

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 3 (CHAPTER 5)

FLIGHT 1



SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 3 (CHAPTER 5)

FLIGHT 2

SCREEN CAPTURES FROM FLIGHT VIDEO FOR GAIN K = 3 (CHAPTER 5)

FLIGHT 3