iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: CAAA-2018 **Issue:** onference **Month of publication:** April 2018

DOI:

Particle Swarm Optimization Based Real-Time Path Planning and Obstacle Avoidance of UAVs Using Low Cost Sensors

Sumana Biswas[1], Sreenatha G. Anavatti[2], Hamish P. Krippner[3]
[1, 2, 3] School of Engineering and Information Technology University of New South Wales at the Australian Defence Force Academy Canberra, Australia

Abstract: This paper presents a real-time path planning combined with obstacle avoidance capabilities for Unmanned Aerial Vehicles navigation. This solution approach utilizes a closed loop design structure for autonomous functionality such as path tracking while avoiding obstacles. The mechanism applies a Particle Swarm Optimization based planning algorithm to find out an optimum path by avoiding static and dynamic obstacles. The methodology implemented for the open loop obstacle detection system integrates a combination of hardware and software elements which is able to detect obstacles, with a detection rate of 98% at five meters for various sized obstacles. Experimental and numerical results for the closed loop path planning combined with Collision Avoidance System demonstrate the UAVs ability to autonomously generate a safe path by detecting obstacles using a low -cost Ultrasonic Sonar sensor in real time. In addition, using the guided loop function, it can successfully avoid any obstacles. The results show the effectiveness of the total system in real-world applications.

Keywords: Real time path planning; obstacle avoidance; unmanned aerial vehicles; particle swarm optimization.

I. INTRODUCTION

Due to technological advancement, the development of Unmanned Aerial Vehicles (UAVs) is rapidly expanding and become more prominent in recent years. Autonomous industries are adopting the UAV technologies due to its cost efficiency for surveying, crop spraying, monitoring, 3D mapping and various other applications. Research conducted by Allianz Insurance Group has projected that there will be 4.7 million UAVs by 2020, three times the current number of UAVs in operation [1]. While the UAVs technology is growing faster in recent years, there are still some challenges in autonomous navigation of UAVs.

Autonomous navigation typically includes the path planning while avoiding obstacles. To improve the autonomy level of UAVs, one of the key challenges is to plan an optimum path by ensuring its safety [2]. Path planning of UAVs refers to plan an optimum path from its present position to the desired goal location in an environment with different obstacle distribution [3]. In high altitude, UAVs usually do not need to avoid obstacle frequently [4]. However, in low altitude environments, the issues related to risk-free planning should be addressed carefully. Therefore, to reduce the number of UAV related incidents, better autonomous functionality in the area of obstacle detection and avoidance capability is desirable.

Over the years a number of research works have been developed on path planning. Heuristic search algorithm such as A* and D* algorithms are widely used in the field of path planning [5]. Due to the need for continual incorporation of new information, these algorithms are ex-pensive in terms of computation time [6]. Dijkstra [7] and D* Lite [8] are other prominent path planning algorithms. Both algorithms are grid based and face some problems associated with time complexity and accuracy [9]. To overcome such types of problems, evolutionary algorithm such as Genetic Algorithm [10], Particle Swarm Optimization (PSO) [11] algorithm etc. are widely used in path planning research. PSO is a heuristic and random search algorithm inspired by the social behaviour of bird flocking or fish schooling [11]. It is simple with fewer control parameters and faster than the Genetic algorithm. In addition, the most significant part of PSO is that it usually gives a good solution for optimization problems [12]. Sumana et al. [13-14] proposed a PSO based algorithm for path planning in dynamic environments.

The ability to ensure safety during planning paths is a great challenge of UAVs [4]. Active research is currently being conducted in this area [15-18]. The development of sensor and vision technology [19-23] is continually assisting research in the area of obstacle detection and avoidance. Sensors like LIDER [24] are generally too large and heavy for the applications of UAVs [4]. With the increasing clutter condition in airspace a universal system needs to be developed so that the UAVs of the future are not susceptible of colliding with obstacles like other aircraft, infrastructure etc. and are therefore capable of evading collisions. Peng et al. [4] utilize a vision-based algorithm for obstacle avoidance. Jun and D’Andrea [25] used probability map to solve the problem of path planning, here the map construction depends on different radar characteristics. However, many of vision based and radar technologies are expensive as well as computationally heavy and require large processing capabilities in order to operate efficiently. This limits the functionality of UAVs, as weight, cost and size are factors that need to be considered in the design and development of UAVs. Therefore, low-cost solution with minimal power and processing consumption capacity is needed to increase the effectiveness of UAVs flight.

Existing path planning problem combined with collision avoidance system is either simulation based or offline mode [2], [26-27]. In most cases, UAVs follow a safe trajectory in offline mode [4], where it moves through provided waypoints within known environments. However, real world scenarios are very demanding and at every step there are uncertainties [13] where pre-existing maps are not very useful [26]. Since there may be different types of static and dynamic obstacles, depending on the circumstances UAVs should be able to replan a path by avoiding all types of obstacles. Hence, for real world performance, a combination of low cost as well as effective hardware that can accurately represent the surrounding in real time and an efficient software approach that can easily find an optimum path to reach the tar-get location by avoiding any collision is needed. Ricardo et al. [6] proposed a low-cost single point LIDER module based real time planning for UAVs. The limitation of their work is that the author only considers static environment. Therefore, real time planning of UAVs in dynamic environments needed to be addressed.

In this paper, a total combination of hardware systems with planning algorithm software is implemented for real time path planning and obstacle avoidance in dynamic environments. Here, a quadcopter with onboard Raspberry pi is embedded with a ground computing plat-form. An onboard low-cost Ultrasonic Sonar Sensor is utilized for obstacle detection. Once the obstacle is detected, it sends information to the ground computer, where the planning algorithm receives the obstacle related information for online processing. After that, the planning algorithm replan its path to avoid obstacles and sends the waypoints information to the quadcopter. Simultaneous Replanning Vectorized Particle Swarm Optimization (SRVPSO) algorithm [13] is used to plan an optimized path by avoiding obstacles. Finally, by following the waypoints the quadcopter is able to modify and update its path by avoiding any types of obstacles in real time outdoor environments.

The rest of the paper is organized as follows: Section II defines the research approach and describe the methodology of the total planning. Section III describes the path generation with obstacle avoidance. Simulation and experimental results are presented in Section IV. The paper is concluded in section V with some indications for the future work.

II. RESEARCH APPROACH AND METHODOLOGY

This section describes the essential hardware components, software and the mechanism of the research approach.

A. Ultrasonic Sonar (US) Sensor

An onboard ultrasonic sonar sensor is used in this research work to detect obstacles. Sonar sensor is a low-cost, light weight and less intensive power option for detecting obstacles. It becomes useful in low and high luminosity situations where vision is obstructed by bright light or the absence of light. US sensor comprises of two components: a transducer and detector. The transducer generates an ultrasonic frequency when an alternating current is applied to it. The detector produces a voltage if the ultrasonic frequency produced by the transducer has been reflected back to the detector. The voltage produced by the detector registers that an object/material has been detected. In order to calculate the distance of the object/material from the sensor Eq. (1) is used, where ‘d’ is the distance in meters, ‘tt’ is the travel time in seconds and ‘S’ is the speed of sound.

$$d = (tt / 2) \times S \qquad (1)$$

The advantages of using US sensor is that it uses minimal power to operate effectively and has an effective range of 2cm-4m with a measurement angle of 15 degrees.

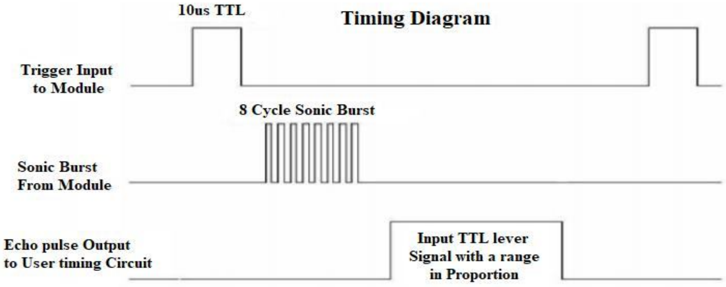


Figure 1. Timing Diagram for Ultrasonic Sonar Sensor

The disadvantages of using US sensors is the fact that porous materials will affect the readings of the sensor, as waves are absorbed. The US sensor works by sending eight pulses at 40 kHz and detects whether the pulse signal has been reflected to calculate the distance of an obstacle, the timing diagram is expressed in Fig. 1.

B. The Quadcopter and The Other Syatem Setup

Figure 2. UAV (quadcopter)

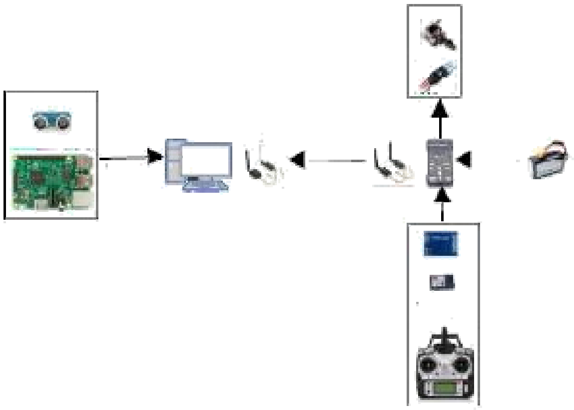


Figure3. System Setup

The quadcopter shown in Fig. 2 utilizes a single HC-SR04 US sensor for range measure-ment of obstacles in normal forward flight. The Raspberry Pi Model 2 which uses an A900MHz quad-core ARM Cortex-A7 CPU with 1GB Ram was used in conjunction with the US sensor to calculate the distance from an obstacle. The other components like RC con-troller, RC receiver, telemetry radio, encoder, remote control etc used in the system set up. The setup for the total system is shown in Fig. 3.

C. Software System

The hardware implemented for the collision avoidance system incorporates an open source software system. The Ground Control Software (GCS) utilized for this project is MAVProxy, which is installed on the Raspberry Pi Model 2 onboard the UAV. MAVProxy is a software package which uses the MAVLink protocol. This is supported with Pixhawk which is the onboard flight computer for the UAV system. The GCS allows for modification and creation of loadable modules, therefore the creation of a module integrating the CAS is possible. The CAS is an independent module which is loaded into the system and is run concurrently whilst the pilot controls the UAV through manual input using a remote control. The GCS can over-ride the pilots control once triggered by detecting an obstacle within the specified risk area from the US sensor, which is specified as less than one meter from the UAV.

D. Mechanism and Implementation

Information from the ground station to the UAV and vice-versa is delivered using a telemetry system operating at 915MHz, allowing for long range communication. The Radio Controller (RC) transmits commands to the UAV over a 2.4GHz communication channel which is processed by a receiver on the UAV. The channels from the receiver are passed through a PPM encoder which provides communication to the Pixhawk flight controller. The Pixhawk is able to process the commands from the RC and converts the received radio frequencies into PWM information which is processed by an electronic speed controller to determine the level of current to be passed through to the 380KV motors to generate lift to the UAV.

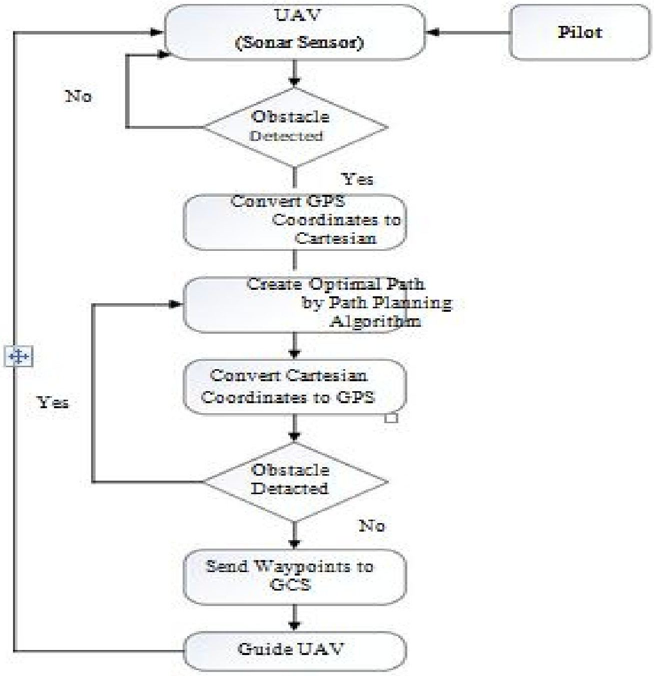


Figure 4: The flow diagram for obstacle avoidance and path planning

In the case of the GCS, guided information is transmitted using the MAVLink protocol through the telemetry system.

This information can then be interpreted by the flight controller and assign the correct way-points for the UAV to follow to avoid an obstacle. The obstacle detection system is used to interact with the quadcopter platform. The Raspberry Pi Model 2 does this by sending a trigger to the sensor, upon receiving an echo, the algorithm calculates the distance between the obstacle and UAV using Eq. (1). Distance units are converted into hexadecimal notation prior to transferring distance information using a UDP packet over a wireless local network. The Raspberry Pi Model 2 connects to the base computer through port 25000 of its IP Address. The computer receives the UDP packets sent from the Raspberry Pi Model 2 and runs an in-dependent algorithm which converts the UDP packet back to its original state, to be read correctly for distance information.

The distance information is processed by the CAS algorithm and triggers the optimized path once the distance between the obstacle and UAV falls below a threshold value of one meter. The algorithm receives the UAVs starting position in Global Positioning System (GPS) coordinates and converts it to the Cartesian coordinates. The algorithm starts optimizing its path based on the obstacles location and size, and plots a number of waypoints along the path using Cartesian coordinates updating the initial position of the UAV, it then converts the way-points to GPS coordinates for the GCS to process. An independent script receives the coordinate information over a local wireless network through port 25001 and writes the latitude and longitudinal information to a binary file using a protocol buffer to structure the data. The CAS module loaded on the GCS continually reads the binary file every 0.2 seconds to see if any information has been written to the file. Upon receiving a waypoint package in the binary file, the CAS module autonomously controls the UAV using a guided function to avoid collision with an obstacle by sending the UAV waypoint commands corresponding to the optimal path. If an object is detected along the optimal path the algorithm activates its reactive planning and creates a new path to avoid collision. Fig. 4 shows the flow diagram corresponding to the CAS algorithm for the clarification of its process. The UAV will have successfully completed its autonomous flight once the CAS algorithm has reached its desired end goal.

III. PATH GENERATION WITH OBSTACLE AVOIDANCE

Path planning algorithm is the key component in delivering a complete autonomous UAV system. Particle Swarm Optimization is one of the popular optimization tools that use the concept of swarm intelligence [13]. This optimization method combines both local and global search by balancing exploration and exploitation. PSO works with a
Population of particles.

Let, a PSO swarm consists of a set of particles (S). Each particle changes its position with time. Position and velocity of the ith particle at time step 't' are respectively-

$$xi(t) = (xi,1(t), xi,2(t), xi,3(t), \ldots\ldots xi,n(t)) \text{ and}$$

$$vi(t) = (vi,1(t), vi,2(t), vi,3(t), \ldots\ldots vi,n(t))$$

Each particle updates its velocity and position according to the following equations:

$$vi(t+1) = \omega vi(t)+c1r1(xi^{pbest}(t)-xi(t))+c2r2(xi^{gbest}(t)-xi(t)) \qquad (2)$$

$$xi(t+1) = xi(t) + vi(t+1) \qquad (3)$$

where, c1 = cognition parameter, c 2 = social parameter, ω = inertia weight factor, and r1, r2 are independent random numbers uniformly distributed in [0, 1]. x^{pbest} is the personal best position achieved by the particle and x^{gbest} is the globally best position achieved by the swarm.

Development in this area in conjunction with obstacle detection systems would create a robust and advanced collision avoidance system. In this research work, Simultaneous Replanning with Vectorized Particle Swarm Optimization (SRVPSO) algorithm by Sumana et al. [13] is used to plan an optimized path. Fig. 5. demonstrates the flow chart of the SRVPSO algorithm.

In this algorithm, the path planner is integrated with a collision avoidance system. To avoid collisions, SRVPSO algorithm works as a reactive planning.

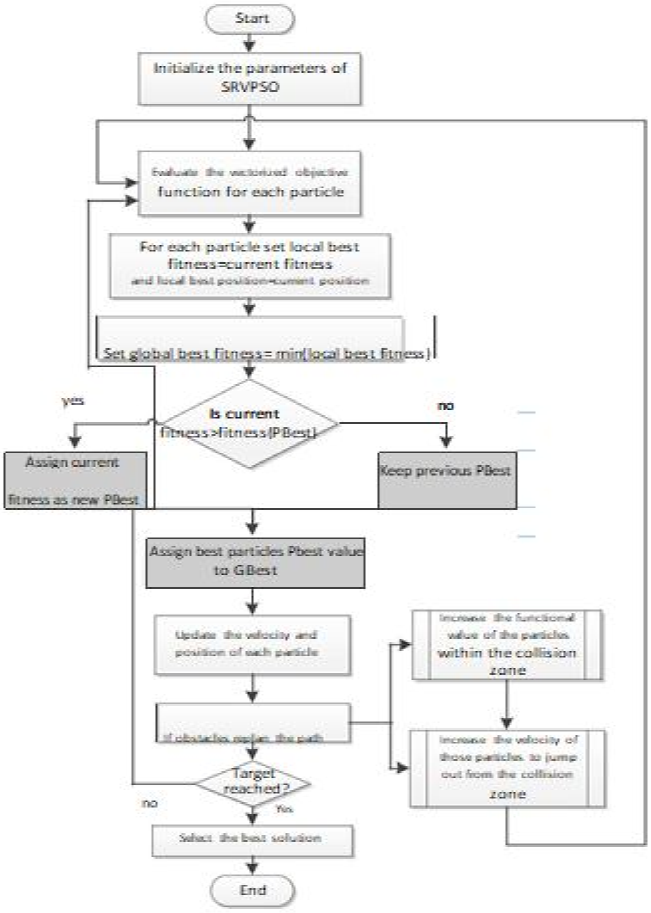


Figure 5. Flowchart of the SRVPSO algorithm

A two-layer strategy for obstacle avoidance is followed in this algorithm. The strategy is given below.
At first layer, each particle calculates the distance from its current location to the obstacles position.
Let, Dp^{obs} is the distance between the obstacles and the particles. Conditions of collisions are:

1) When $Dp^{obs} > 0$, no collision, move to next step.

2) When $Dp^{obs} \leq 0$, collision is likely to take place, replan the path.

Here, the negative cost of the distance means that the locations of the particles are inside the collision zone. If any risk of collision arises, the trapped particles in the collision zone are ignored and the path within the zone is considered as infeasible.
In the next layer, the trapped particles are recovered by increasing the particles velocity, to overcome the unbalanced scenario of the swarm population.
The parameters of PSO used in the tests have been chosen by running several tests with different combination. The preferred combinations of parameters are as follows:

$\omega = 1.0$, $c1 = 0.5$, $c2 = 2.0$, Population size of swarms = 25 and Maximum iterations = 50

Parameters considered for collision avoidance strategy are as follows:

$\omega = 1.0$ and $c1 = 3.0$, $c2 = 4.0$; these increased parameter values help the trapped particles to recover.

The algorithm is able to successfully convert the Cartesian coordinates into Decimal GPS co-ordinates and vice versa. The GPS coordinate system was based on the WGS84 earth centered, earth-fixed terrestrial reference system with an earth radius set to 6378137m. The bearing for forward momentum corresponded to a northern bearing at 360 degrees for the UAV as this is what depicted its flight path.

IV. SIMULATION AND EXPERIMENTAL RESULTS

Tests have been performed both in simulation and with a real quadcopter in outdoor environments. The CAS showed success in its application. 40 tests have been completed using numerous sized obstacles and different path goals. In all test runs, the UAV is able to success-fully detect and maneuver around all the static and dynamic obstacles to avoid collisions. Fig. 6 (a) and 6 (b) displays the result of the simulation. It is found that, the waypoints correctly depict the optimal path that is generated using the SRVPSO algorithm and the UAV is able to reach its goal location (40, 120). This technique was able to be manipulated through conversion to depict a real-world interface. It is found in Fig. 5 that the UAV is able to autonomously move around the obstacle based on the local coordinates provided by the SRVPSO algorithm. Out of the waypoints sent from the VSPO algorithm, 97.41% were received by the GCS. This meant that in almost all cases the UAV is able to avoid an obstacle without causing a collision. In the 2.59% cases where the packets are not received the data is corrupted by giving incorrect GPS co-ordinates.

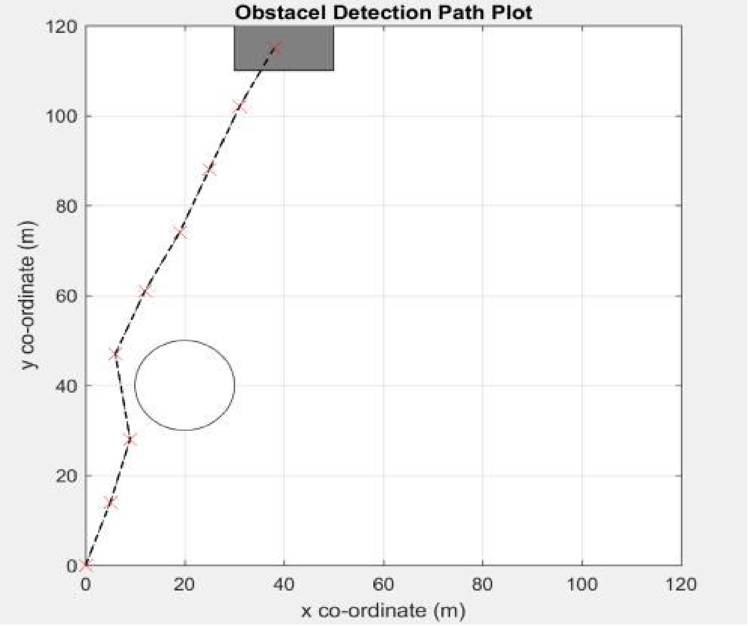
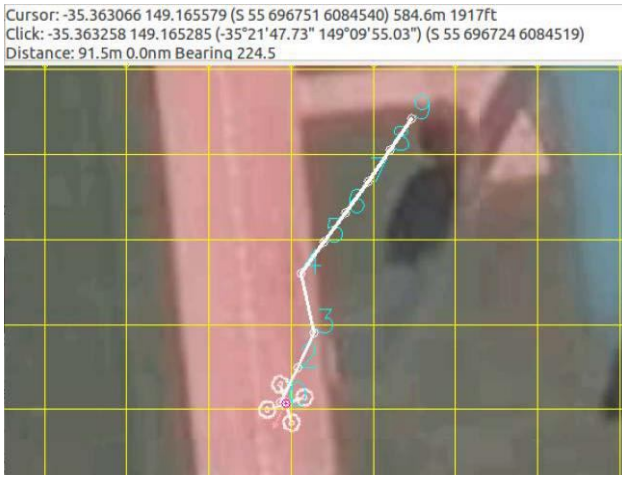


Figure: 6(a) Waypoint plots from Simuleted



Cursor: -35.363066 149.165579 (S 55 696751 6084540) 584.6m 1917ft
Click: -35.363258 149.165285 (-35°21'47.73" 149°09'55.03") (S 55 696724 6084519)
Distance: 91.5m 0.0nm Bearing 224.5

Figure 6 (b). real-world coordinates

For this reason, the UAV being unable to reach its goal destination. Fig. 6 (b) highlights the GPS waypoints to achieve the desired goal point of the UAV.

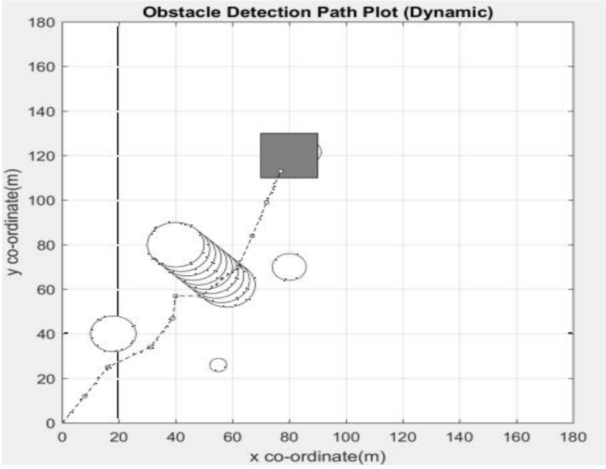


Figure 7. Obstacle Detection Plot (Pre-Determined)

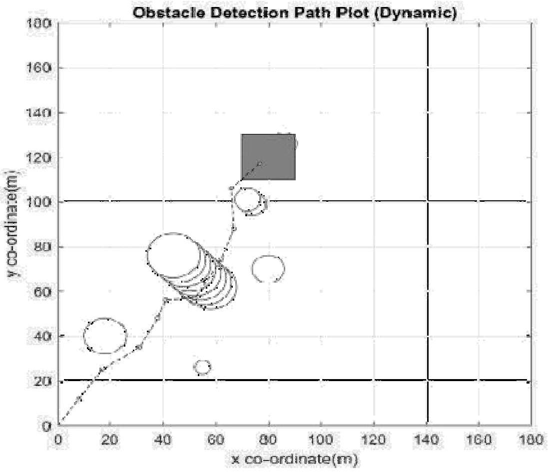


Figure 8. Path planning in dynamic environments

Fig. 7 and 8 shows the path planning by avoiding both static and dynamic obstacles. Fig. 7 shows the original path where the position and movement of the static and dynamic obstacles are predetermined. The obstacles are varied in size, an obstacle was placed at (18, 40) with a radius of 8 meters, another placed at (55, 26) with a radius of 3 meters and a dynamic obstacle placed at (60, 60) location with a radius of 10 meters that moved 2 meters in the negative X and positive Y coordinates per frame. In Fig. 8 it is found that the CAS is able to successfully identify the sudden presence of new obstacles and replan its path accordingly to avoid the potential risk of collisions.

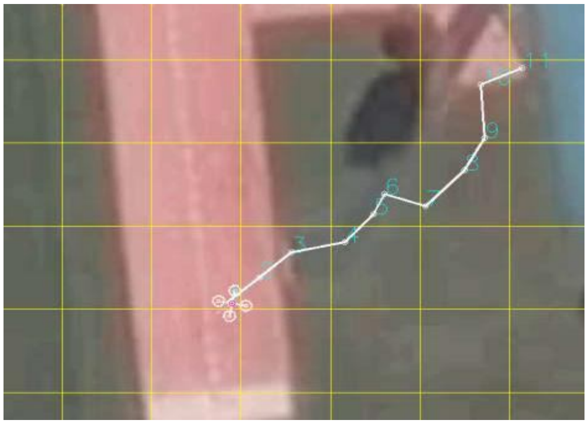


Figure 9. UAV Waypoints during dynamic obstacles avoidance

Figure 10. Flight test of UAV in outdoor environment

The new waypoints to evade the detected obstacle shown in Fig. 9 are transmitted to the GCS and the UAV mirrored this path as expected by avoiding collisions. Fig. 10 shows the photograph of the experimental flight test of the quadcopter during avoiding collisions at outdoor environments. The black circle in the figure indicates the position of the quadcopter.

From the above results, it is found that the total integrated system performs robustly to find an optimum path. The quadcopter successfully detects and avoid obstacles including new obstacles on its way and provides evasive maneuvers in an autonomous application to find the best path available to reach the desired goal location.

V. CONCLUSION AND FUTURE WORK

In this work, an autonomous navigation system with the real-time collision avoidance strategy of UAVs is developed. The simulation and experimental results with the real quadcopter validate the total approach. The total system is simple and can effectively detect and avoid obstacles in dynamic environments. This work shows that the use of a low-cost hardware in UAVs to avoid obstacles is feasible and can be used as a viable low-cost alternative to current market offerings. Despite the positive results and the successful application of the obstacle detection and CAS, there are still limitations in the application of the CAS. The CAS is limited in its ability to determine the UAV headings and as such needs to be manually changed. Future work could extend the ability of the CAS to factor in the UAVs bearing to provide better accuracy in its path optimization. In addition, in future this approach will be applied for a multi-agent path planning and obstacle avoidance in real-time applications.

VI. ACKNOWLEDGMENT

The authors of this paper are grateful to The University of New South Wales at the Australian Defence Force Academy, Canberra, Australia. The Research is supported by the Australian Government Research Training Program Scholarship.

REFERENCES

[1] Allianz, "Rise of the Drones: Managing the Unique Risks Associated with Unmanned Aircraft Systems", Allianz Global Corporate & Speciality SE., Joel Whitehead, Munich, 2016.

[2] Yao, Peng, Honglun Wang, and Zikang Su. "UAV feasible path planning based on disturbed fluid and trajectory propagation." Chinese Journal of Aeronautics vol. 28, no. 4, pp. 1163-1177, 2015.

[3]] Liu, Wei, Zheng Zheng, and Kaiyuan Cai. "Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval." Chinese Journal of Aeronautics vol. 26, no. 3, pp. 646-660, 2013

[4] Peng, Xin-Zhong, Huei-Yung Lin, and Jyun-Min Dai. "Path planning and obstacle avoidance for vision guided quadrotor UAV navigation." In Control and Automation (ICCA), 2016 12th IEEE International Conference on, pp. 984-989. IEEE, 2016.

[5] M. Nosrati, R. Karimi, H. A. Hasanvand, "Investigation of the *(star) search algorithms: characteristics, meth-ods and approaches," World Applied Programming, vol. 2, no. 4, pp. 251-256, 2012.

[6] Ricardo, Shawn, Doina Bein, and Anand Panagadan. "Low-cost, real-time obstacle avoidance for mobile ro-bots." In Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual, pp. 1-7. IEEE, 2017.

[7] E. A. Shaikh, A. Dhale, "AGV path planning and obstacle avoidance using Dijkstra's algorithm," International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 2, issue. 6.
pp. 77-83, June 2013.

[8] S. Koenig and M. Likhachev, "D* Lite", Menlo Park, California, American Association for Artificial Intelli-gence, 2002.

[9] Z. Zhang, Z. Zhao, "A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm," International Journal of Smart Home, vol. 8, no. 3, pp. 75-86, 2014.

[10] J.H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI, 1975.

[11] J. Kennedy, R. Eberhart, "Particle swarm optimization," Proceedings IEEE International Conference on neural Networks, pp. 1942-1948, 1995.

[12] R. Poli, "Analysis of the publications on the applications of particle swarm optimization," Journal of artificial Evolution and Applications, Article ID: 685175, 2008, DOI: 10.1155/2008/685175.

[13] S. Biswas, A. G. Sreenatha, M. A. Garratt, and M. Pratama. "Simultaneous replanning with vectorized particle swarm optimization algorithm," In Control, Automation, Robotics and Vision (ICARCV), 2016 14th Interna-tional Conference on, pp. 1-6. IEEE, 2016.

[14] S. Biswas, S. G. Anavatti, and M. A. Garratt, "Obstacle avoidance for multi-agent path planning based on vec-torized particle swarm optimization," In Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposi-um, IES 2016, Canberra, Australia, November 2016, Proceedings, pp. 61-74. Springer International Publishing, 2017.

[15] Z. Gosiewski, J. Ciesluk and L. Ambroziak, "Vision-based obstacle avoidance for unmanned aerial vehicles", 2011 4th International Congress on Image and Signal Processing, 2011.

[16] C. Chiu and C. Lo, "Vision-Only Automatic Flight Control for Small UAVs", IEEE Transactions on Vehicular Technology, vol. 60, no. 6,
pp. 2425-2437, 2011.

[17] J. Lee, K. Lee, S. Park, S. Im and J. Park, "Obstacle avoidance for small UAVs using monocular vision", Aircraft Engineering and Aerospace Technology, vol. 83, no. 6, pp. 397-406, 2011.

[18] N. Gageik, P. Benz and S. Montenegro, "Obstacle Detection and Collision Avoidance for a UAV With Comple-mentary Low-Cost Sensors", IEEE Access, vol. 3, pp. 599-609, 2015.

[19] "Ultrasonic Ranging Module HC - SR04", micropik, 2017. [Online]. Available: http://www.micropik.com/PDF/HCSR04.pdf. [Accessed: 01- May- 2017].

[20] "IR (Infrared) Obstacle Detection Sensor Circuit", Electronics Hub, 2017. [Online]. Available: http://www.electronicshub.org/ir-sensor/#Principle_of_Working. [Accessed: 19- May- 2017].

[21] "GP2Y0A710K0F", Sharp, 2017. [Online]. Available:http://docs-asia.electrocomponents.com/webdocs/0d1b/0900766b80d1bdcd.pdf [Accessed: 08-May-2017].

[22] "Kinect for Windows Sensor Components and Specifications",
Msdn.microsoft.com, 2017. [Online]. Available:
https://msdn.microsoft.com/en-us/library/jj131033.aspx. [Accessed:
10- May- 2017].

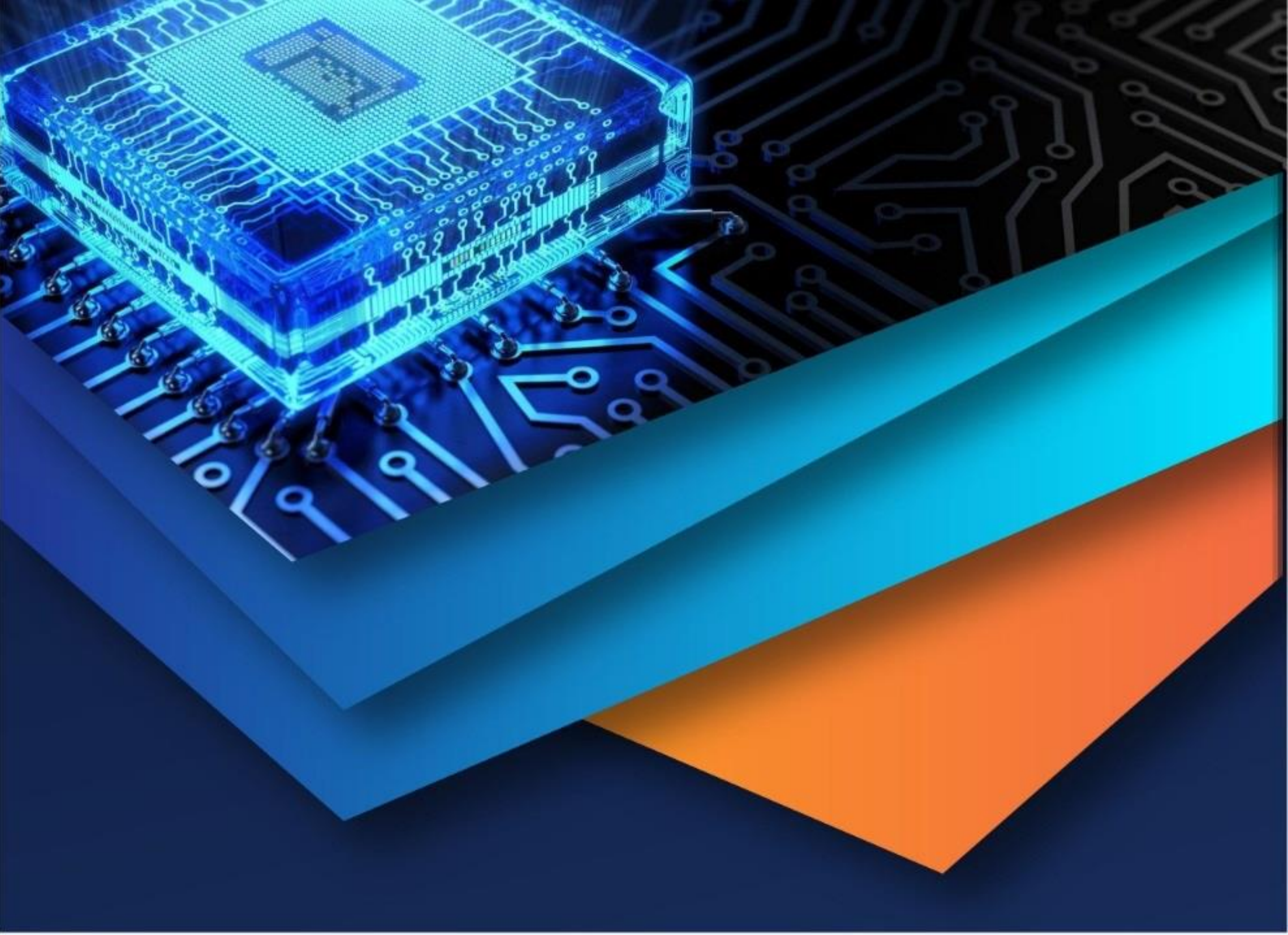
[23] ZED - Depth Sensing and Camera Tracking", Stereolabs.com, 2017.
[Online].Available:https://www.stereolabs.com/zed/specs/.
[Accessed: 11- May- 2017].

[24] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: Methodology and experiments," Int. J. Rob. Res., vol. 27, no. 5, pp. 549–574, May 2008. [Online]. Available: http://dx.doi.org/10.1177/0278364908090949.

[25] Jun, Myungsoo, and Raffaello D'Andrea. "Path planning for unmanned aerial vehicles in uncertain and adver-sarial environments." In Cooperative control: models, applications and algorithms, pp. 95-110. Springer, Bos-ton, MA, 2003.

[26] Nguyen, Phuong DH, Carmine T. Recchiuto, and Antonio Sgorbissa. "Real-time path generation and obstacle avoidance for multirotors: A novel approach." Journal of Intelligent & Robotic Systems, vol. 89, no. 1-2 pp. 27-49, 2018.

[27] Gupta, Durgesh Kumar, and Anant Kumar Jaiswal. "Path Planning with Real Time Obstacle Avoidance." Inter-national Journal of Computer Applications, vol. 70, no. 5, 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY