Design and Development Autonomous Unmanned Aerial Vehicle Software

S.Anirudh, Avinash M.G., Suriya Prakash S., G. Y. Rajaa Vikhram

Abstract: Unmanned aerial vehicles are the cutting edge technology which is used in various arduous applications and emergency scenarios. But human operators find it burdensome and experience a lot of physical and mental stress while operating the aerial systems in critical and emergency scenarios such as rescue operations, mine inspection, and surveillance. Our proposed idea is to provide the autonomous capability and features to these automatons by developing a mission-planning application that can autonomously guide UAV operations even in GPS denied environments by implementing SLAM (Simultaneous Localization and Mapping). With autonomous capability, aerial systems can help to plummet the stress on human operators or may even perform the process or mission efficiently without human intervention in numerous applications. Applications involving autonomous unmanned aerial systems have increased in recent times and are being applied in a wide range of fields such as infrastructure, transport, agriculture, mining, media, and transport. This paper covers the working of the autonomous navigation algorithm, artificially intelligent object detection algorithm and the mission planning API (Application Programming Interface).

Keywords: API Application Programming Interface, GNSS Global Navigation Satellite System, GPS Global Positioning System, ROS Robot Operating System, SLAM Simultaneous Localization and Mapping, UAV Unmanned Aerial Vehicle.

I. INTRODUCTION

The use and the need for autonomous unmanned aerial systems have been increased in recent times and has been applied in various fields. Human operators find it arduous to operate these unmanned aerial systems in critical and emergency scenarios.

S.Anirudh*, Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India. Email: anirudh_sundararajan@srmuniv.edu.in

Avinash M.G., Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India. Email: avinashmg@technologist.com

Suriya Prakash S, Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India. Email: suriyaprakashsiva.eie@gmail.com

Dr. G. Y. Rajaa Vikhram, Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India. Email: rajaaviy@srmist.edu.in

These human operators are subjected to immense physical and mental stress and are likely to commit errors while operating the unmanned aerial systems. With an Intelligent autonomous aerial system, the human interference and errors in the flight control process can be drastically reduced. Our proposed idea is to develop a mission-planning application that can autonomously guide UAV operations even in GPS denied environments [7]. And to design an autonomous navigation algorithm using SLAM and computer vision to guide a drone from the start to destination [2], to design an artificially intelligent obstacle detecting algorithm to detect obstacles in the mission path and avoid them by planning alternate paths when encountering them.

The ROS framework is used to design and develop an unmanned aerial system [6]. The PX4 flight stack firmware is used to operate and control the unmanned aerial system operations. Gazebo simulator is used to simulate the autonomous unmanned aerial system operations in the 3D rendered complex GNSS denied environments [6]. The ROS framework and PX4 firmware are then integrated and the Gazebo 3D simulator is used to simulate the working of autonomous unmanned aerial systems in GNSS denied complex environments. Further features will be added to the API such as visualization of flight data and path followed by the autonomous unmanned aerial system.

Intelligent aerial systems powered by computer vision are capable of executing autonomous missions and path planning with precision. The availability of open-source frameworks like ROS enables developers to program and control robotic systems.
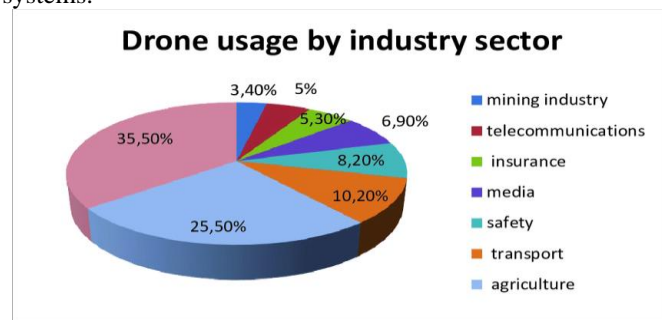

Fig 1.1. Contribution of UAV in Different Sectors

The pie chart in Fig 1.1 depicts the usage and application in various sectors. The UAVs are used in the infrastructure sector to inspect and monitor buildings' structural quality. These UAVs are also used in the transportation sector to achieve the last-mile delivery. Small and commercial UAVs also find numerous applications in the Media and entertainment industry.

UAVs are also extensively used in mining operations such as mine inspection and ore or sample collection. And finally, the UAVs are also used in India's largest sector i.e., agriculture for soil and field analysis, irrigation monitoring and crop monitoring. With autonomous navigation, object detection, and environment mapping capability, the UAV can navigate in GNSS denied environments and detect the obstacles and maneuver over them efficiently. With autonomous capability, aerial systems can perform the process or function efficiently without human intervention in the above mentioned and other numerous applications.

II. COMPONENTS

A. Object Recognition

Object detection is the task of finding what's present in an image and where it is present for a variety of object classes. In recent years, deep learning methods for object detection and recognition have become more popular due to their state-of-the-art accuracy. Faster RCNN, Mask R-CNN, and many other models top the mean Average Precision (mAP) charts [1],[9]. But these models are memory and computationally expensive to run on edge devices.

There has also been a rise in computationally lightweight and compact neural network architectures. This has accelerated the use of deep learning models on limited-compute edge devices such as mobile phones, single board computers, etc. This expands the horizon for applying deep learning to a variety of problem spaces like computer vision powered small robots and drones. One such efficient architecture that we employ is a Single Shot Detector (SSD) [11]. The SSD model trained on the COCO dataset can detect 90 different classes like people, vehicles to name a few [3]. Deploying this model on an autonomous UAV adds intelligence to it. For example, as the UAV navigates through a GPS denied building during a disaster, it can scan the environment for people stuck inside and can report to the rescue personnel. The object detection model is implemented in Tensorflow and deployed on a ROS node [4]. This node subscribes to the camera feed ROS topic and uses the frames obtained as inputs to the object detection model. After inference, the result is relayed back to the interface along with the feed.

B. Navigational Algorithm

For a drone to navigate an unknown environment autonomously, it requires a path planning algorithm that computes the shortest collision-free path to the destination. The A* algorithm is used for traversing graphs to find the shortest path between a starting and a destination node. We chose to use this over other algorithms such as Djikstra as it is faster.

C. Px4 Flight Stack

PX4 is open-source architecture for both manual and autonomous flight control systems that expose drivers for additional sensors and peripherals [8]. It also provides options to simulate UAVs in 3D environmental models. The flight stack consists of various low-level control loops and also supports high-level control tasks such as altitude hold, position hold, takeoff to a specific altitude and more. The flight stack uses a messaging bus called uORB to communicate between different components of PX4.

D. Mavlink

MAVLink is a communication protocol that can be used to communicate between the flight control board and a ROS flight control node. It also supports communication between multiple flight controllers and control nodes. For our work, we use MAVLink for communication between the simulated UAV with PX4 flight stack in the Gazebo environment and the ROS node that controls the UAV. We use a package called MAVROS which converts the MAVLink messages sent by the PX4 to ROS messages and ROS messages sent by the control node to MAVLink type.

E. Stereo Camera

A typical stereo camera consists of two or more image sensors which allows the device to simulate the binocular vision of humans to perceive the sense of depth. The depth is determined by calculating the stereo disparity and correlation between the two 2D images taken from different positions. Then the triangulation method is used to find the distance of the object from the camera. In our work, we simulate a UAV fitted with a stereo camera in the Gazebo environment [2], [6].

F. Graphical User Interface

A graphical user interface is made with languages JavaScript, HTML, CSS. The HTML is used to build the basic structure of the user interface with input elements and buttons. CSS is used for styling the elements and JavaScript is used as a scripting language that lets the user interface communicate with the UAV.
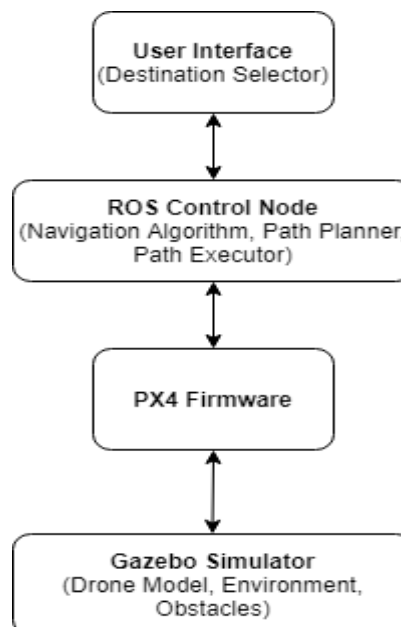
III. WORKING

Fig.3.1. Architectural Overview

The components described above work in sync to guide the UAV in an unknown environment. The ROS control node runs the navigation A* algorithm and it sends control signals to the UAV in the Gazebo simulator using MAVROS.

The ROS nodes are implemented in Python programming language owing to its simplicity [10]. But there's also an option to implement ROS nodes in C++. First, a shell script launches the Gazebo simulator with a PX4 powered UAV in it. The simulated world contains obstacles such as walls, household objects to form a complex environment [6]. Then, the ROS control node and the object recognition node are launched by the shell script. A program to communicate to remote computers using web protocols to get the user input is also launched.

Finally, a user interface is launched to view the mission progress. The web based user interface can be opened from a remote computer that is connected in the network. The user can update the mission parameters by setting a target location which the UAV has to reach by navigating through the obstacles. The user interface prompts for the relative position of the target location with respect to the initial position of the UAV. Target is specified by entering the distance towards the target as distance in north direction and distance in east direction. South and west direction can be given in negative values of north and east respectively.

Once the user starts the mission by setting a target location, 100 m to the North of the UAV, for example, the control node will start sending control messages to the PX4 flight stack on the UAV. As we are demonstrating in a simulated GPS/GNSS denied environment, the UAV relies solely on the stereo camera for navigation and obstacle detection. The generated map of the UAV's environment is used to detect obstacles on its course to the user given target location. This data of where the obstacles are is used by the control node to plan a path around it if one exists.

Fig. 3.1. GUI Input for target location

Fig. 3.2. GUI for displaying UAV mission status

IV.   APPLICATIONS

The automated UAV is applicable in wide variety of cases such as:

A. Transportation Of Goods.

In cases where the objective is to transport goods from location A to location B, where there are unknown number of obstacles and unavailability of GPS signal, this automated UAV can be used to fulfil the task.

B. Search And Rescue Operations.

In emergency situations such as earthquakes, and floods, the search and rescue service could be done using these automated UAVs since they can move through spaces that are having unknown obstacles and problems with GPS/GNSS signals.

V.   RESULTS AND DISCUSSION

We simulated the UAV in a GPS denied environment in the Gazebo environment. The target location was set 10 m in front of the drone and the walls in front of the drone obstruct the direct flying path. Once the mission is initiated the UAV takes off to an altitude of 1 m. Once it reaches that point, the control node starts running the A* navigation algorithm to check for available paths to the destination. If the algorithm finds a collision-free path, the drone follows the path and reaches the destination. If it doesn't, the UAV scans the environment by yawing to the left and right or by increasing the altitude to find any available shortest path to the destination. In the mission shown in Fig 5.2 (b), the UAV finds that there's the shortest path available to the destination if it raises its altitude. Rather than going through the open area on the wall on its left, it chooses to fly over it since that path is shorter compared to the other one.
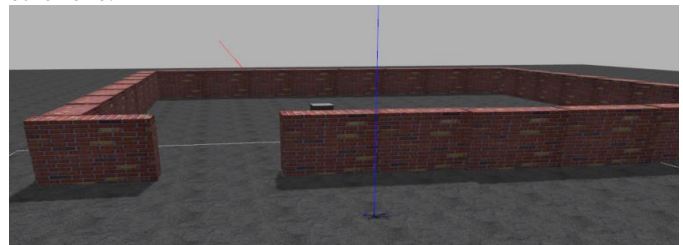

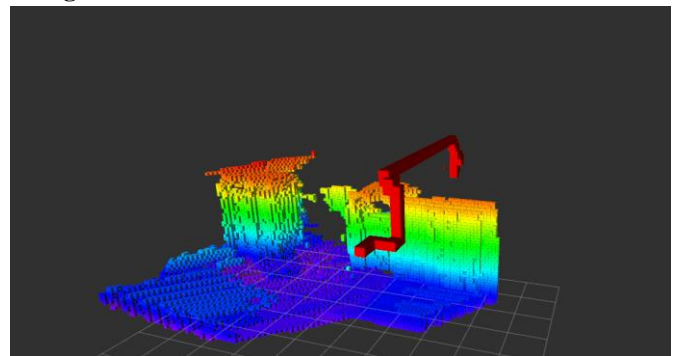**Fig. 5.1 3D model world with a UAV and an obstacle**

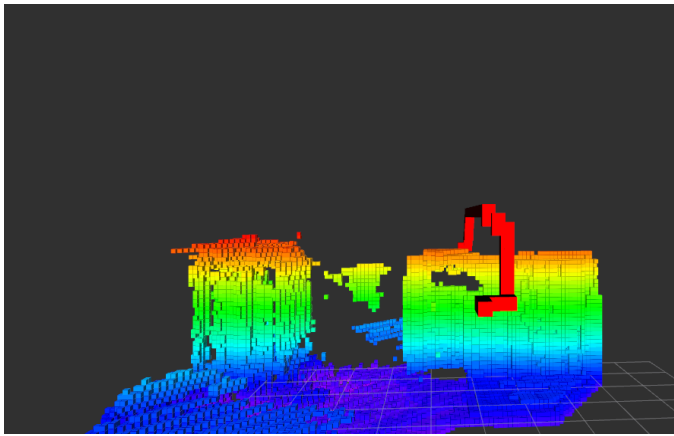**Fig. 5.2. (a) The path planned by the A* algorithm to the destination**

Fig. 5.2. (b) The path planned by the A* algorithm to the destination (Front View)

VI. CONCLUSION

The automated UAV with path planning and mapping can be useful in improving the existing applications of UAV. As we have demonstrated in this paper, the autonomy of the drone proves to be effective in planning in an obstacle filled environment. With improved sensors, flight systems and more sophisticated algorithms, there's room to improve our work in the future. This system can be improved in future by adding additional sensors for a much more robust operation. The path planning software can be distributed among multiple UAV via networks and much more complex tasks can be accomplished by UAVs. This will open up more field application like coordinated military ops, civilian surveillance, large area search and rescue.

REFERENCES

1. K. He, G. Gkioxari, P. Doll´ar, and R. B. Girshick.Mask R-CNN. CoRR, abs/1703.06870, 2017.
2. R. Mur-Artal and J. D. Tard´os. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics, 33(5):1255–1262, 2017.
3. T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick. Microsoft COCO: common objects in context. CoRR, abs/1405.0312, 2014.
4. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane,´ R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, ´ M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
5. F. Chollet. Keras. https://github.com/fchollet/keras, 2015
6. Meyer J., Sendobry A., Kohlbrecher S., Klingauf U., von Stryk O. (2012) Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo. In: Noda I., Ando N., Brugali D., Kuffner J.J. (eds) Simulation, Modeling, and Programming for Autonomous Robots. SIMPAR 2012. Lecture Notes in Computer Science, vol 7628. Springer, Berlin, Heidelberg
7. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments. In: Robotics: Science and Systems Conference (2008)
8. L. Meier, D. Honegger and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, 2015, pp. 6235-6240.
9. R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1440-1448.
10. G. van Rossum, Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.
11. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.- Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016

AUTHORS PROFILE

Anirudh S. became a member of IET in 2016. Anirudh was born in Chennai, Tamilnadu, India. He is currently pursuing Bachelor's Degree B. Tech in the stream of Electronics and Instrumentation Engineering from SRM Institute of Science and Technology, Chengalpattu, India. He has also participated in many industry level hackathons such as TATA hackathon and AXIS AI Hackathon. He is also an honored member of SRM SCRO copter team.

Avinash M.G. became a member of IET in 2016. Avinash was born in Kerala,India. He is currently pursuing Bachelor's Degree B. Tech in the stream of Electronics and Instrumentation Engineering from SRM Institute of Science and Technology, Chengalpattu, India. He has been part of SRM's technical team like NextTech Lab and SCRO Copter Team. He has also participated in many national level hackathons such as NEC AI hackathon and NEC Innovation hackathon.

Suriya Prakash S became a member of IET in 2016. Suriya was born in Puducherry, India. He is currently pursuing Bachelor's Degree B. Tech in the stream of Electronics and Instrumentation Engineering from SRM Institute of Science and Technology, Chengalpattu, India. He has been part of SRM's technical team NextTech Lab. He has also participated in national level hackathon such as Smart India Hackathon and TechGig Code Gladiators Hackathon.

Dr. G. Y. Rajaa Vikhram is working as Assistant Professor at Department of Electronics and Instrumentation Engineering, SRM IST. He had completed his Bachelor's degree in Electrical and Electronics Engineering from K.L.N. College of Engineering in 2009 and Master's degree in Control and Instrumentation from Thiagarajar College of Engineering, Madurai in 2011. He also got his doctorate in Electrical and Electronics Engineering from Thiagarajar College of Engineering, Madurai in 2017. He has published various papers on the topic of Power System Control. He has received the Best Outgoing Student award for M.Tech (Control & Instrumentation) in Thiagarajar College of Engineering, Madurai for the year 2011.