COLLISION AVOIDANCE. SYSTEM FOR DRONES

EXPLORATORY PROJECT Report

2022

JANUARY-MAY

IIT BHU

PRESENTED TO
JP MISRA

COLLISION AVOIDANCE SYSTEM FOR DRONES

EXPLORATORY PROJECT REPORT

by

KARTIK KANKURTE - 20135058

HIMANSHU KUMAR - 20135048

AYUSH LALL - 20135024

KOUSTUBH MANE - 20135063

ANKIT BARWAR - 20135019

Department of Mechanical Engineering

Indian Institute of Technology (BHU) Varanasi

April 2022

Table of Content

1.ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We owe our deep gratitude to our project guide Dr. JP Misra, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system also for their encouragement and more over for their timely support and guidance till the completion of our project work.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Mechanical Engineering which helped us in successfully completing our project work

KARTIK KANKURTE	-	20135058
HIMANSHU KUMAR	-	20135048
AYUSH LALL	-	20135024
KOUSTUBH MANE	-	20135063
ANKIT BARWAR	-	20135019

2.Abstract

In this study we researched and implemented best approach presently available for collision avoidance system. In real world drones are still facing many challenges, one of which is collision avoidance in dynamic and cluttered environment. Today drones are present everywhere at the war front, in delivering products, photography, search and rescue, detecting a fault in high power lines, and many more, increasing at a rapid pace. The emerging global market for business services using drones is valued at over $127B, according to PricewaterhouseCoopers.

Due to rapid advancements in drone technologies, we can see that the number of drones present is skyrocketing. However, most of the drones are controlled by humans. So, in the real world's dynamic environment, there is always a possibility of collision with trees, birds, other drones, and worst, with another human. Therefore, for a collision free motion to the goal, the global path planning (SLAM) is to be associated with a local obstacle handling that involves obstacle detection and obstacle avoidance. Our simulated quadrotor platform is equipped with a forward-looking three-dimensional (3D) light detection and ranging (lidar) sensor to perceive the environment. Specifically, the computer estimates. the current pose of the UAV, maintains a local map (time accumulated point clouds), and computes a safe trajectory using BUG algorithm to the goal point.

KEYWORD: - Bug Algorithm; QUAV; LIDAR; Point clouds; Obstacle avoidance; ROS.

3. Introduction

In the last years, the use of UAVs (unmanned aerial vehicles), also known as drones, has increased exponentially. Their reduced cost and versatility are the main reasons for the increase in their use. Unmanned aerial vehicles (UAVs) hold good promises for performance in both civilian and military operations. A broad range of meaningful tasks such as aerial surveillance, simultaneous arrival and attack, search and rescue, flight formation, etc. can be accomplished using the application of QUAVs. However, for UAVs to perform successfully in their respective missions and operations, collision avoidance system is necessary to be imparted into them. Without collision avoidance system, an UAV during operation would collide into unexpected objects such as wall, flying objects, flying animals, humans and others, resulting in mission failure, destruction of UAV, damage or injury to the collided objects or a combination of all. In most cases, UAVs are used for outdoor applications using remote sensing payloads such as RGB cameras or LiDAR sensors, but also for indoor applications. Collision avoidance aims at finding a proper safety route, which contains two parts: collision avoidance model and collision avoidance planning. The focus of collision avoidance model is to build the appropriate architecture for gaining collision avoidance principle; Collision avoidance planning is to further optimize the strategy selection based on the analysis results of previous collision avoidance model, getting the safety, real-time, smoothness and seaworthiness route, through a variety of specified constraints and algorithm design.

The Related Work section describes related works in the domain. The Materials and Methods section describes the datasets, network architectures, and experimental analysis methods used in this work. The Discussion section relates the experimental findings to previous work and proposed future work.

4. Related Work

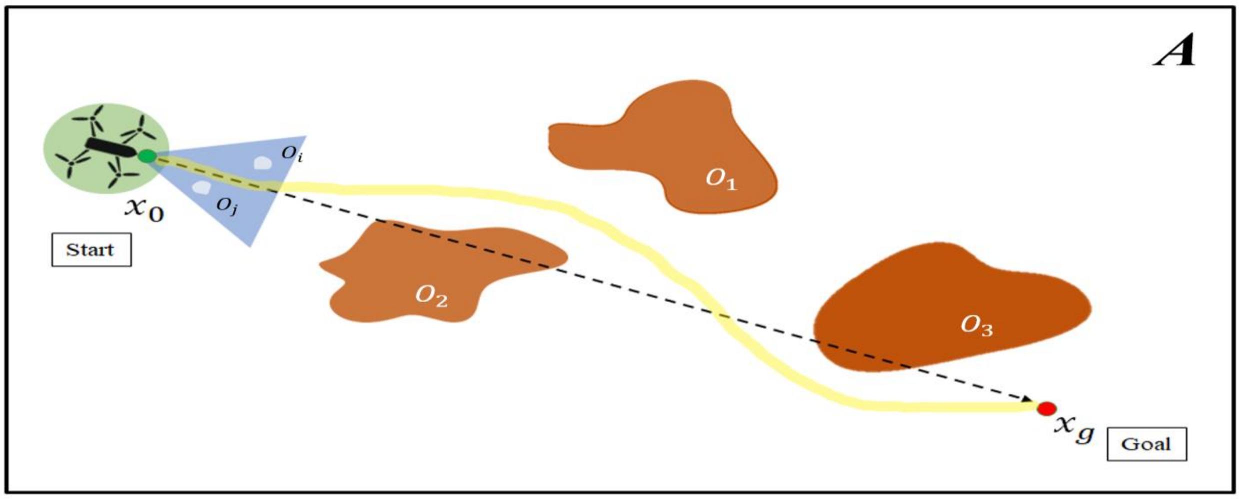
Today, commercial QUAVs are mostly teleoperated, use GPS for navigation, and have limited ability to automatically avoid obstacles. Nevertheless, the research community has made significant strides toward introducing more visual based processing to facilitate obstacle avoidance and state estimation. Alvarez et al. [1] use a structure-from-motion(SfM) algorithm to estimate depth, and hence avoid obstacles, from a single forward-facing camera. A method to compute structure from motion that doesn't need an accurate calculation of optical flow of each feature point is introduced to precisely compute an individual optical flow value for each feature point is not easy to achieve this is owing to the image noise effect and the aperture problem. Alternatively, they proposed the idea of optical flow probability distribution. The proposed method can be appropriate in case of points with sharp edges. Structure from motion is calculated using two different approaches, linear structure from motion, to overcome nonlinear optimization problem and optimal structure from motion, which is designed to handle the existence of noise. To sum up all, the covariance estimate (how accurate the optical flow detection is in any direction) should be a necessary step for any optical flow technique. Bachrach et al. [2] build a map using an RGBD camera, which is then used for localization and planning. Many other methods depending on monocular cue can be found in literature for instance, texture gradient: the closer the object the more texture information we can notice on its surface, occlusion: objects which partially or fully block the appearance of other objects are sensed as closer in depth, haze: because of the atmospheric light scattering the farther the object the hazier it look so distal objects can be perceived as blurred, shadow also plays an important role to determine the shape of the object and its size in space. These approaches require good features for defined objects.Stereovision is to extract 3D information from digital images by means of triangulation and use this information to estimate position of objects in two images. Stereovision requires sufficient base line separation between the lenses as a small error in triangulation results in a large error in distance calculation Bry et al. [5] combine an inertial measurement unit (IMU) with a laser range finder to localize a fixed-wing MAV, enabling reliable flight in indoor environments. Faessler et al. [9] apply the SVO algorithm [10] to a downward-facing camera and IMU to build a map of the environment, which is then used for planning. Fraundorfer et al. [14] fuse a forward-facing stereopair of cameras with a downward-facing camera for map building and state estimation. Scaramuzza et al. [4] also combine IMU and three cameras for obstacle avoidance and map building.

Some of the challenges of flying low among obstacles are described by Scherer et al. [6]. Recent attention has also been paid toward using deep reinforcement learning (DRL) for MAVs. In foundational work, Bhatti et al. [3] describe a system that combines visual. Many obstacle avoidance algorithms have been proposed according to the geometrical relationships between aircraft and collisions. Sasongko et al.[7] proposed a method which calculated a set of avoidance waypoints to avoid collisions based on an obstacle model and an aircraft's speed vector. Zheng et al.[13] established fuzzy rules to obtain avoidance maneuver options according to the aircraft's forward speed and the distance between the vehicle and obstacles. Al-Kaff et al.[12] proposed an avoidance method through tracking along the safe flight boundary consisting of a set of obstacle feature points, which were obtained by determining the coordinate relationship between the aircraft and obstacles in real time. Overall, these obstacle avoidance methods based on geometrical relationships are easy to implement and test, but they are difficult to apply in actual complex flight environments.

The two leading approaches are model-based and learning-based system design. The model-based approach follows a classical sense-plan-control scheme, which is modular and requires very accurate knowledge about the drone dynamics, the drone's state, and the ability to perform low-latency minimum-time control onboard. Indeed, this approach has been very successful and has been able to outperform experienced drone racing pilots on challenging race maneuvers in highly controlled environments [8]. However, model-based approaches often require external sensing and highly accurate systems knowledge, pre-planned trajectories, and do not generalize to unknown environments or noisy sensory inputs. The alternative is infusing learning-based methods into systems design, where sensing, planning, and control tasks are performed by a single neural network. These so-called end-to-end neural networks have been successfully trained and deployed for quadrotor flights of acrobatic maneuvers [11], obstacle avoidance in the wild , and simulator-based drone racing .

This work is aimed at solving the problem of real-time autonomous obstacle avoidance for UAVs in three-dimensional space, and an innovative obstacle avoidance algorithm based on BUG algorithm is proposed according to the geometric relationship between UAVs and obstacles through obstacle modeling.

Many other methods depending on monocular cue can be found in literature for instance, texture gradient: the closer the object the more texture information we can notice on its surface, occlusion: objects which partially or fully block the appearance of other objects are sensed as closer in depth, haze: because of the atmospheric light scattering the farther the object the hazier it look so distal objects can be perceived as blurred, shadow also plays an important role to determine the shape of the object and its size in space. These approaches require good features for defined objects.



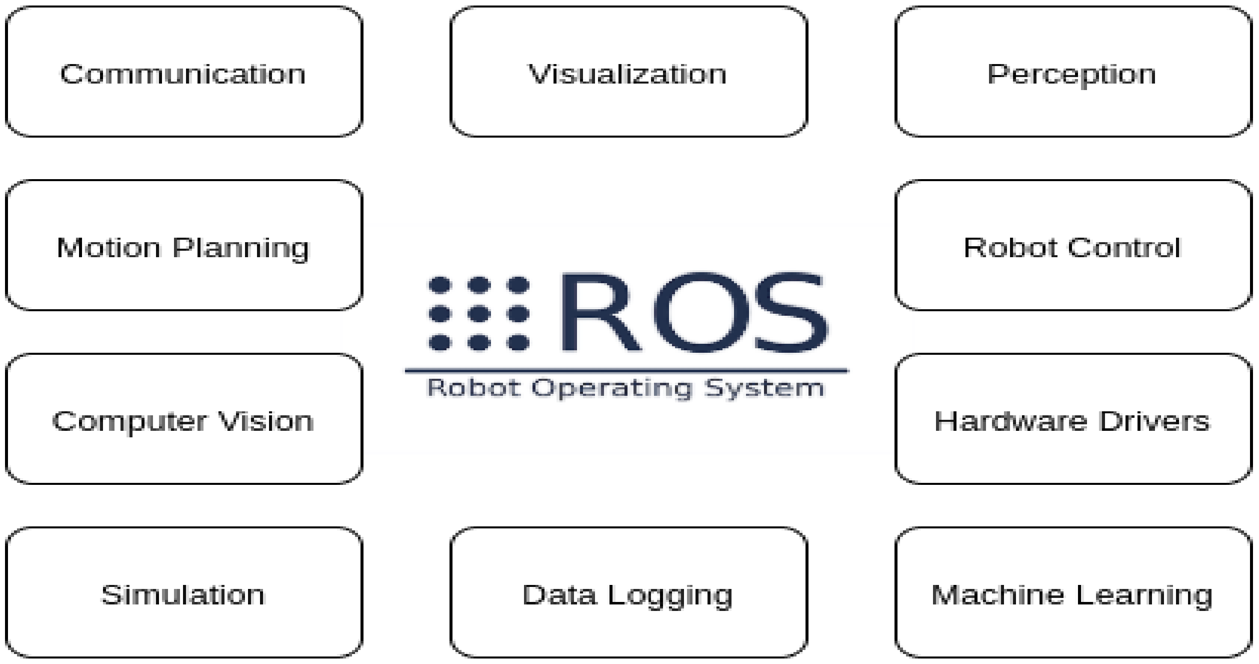
5. Literature

5.1 ROS: - ROS is an open-Lsource, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers. ROS is similar in some respects to 'robot frameworks. The primary goal of ROS is to support code reuse in robotics research and development. ROS is a distributed framework of processes (aka Nodes) that enables executables to be individually designed and loosely coupled at runtime. These processes can be grouped into Packages and Stacks, which can be easily shared and distributed

. ROS also supports a federated system of code Repositories that enable collaboration to be distributed as well. This design, from the filesystem level to the community level, enables independent decisions about development and implementation, but all can be brought together with ROS infrastructure tools.

Distributed applications are designed as units known as Nodes. In a robot system, sensors (Lidar, camera) motion controllers (motors providing motion), algorithm components (route planner) can be nodes each. ROS 2 separates the node concept from the operating system level process structure. If desired, multiple nodes can be created within a process and these nodes can communicate independently with other nodes. All nodes in the system can be run on a single computer or they can be distributed and run across multiple computers.

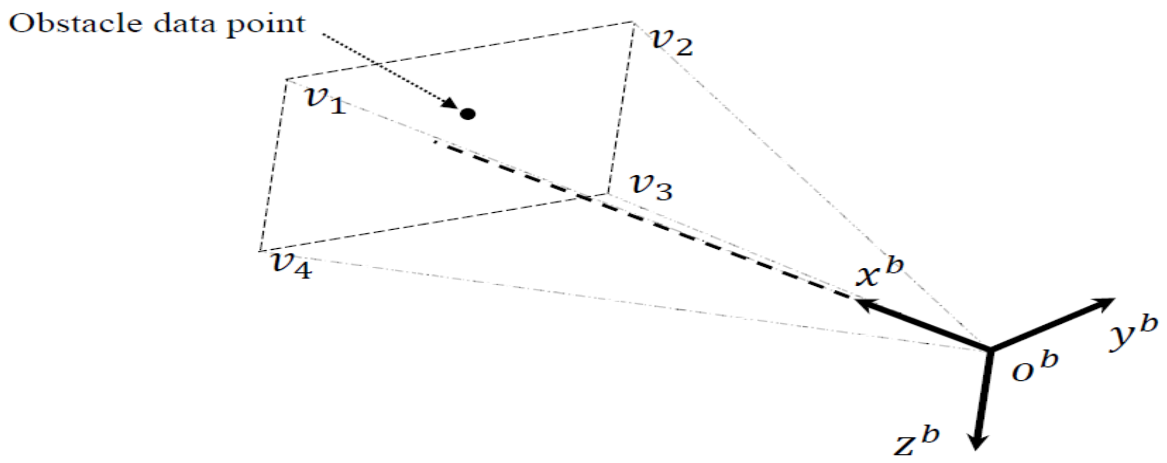
Nodes communicate with each other via Topic, Service Invocation and Actions. The topic-based communication has published subscribe architecture where the data produced and published by a node is subscribed by more than one node and similarly, one topic data can be produced to more than one node



5.2 LiDAR: - Lidar, which is commonly spelled LiDAR and also known as LADAR or laser altimetry, is an acronym for light detection and ranging. It refers to a remote sensing technology that emits intense, focused beams of light and measures the time it takes for the reflections to be detected by the sensor. This information is used to compute ranges, or distances, to objects. In this manner, lidar is analogous to radar (radio detecting and ranging), except that it is based on discrete pulses of laser light. The three-dimensional coordinates (e.g., x,y,z or latitude, longitude, and elevation) of the target objects are computed from 1) the time difference between the laser pulse being emitted and returned, 2) the angle at which the pulse was "fired," and 3) the absolute location of the sensor on or above the surface of the Earth.

In this study, LiDAR sensor was used to measure the exact direction and distance of the obstacle. the distance is calculated through the time from when the laser pulse signals are emitted and reflected pulses reflected from the object are received by the receiver.

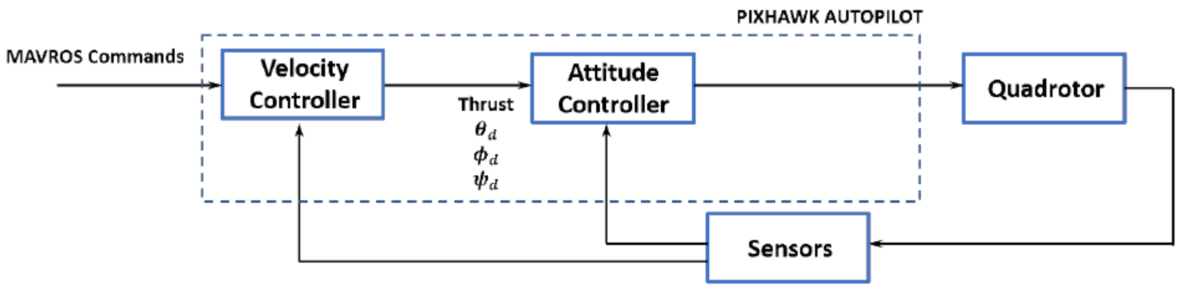
5.2.1 LiDAR Data Sensing: - Let us assume that an obstacle consists of a point cloud that represents its external surface. The LiDAR with limited FoV is mounted on the QUAV and directed along the +xb-axis to acquire obstacle data. The visibility of the obstacle point by the LiDAR can be determined by examining whether the obstacle data point is inside a rectangular pyramid whose vertices are ob, v1, v2, v3, and v4, as shown in Figure 2. v1, v2, v3, and v4 in the body-fixed coordinate system can be obtained as follows.



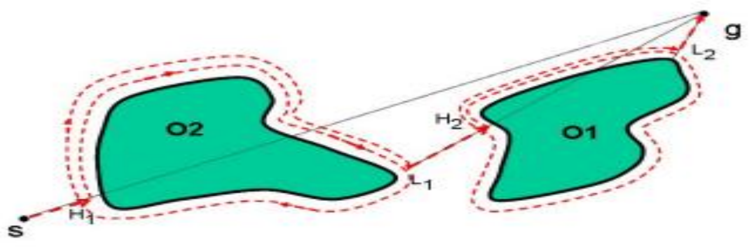
$$
\begin{aligned}
v_1 &= \begin{bmatrix} d_a & -d_a \tan(H_{fv}/2) & -d_a \tan(V_{fv}/2) \end{bmatrix}^T \\
v_2 &= \begin{bmatrix} d_a & d_a \tan(H_{fv}/2) & -d_a \tan(V_{fv}/2) \end{bmatrix}^T \\
v_3 &= \begin{bmatrix} d_a & d_a \tan(H_{fv}/2) & d_a \tan(V_{fv}/2) \end{bmatrix}^T \\
v_4 &= \begin{bmatrix} d_a & -d_a \tan(H_{fv}/2) & d_a \tan(V_{fv}/2) \end{bmatrix}^T
\end{aligned}
$$

where Hf v and Vf v are the horizontal and vertical FoV of the LiDAR, respectively, and da is a sensing range of the LiDAR. The QUAV performs its mission with constant obstacle data acquisition from the LiDAR.

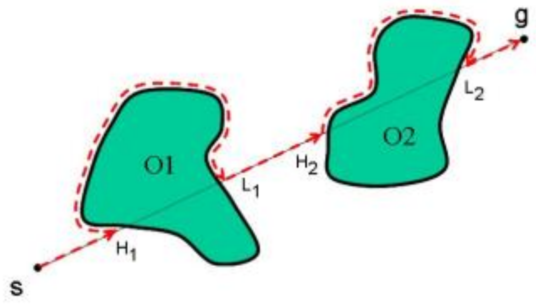
5.3 Flight Control: - The algorithm is designed to send high-level commands to the quadrotor – (1) forward translational speed, (2) constant height, and (3) rotational velocity (heading/desired yaw). The forward velocity of the complete system is constant throughout the whole mission and can be modified by the user based on the application of the vehicle. Fig. 3 shows the schematic diagram of the flight control used for this algorithm. As mentioned, the required high-level commands like the desired yaw, height of the vehicle, etc., are sent to the velocity controller present in the Pixhawk autopilot. These velocity commands are sent via may link protocol while using the MAVROS package to the autopilot. The Pixhawk is connected to the on-board computer via a serial connection. The outer loop uses a PID controller for velocity control. It computes the desired thrust, roll, pitch and yaw and sends it to the attitude controller. The PID control of the attitude controller then realizes the required motor speed on the quadrotor for the optimal motion. The above-mentioned process is completely realized inside the Pixhawk autopilot of the quadcopter.



5.5 The BUG Algorithms: - In its original version Bug 1, [1], or in the improved version Bug 2, [2], are simple ways to overcome unexpected obstacles in the robot motion from a start point s, to a goal point g. The goal of the algorithms is to generate a collision free path from the s to g with the underlying principle based on contouring the detected obstacles. The two versions of the algorithm differ on the conditions under which the border-following behavior is switched to the go-to-goal behavior. Consider that the robot is a point operating in a plane, moving from s to g, and that it has a contact sensor or a zero-range sensor to detect obstacles.



In the Bug 1 algorithm, as soon as an obstacle I is detected, the robot does a full contour around it, starting at the hit point Hi. This full contour aims at evaluating the point of minimum distance to the target, Li. The robot then continues the contouring motion until reaching that point again, from where it leaves along a straight path to the target. Li is named as the leave point. This technique is very inefficient but guarantees that the robot will reach any reachable goal. Figure 1 represents a situation with two obstacles where H1 and H2 are the hit points and L1 and L2 the leave points.



In the Bug 2 algorithm the obstacle contour starts at the hit point Hi but ends whenever the robot crosses the line to the target. This defines the leave point Li of the obstacle boundary-following behavior. From Li the robot moves directly to the target. The procedure repeats if more obstacles are detected. Figure 2 represents the path generated by Bug 2 for two obstacles.

The simplicity of these algorithms leads to a number of shortcomings. None of these algorithms take robot kinematics into account which is a severe limitation, especially in the case of non-holonomic robots. Moreover, and since only the most recent sensorial data is taken into account, sensor-noise has a large impact in the robot performance.

<u>5.6 Potential Field Method</u>: - Path planning using artificial potential fields is based on a simple and powerful principle, first proposed by O. Khatib in [3]. The robot is considered as a particle that moves immersed in a potential field generated by the goal and by the obstacles present in the environment. The goal generates an attractive potential while each obstacle generates a repulsive potential. A potential field can be viewed as an energy field and so its gradient, at each point, is a force. The robot immersed in the potential filed is subject to the action of a force that drives it to the goal (it is the action of the attractive force that results from the gradient of the attractive potential generated by the goal) while keeping it away from the obstacles (it is the action of a repulsive force that is the gradient of the repulsive potential generated by the obstacles). The robot motion in potential field-based methods can be interpreted as the motion of a particle in a gradient vector field generated by positive and negative electric particles. In this analogy, the robot is a positive charge, the goal is a negative charge and the obstacles are sets of positive charges. Gradients in this context can be interpreted as forces that attract the positively charged robot particle to a negative particle that acts as the goal. The obstacles act as positive charges that generate repulsive forces that force the particle robot away from the obstacles. The combination of the attractive force to the goal and the repulsive forces away from the obstacles drive the robot in a safe path to the goal.

The potential field approach herein presented is a simple path planning technique that has an intuitive operation principle based on energy type fields. For a static and completely known environment, the potential can be evaluated off-line providing the velocity profile to be applied to a point robot moving in the energy field from a starting point to a goal. Moreover, the technique can be applied in an on-line version that accommodates an obstacle avoidance component. The repulsive potential may result from known environment obstacles present in an a priori map or obstacles that are detected by the robot on-board sensors during its motion.

A large number of improvements and extensions of the potential field have been published since the early version presented by O. Khatib in [3]. M. Khatib and Chatila in [10] proposed an extended version of the potential field-based methods, where the repulsive potential is changed in two different ways. The repulsive force due to an obstacle is not only considered as a function of the distance to the obstacle but also of the orientation of the robot relative to the obstacle. This is a reasonable change since the urgency of avoiding an obstacle parallel to the robot motion is clearly less than the one that arises when the robot moves directly facing the obstacle. The second extension of the repulsive potential does not consider the obstacles that will not closely affect the robot velocity. For example, it is irrelevant to consider the repulsive force generated by an obstacle in the back of the robot, when it is moving forward.

6.Methodologies

First we installed Ardupilot for drone software development and then run it in the SITL(Sofware in the loop) simulation. We installed MAVProxy for communicating with drone.

Ardupilot is an open-source software suite for autonomous drones' development. It has a number of advance features available. We use Ardupilot and Qgroundcontrol in conjuction with each other for simulation.But we were unabled to see the 3D drone and environemnt thus we installed Gazebo which is one of the open source robotic simulator present. In GAzebo we can model different environemnts as well as integrate different sensors with our drone

Then we setup ROS environment in our linux system for drone development

As ROS has 2000+ packages available we used them rather wasting time in reinventing the wheel from scratch.

We added subscriber for LIDAR node for retrieving sensor data

ros::Subscriber collision_sub = n.subscribe<sensor_msgs::LaserScan>("/spur/laser/scan", 1, scan_cb);

Then we initialised all the publisher/subscriber and all the commands required for takeoff.

After which we used code in Scan_cb function which is the application of repulsive potential between two charges

$$U_{rep_i}(q) = \begin{cases} \frac{1}{2}k_{obst_i}\left(\frac{1}{d_{obst_i}(q)} - \frac{1}{d_0}\right)^2 & if \quad d_{obst_i}(q) < d_0 \\ 0 & if \quad d_{obst_i}(q) \geq d_0 \end{cases}$$

where dobsti (q) is the minimal distance from q to the obstacle i, kobsti is a scaling constant and d0 is the obstacle influence threshold.

The negative of the gradient of the repulsive potential, $F_{rep_i}(q) = -\nabla U_{rep_i}($
is given by,

$$F_{rep_i}(q) = \begin{cases} k_{obst_i}\left(\frac{1}{d_{obst_i}(q)} - \frac{1}{d_0}\right)\frac{1}{d^2_{obst_i}(q)}\frac{q-q_{obst}}{d_{obst_i}} & if \quad d_{obst_i}(q) < d_0 \\ 0 & if \quad d_{obst_i}(q) \geq d_0 \end{cases}$$

Above equation gives us the imaginary repulsive force experienced by the drone and obstacle which gives us our path planning algorithm.

7.Conclusion

Doing this project we came through various techniques, algorithms, development environments and software. We faced many challenges like non-compatible environments, version mismatches, setting up complicated development tools, steep learning curve, limited time duration to name few but we overcame them and made this project.

Advantages:-We found that BUG algorithm utilizes limited computational resources available onboard the QUAV very efficiently unlike DNN based approach which takes considerable time in processing the image captured via camera. Our simulated QUAV achieved all the goals we defined earlier like avoiding obstacles and reaching the pre-defined target position in limited time.

Disadvantages:-In simplest form of BUG algorithm and potential field-based path planning has few shortcomings such as irrelevant force experienced by drone when it has already passed the obstacle, it is sensitive to the robot oscillatory behavior in narrow passages. As for decisions we take most recent sensor data in account BUG algorithm is prone to noise in sensor data.

Overall, it worked out great and we solved the problem defined earlier in problem statement.

8.Future Prospects

We can use extended artificial potential field-based methods as suggested in O. Khatib, M. Khatib and Chatila in their paper which accounts for force as function of distance as well as orientation of drone and obstacle.

We can use BUG2 algorithm which can reduce the travelling time of drone.

For collision avoidance system in future, we can use DNN based approach which is most promising as computational power available increases.

9.References :-

[1] H. Alvarez, L. Paz, J. Sturm, and D. Cremers. Collision avoidance for quadrotors with a monocular camera. In International Symposium on Experimental Robotics (ISER), pp. 195–209, Springer, 2016

[2] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy. Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. Intl. J. Robotics Research, 31(11):1320–1343, 2012.

[3] S. Bhatti, A. Desmaison, O. Miksik, N. Nardelli, N. Siddharth, P. H. S. Torr. Playing Doom with SLAM-augmented deep reinforcement learning. arXiv preprint, arXiv:1612.00380, 2016.

[4] D. Scaramuzza et al. Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. IEEE Robot. Auton. Mag., 21(3):26–40, Sep. 2014.

[5] A. Bry, A. Bachrach, and N. Roy. State estimation for aggressive flight in GPS-denied environments using onboard sensing. In International Conference on Robotics and Automation (ICRA), 2012.

[6] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma. Flying fast and low among obstacles: Methodology and experiments. International Journal of Robotics Research, 27(5):549–574, 2008.

[7] R. A. Sasongko, S. S. Rawikara, and H. J. Tampubolon, “UAV obstacle avoidance algorithm based on ellipsoid geometry,”Journal of Intelligent & Robotic Systems, vol. 88, no. 2-4,

pp. 567–581, 2017.

[8] Foehn P, Romero A, Scaramuzza D. Time-optimal planning for quadrotor waypoint flight. Science Robotics. 2021;6(56):eabh1221.doi:10.1126/scirobotics.abh1221.

[9] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. Journal of Field Robotics, 33(4):431–450, June 2016.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semidirect monocular visual odometry. In International Conference on Roboticsand Automation (ICRA), 2014.

[11]. Kaufmann E, Loquercio A, Ranftl R, Müller M, Koltun V, Scaramuzza D.Deep Drone

Acrobatics. RSS: Robotics, Science, and Systems. 2020;.

[12]. A. al-Kaff, F. García, D. Martín, A. de la Escalera, and J. Armingol, "Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs," Sensors, vol. 17, no. 5, pp. 1061–1082, 2017.

[13] J. Zheng, B. Liu, Z. Meng, and Y. Zhou, "Integrated real time obstacle avoidance algorithm based on fuzzy logic and L1 control algorithm for unmanned helicopter," in 2018 Chinese Control and Decision Conference (CCDC), pp. 1865–1870, Shenyang, China, 2018.

[14] F. Fraundorfer, H. Lionel, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor MAV. In International Conference on Intelligent Robots and Systems (IROS), Nov. 2012.