Assisted Teleoperation of a Quadrotor using Active Perception [1]

João Mendes
jmendes@isr.ist.utl.pt
Rodrigo Ventura
rodrigo.ventura@isr.ist.utl.pt

Institute for Systems and Robotics
Instituto Superior Técnico
Av. Rovisco Pais, 1
Lisbon, Portugal

Abstract

The teleoperation of UAVs often demands extensive training, since even well trained pilots are prone to mistakes, resulting frequently in collisions of the vehicle with obstacles. This paper presents a method to assist the teleoperation of a quadrotor using an obstacle avoidance approach. The target scenario is unknown, unstructured, and GPS-denied. A short-term rough map of the nearby environment is constructed using sonar sensors. This map is constructed using FastSLAM to allow tracking of the vehicle position with respect to the map. A danger classification method is then applied to choose the appropriate action for each particular, and potentially dangerous, situation. A simple active perception routine is used to orient one of the sensors to an unknown area, in case the UAV is ordered to move towards an unmapped area. Real world results are presented allowing a preliminary validation of the proposed methods.

1 Introduction

Quadcopters have several characteristics that make them well fit for situations where hovering is crucial, such as indoor usage. In particular, its high agility, small size and hover capacity make this type of vehicles useful for, as example, remote inspection of areas where ground vehicles are unable to reach. However, its teleoperation, particularly in confined environments, is far from trivial.

In this paper we address the problem of *Assisting Teleoperation* of a quadcopter. We define assisted teleoperation as the process of overriding the operator input, either by modulation, inhibition, or replacement with a different input. Our goal is to use sensor data to determine an appropriate assisted teleoperation in order to guarantee a safe flight.

The presented solution is based on a FastSLAM [5] approach using an occupancy grid map [1]. Since the purpose of this work is not a detailed map of the environment, the problem can be efficiently addressed by knowing the relative position of the quadrotor in relation to nearby obstacles. After knowing the vehicle's position and map, a decision making based on danger assessment is applied. This classifier overrides the user's inputs if they compromise the quadcopter's physical integrity in the near future. Overriding may range from simple velocity reduction to, in extreme cases, an evasive maneuver. As our aim is to ensure the vehicle's safety at all times, an active perception methodology is applied to address map uncertainty whenever it is necessary. Unlike a purely reactive methodology, if the map is kept in memory it is possible to avoid crashes in sonar's blind spots. The full architecture is presented in Figure 1.

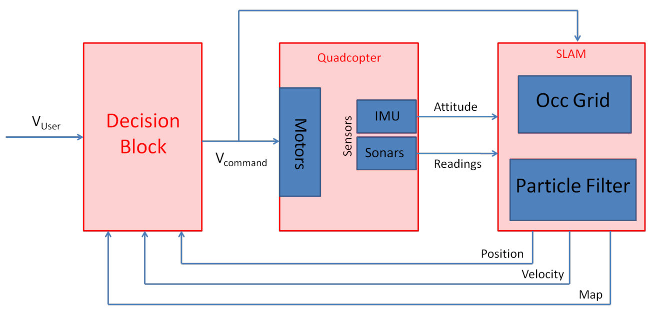

Figure 1: Full architecture of the proposed approach

3D Simultaneous Localization and Mapping (SLAM) in Unmanned Aerial Vehicles (UAV) using lasers has been studied but typically including techniques, such as loop closure algorithms [4]. As for obstacle avoidance methodologies for UAVs, literature mostly addresses path re-planing topics [2]. This paper differs from the above in that we aim at a rough and low complexity map, and thus more time efficient.

2 Methodology

2.1 FastSLAM

Correct attitude is assumed to be maintained at all times by an onboard IMU. Since accurate attitude estimations can be provided by a commercial solution, the 6D problem (position and attitude) is reduced to a 3D problem (position only). The objective of SLAM is to estimate the robot's position and the 3D map of the environment simultaneously and hereby solved by the FastSLAM approach proposed by Montermerlo et al. [3]. The conditional independence propriety of the SLAM problem allows for a factorization of the posterior and decompose SLAM into a path estimation problem and a mapping problem. This is solved by a combination of a Particle Filter with an occupancy grid mapping algorithm.

2.2 Decision Block

The inputs for the Decision Block are the position estimation and a map m. Each map cell i is classified in one of 3 states, $c^i \in \{F, U, O\}$, corresponding to Free, Unknown, Occupied. The classifier is based on thresholding the occupancy probablities, according to:

$$c^i = \begin{cases} F & \text{if } P(m^i_{occupied}) < 0.5 \\ U & \text{if } P(m^i_{occupied}) = 0.5 \\ O & \text{if } P(m^i_{occupied}) > 0.5 \end{cases} \tag{1}$$

All cells are initialized with $P(m^i_{occupied}) = 0.5$ thus classified as Unknown. After being mapped, they become classified as either Free or Occupied.

The global flow chart of the Decision Block is presented in Figure 2. When the user inputs a desired velocity to the quadcopter, the Active Perception block, Figure 3, validates if the commands will be applied to the vehicle without any constraint. To do so, the inputs are firstly used to compute the desired movement direction and the distance to the first non-Free cell. If the closest cell is Occupied, the inputs are subject to a confirmation that they do not compromise the vehicle's safety in a near future and then applied to the vehicle. If the closest cell is Unknown and if the distance to it is larger than a certain threshold (TH) the algorithm checks whether there is any sonar aligned with the desired direction. If not, the algorithm will autonomously rotate the vehicle and then apply the same norm of velocity the user demanded. The algorithm is, therefore, able to avoid flying in unknown areas as the sonar is pointing towards the movement direction and still perform the order the user requested. By applying active perception it is possible to guarantee that the vehicle is not allowed to fly towards unknown areas.

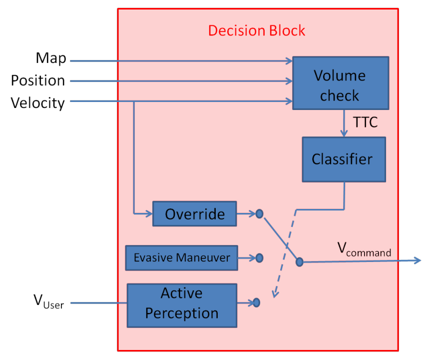

Figure 2: Flow chart of the Decision Block.

[1] This work was supported by the FCT projects [PEst-OE/EEI/LA0009/2011] and [PTDC/EIA-CCO/113257/2009].

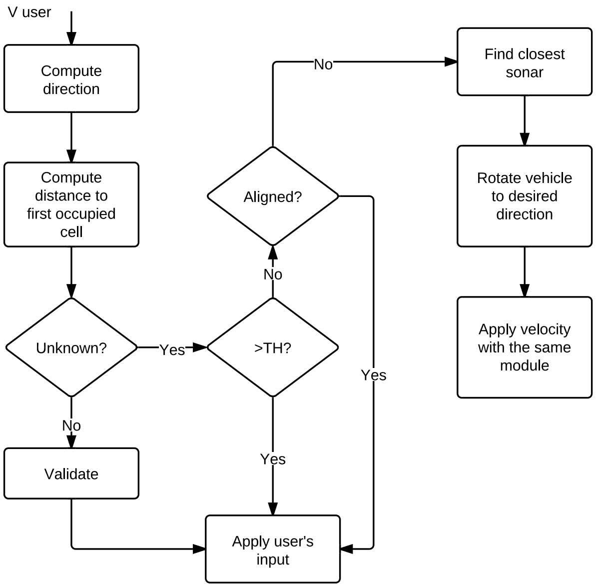

Figure 3: Flow chart of the Active Perception block.

An obstacle's position is computed by a method designated here by *volume check*. The algorithm applies an extrusion of a square centered on the quadcopter's position, along the velocity vector (, as illustrated in Figure 4. The size of this square, *b*, encompasses the quadrotor volume while *a* is the length of the volume in which obstacles are searched. This volume enables the algorithm to know which grid cells are in the near-future path of the vehicle, to determine the position of the closest occupied cell, and to compute the distance between that cell and the vehicle. It is then possible to predict how long it takes — if we maintain the current speed — to collide with it. This concept is known as Time To Collision (TTC) and is a crucial step in the classification of the danger levels.

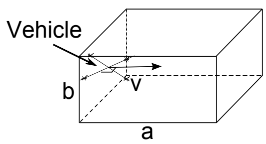

Figure 4: Graphical definition of the volume check. V represents the velocity vector.

The classifier block acts as a multiplexer by choosing an input, and forwarding it to the vehicle, given a certain TTC. The threat levels differ in the action performed. This action may be allowing the user to fully control the vehicle or to impose an override to the user's inputs. This override ranges from ordering the vehicle to lower its velocity if the danger is low to, in extreme cases, an evasive maneuver.

3 Real world results

In this section, a preliminary evaluation to the architecture, combining the FastSLAM and the decision block in a real world experiment, is presented. The available quadrotor is equipped with an IMU capable of providing filtered values of yaw, pitch and roll. Four sonars were used and placed above each of the propellers. The sonars maximum range is three meters. All readings were recorded with the quadrotor's motors off and the vehicle manually placed in the different poses throughout the tests.

The presented example considers 5 different stages illustrated in Figure 5. It is considered that the user is ordering a constant movement towards the wall through the entire test. With this set-up, we aim at a possible and common movement where a purely reactive methodology would not manage to avoid the collision.

It is observable that in position 1 the vehicle has $pitch = 0^\circ$ and is able to map the area where it is going to be in position 2. The vehicle then moves to the previously mapped and now known to be free area. Since the distance to the nearest obstacle — in this case an unknown position — is still high enough (Table 1) the algorithm allows the user to maintain control of the quadcopter. When the vehicle reaches position 3, the distance between its position estimation and the first obstacle in its own map will lower and the TTC decreases. As a consequence, the user inputs

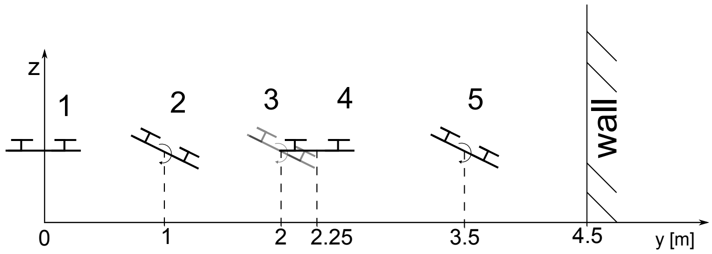

Figure 5: Position and attitude in each of the 5 phases.

will be overriden by the Decision Block and the vehicle ordered to slow its movement. When the velocity is lowered — position 4 — the vehicle regains sight of its moving direction and re-updates the map. When the distance to the obstacle is updated, the algorithm allows the user to input commands once again. This situation can occur multiple times until the vehicle reaches a real obstacle.

In Table 1, the first line shows the distance directly reported by sonars in each situation while the second one shows the distance between the position estimation and the first obstacle in its own map. The proposed solution keeps a constant track of the distance to the obstacle allowing the classifier to choose the best action in each situation. It is also noticeable that the algorithm, due to the acquired attitude, only updates the distance to the obstacle when the sonar is pointed to the direction of movement — position 1 and 4. In the last two rows it is possible to see the classification C given to the closest non-free cell i and the consequent action performed on the command inputs.

Table 1: Comparison between the distance both methods and consequent classifier decision.

	1	2	3	4	5
$d_{reactive}$	3	–	–	2	–
$d_{mapping}$	3.25	2.34	1.38	2.28	1.11
C^i	U	U	U	O	O
Classifier decision	User	User	Override	User	Override

4 Conclusion

This paper presented an assisted teleoperation method for UAVs, based on short-term, rough mapping of the nearby environment. In particular, we targeted quadcopter vehicles operating in unstructured, unknown, GPS-denied environments. By knowing the environment, the algorithm is capable of overriding user inputs whenever the vehicle faces a potentially dangerous situation. Whenever confronted with an unknown area, the active perception routine forces the vehicle to point a sensor towards that area. The main objective was successfully achieved in simulation and partially evaluated in a real world scenario. As for future work, we are currently working in a full real world evaluation of the proposed method.

References

[1] A. Elfes. *Occupancy grids: A probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.

[2] Zhihai He, Ram Venkataraman Iyer, and Phillip R Chandler. Vision-based uav flight control and obstacle avoidance. *2006 American Control Conference*, pages 2166–2170, 2006.

[3] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

[4] Cyrill Stachniss, Dirk Hï£¡hnel, Wolfram Burgard, and Giorgio Grisetti. On actively closing loops in grid-based fastslam. *ADVANCED ROBOTICS*, 19:2005, 2005.

[5] Anthony Stentz, Dieter Fox, Michael Montemerlo, and Michael Montemerlo. Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2003.