AN OCTOMAP COLLISION AVOIDANCE APPROACH ASSISTING REMOTELY OPERATED QUADROTORS

Bruno Giovanini*, Raphael Teixeira*, Paulo F F Rosa*

*Instituto Militar de Engenharia, Rio de Janeiro, Brazil

Emails: bsgiovanini@gmail.com , raphael.teixeira@gmail.com, rpaulo@ime.eb.br

Abstract— The use of quadrotors to civilian and military missions has been increased and the challenges involved on controlling it, mainly in indoors and restricted environments, has been attracting robotics researchers. Thus, an automatic collision avoidance approach is of utmost importance in this scenario, given the difficulty of control and the risk of accidents involved in the use of these vehicles. This paper presents an approach for obstacle avoidance of a manually controlled quadrotor automatically, allowing the operator to keep focus on the overall mission. The method is based on constantly estimating its future path considering its dynamics, current status, current control and distances measured by four on-board sonar sensors. Simultaneously, the pose is estimated based on the quadrotor odometry and an Octomap occupation grid representation of the nearby environment is constructed using the sonar sensors measurements. All that information is used to determine an imminent collision and overrides the user control, if necessary, keeping its last safe position. All the solution was evaluated in a simulator and fulfilled its purpose by avoiding obstacles in a safe manner. The solution is ready to be embedded in an real quadrotor.

Keywords— indoor environment, collision avoidance, trajectory estimation, mapping, PID control

1 Introduction

The use of quadrotors to civilian and military missions has been increased and its related challenges have been attracting robotics researchers in a variety of areas such as surveillance and rescue in remote areas.

The usual platform with four rotors has the suitable ability to perform vertical take off and landing, hovering and performing fast maneuvers in different environment conditions. However, controlling a quadrotor is not a simple task when a safe flight is desired. Even extensively trained pilots may face situations where it is hard to guarantee a safe flight due to, for instance, loss of visual contact. So, an automatic collision avoidance method to assist the operation of the platform is required.

Our solution consists in an algorithm to assist the operation of a quadrotor avoiding obstacles based on the Simultaneous Localization and Mapping approach (SLAM), with the global position being calculated continuously during the flight through its odometry data, while on-board sonar sensors measure the relative distances to nearby obstacles and map the environment as an occupancy grid map (Thrun et al., 2005). Then, the user input is overridden if a possible collision is detected in a near future based on its calculated future trajectory and obstacle positions. A Proportional-Integral-Derivative controller (PID) is used to override the user input while the platform is in the imminence of a collision with an obstacle.

2 Related Work and State-of-the-art

Recently, unmanned aerial vehicles have received greater attention from the robotics community. Many authors have focused on the modelling an control of these vehicles (Ross et al., 2013), particularly, on quadrotors (Lugo and Zell, 2014). In this context, comprehensive studies about the platform configuration, modelling methodologies, comprehensive nonlinear modelling, aerodynamic effects, identification and simulation were performed (Zhang et al., 2014). For being considered a nonlinear complex system, techniques for quadrotor control also have been widely studied. (Salih et al., 2010) developed a PID control for a quadrotor to stabilize it in flight. On the other hand, (Nicol et al., 2008) proposed an adaptive neural network to stabilize the quadrotor taking into account wind modelling errors and disturbances.

Several applications are under study on the quadrotor platform. For the majority (e.g., autonomous navigation, multi-agent tasks and obstacle avoidance), estimating the vehicle state is an well established challenge (Ito et al., 2014). Regarding to indoor environments, computer vision, which makes use of cameras to image processing, is one of the researching area about the problem (Shen et al., 2013). Another area is about using stereoceptive sensors for this task, such as sonars (Mendes and Ventura, 2013) and laser (Grzonka et al., 2012). For outdoor environments, however, Global Positioning System (GPS) is available and some applications using this technology have been developed, such as (Kerns et al., 2014) and (Roberts and Tayebi, 2013).

Specifically, with regards to collision avoidance problem, (Israelsen et al., 2014) conducted an approach to head quadrotor off the obstacle to a safe position. However, the referred method considers the environment and obstacle positions previously known and being

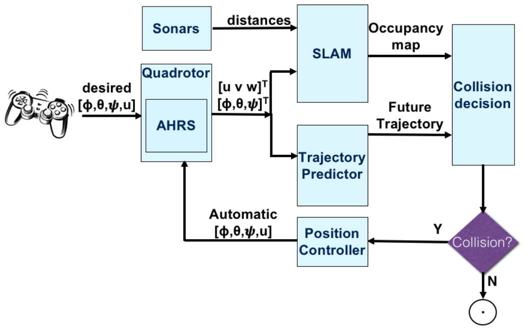

Figure 1: An overall representation of our proposed architecture.

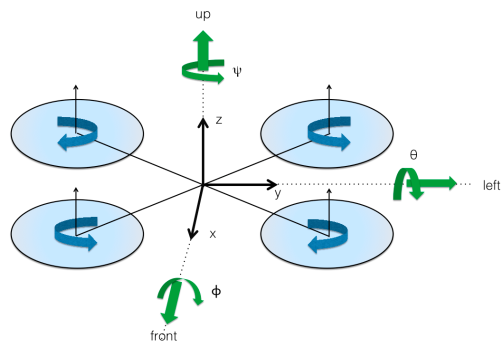

Figure 2: A simplified quadrotor model with arrows indicating the rotors' directions in blue and the angular (ϕ,θ,ψ) and linear (x,y,z) moviments in green.

already geometrically preprogrammed, not using real time measurements obtained from on-board sensors. (Grzonka et al., 2012) also developed a fully autonomous quadrotor in an indoor environment using, for obstacle avoidance, a laser scanning sensor miniature. (Becker et al., 2012) used five ultrasonic sensors for that, but both (Grzonka et al., 2012) and (Becker et al., 2012) used a very well-known platform developed and studied in laboratory for years. (Mendes and Ventura, 2013) proposed a real time method, using fast simultaneous localization and mapping (FastSLAM) with occupancy grid mapping (Thrun et al., 2005) to localize the quadrotor and map the environment in a simultaneous way, however, using an off-board approach and validating it in a simulator.

3 Collision avoidance: our approach

Indeed, there are many steps in implementing our proposed method. An overview of its entire architecture is presented in Figure 1. Firstly, the user's desired control is inputed to the quadrotor, which provides its attitude and velocity through an embedded Attitude Heading Reference System (AHRS). That information is used by both the trajectory estimator and SLAM modules that output the future trajectory and an occupation grid representation of the environment, respectively. Both information are confronted each other by a collision decision block which detects an imminent collision. If negative, the system does nothing. Otherwise, a position controller determines the automatic control necessary to keep the last safe position. Finally, the user control is overridden and the automatic control is passed to the quadrotor. Following are the topics supporting this architecture.

3.1 The Quadrotor Platform

A quadrotor is classified as an under-actuated system. It receives four inputs, given by the thrusts generated vertically by its four rotors, and produces six outputs, represented by the three angular movements roll (ϕ), pitch (θ) and yaw (ψ) and the three linear movements (x,y,z) (Gupte et al., 2012). The movements are generated combining the variation of each rotor's thrust, with one pair of opposed rotors rotating clockwise and the other one counter-clockwise to balance the total torque of the system (Mian and Daobo, 2008). Those characteristics are shown in Figure 2.

In order to develop our method, it was used a robust small quadrotor with on-board electronic stabilization, equipped with a front-camera and a down-looking camera, making use of the latter to improve stability. Furthermore, it is equipped with a sonar pointing downwards to measure altitude and an AHRS measuring (ϕ,θ,ψ) and linear velocities along all axes. The vehicle is controlled by sending commands over a WI-FI connection and outputs its navigation data, including acceleration, velocity and attitude at frequency of $200Hz$ (Dijkshoorn and Visser, 2011).

3.2 Future Trajectory Estimation

The quadrotor dynamics supports this context. Considering that we are working with a quadrotor that gives its acceleration, velocity and attitude over the time, the source of that data is neglected and only its kinematics is considered.

The transformations related to the study of kinematics involve one translation and successive rotations from the inertial referential $\mathcal{F}^i$ until the body-fixed referential $\mathcal{F}^b$. Figure 3 shows the transformations step-by-step from $\mathcal{F}^i$, passing by three intermediaries referentials, including one translation from $\mathcal{F}^i$ to $\mathcal{F}^v$ and three rotations ψ-θ-ϕ for the sequence $\mathcal{F}^v$-$\mathcal{F}^{v1}$-$\mathcal{F}^{v2}$-$\mathcal{F}^b$ (Beard and McLain, 2012).

The rotation matrix $R_v^b(\phi,\theta,\psi) = R_{v2}^b(\phi)R_{v1}^{v2}(\theta)R_v^{v1}(\psi)$ represents the

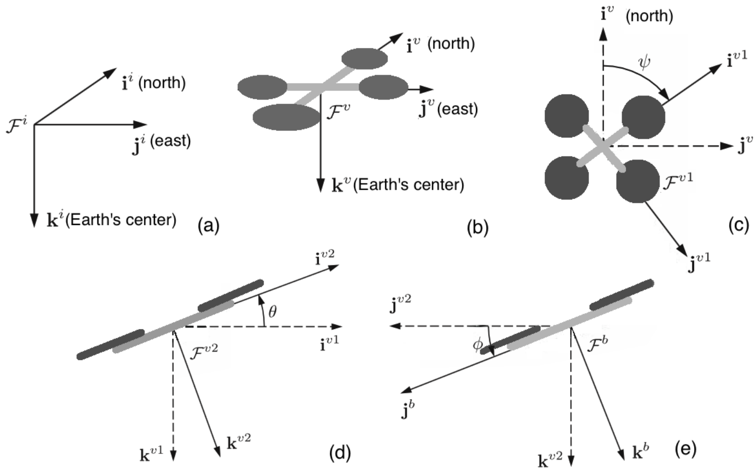

Figure 3: Referentials related with the study of the quadrotor dynamics from the inertial referential $\mathcal{F}^i$(a), with a translation to $\mathcal{F}^v$(b) and three sucessive rotations ψ, θ, ϕ to F^{v1}(c), $\mathcal{F}^{v2}$(d) and the body-fixed referential $\mathcal{F}^b$(e), respectively.

transformation from $\mathcal{F}^v$ to $\mathcal{F}^b$ and the matrix W describes the relation between the vehicle's angular velocity ω and Euler's Angles variation $\dot{\theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ (Beard and McLain, 2012). Given the global quadrotor's current position $\mathbf{x}$ and its current measurements: linear acceleration $\ddot{\mathbf{x}}$, linear velocity $\mathbf{v}$, attitude θ, angular velocity ω and angular acceleration $\dot{\omega}$, the trajectory is calculated by an integration over time, considering the previous quantities as initial conditions. Both accelerations $\ddot{\mathbf{x}}$ and $\dot{\omega}$ are constant during the trajectory. That is reasonable given the short period of time for trajectory estimation. One step of the prediction is presented in equation (1) (where dt is a constant representing the duration time between two steps).

$$\begin{aligned} \omega_t &= \omega_{t-1} + dt * R_b^v * \dot{\omega} \\ \dot{\theta}_t &= W^{-1} * \omega_t \\ \theta_t &= \theta_{t-1} + dt * \dot{\theta}_t \\ \dot{x}_t &= \dot{x}_{t-1} + dt * R_b^v * \ddot{\mathbf{x}} \\ x_t &= x_{t-1} + dt * \dot{x}_t \end{aligned} \tag{1}$$

3.3 Simultaneous Localization and Mapping

Our collision avoidance problem considers the fact that we are in an unknown environment and under sonar measurements uncertainly. The objective of SLAM is to estimate position of a robot and its nearby map simultaneously in a probabilistic fashion (equation (2)). Let $x_{1:t}$ denote the path of the robot, m the map, $z_{1:t}$ all measurements and $u_{1:t}$ all control inputs where $1 : t$ represents the time step from 1 to t. The conditional independence propriety of the SLAM problem (Thrun et al., 2005) allows us to perform a factorization that decomposes SLAM into a localization estimation and mapping problem (equation (3)).

$$p(x_{1:t}, m | z_{1:t}, u_{1:t}) \tag{2}$$

$$p(x_{1:t} | z_{1:t}, u_{1:t}) p(m | z_{1:t}, x_{1:t}) \tag{3}$$

Getting the data given by the quadrotor at a frequency of $200Hz$ as shown in section 3.1, it is possible to integrate over the time and get its global location. Given x_{t-1} the position in $t-1$, v_t the velocity and a_t the acceleration provided by the quadrotor, the position x_t is calculated as demonstrated in the equation 4.

$$x_t = x_{t-1} + v_t * \Delta t + a_t * 0.5t^2 \tag{4}$$

In general, the imprecision and noise of sensors' data in calculating distances, including sonars, lead us to adopt probabilistic approach for mapping. Occupancy grid maps address the problem of generating consistent maps from noisy and uncertain measurement data, given the robot pose (Thrun et al., 2005).

We adopted the Octomap to generate volumetric 3D environment models, based on octrees and probabilistic occupancy estimation. As input, it receives measurements from the embedded sonars and the desired resolution and outputs the nearby environment grid representation generated after a flight. It explicitly represents not only occupied space, but also free and unknown areas. Its central property is its efficiency and the ability of promoting probabilistic updates of occupied and free space while keeping memory consumption at a minimum (Hornung et al., 2013).

3.4 Our approach for Collision Detection

A possible collision is verified based on an intersection between the region occupied by the obstacle and the quadrotor. Here, the

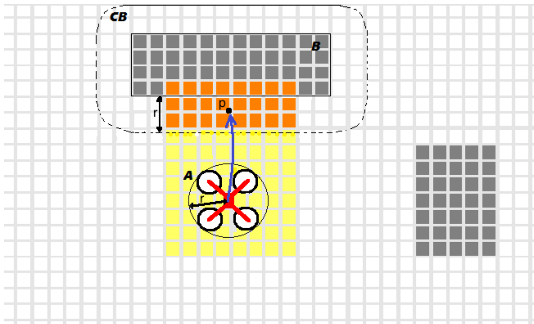

Figure 4: Illustration of the intersection (in orange) between the collision scanning area (in yellow) and the obstacle region (**CB**) during flight. The quadrotor is represented by the sphere region **A** with radius r.

quadrotor's region is simplified as a sphere and the obstacle's region includes unknown areas and the area increased by the quadrotor sphere radius. During the flight, the scanning collision area is constrained as the possible affordable positions in the interval $t \in [0, \tau]$ while the future trajectory is constantly calculated. Both are based on the quadrotor dynamics and state, including position and velocity.

Given the vehicle position $\mathbf{x} = (x, y, z)$, the scanning collision area is delimited as follows. Initially, the affordable distance from its position $\mathbf{x}$, denotated by Δ_x, which is defined by $\Delta_x = v_{max} * \tau$ where v_{max} is a fixed value representing the vehicle maximum affordable velocity and τ is the configured forwarding time. Afterwards, all the map grid cells are covered in a loop through the volume formed by the interval $[\mathbf{x} - \Delta_x/2, \mathbf{x} + \Delta_x/2]$. Furthermore, each future trajectory step is evaluated against the current grid cell, looking for an occupied area indicating a possible collision with the quadrotor area. The Figure 4 illustrates our approach. The environment is depicted as a grid. Region **A** represents the quadrotor's sphere with radius r and **CB** is the obstacle region. The collision scanning area is in yellow and its intersection with **CB** is in orange showing all possible positions for a potential collision. A future trajectory is represented by the blue arrow leading to the future position p at time τ, which demonstrates an imminent collision.

3.5 Controlling to Safe Position

The last known safe position is determined when a future trajectory is estimated and a colliding path is found. In this case, an automatic control is enabled and overrides the user input for 3 (three) seconds. This approach helps to maintain the user's experience letting the quadrotor stop while keeping its attitude and position.

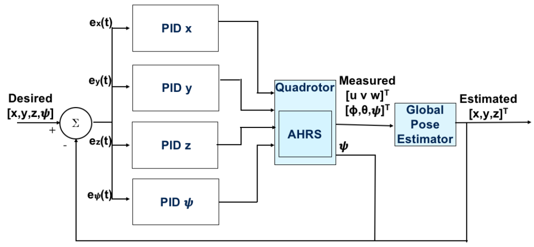

Figure 5: PID controller scheme to lead the quadrotor to its last safe pose. The platform is controlled by four PID controllers for x, y, z and ψ, which receive feedback from the global position estimator.

Four PID Controllers were implemented as a combination in order to control the quadrotor global position (x,y,z) and yaw (ψ) (Lugo and Zell, 2014). Figure 5 shows the controller scheme. Error inputs are calculated regarding to the difference between the desired safe position and ψ and its current data obtained from a global position estimator using the quadrotor AHRS measurements. Then, they are mapped to velocity commands accordingly to the range of acceptable vehicle commands. The control is calculated at the same frequency that the platform sends its navigational data ($200Hz$) during 3 (three) seconds. The proportional, integrated and derivative constants k_p, k_i and k_d were measured empirically.

4 Experimental Setup and Results

The solution was implemented in C++ language. An Octomap cell resolution of $0.1m$ was chosen allowing to have a good environment representation regarding to the solution performance. Tests were executed to determine efficiency and its correctness. All data used in our experiments was obtained through the TUM Simulator over the Robotics Operation System (ROS), which models the quadrotor with fidelity and allow us to communicate with the vehicle over WI-FI and to get its navigation data in real time. Figure 6 illustrates the simulator running our solution showing the scanning collision areas being mapped over the time (in yellow) while the quadrotor trajectory towards to the obstacle (arrows in green). In an imminent collision (orange area), an automatic control is generated (arrow in red) and stops the vehicle safely.

The forwarding time τ has influence in the solution behavior as a configuration parameter. The smaller is τ, the shorter will be the scan collision area and closer to the obstacle the quadrotor will stop. Table 1 gives the distances obtained when τ equals to $0.5s$, $1.0s$, $1.5s$ and

ISSN 2525-8311

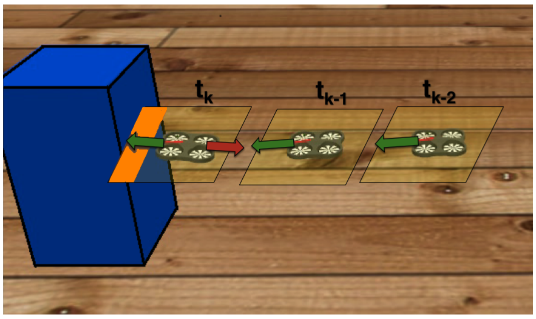

Figure 6: Simulator running our solution while the quadrotor's trajectory towards to the obstacle and scanning collision areas are mapped.

Table 1: Distance to obstacle when the quadrotor stops considerating different fowarding times τ.

τ (s)	Distance to obstacle (m)
0.5	0.031915
1.0	0.104508
1.5	0.518959
2.0	0.907943

$2.0s$. Due to quadrotor's sensors and sonars imprecisions, it was considered a τ equals to $1.0s$.

The automatic control was calibrated by tunning empirically the four PID controllers' constants k_p, k_d and k_i. An experiment, with the quadrotor in idle state at position $x = 0$, $y = 0$, $z = 0$ and $\psi = 0$, was conducted to lead it to the position $x = 0.5$, $y = 0.5$, $z = 0.5$ and $\psi = 1rad$. Four combinations of k_p, k_d and k_i were evaluated to determine which one has major influence on the automatic control behavior over time. The faster is the stabilization time combined with a lower oscillation, the better is the automatic control response time with minor risk to assume an unknown position. For both x, y and yaw, $k_p = 2.0$, $k_d = 0.0$ and $k_i = 0.0$ stabilized first with a lower oscillation. In case of z, $k_p = 2.0$, $k_d = 2.0$ and $k_i = 0.0$ presented a better result.

In our experiment, the program managed to fulfill its objective and avoid the crash despite the orders from the user to continue its trajectory towards the obstacle. The quadrotor was represented as sphere model with radius of $0.26m$ and a security distance of $0.1m$ from the obstacle was considered. Figure 7 presents the satisfactory behavior of our solution when the vehicle is headed toward the obstacle while the controller is managed at full throttle. The distance is represented in blue (Figure 7a) and keep decreasing until our security distance of $0.1m$ from the obstacle (in orange). From that moment, the distance is kept constant even when the maximum velocity command is applied. Figure 7b shows the velocity command

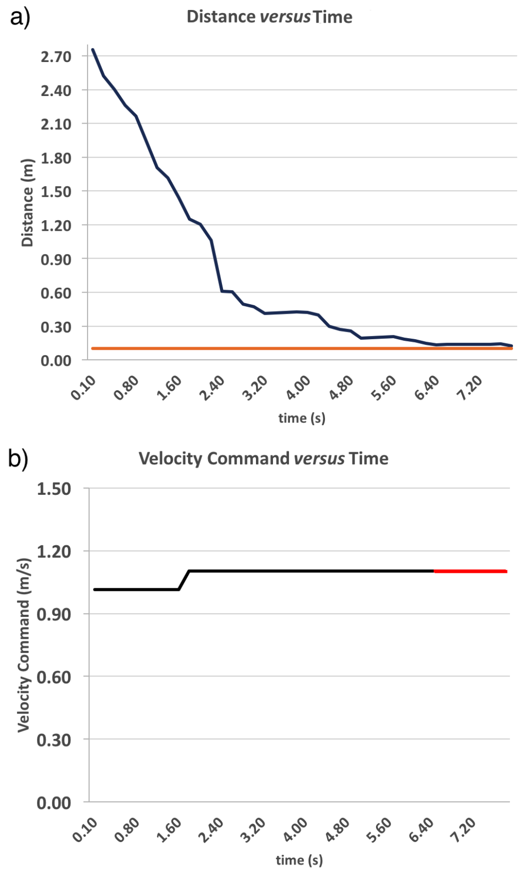

Figure 7: Experiment showing the system behavior when the quadrotor is towarding the obstacle.

magnitude of approximately $1.1m/s$ being applied to the quadrotor during the trajectory practically constant over the time. In red, it is possible to verify the period when the automatic control is enabled. This, fortunately, is our desired accomplishment.

The obtained results, even considering some imprecision from the quadrotor's sensors and sonars, match our expectations and can be embedded on the platform to complete the implementation.

5 Conclusion

In this paper, we presented an approach to avoid collision automatically assisting the manual control of a quadrotor. Our experiments in a simulated environment accomplished the objective being capable to avoid collisions in totally unknown environments even when the user's commands were contrary or unable to do so. When at risk, the inputs are overridden and an automatic control assumes for some time keeping the quadrotor in its last safe known position. As for future work, we are currently working at embedding our solution on the quadrotor, enabling it suitably to complete missions in a much safer manner.

6 ACKNOWLEDGMENTS

This research is partially funded by CAPES (Brazilian Research Agency) through Edital Pro-Estrategia N^o 050/2011 and Demanda Social and Petrobras SA.

References

Beard, R. W. and McLain, T. W. (2012). *Small unmanned aircraft: Theory and practice*, Princeton University Press, Princeton. isbn: 978-0691149219.

Becker, M., Sampaio, R. C. B., Bouabdallah, S., Perrot, V. d. and Siegwart, R. (2012). In-flight collision avoidance controller based only on OS4 embedded sensors, *Journal of the Brazilian Society of Mechanical Sciences and Engineering* **34**(3): 294–307.

Dijkshoorn, N. and Visser, A. (2011). Integrating sensor and motion models to localize an autonomous ar. drone, *International Journal of Micro Air Vehicles* **3**(4): 183–200.

Grzonka, S., Grisetti, G. and Burgard, W. (2012). A fully autonomous indoor quadrotor, *Robotics, IEEE Transactions on* **28**(1): 90–100.

Gupte, S., Mohandas, P. I. T. and Conrad, J. M. (2012). A survey of quadrotor unmanned aerial vehicles, *Southeastcon, 2012 Proceedings of IEEE*, IEEE, pp. 1–6.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C. and Burgard, W. (2013). Octomap: An efficient probabilistic 3D mapping framework based on octrees, *Autonomous Robots* **34**(3): 189–206.

Israelsen, J., Beall, M., Bareiss, D., Stuart, D., Keeney, E. and van den Berg, J. (2014). Automatic collision avoidance for manually tele-operated unmanned aerial vehicles, *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6638–6643.

Ito, K., Han, J. and Ohya, A. (2014). Localization using uniaxial laser rangefinder and IMU for MAV, *System Integration (SII), 2014 IEEE/SICE International Symposium on*, IEEE, pp. 712–717.

Kerns, A. J., Shepard, D. P., Bhatti, J. A. and Humphreys, T. E. (2014). Unmanned aircraft capture and control via GPS spoofing, *Journal of Field Robotics* **31**(4): 617–636.

Lugo, J. J. and Zell, A. (2014). Framework for autonomous on-board navigation with the AR. Drone, *Journal of Intelligent & Robotic Systems* **73**(1-4): 401–412.

Mendes, J. and Ventura, R. (2013). Assisted teleoperation of quadcopters using obstacle avoidance, *Journal of Automation Mobile Robotics and Intelligent Systems* **7**(1): 54–58.

Mian, A. A. and Daobo, W. (2008). Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter, *Chinese Journal of Aeronautics* **21**(3): 261–268.

Nicol, C., Macnab, C. and Ramirez-Serrano, A. (2008). Robust neural network control of a quadrotor helicopter, *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 1233–1237.

Roberts, A. and Tayebi, A. (2013). A new position regulation strategy for VTOL UAVs using IMU and GPS measurements, *Automatica* **49**(2): 434–440.

Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A. and Hebert, M. (2013). Learning monocular reactive UAV control in cluttered natural environments, *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, pp. 1765–1772.

Salih, A. L., Moghavvemi, M., Mohamed, H. A. and Gaeid, K. S. (2010). Flight PID controller design for a UAV quadrotor, *Scientific Research and Essays* **5**(23): 3660–3667.

Shen, S., Mulgaonkar, Y., Michael, N. and Kumar, V. (2013). Vision-based state estimation for autonomous rotorcraft MAVs in complex environments, *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, pp. 1758–1764.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic robotics*, MIT press. isbn: 978-0262201629.

Zhang, X., Li, X., Wang, K. and Lu, Y. (2014). A survey of modelling and identification of quadrotor robot, *Abstract and Applied Analysis*, Vol. 2014, Hindawi Publishing Corporation.