Vision-based Guidance and Navigation for Autonomous MAV in Indoor Environment

Mahammad Irfan
CSIR-Central Electronics Engineering Research Institute
Rajasthan, India 333031
mahammad.irfan1@gmail.com

Sagar Dalai
CSIR-Central Electronics Engineering Research Institute
Rajasthan, India 333031
mr.sagardalai@gmail.com

Kaushal Kishore
CSIR-Central Electronics Engineering Research Institute
Rajasthan, India 333031
kaushal@ceeri.res.in

Samarth Singh
CSIR-Central Electronics Engineering Research Institute
Rajasthan, India 333031
samarth@ceeri.res.in

S.A Akbar
CSIR-Central Electronics Engineering Research Institute
Rajasthan, India 333031
saakbar@ceeri.res.in

***Abstract*— The paper presents an autonomous vision-based guidance and mapping algorithm for navigation of drones in a GPS-denied environment. We propose a novel algorithm that accurately uses OpenCV ArUco markers as a reference for path detection and guidance using a stereo camera. It enables the drone to navigate and map an environment using vision-based path planning. Special attention has been given towards the robustness of guidance and controlling strategy, accuracy in the vehicle pose estimation and real-time operation. The proposed algorithm is evaluated in a 3D simulated environment using ROS and Gazebo. The results have been presented for drone navigation in a maze pattern indoor scenario. Evaluation of the given guidance system in the simulated environment suggests that the proposed system can be used for generating a 2D/3D occupancy grid map autonomously without the use of high-level algorithms and expensive sensors such as lidars.**

Keywords—ROS, Gazebo, Pose estimation, Quadcopter, Vision Guidance, SLAM.

I. INTRODUCTION

Micro Aerial Vehicles (MAVs) commonly known as drones playing an important role in achieving various tasks in both civilian and military areas because of their autonomy, high versatility, low cost, extreme agility, easy to deploy, and the capability for performing complex maneuvres in both structured and unstructured environment [1]. However, a lot of work has been progressed during the past recent years, research on their deployment and control efforts are still focused on achieving fully autonomous drones that can perform high-level missions particularly in GPS denied environment [2]. Estimating and mapping of the surrounding 3D environment is one of the key challenging tasks for fully autonomous MAVs to navigate safely and operate high-level tasks [3].

The proposed research work aims at developing an algorithm for vision-based path following and navigation approach in indoor application. In this paper, we have deployed an online ArUco based OpenCV tracker, which helps the MAV to autonomously track, detect and find its position with reference to the ArUco marker for the online navigation application. Whenever an ArUco tracker is detected, the controller flag is set to high which results in the position controller module to receive the sensor feedback data in order to generate (x- y) relative positions of the MAV with respect to the tracker (shown in Fig. 1). The reference value r which is relative distance to the marker of controller is set to zero. In order to achieve the proposed task, the localization relies on the drone vision-based technique using ArUco markers. Finally, the simultaneous localization and mapping (SLAM) based model using cartographer ROS has been implemented for mapping 2D/3D surrounding. The test is performed in a virtual environment created by gazebo and Rviz by placing different dynamic ArUco markers for different destinations on the map.

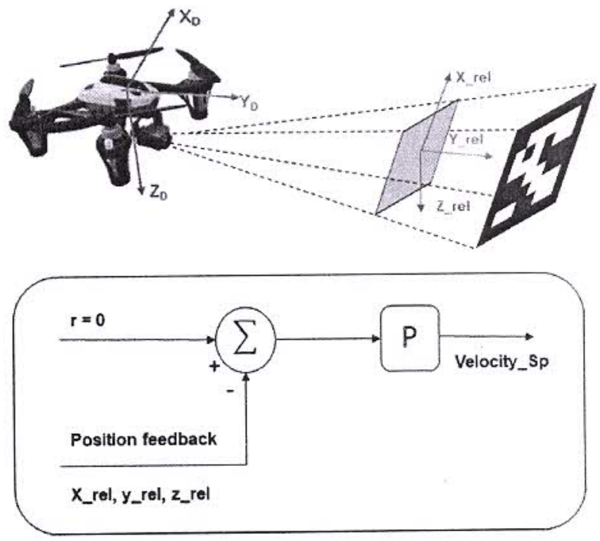

Figure 1. A vision-based navigation control law

This research aims to contribute to the emerging field of research of MAVs in a wide range of different applications by introducing a low-cost vision-based intelligent indoor pathfinding and mapping technique. The novelty of the developed algorithm relies on its simple yet accurate autonomous real-time online navigation and mapping with no prior information from external infrastructure.

II. RELATED WORK

The real-time vision-based navigation and mapping approach is the research of interest in many areas from augmented reality to intelligent robotics and autonomous aerial vehicle application. It collects the visual information from vehicles using path trackers like ArUco [4], [5]. The other type works without using such markers but considers the 3D surrounding information [5], [6]. There are many techniques that are already implemented to control the MAV attitude/altitude and position [7], which includes PID [8], LQR [9], H∞ [10], and model predictive [11] controllers. The ideal purpose of such a controller is to maintain a suitable accurate position of the vehicle, however, the position control is used to drive the drone to navigate a predefined indoor tracking path. To enhance the performance of the controller, vehicle pose feedback is an important parameter. GPS signals are widely used for outdoor in the robotics control tasks because inertial measurement unit (IMU) sensors have accumulated errors. Furthermore, in an indoor environment, GPS signals are undetectable many times, which leads to new research scope to the design of autonomous MAV controllers [9, 12].

The traditional monocular camera is used because of its low-cost solution to determine the vehicle pose estimation in GPS denied environment. As resulted in vision loss depth qualities as compared to binocular camera vision and some additional features required to estimate the vehicle pose [13,14,15]. Simultaneous localization and map (SLAM) algorithm use depth information for MAV pose estimation [16], however this method require a larger memory system and high computation resources. An OpenCV ArUco library initially developed for augmented reality but later it uses in different robotics applications [17]. It also used as intelligent markers for MAV pose estimation [18, 19]. Google cartographers are used for mapping purposes. SLAM provides autonomous vision-based guidance and mapping work [20].

In this paper, we present a high-level comprehensive vision-based position control algorithm in a complex indoor 3D environment. This results in robust autonomous navigation in GNSS denied environment. An iris camera equipped in the gazebo model-based quadcopter to observe ArUco trackers which are placed on the wall, and estimate the pose of the MAV. To improve the accuracy and eliminate the error of pose estimation, an intelligence PID controller [21] is implemented for position control of the vehicle. A 3D ROS based Gazebo simulation platform has been created to verify the proposed control algorithm. Different navigation trials have been performed and obtained simulated results are able to provide an accurate vision-based algorithm for real-time MAV pose calculation while mapping the surrounding seen at the same time.

III. VISION-BASED GUIDANCE ALGORITHM

The proposed research work consists of three main segments. Firstly, an OpenCV library ArUco marker-based technology for indoor navigation purposes. Second, a stochastic method to track and generate a path to a goal location, taking into account for safe navigation. This method is efficient because it only uses simple ArUco markers to find its position for autonomous pre-defined path tracking and does not require any high-level optimized method and sensors. Lastly, a computationally efficient algorithm for creating the map of its surrounding while navigating. All the above tasks can be achieved using a single RGB-D fixed non-tilting frontal camera interfaced with the onboard processor. The proposed vision-based guidance algorithm consists of several segments of ROS compatible packages namely, ArUco_ROS package and MAV pose estimation method to determine the MAV pose, design of intelligent PID controller for safe position control of the vehicle and SLAM based mapping algorithm using ROS_cartographer.

A. ArUco_ROS Trackers

For autonomous target detection, pose estimation, and path tracking purposes, we have used ArUco_ROS package from OpenCV library, which is commonly used in robotics, augmented reality and camera calibration applications [22].

The ArUco tracker consists of several binary bits having five bits in each row, while the second and fourth bits are known as information bits and the rest three bits act as error control bits. There are different types of ArUco markers as shown in Fig.2. Each ArUco markers has its unique ID and each marker ID's are recognized by the information present in bits. It consists of a total of 1024 different patterns. The proposed system uses 7x7 dictionary-based markers (7x7 bits) for better detection, resulting in reduced false-positive identification of the markers.

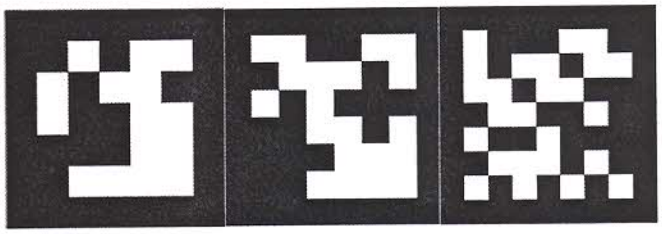

Figure 2. ArUco square markers of 5x5, 6x6 and 7x7 dictionary size respectively.

B. MAV pose estimation

The vehicle pose can be estimated by using any calibrated camera which calculate minimum of four co-planer and non-co-linear elements by the obtained image. Let us assume that *P (A, B, C)* be the location of the camera in space. The camera coordinates can be projected with 3D point in a condition such that translation, (t) and rotation, (R) values should be known already. Given coordinate location of the point *P(X, Y, Z)* by the r200 camera can be determine using the obtained eqn. (1) and (2)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} A \\ B \\ C \end{bmatrix} + t \quad \text{.................................} (1)$$

After expanding equation (1), we will get

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ 1 \end{bmatrix} \quad \text{..................} (2)$$

Now, the value of real world coordinates (A B C) are obtained and points (X Y Z) are unknown. The (X, Y) point can be calculated from eqn. (2).

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \text{..........................} (3)$$

From the above obtained equation (3), f_x and f_y are known as focal length of drone camera in x and y direction respectively, whereas, 's' considered as unknown factor and c_x and c_y are known as optical centers of drone camera.

Whenever an image from 3D camera is projected towards a 2D image plane, it results loss of depth information, therefore scale factor 's' can be obtain from X, Y, Z image. After getting rotation and translational values, it is easy to determine the camera projected point as well as the image points (X, Y) in a camera matrix. Rotation and translation are arbitrarily selected and the obtained image point will be linked to projected X-Y.

C. Intelligent controller design

In this section, a vision-based hybrid intelligent controller is designed and developed to provide accurate position control mechanism for MAV. The PID controller take difference between pose feedback information and generated reference trajectory to calculate the vehicles 3D velocity.

A set of PID controllers were implemented to control the vehicle position by guiding thrust and attitude set point to the flight controller. The controller system consists of three parameters, taken as constant Kp, Ki, and Kd which are being used respectively to tune the vehicle proportional, integral and differential unit respectively. Output Control equation in continuous form (3) and in discrete form (4) and it is structured in Fig. 3.

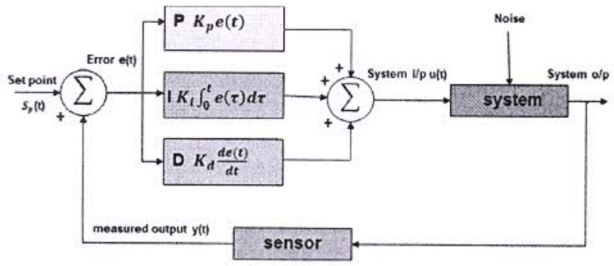

Figure 3. Block diagram for PID controller

PID is the most widely used feedback gain especially in aerial robotics guidance and control application. The landing phase of MAV is controlled by PID controller, with desired acceleration, the control output $u(t)$ is defined as:

$$u(t) = K_p e(t) + K_i \int_o^t e(t)dt + K_d \frac{de(t)}{dt} e(t) \quad \text{..........} (3)$$

$$u(s) = (K_p + \frac{1}{s} K_i + K_d \frac{N}{1+N\frac{1}{s}}) \, E(s) \quad \text{....................} (4)$$

D. SLAM Mapping using cartographer

In order to fly autonomously in an unknown environment, a drone must localize itself and understand it's surrounding, this can be possible using sensors such as depth LIDAR's and camera etc. In this proposed research, sensors like RGB-D camera and IMU (inertial measurement unit) are used. By using generated ArUco pose and cartographer mapping algorithm the MAV can locate itself inside the map. SLAM is based on extended kalman filter algorithm which helps quadcopter to build a map, estimate the pose and localize itself simultaneously [16]. Cartographer is a technology that provides real-time SLAM in 2D and 3D across multiple platform and sensor configuration [20].

IV. EXPERIMENT

A. Software in the loop simulation (SITL) setup

ROS (Robot Operating System) is a Linux based open source robotics platform consisting of a set of tools and software libraries that help robotics researchers to develop and build robotic applications [23]. ROS can be used with px4 and the Gazebo simulator [24]. SITL is useful to check new mission performance or control algorithms before actually flying the MAV and possibly damage it.

In order to communicate with flight controller PX4 autopilot, MAVROS MAVLink protocol nodes are used [25]. The ROS and Gazebo integrate with autopilot to communicate with the drone platform and receive camera output, IMU, barometer, magnetometer and GPS sensor data with noise from the simulated world and sends this data to generate motor/actuator thrust command values which are sent back to Gazebo physics simulator. This can also communicate with Ground Control Station and Offboard API to guide telemetry data with simulated gazebo and accept commands. The comprehensive configuration model of proposed SITL is shown in Fig. 4.

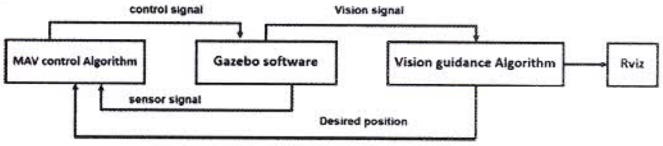

Figure 4. SITL configuration for MAV navigation and visualization

The results provided by this paper were generated using SITL simulations. The simulation model used here is an iris-based quadcopter equipped with an IMU sensor for six degrees of freedom position estimation, a barometer for altitude control and a monocular RGBD camera Intel realSense r200 for SLAM based mapping. The resulted image topics published by ROS node is processed using ArUco libraries for target path tracking. The propose testing environment is shown in the Fig. 5.

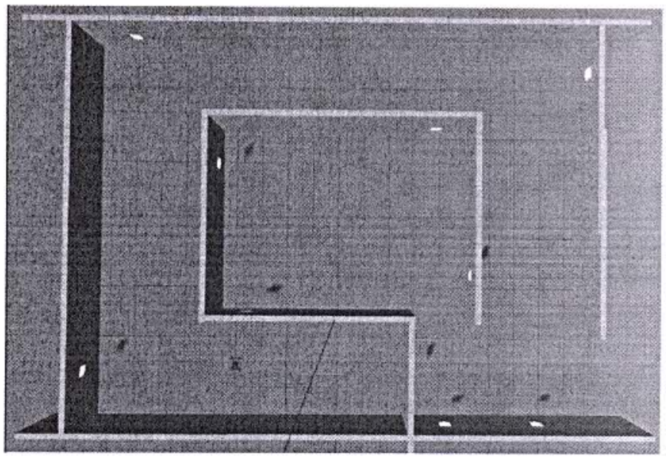

Figure 5. Testing environment

B. Implementation

This following section explains how the Algorithm 1 works. ROS based Gazebo environment has been created, models such as ArUco trackers, IRIS drone equipped with real sense r200 camera is imported into the simulation platform. ROS provides inter process communication mechanism, each running process in ROS is called as a node. Every node can able to communicate with other nodes by publishing/subscribing message passing scheme. This allows MAV software which composed of independent programs to talk each other.

The quadcopter flies using its fixed non-tilting frontal camera, searching navigation path identification only by the intelligence ArUco path trackers placed inside the indoor environment. When visual tracker is identified, the vision based algorithm and image processing library computes the 6-DOF relative pose between the marker itself and the MAV. The obtained 3D x y z pose is used to help the drone approach towards trackers with the right orientation. An intelligence PID controller is designed to reduce the error on x-, y- and z-axis along with yaw angle. The drone continuously fly and keep tracking the marker provided at a given threshold value (distance of 50 cm and an angle of ± 5 degree). Every tracker is coded with some pre-defined task (perform the next movement) done by the MAV, as it track the ArUco and reaches near at a fixed distance, MAV read the coded information and perform the given task to explore the pre-defined path created in indoor environment by controlling the drone position. While tracking the ArUco, MAV create a SLAM based map of the surrounding environment in Rviz.

Algorithm 1 : Proposed Algorithm

```
controller.command(takeoff)
landed = False
start_SLAM_Mapping()
while not landed do :
       MARKER_POSITION = None
       if MarkerVisible then
          MARKER_CODE = detected_maker_code
          MARKER_POSITION = detected_maker_pos
          if  MARKER_POSITION > thresold_distance_angle_xy then
              controller.command(move_xy)
          else if MARKER_CODE == landing_code  then
              next_move = Predefine_Next_move(detected_maker_code)
              controller.command(next_move)
              controller.command(land)
              landed = True
              stop_SLAM_Mapping()
          else
              next_move = Next_move(detected_maker_code)
              controller.command(next_move)
          end if
       else
          controller.command(yaw_move)
       end if
end while
```

C. Simulation and results

In this section, ROS based gazebo simulation experiment has been performed to validate the efficacy of each proposed research work and the developed SITL configuration which is constructed in Fig.4. The simulation model used here is an iris based MAV equipped with r200 camera as shown in the Fig.6 which describes an ArUco marker is detected by an iris quadcopter and its 3D pose is obtained by using published data from camera sensor node in order to track the desired position.

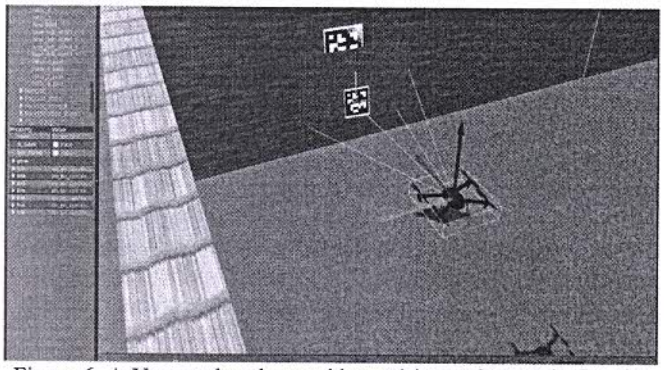

Figure 6. ArUco marker detected by an iris quadcopter in Gazebo simulation

The results as reported in Fig. 7. Showing 2D and 3D trajectories of MAV during a complete experiment performed in ROS based Gazebo simulation by implementing proposed vision Algorithm 1. The obtained result explain that the quadcopter recognize ArUco markers accurately and navigate in a 3D maze scenario from its initial take-off (position marked as green) to the final desired location (marked as red colour).

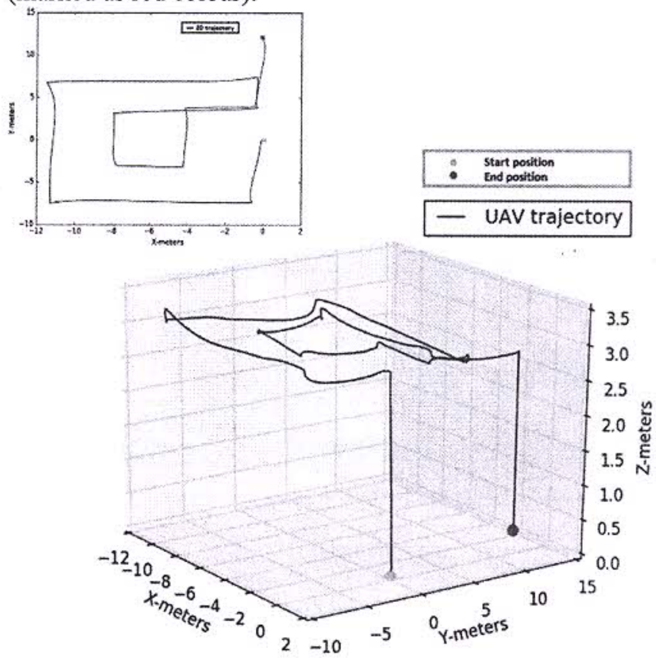

Figure 7. 2D and 3D MAV trajectory

The Fig.8, shows the initial generation of the map using SLAM algorithm in Rviz, when launched. The vehicle moved in a pre-defined path coded in the ArUco trackers and move until a full map is created using the "auto_nav_indoor_pkg" package created in ROS platform, where the drone is controlled fully autonomously

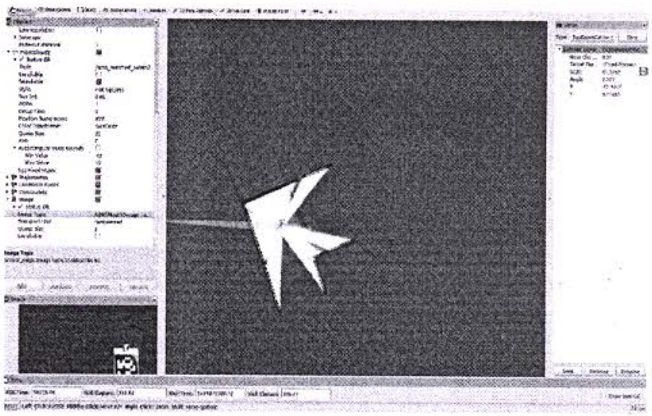

Figure 8. Initial scan of the map in Rviz

The Fig 9. illustrate the final constructed octo map in the Rviz after MAV successfully navigates through the testing environment shown in Fig. 5 and reaches at the target destination, similar to the created environment in the Gazebo. The point cloud information collected from R200 camera which further processed to generate 3D octomap of the indoor environment.

Figure 9. Scan octo map of the indoor environment

V. CONCLUSION

We have designed and developed an autonomous MAV online vision based path navigation algorithm using ArUco tracker- a visual markers for self-localization technology in indoor scenario and mapping of 2D/3D surrounding environment which is based on SLAM and cartographer technology. We have used several open source known technologies by adapting these resources in our purposed research work the system has been successfully validated on ROS based Gazebo simulator as well as Rviz and results were obtained. An intelligent PID controller has been integrated and the effectiveness of controller performance shown through experimented simulation results.

Future lines involve the integration into real hardware platform as well as implementation with other vision based auto-localization algorithms, control approaches and replacing them with other sensors without using the marker coded information at all. Furthermore, sensor data fusion could be implemented in order to get exhaustive position control of MAV.

ACKNOWLEDGMENT

The authors would like to thank the Department of Science and Technology (DST) Govt. of India as part of Indo-Korea Joint Network Centre on Robotics (IKJNCR) for its financial support (Grant Number HRD/171030/2018, IIT Delhi) and CSIR-Central Electronics Engineering Research Institute, India (GAP-3133).

REFERENCES

[1] H. Shakhatreh et al., "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," in IEEE Access, vol. 7, pp. 48572-48634, 2019.

[2] R. Shakeri et al., "Design Challenges of Multi-UAV Systems in Cyber-Physical Applications: A Comprehensive Survey and Future Directions," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3340-3385, Fourthquarter 2019.

[3] Yousif, K., Bab-Hadiashar, A. & Hoseinnezhad, R. An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intell Ind Syst* **1,** 289–311 (2015).

[4] S. Garrido-Jurado, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 4, no. 6, pp. 2280–2298, 2014.

[5] E. Marchand, H. Uchiyama, F. Spindler, E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality : a hands-on survey," IEEE Trans. on Visualization & Computer Graphics, vol. 1, 2016.

[6] "Visual odometry: Part II: Matching, robustness, optimization, and applications," IEEE Robot. Autom. Mag., vol. 19, no. 2, pp. 78–90, 2012.

[7] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," Journal of Field Robotics, vol. 29, no. 2, pp. 315–378, 2012.

[8] A. Salih, M. Moghavvemi, H. Mohamed, and K. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," IEEE International Conference on Automation Quality and Testing Robotics, 2010.

[9] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," IEEE Control Systems Magazine, vol. 28, no. 2, pp. 51–64, 2008.G. Raffo, M. Ortega, and F. Rubio, "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter," Automatica, vol. 46, no. 1, pp. 29–39, 2010.

[10] G. Raffo, M. Ortega, and F. Rubio, "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter," Automatica, vol. 46, no. 1, pp. 29–39, 2010.

[11] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Model predictive quadrotor control: Attitude, altitude and position experimental studies," IET Control Theory and Applications, vol. 6, no. 12, pp. 1812–1827, 2012.

[12] S. Ahrens, D. Levine, G. Andrews, and J. How, "Visionbased guidance and control of a hovering vehicle in unknown, GPS-denied environments," IEEE International Conference on Robotics and Automation, 2009.

[13] E. Royer, M. Lhuillier, M. Dhome, and J. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," International Journal of Computer Vision, vol. 74, no. 3, pp. 237–260, 2007.

[14] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, "A Robust visual odometry and precipice detection system using consumer-grade monocular vision," IEEE International Conference on Robotics and Automation, 2006

[15] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for MAVs using one single circular landmark," Journal of Intelligent and Robotic Systems, vol. 61, no. 1–4, pp. 495–512, 2011.

[16] A. Davison, I. D. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

[17] S. Garrido-Jurado and F. Madrid-Cuevas, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 47, no. 6, pp. 2280–2292, 2014.

[18] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," International Journal of Computer Vision, vol. 109, no. 3, pp. 252–286, 2014.

[19] T. Krajnik, M. Nitsche, J. Faigl, P. Vanek, M. Saska, and L. Preucil, "A practical multirobot localization system," Journal of Intelligent and Robotic Systems, vol. 76, no. 3–4, pp. 539– 562, 2014.

[20] Google cartographer using ROS : https://google-cartographer-ros.readthedocs.io/en/latest/

[21] G. Welch and G. Bishop, An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, 2001.

[22] ArUco ROS wrapper: https://github.com/pal-robotics/aruco_ros

[23] Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software, 2009, p. 5.

[24] 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.

[25] PX4 Pro Drone Autopilot, https://github.com/PX4/Firmware, last accessed 2017/06/06.