Exploring AprilTag use with Quadrotors

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Prithvi Romil Kinariwala

Spring, 2020.

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Nicola Bezzo, Department of Engineering Systems and Environment & Department of Electrical and Computer Engineering

Introduction

To fulfill the technical requirement of the Bachelor of Science of Computer Science thesis, I was on Professor Nicola Bezzo's Autonomous Mobile Robot Lab research team as an undergraduate researcher. As the sole undergraduate with a lab of graduate students, my work consisted of an educational experience. This paper outlines the semester of my understanding of the technologies in the lab. This paper begins with an introduction to quadrotors followed by my exploration of the use of AprilTags in quadrotors adapting the use of Robot Operating System (ROS).

Professor Bezzo's lab focuses on technologies to "increase resilience, autonomy and assured safety" of robotic and cyber-physical systems (Bezzo, n.d.). The NSF defines cyber-physical systems to "integrate sensing, computation, control, and networking into physical objects and infrastructure." In other words, such systems connect the gap between existing solo-standing robotic systems to greater capabilities, often with the incorporation of greater algorithms and software. To that extent, this paper first covers the physical attributes of quadrotors, and then the added functionality of ROS's apriltag_ros package.

Quadrotors Basics

Quadrotors are the four-rotor variant of multirotor aircraft. Niemiec & Gandhi (2016) claim multirotor aircraft—compared to single-rotor aircraft—use rotors "in lieu of a large main rotor controlled with collective and cyclic pitch inputs." This added control is made due to the use of "multiple fixed pitch, variable RPM propellers" to produce controlled thrust. Due to greater maneuverability afforded by multiple rotors, multirotor aircraft have become subject to greater research and hobbyist interest.

According to Niemiec & Gandhi (2016), quadrotors are the simplest variant of multirotor aircraft. Simple quadrotors can be distinguished by "four rotors connected to the fuselage via booms, generally arranged in a square pattern." Quadcopter consist of two sets of counter-rotating rotors arranged such that adjacent rotors spin in opposite directions. Counter rotating rotors minimize the moment of inertia generated by the other. There are two ways quadrotors can be flown, each way having a characteristic manner of which rotors lead the aircraft. Figure 1, adapted from Niemiec & Gandhi, illustrates the two configurations of quadrotors. The "plus" configuration is characterized by a single rotor leading the aircraft whereas the "cross" configuration can be characterized by two rotors leading the aircraft.

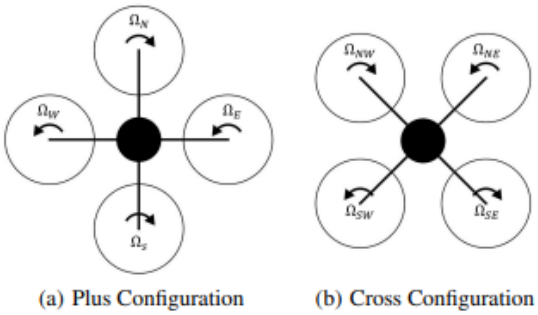

Figure 1. Quadrotor Flight Configurations. Adapted from Niemiec & Gandhi (2016).

According to G, J. & J.P. (2016), quadrotor orientation can be determined by a combination of roll angle (ϕ), pitch angle (θ), and yaw angle (Ψ). Figure 2, adapted from the same source, illustrates these angles on a quadrotor. Independent thrusts from each rotor create the toral thrust force and three torques determined by the respective angles: roll torque, pitch torque, and yaw torque. Quadrotors are controlled by a combination of these torques on each rotor. Figure 3 illustrates common quadrotor maneuvers by the rotation of each rotor.

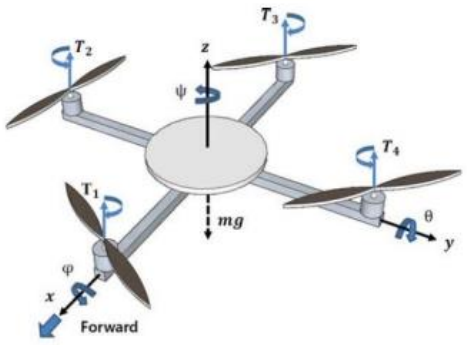

Figure 2. Free Body Diagram of a Quadrotor. Adapted from G, J. & J.P. (2016).

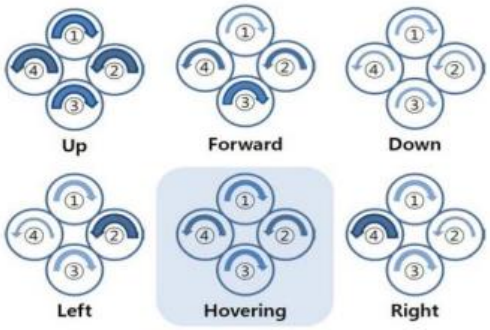

Figure 3. Quadrotor Rotor Actions for Common Maneuvers. Adapted from G, J. & J.P. (2016).

The scope of quadrotor projects for the semester was to the extent of basic operation. The bulk of the summer's work consisted of exploring the use of Robot Operating System and AprilTag technologies with basic quadrotor technologies.

Robot Operating System (ROS) and the AprilTag Fiduciary System

Introduction to Robot Operating System (ROS)

As per its main documentation, ROS is an open-source middleware that aims to help build better robot applications. ROS consists of drivers, state-of-the-art algorithms, and developer tools. ROS itself is not an operating system for robots, but instead is a collection of software frameworks for software development. ROS is developed exclusively for Linux and OSX operating systems. High-level ROS processes are represented as nodes in a graph. In this "graph," edges are denoted to be topics. In common applications, nodes pass messages to one

another through topics. ROS Master, a high-level process within this graph registers nodes, creates node-to-node communication for topics and controls parameter server updates.

Additionally, ROS comes with an array of tools to allow developers to visualize and record robotic data. Some examples include rviz, a three-dimensional visualizer used to visualize robots and their respective environments; rosbag, a tool used to record and playback message data; catkin, a ROS build system; rosbash, a ROS equivalent of a bash shell; and finally, roslaunch, a tool used to launch ROS nodes locally and remotely.

Introduction to the AprilTag Fiduciary System

The AprilTag fiduciary system is a system of QR code-like tags that can are detected by robots and autonomous systems. Maintained by the APRIL Robotics Lab at The University of Michigan, AprilTags can be utilized for various tasks, including but not limited to augmented reality, robotics, and camera calibration. One of its specializations, giving aid to the calculating angle of attack, allows for greater incorporation into robotics applications. Figure 4 depicts six sample AprilTags taken from the AprilTags documentation hosted by the University of Michigan. AprilTags can be attached to surfaces or even other autonomous robots.

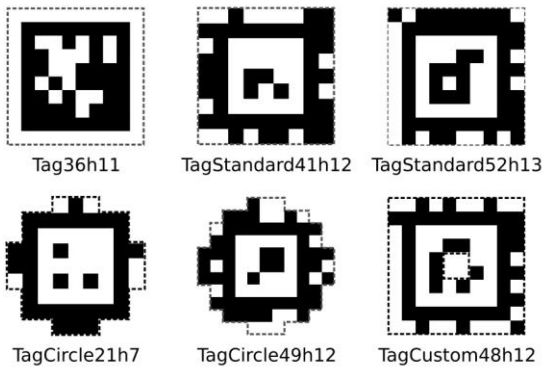

Figure 4: Collection of six AprilTags. Adapted from University of Michigan (n.d.).

ROS utilizes a collection of packages giving additional functionality to existing ROS capabilities. Similar to libraries created for Python, these packages—also referred to as "wrappers"—bridge external capabilities with core ROS features. Pertinent to this research experience, apriltag_ros is a ROS package providing access to core AprilTag 3 algorithms for visual detection. As cited in the package's documentation, the package allows for the detection of not only tags but also tag bundles.

apriltag_ros has steps to initialize written on the website. After following the tutorials on the documentation to set up the package, basic detection can occur. However, to maximize the extent of accuracy and precision when detecting the robot's view of an AprilTag, numerous tags are assembled in specific formations to form a "bundle" stored in a ROS datatype called rosbag. Figure 5 illustrates an example of a bundle of AprilTags. After this, the video stream can be inputted using the appropriate setups from the detector, which is later run. These steps are outlined on the package's ROS Wiki Page. These tags could then be scanned with AprilTag's AprilTag3 iPhone application. After completing the steps for the ROS package, the information could then be run atop a quadrotor operating the ROS wrapper. This merged the functionalities of AprilTag, ROS, and finally basic quadrotors.

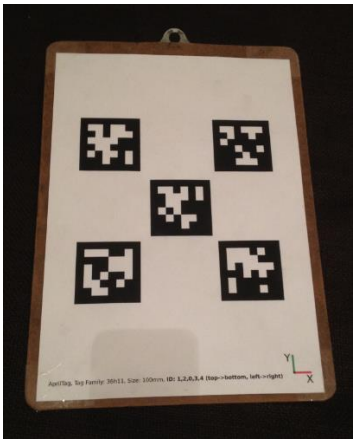

Figure 5. Sample AprilTag Bundle. Adapted from ROS Wiki (n.d.).

Conclusion

This research experience exposed me to graduate-level research while still an undergraduate. My time as an undergraduate was encapsulated with new technologies as well as hands-on experience with lab members working on projects of their own. The educational experience also encompassed learning quadrotor trajectory planning, three-dimensional reconstruction, and autonomous operation of industry-grade drones. As I plan on perusing my Master's in Computer Science at the University of Virginia, I will continue researching quadrotors.

Works Cited

Bezzo, N. (n.d.). *Research*. UVA Bezzo Robotics. Retrieved May 9, 2022, from https://www.bezzorobotics.com/research

G, J., & R, J. P. (2016). Quadrotor modelling and Control. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. https://doi.org/10.1109/iceeot.2016.7754868

National Science Foundation. Cyber-Physical Systems | National Science Foundation. (n.d.). Retrieved May 8, 2022, from https://www.nsf.gov/news/special_reports/cyber-physical/

Niemiec, R., & Gandhi, F. (2016). A comparison between quadrotor flight configurations.

Open Robotics. (n.d.). *apriltag_ros*. ROS Wiki. Retrieved May 9, 2022, from http://wiki.ros.org/apriltag_ros

Robot operating system. ROS. (n.d.). Retrieved May 8, 2022, from https://www.ros.org/

University of Michigan. (n.d.). *AprilTag*. April Robotics Laboratory. Retrieved May 1, 2022, from https://april.eecs.umich.edu/software/apriltag#:~:text=AprilTag%20is%20a%20visual%20fiducial,tags%20relative%20to%20the%20camera.