Control Writing Document

As the field of robotics and its impact on society grows, there will be increased need for simplified and easily accessible controls simulation. One type of robotic system, the quadcopter, has grown in popularity due to its simplicity, easy testing procedure, and lack of external requirements [1]. The quadcopter platform can be utilized in a variety of applications including: education [1]–[5], advertisement, tourism, cinematography and photography, military and defense, lighting, search and rescue, and recreational drone flight are all widely utilized [6].

Of course, simulations already exist for a variety of quadcopter types. Coming in nearly every format, these simulations can be found on GitHub among other software-sharing platforms in software packages utilizing ROS, Python, and MATLAB. Unfortunately, the programs most widely used like ROS and Python typically require large amounts of user knowledge to install programs, source code, and all associated dependencies. For example, Open-Source Drone Programming Course for Distance Engineering Education [3] requires installation of ROS (Robot Operating System), Python, with an unlisted number of libraries. While it is an impressive compilation of educational material, it lacks the simplicity and clear instructions required for quick implementation. This paper will pose as a resource for quick simulation with no associated blackbox.

There is a need for simplified, and easily changed control software. Many students, instructors, hobbyists, and even companies are interested in simulations relating to physical systems. While many papers [2], [5] attempt to incorporate drone control into the classroom, these papers often require complex and difficult to acquire software. Control theory is a complex topic, and many students do not learn the tools needed to understand control theory until the university level. As such, this paper and its associated work will focus on implementation at both the university, hobbyist, and company level. Dyanmics, control theory, associated control laws, and implementation of this within an easily defined and publicly available software will be provided. Additional information and guides will be provided for those who do not have access to software utilized.

Outline of paper

1. Need for simplified drone control architecture
2. Methodology and the theory behind the work
   1. Newton’s laws for XYZ
   2. Newton’s laws for orientation
   3. PID control
   4. PID tuning
3. Description of how to modify, and utilize this system

Tuning PID controllers in a system is often tedious and can be both dangerous and difficult in many situations. With robotic arms cable of moving entire automobiles smooth and stable control is very important.

Drones, specifically quadcopters, have seen increasing use in our society. Having high maneuverability coupled with relatively inexpensive actuation, quadcopters are becoming more and more utilized. Controlling these quadcopters often requires fast and precise tuning values to accurately control the system. When controlling attitude of quadcopters, the difference between an unstable and fully controllable system is often minute.

While tuning PID controllers knowing where to start can be a huge benefit. Fortunately, MATLAB has a built in toolbox which offers PID tuning. Utilizing MATLAB’s toolbox is quite simple, but it requires a fully functional simulation of free body dynamics. While there exists extensive research in strategies for quadcopter control, there exists very little research into tuning a quadcopter based on simulation results. This paper and its associated work will address that goal.

Before addressing the nested control strategy used in this paper, a proper understanding of the dynamics governing motion must be understood and realized. Research into these dynamics and the associated assumptions can be found in the following sources: \_\_\_\_\_\_. The reader is encouraged to pay close attention to source \_\_\_ regarding

Simulations should be performed with digital control, but utilize continuous free body dynamics.

Statregies for tuing PID controllers include, the Ziegler-Nichols method, Guess and check PID controlling, and even

[1] B. Michini, J. Redding, N. Kemal Ure, M. Cutler, and J. P. How, “Design and flight testing of an autonomous variable-pitch quadrotor,” in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 2978–2979. doi: 10.1109/ICRA.2011.5980561.

[2] M. F. A. Rahman, A. I. C. Ani, S. Z. Yahaya, Z. Hussain, R. Boudville, and A. Ahmad, “Implementation of quadcopter as a teaching tool to enhance engineering courses,” in *2016 IEEE 8th International Conference on Engineering Education (ICEED)*, Dec. 2016, pp. 32–37. doi: 10.1109/ICEED.2016.7856089.

[3] J. M. Cañas *et al.*, “Open-Source Drone Programming Course for Distance Engineering Education,” *Electronics*, vol. 9, no. 12, Art. no. 12, Dec. 2020, doi: 10.3390/electronics9122163.

[4] U. Veyna, S. Garcia-Nieto, R. Simarro, and J. V. Salcedo, “Quadcopters Testing Platform for Educational Environments,” *Sensors*, vol. 21, no. 12, Art. no. 12, Jan. 2021, doi: 10.3390/s21124134.

[5] P. N. Beuchat, Y. R. Stürz, and J. Lygeros, “A Teaching System for Hands-on Quadcopter Control,” *IFAC-Pap.*, vol. 52, no. 9, pp. 36–41, Jan. 2019, doi: 10.1016/j.ifacol.2019.08.120.

[6] G. Ostojic, S. STankovski, B. Tejic, N. Dukic, and S. Tegeltija, “Design, Control and Application of Quadcopter,” *Int. J. Ind. Eng. Manag.*, vol. 6, no. 1, pp. 43–48, Feb. 2015.