

Design of Multicopter Test Bench

Srikanth Govindarajan, Tarun Agarwal, Sai Kishan R., C. S. Suraj, G. Ramesh, and Veena Devi

Abstract—In this paper we have designed and implemented a complete system of quad rotor stability mechanism. Starting with a Single-axis Implementation of a Quad rotor, we proceeded to create a Quad rotor system inclusive of Quad rotor model, a mathematically modeled and simulated MatLab module along with a LabView graphics interface userface. The purpose of this project is to design and implement a Quad Rotor stability control system that is capable of better stability augmentation and control authority for different load capacities. The magnitude of throttle applied and the corresponding shift in the position of the Quad Rotor from equilibrium must be accurately obtained for better control authority. The project consists of the development of four main areas, the first being the on-board measurement systems including gyroscopes and Accelerometers, the communication systems, the control of the individual rotors and the main processor which pull all these together. The project also required the development of a method of allowing the user to communicate to the Quad Rotor which will be achieved using LabView. In addition to these, a software implementation of the mathematical model of the system will be designed using Matlab. Finally, the stability of the Quad Rotor will be achieved using a customized PID software code.

Index Terms—Graphic user interface, mathematical model, quad rotor, simulation, stability.

I. INTRODUCTION

This paper is about modeling, design and control of Miniature Flying Robots (MFR) with a focus on stability of the system specifically, micro quadrotors. It introduces a mathematical model for simulation and control of such systems. Based on the mathematical model, linear and nonlinear control techniques are used to design and simulate various controllers along this work. The dynamic model and the simulator evolved from a simple set of equations, valid only for hovering, to a complex mathematical model with more realistic aerodynamic coefficients and sensor and actuator models [1]. Control of vehicle motion is achieved by altering the pitch and rotation rate of one or more rotor discs, thereby changing its torque load and thrust characteristics.

Our experiment was intended to optimize the stability of a quadrotor. We have used Proportional-Integral-Derivative (PID) control algorithm to stabilize the three basic parameters of aerodynamics namely, pitch, roll and yaw. As pitch and roll have similar traits, we implemented a single-axis version of a quadrotor. Further, with pitch and roll values calibrated, we extended the model to a 3-D version to incorporate yaw.

A mathematical model similar to our hardware was implemented on MatLab [2]. In addition to this, a GUI using

LabView was developed. This methodology can be extended to multi-copters.

II. SINGLE AXIS IMPLEMENTATION

Methodology: Our model is a single axis realization of a Quadcopter. The central housing was made of High Density Reinforced Plastic (RPT) to house the APM2, Xbee Module and the receiver. This was affixed onto the wooden framework with the help of ball bearings and aluminum rods. RPT motor holders were press fitted onto the carbon fiber rods[3]. With the help of APM Planner software, we were able to get the run time values of Altitude, Pitch, roll, Yaw, PID values, Throttle as well as the GPS data.

Specification:

- 1) Arm Length – 23 cms
- 2) Central Housing – Reinforced Plastics
- 3) ESC – 20Amps
- 4) Xbee – ISM 2.4 Ghz

Testing: Thrust calculation for each motor was done with the help of ‘Medussa’. [Table I]. It was observed that, at 50% throttle sufficient thrust was generated for the Quadcopter to liftoff. Next, we changed the run time values of P, I and D with the help of the APM Planner. Using iterative algorithm and Ziegler Nichols Method, we altered the values of P, I and D, one after the other and tabulated the same. In each case, we kept 2 parameters constant and varied the 3rd. Thus, we obtained a range of values, for which our model was stable.



Fig. 1. Single axis implementation

TABLE I: MEDUSSA THRUST CALCULATION

Throttle (%)	Volts (V)	Amps (A)	RPM	Thrust (g)
40	12.37	1.17	4900	131
45	12.36	1.48	5350	154
50	12.35	1.8	5700	180
55	12.34	2.1	6025	204
60	12.32	2.46	6300	228
65	12.32	2.78	6550	249
70	12.3	3.42	6975	289

Manuscript received April 20, 2013; revised June 21, 2013.

Srikanth Govindarajan is with Rashtreeya Vidyalyaya College of Engineering, India (e-mail: srikanthjg123@gmail.com).

Observation: The experiment yielded the optimum stability for the propellers we used.

Input type: Step Input

- For 10x4.7" Prop:
P =0.231 , I =1.0 , D =0.015
- For 8x3.8" Prop:
P =0.37, I =0.7, D =0.015



Fig. 2. Response of 10x4.7"

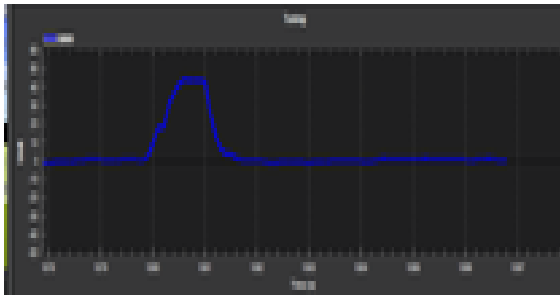


Fig. 3. Response of 8x3.8"

The results obtained from our experiment are suitable only for pitch or roll axis individually. The 'Yaw Axis' is not considered. Hence, we plan to extend our model to a 3-D system to find out the PID values for each axis (Pitch, Yaw and Roll).

TABLE II: PID INFERENCE

Response	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	Decreases	Increases	-	Decreases
K_i	Decreases	Increases	Increases	Eliminate
K_d	-	Decreases	Decreases	-

We intend to incorporate a "Ball And Socket" joint in the RPT Central Housing for a free 3d rotation. Our model could be used to determine the PID values for Different Lengths, Propeller Size and different Motors; and we can also arrive at a formula using 'Regression or ANN Algorithms'.

III. HARDWARE

Our model is mounted on a ball socket joint tripod stand. This allows it to have 3 degrees freedom along the pitch, roll and yaw axis.

Specifiacation:

- 1) Arm Length : 23cm
- 2) Propeller size : 10x4.5' and 8x3.8'
- 3) ESC: 20A



Fig. 4. Hardware model

IV. RC PRECISION CONTROL

LabView is a highly productive development environment used for graphical programming and unprecedented hardware integration to rapidly design and deploy measurement and control systems. By abstracting low-level complexity and integrating the tools needed to build any measurement or control system, the graphical system design software of LabView provides a platform to expedite processes and reach results faster.

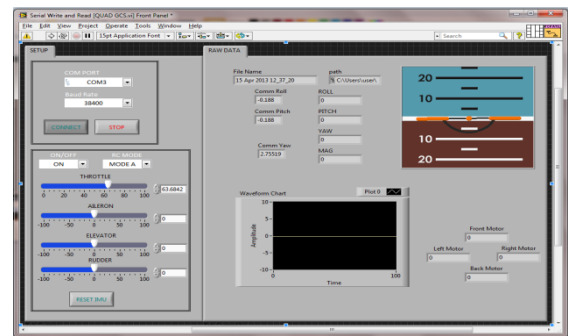


Fig. 5. LabView GUI

The LabView Graphical User Interface (GUI) controls the Quadcopter in run-time. It also has a graphical attitude indicator. LabView is preferred over RC remote Control for its precision in the command output. The interface is used to acquire the RC command data and the IMU data which in turn is sent to the Control Mixer. The revised motor PWM values are sent to the ESCs. These values are also fed back to correct the error in the system which will be displayed in runtime as motor values in the interface.

V. DESIGN METHODOLOGY

The design process starts by setting three design constraints from the application definition: This gives a good idea about the propeller diameter. In practice, the propeller span defines the overall span of the helicopter. Using the propeller diameter, one can estimate the characteristics of the propeller in term of thrust, drag and power for a range of angular speeds [4]. So, the mass, the drag moment and the thrust/weight ratio are enough to fully define the motor power requirements. This allows the algorithm to select from the database a list of candidate actuators which offer the required power.

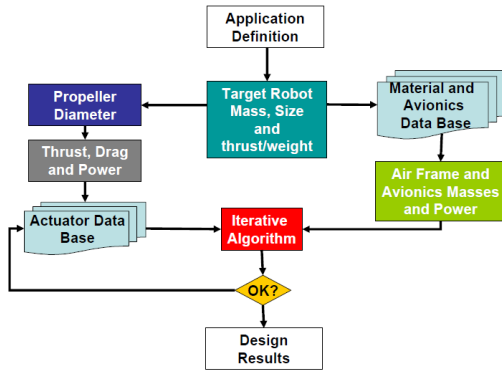


Fig. 6. Methodology flow

VI. MATHEMATICAL MODELING

Modeling the rotor dynamics: This step is carried out, investigating the step responses of the Simulink vehicle model [5]. Because of aerodynamics properties, the rotors are far from being linear. For example, a positive step applied on the four voltages doesn't give rise to the same response as a negative one.

Thus, it is firstly necessary to find out the four voltage combinations which are going to be used to controlling the vehicle's motion in order to model, then, the relations between these voltage combinations and the famous variables $u1, u2, u3, u4$.

Using the model's notation, the quadrotor is controlled by:

- Vertical thrust (sum of the four thrusts):

$$u1 = T1 + T2 + T3 + T4$$
- Rolling moment(thrust difference):

$$u2 = l(T4 - T2)$$
- Pitching moment(thrust difference):

$$u3 = l(T1 - T3)$$
- Yawing moment(algebraic sum of four torques):

$$u4 = Q1 + Q2 + Q3 + Q4$$

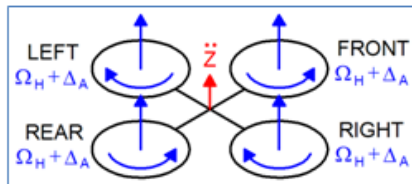
Thus the voltage combinations are:

- Vertical thrust(z axis): $V1 + V2 + V3 + V4$
- Rolling moment(y axis): $V4 - V2$
- Pitching moment(x axis): $V1 - V3$
- Yawing moment(control of psi): $V1 - V2 + V3 - V4$

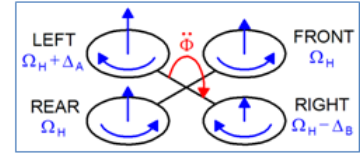
From now on, we can work out the relations between the voltage combinations and $u1, u2, u3, u4$ by looking at 4 different step responses [6].

The following gives a detailed explanation of various feedback variables that need to be changed to bring out a vertical thrust or a pitch or a roll or a yaw.

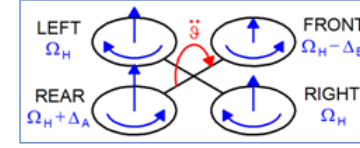
1. THROTTLE



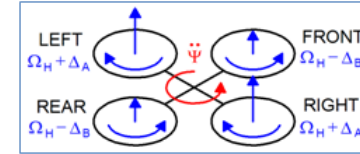
2. ROLL



3. PITCH



4. YAW



VII. IMPLEMENTATION ON MATLAB

We used Matalab to implement the model and check the simulation results and compare it with our physical model results by tweaking the gain parameters[7]. This simulink model is to simulate the actual movement of a quadcopter. The model simulates 3 steps:

- Acend
- Hover
- Decend

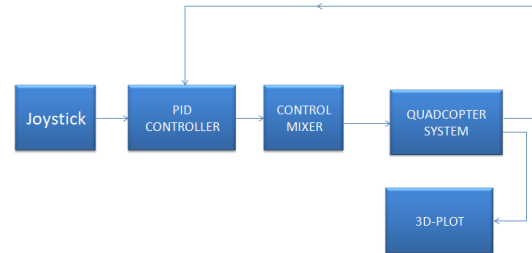


Fig. 7. MatLab simulation flow

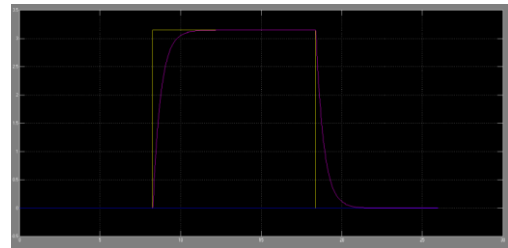


Fig. 8. Joystick pitch response

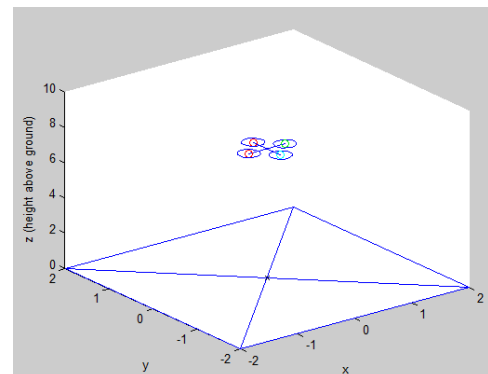


Fig. 9. Quad plot (3D)

The Graphs below shows a step input that is given to the system and its response.

VIII. TUNING

A. Manual tuning

The tuning method that we followed is to first set K_i and K_d values to zero. Increase the K_p until the output of the loop oscillates, and then the K_p should be set to approximately half of that value for good response. Then increase K_i until any offset is corrected in sufficient time for the process. Finally, increase K_d , if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much K_d will cause excessive response and overshoot [8]. A fast PID loop tuning usually overshoots slightly to reach the set point more quickly; however, some systems cannot accept overshoot, in which case an over-damped closed-loop system is required, which will require a K_p setting significantly less than half that of the K_p setting that was causing oscillation. As the Quadcopter requires a quick come-back from any disturbance, we would desire least rise time without an overshoot

B. Simulation Based Tuning

The Matlab Simulink replicates the real Quadcopter model with few parameters included in its code namely,

Propeller: Moment of inertia of the propeller, Drag factor, Thrust factor;

Quadcopter : Weight, length of arm , rotor rpm, Quadrotor Dynamics ;

Atmospheric Conditions: Gravity, Wind Models.

Once these parameters are included in the simulation, based on the run-time graphical response generated the PID values can be verified for its desired stability and adjusted accordingly. Once a desired output is obtained, the PID values in the simulation will be very close to the values in real model (not necessarily the same). If these PID values do not exactly stabilize the system then adopt manual tuning. Tweaking the PID values close to the simulated values in the manual tuning method will stabilize the system.

IX. POSSIBLE IMPROVEMENTS

A. Re-Correcting PID Mechanism in Software

Present PID controllers implemented in software does not generate the PID values for the system. We plan to implement a re-correcting mechanism in the software which can generate the PID values for different parameters as mentioned in simulation based tuning.

B. Machine Learning

The correction mechanism can also be done using machine learning tool like Artificial Neural Network. For this, a set of experiments should be conducted by varying one parameter and keeping rest of the parameters constant. The parameters to be included are mentioned in the Simulation Based Tuning. These parameters are given as input and experimented PID values are given as target

output. Once the network is trained, the output can be predicted within the range of the trained values.

X. CONCLUSION

In present day technology, the efficacy of PID controller is debatable. The control algorithm is not easily adaptable to complex systems. To counter this certain PID control softwares like APM planner have introduced inner PID loops for better performance. Further PID controllers don't have auto correction mechanisms incorporated in them. Thus tuning of parameters using PID values have to be done using trial and error mechanisms like Ziegler Nichols method or Manual Tuning. Ziegler Nichols method of tuning is too aggressive and crude. Thus we are mainly left with Manual Tuning of PID values for parameters. This method is cumbersome and is not safe for practical run-time flight testing. Our system is inclusive of a hardware model and a software implementation of the hardware in MatLab, thus enabling any run time tests to be simulated in software before implementation in hardware. Further any modification to the present system can be simulated in MatLab before hardware testing, making it safe and adaptive to complex systems. An extension of this system could be an implementation of auto corrective PID algorithms that have self PID correction mechanisms instead of Manual Tuning or other heuristic techniques. This methodology can be extended to multi-copters.

ACKNOWLEDGMENT

We are grateful to our project guides, Dr. G. Ramesh, Head MAV Unit, Mr. Suraj C S, Scientist, MAV Unit, CSIR-NAL and Mr. Roshan Antony, Scientist, MAV Unit, CSIR-NAL, Bangalore for their guidance and kind co-operation to complete this project.

We would also like to place on record our profound sense of gratitude to Mrs. Veena Devi, Asst. Professor, R V College of Engineering, and Dr. M Uttara Kumari, Head of Electronics and Communication Engineering Department, R.V.C.E, Bangalore, for providing us an opportunity to undertake a project on "An Optimized Tuning Of A Quadcopter To Attain Stability".

REFERENCES

- [1] S. Bouabdallah *et al.*, "Design and control of an indoor micro quadrotor," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'04)*, (New Orleans, USA), 2004.
- [2] P. Samuel *et al.*, "Design and testing of a rotary wing mav with an active structure for stability and control," in *Proc. AHS Annual Forum 61*, (Grapevine, USA), 2005.
- [3] M. H. Jaffery, "Precision Landing and Testing of Aerospace Vehicles," Surrey Space Center, University of Surrey, Guildoford, PhD Thesis 2012.
- [4] J. Horton, "Low Cost QuadRotor Design," Department of Electrnics Engineering, University of Surrey, Guildoford, Surrey, UK, Thesis 2012.
- [5] N. Guenard *et al.*, "Control laws for the tele operation of an unmanned aerial vehicle known as an x4-flyer," in *Proc. (IEEE) International Conference on Intelligent Robots (IROS'06)*, Beijing, China, 2006.
- [6] D. Griffiths and J. Leishman, "A study of dual-rotor interference and ground effect using a free-vortex wake model," in *Proc. the 58th Annual Forum of the American Helicopter Society*, (Montréal, Canada), 2002.

- [7] A. Benallegue, A. Mokhtari, and L. Fridman, "Feedback linearization and high order sliding mode observer for a quadrotor UAV," in *Proc. the 2006 International Workshop on Variable Structure Systems*, June 2006, Alghero, Italy, pp. 365.
- [8] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," *Swiss Federal Institute of Technology*, 2004.



Srikanth Govindarajan was born on August 11th 1991, Bangalore has done his Bachelors in Engineering(B.E) in the field of Electronics and Communication in R.V College of Engineering , Bangalore, India (year of pass out 2013). The author's major fields of studies are embedded systems, artificial intelligence and avionics.

He has worked as an intern in Indian Innovation Labs(IIL). The field of research at IIL was "Collective Intelligence". He has worked as an intern in CSIR-National Aerospace Laboratories in the field of Unmanned Aerial Vehicle (UAV).