

# PID based Stabilization of Gesture Controlled Drones using HIL Simulation

Vitul Varshney, Melvin Wilson, Sakthivel Sivaraman

**Abstract**— In order to understand the balancing dynamics of a quadcopter, a hardware-in-the-loop simulation (HIL Simulation) using NI ELVIS II+ was undertaken. The purpose of this project was to implement collision avoidance on a quadcopter prototype in a controlled environment. The extra mile was run by simulating PID control for the motor actions in the balancing of the quadcopter. Three modules were developed to simulate the control, which upon implementation, provoked an angular change in the quadcopter position. Upon removal of the control, the balancing aspect comes into the picture. PID control was used to simulate the balancing procedure and its constituting effects. The project was approached with modular programming and project approach in mind to incorporate a readable, maintainable and fool-proof environment..

**Index Terms**— Balancing, Collision avoidance, Gesture control, PID.

## I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) industry is a fast growing sector. This exemplary growth can be attributed to the fact that these vehicles are very versatile in contrast to their lower complexity. More recently, studies on UAV has attracted the attention of the academic community due to its broad applications. Quadcopter or quad rotor aircraft is one of the UAV that are major focuses of active researches.<sup>[1]</sup> The research related to Quadcopter covers the areas of design, control, stability, communication systems and collision avoidance. In this paper, a low-cost gesture based hover control mechanism is developed and implemented using low-cost microcontroller for quadcopter aircraft configuration. Flight control becomes simpler as the quadcopter hovers at a constant level from ground by itself, at the same time allowing anyone to easily maneuver it at that height and perform tasks such as imaging. When successfully implemented, the proposed hover control design will simplify the flight control of a quadcopter, especially for beginners and unskilled individuals. The experiment can be divided into three modules:

**Module 1 - Gesture Control:** In this module, we introduce to the user to gesture based control. This module constitutes the use of Gyroscopes and Accelerometers for the gesture based control. Based on calibration, the change in values of the Gyroscope and Accelerometer cause a change in angle of the quadcopter (pitching or rolling).

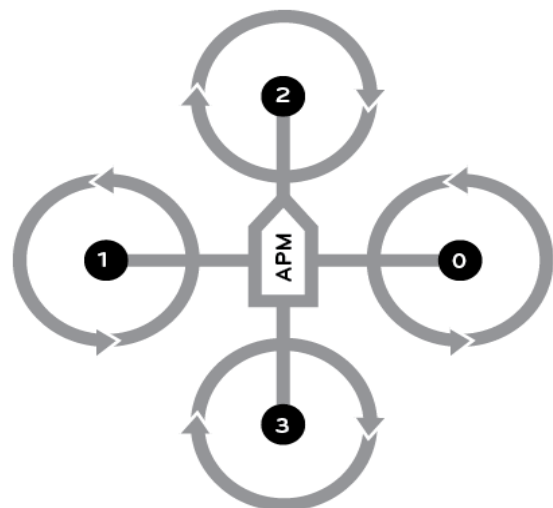
In this rudimentary model, a pre-set angular displacement is taken. Once the gesture control is used to simulate the directional movements, the quadcopter then uses PID control to self-balance and stabilize.

**Module 2 - Manual Control:** In this module, we introduce to the user to the LabVIEW front panel with a user friendly arrangement of constituting inputs and displays. The user may use the Push Buttons (analogous to the keypad of a joystick) to simulate the quadcopter rotation (pitching and rolling) by a certain angle. Upon release of the control, the quadcopter then uses PID control to self-balance and stabilize.

**Module 3 - Obstacle Avoidance:** In this module, we introduce to the user to the quadcopter's reaction to obstacles posed in its path. The main aim here lies to stimulate on-coming obstacles to a stationary (hovering) quadcopter, its reaction to the stimulus and its ability to self-balance and stabilize using PID control once the obstacle is removed. Self-made proximity sensors are used to simulate obstacles a quadcopter may face (birds, buildings, etc.). Once stabilized, various plots are used to visualize the response characteristics.

## Quadcopter Configuration

Quadcopter is a multi-rotor helicopter that is lifted and propelled by four rotors placed in cross configuration.<sup>[2]</sup> The stability of the vehicle is achieved by the configuration where two of the propellers spin in one direction (clockwise) and the other two spin the opposite direction (counterclockwise).<sup>[3]</sup>



**Fig. 1 Quadcopter Motion Concept**

Alteration of the rotation rate of one or more rotor discs changes the torque load and thrust/lift characteristics, whereby control of vehicle motion can be achieved.

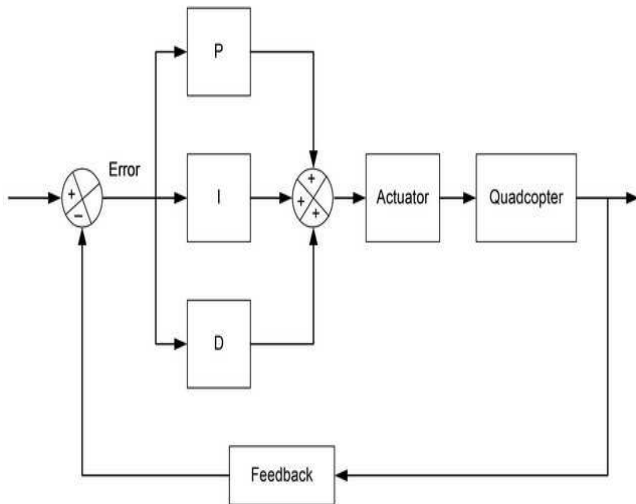
**Manuscript Received on August 2014.**

**Vitul Varshney**, Department of Mechanical Engineering, National Institute of Technology Karnataka, Surathkal, India.

**Melvin Wilson**, Department of Mechanical Engineering, National Institute of Technology Karnataka, Surathkal, India.

**Sakthivel Sivaraman**, Department of Mechanical Engineering, National Institute of Technology Karnataka, Surathkal, India.

**PID Control**



**Fig. 2 Block of Quadcopter PID Controller**

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism that calculates an error value as the difference between a measured process variable and a desired setpoint. The controller adjusts the process through use of a manipulated variable thus minimizing the error.<sup>[4]</sup> If we consider  $u(t)$  as the controller output and  $e$  as the error at instantaneous time  $t$ , the algorithm can be formulated as :

$$u(t) = P_e(t) + I_e \int_0^t e(\tau) d\tau + D_e \frac{d}{dt} e(t)$$

Where the PID controller algorithm involves three separate constant parameters, which can be interpreted in terms of time

- Proportional values ( $P$ ),  $P$  depends on *present* error.
- Integral values ( $I$ ),  $I$  on the accumulation of *past* errors
- Derivative values ( $D$ ),  $D$  is a prediction of *future* errors

The weighted sum of these three actions is used to adjust the process via a control element, in our project, it is the voltage supplied to the actuators (DC Motors).<sup>[2]</sup> By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements.<sup>[5]</sup>

**II. EXPERIMENTAL SETUP**

The frame of the prototype quadcopter used in this experiment for testing purposes was designed and constructed using closed-cell extruded polystyrene foam, mounted with 4 DC motors which powers the propellers. A servo motor is added at the center which can be used to control position of the camera mounted on the copter. Modular programming was implemented using LabVIEW and an intelligible GUI was developed which made manual control easier to implement and also displayed the relevant waveform charts. Arduino UNO microcontroller was used primarily throughout the process, whose results were compared to readings obtained from repeating the process on an NI ELVIS II+ platform. Arduino UNO provided the advantage of working on an incomplex platform thus providing flexibility to the test conditions the prototype could be subjected to. Further tuned PID results could be obtained when the experiment is repeated using NI ELVIS II+ on NI ELVISmx Instrument

Launcher platform. A gyroscope was attached to a glove that could be worn by the user. The detected signal is transmitted to the microcontroller which actuates the DC motors to provide quadcopter motion corresponding to the gesture detected. Four proximity sensors are also placed for obstacle detection. Proximity sensors are developed using photodiodes, IR sensors and resistors. Functionality of the accelerometer, gyroscope, NI ELVIS II+, and microcontroller (Arduino UNO) are tested by running example software and readings from the gyrometer and accelerometer are obtained. Once the initial testing and installation phase was completed, the gyroscope and accelerometer were interfaced to the DC motors of the Quadcopter setup and calibration is carried out. This phase included the development of codes/programs and alterations that will be used for control purposes. Multiple PID constants are calculated according to Ziegler–Nichols tuning method (Refer to Table 1).<sup>[6]</sup> Once successfully completed using the NI ELVIS II+ domain, LabVIEW interface was brought into play. The creation of user interface was tied with the development cycle. The data was acquired, analysed, and put into perspective from the gyroscope, accelerometer and the proximity sensors. The necessary block diagrams<sup>[7]</sup> were created and through the function diagram, the VI was created for the necessary interfacing. PID Control and its implementation was undertaken. Finally, optimization and improvement features were added (i.e. thresholding techniques, smoother operation, user friendly GUI) to the Quadcopter setup so as to improve our project functionality.

**Table 1. Ziegler–Nichols Method**

Control type	Kp	Ki	Kd
P	0.5Ku	-	-
PI	0.45Ku	1.2Kp/Tu	-
PD	0.8Ku	-	KpTu/8
Classic PID	0.60Ku	2Kp/Tu	KpTu/8
Pessen Integral Rule	0.7Ku	0.4Kp/Tu	0.15KpTu



**Fig. 3 Experimental Setup (Hardware)**

**III. GESTURE CONTROL MODULE**

In this module, we introduce to the user to gesture based control. This module constitutes the use of Gyroscopes and Accelerometers for the gesture based control. Based on calibration, the change in values of the Gyroscope and Accelerometer cause a change in angle of the quadcopter (pitching or rolling). In this rudimentary model, a pre-set angular displacement is taken. Once the gesture control is

used to simulate the directional movements, the quadcopter then uses PID control to self-balance and stabilize. The following displays a block diagram of the functioning of the Quadcopter. A single accelerometer was used to control both the thrust of the Quadcopter and the motion of the mounted servo motor (represents a mounted camera on the quadcopter). This accelerometer used readings from two different axis to obtain the desired values. Calibration was then undertaken to achieve desired motion. If we consider  $V_t$  to be the current voltage reading,  $V_{min}$  to be the minimum voltage reading,  $\Phi_{max}$  to be the desired maximum angular displacement,  $R_o$  be the range of values obtained from the microcontroller for a particular axis, then the angular displacement turned  $\Phi$  can be found by the relation:

$$\phi = (V_t - V_{min}) \times (\phi_{max}) / (R_o)$$

The gyroscope was used to obtain readings corresponding to which the quadcopter was directed to either pitch or roll. These values were fed to the PID control using which the designated voltage values were appended to the thrust values and together fed to the motors. Thus, corresponding to a left turn, the motors F1 and F2 maintained a constant thrust, whereas in the case of motors F2 and F4, F2 was given a higher voltage (faster spin of DC motor) and the motor F4 was given a lower voltage (slower spinning of the DC motor)(Figure 4). Thus, this caused a pitching motion of the quadcopter and hence a leftward transverse.<sup>[8]</sup> The readings are constantly fed to the PID control from the gyroscope and the motors respectively adjust to the voltage applied to simulate the motion. In the case an obstacle is present, motion in that direction is stopped immediately and the quadcopter is stabilized. Further motion in the direction of the obstacle is disabled until the obstacle is removed. The front panel constitutes all the necessary indicators which correspond to the information above.

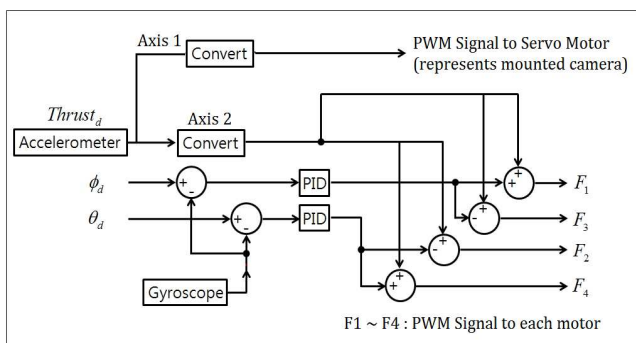


Fig. 4 Gesture based Control using Accelerometer

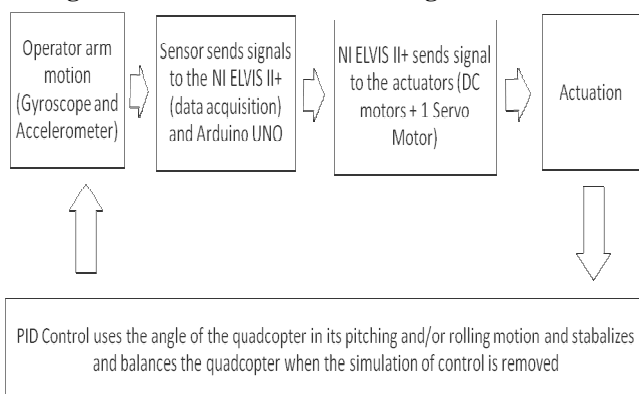


Fig. 5 Block Diagram of Gesture Control Process

#### IV. MANUAL CONTROL MODULE

In this module, we introduce to the user to the LabVIEW front panel with a user friendly arrangement of constituting inputs and displays. The user may use the Push Buttons (analogous to the keypad of a joystick) to simulate the quadcopter rotation (pitching and rolling) by a certain angle. Upon release of the control, the quadcopter then uses PID control to self-balance and stabilize.<sup>[9]</sup> This case is similar in functioning as the Gesture Control Module; however, the simulation of directional movement of the quadcopter is via the buttons present on the front panel of the LabVIEW program. By pressing the button corresponding to leftward motion, the out would be the same as when the Gyroscope was tilted leftward. The altitude (or thrust) of the quadcopter is manually controlled as well using a slider.

#### V. OBSTACLE AVOIDANCE MODULE

In this module, we introduce to the user to the quadcopter's reaction to obstacles posed in its path. The main aim here lies to stimulate on-coming obstacles to a stationary (hovering) quadcopter, its reaction to the stimulus and its ability to self-balance and stabilize using PID control once the obstacle is removed. Self-made proximity sensors are used to simulate obstacles a quadcopter may face (birds, buildings, etc.). Once stabilized, various plots are used to visualize the response characteristics.<sup>[10]</sup> No external control of the quadcopter is exhibited. Once an obstacle is present (that is, when the user covers one of the proximity sensors which their hand) the quadcopter (initially stable) is directed to move away from that obstacle. It continues to do so as long as the obstacle is present. Once all obstacles are removed, the quadcopter again stabilizes and balances itself. Throughout the process, classic PID control type was selected. Using Ziegler-Nichols we calculate the  $K_p$ ,  $K_i$ , and  $K_d$  values to be 0.8, 25, and 0.0064 respectively ( $K_u = 1.333$  and  $T_u = 0.0064$ ).<sup>[11]</sup>

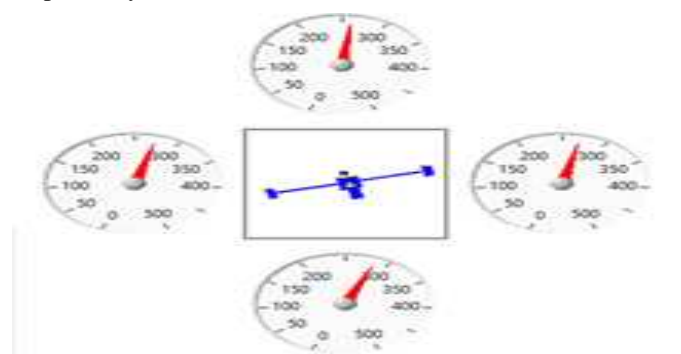


Fig. 6 Simulated Orientation of the UAV and Motor rpms

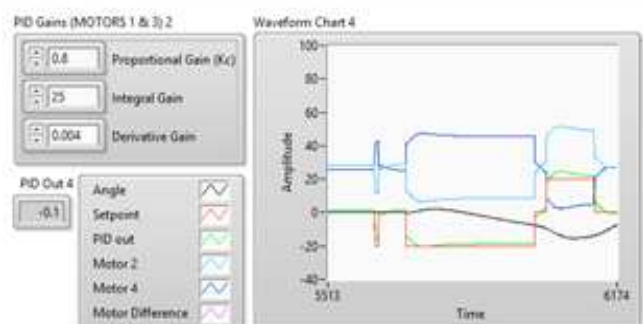
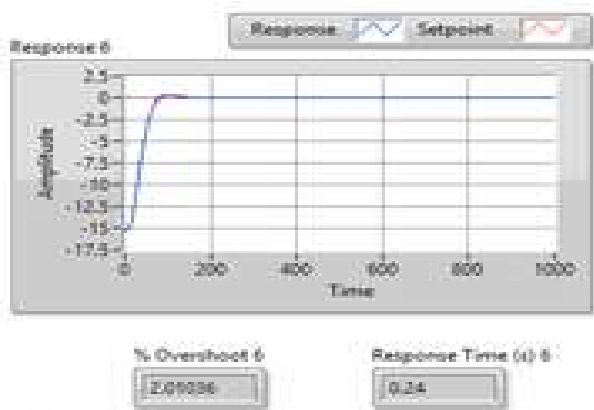


Fig. 7 UAV Performance during Simulation



**Fig. 8 Response Time Graph**

The results derived from the experiments can be observed from Fig 6, 7, and 8. Fig 8 presents the response time achieved using PID control for stabilization during obstacle avoidance tests.<sup>[12]</sup>

## VI. CONCLUSION

The study was conducted to explore the different control strategies that can be implemented to facilitate a more versatile approach to quadcopter control. Detailed research on fine tuning of the control algorithm of each module can provide major advancements in this domain. Analytical optimization method is used to tune a conventional PID controller for stabilization and disturbance rejection of quadcopter. The response time simulations of the prototype using LabVIEW software illustrate the efficient of using different control methods. Future works will focus on predictive obstacle avoidance control design for quadcopter to have ideal tracking and stabilization.

## REFERENCES

- [1] Dirman Hanafil, Mongkhun Qetkeaw1, Rozaimi Ghazali1, “Simple GUI Wireless Controller of Quadcopter,” in Int. J. Communications, Network and System Sciences, 2013, 6, 52-59
- [2] Atheer L. Salih1, M. Moghavvemi1, Haider A. F. Mohamed, “Flight PID controller design for a UAV quadrotor” Scientific Research and Essays Vol. 5(23), pp. 3660-3667, 4 December, 2010.
- [3] Andrew Gallagher “Surveillance UAV” in Worcester Polytechnic Institute, 1May, 2014
- [4] Manju Kurien, “Overview of different approach of PID controller tuning” in International Journal of Research in Advent Technology *E-ISSN: 2321-9637*.
- [5] Ajanta Ganguly, Abinash Banerjee Lane, Yuni Awan Wijonarko, “Design of Tuning method for PID controller using Fuzzy Logic” International Journal of Emerging trends in Engineering and Development., Volume Issue 3, Vol.5 (September 2013).
- [6] Damir Vrančić, “A New PID Controller Auto-tuning Method based on Multiple Integrations” *J. Stefan Institute, Jamova 39, SI-1001 Ljubljana, Slovenia.*, submitted for publication.
- [7] Edwin Chobot, Daniel Newby, Renee Chandler, “Design and Implementation of Wireless Sensor and Actuator Network for Energy Measurement and Control at Home” *International Journal of Embedded Systems and Applications (IJESA) Vol.3, No.1, March 2013*.
- [8] T. Jirinec, “Stabilization and control of unmanned quadcopter,” Master’s thesis, CZECH TECHNICAL UNIVERSITY IN PRAGUE, 2011.
- [9] K. Ogata, Modern Control Engineering. Prentice Hall - Br, 1999.
- [10] R. H. B. Richard C. Dorf, Modern Control Systems. Addison-Wesley, 1995
- [11] R. D. H. Charles L. Phillips, Basic Feedback Control Systems, 1990.
- [12] Cooper Bills, Arjun Prakash, “Vision-based obstacle detection and avoidance,” submitted for publication.