

PROJECT REPORT

Position Control of Quadrotor Helicopter



SURUTHI RAMAN(11012454)

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO.** |
| 1 | Abstract | 03 |
| 2 | Introduction | 04 |
| 3 | Simulink | 07 |
| 4 | Controller | 14 |
| 5 | Simulation | 18 |
| 6 | Conclusion | 20 |
| 7 | References | 21 |

# ABSTRACT

In this project we are working on designing the controller for the 2D Quadcopter model. The controller is finely tuned using PID in order to find the best parameters for the controller which guarantees the robustness of the model. In the beginning of work, the fundamental equations of motion and forces of the Quadcopter are derived and the design parameters for the given Quadcopter are chosen. We have created a 2D model for the Quadcopter based on the equation of motion and forces of moment. Generally, the 2D model will be linearized at a given point. In this project, the model is made to hover at an altitude where the non-linear model is linearized. Based on the inputs, the response of the linear and non-linear model are analysed, it is made sure that both the models behaves in the same manner, if not then the non-linear model and the linearization condition is checked. Once the output for both the model is matched, the research work is further preceded. The controller for the 2D model is designed and the results are analysed. Finally, the controller is finely tuned using PID Tuning for best controller parameters. The best parameter with the effective results for the given Quadcopter is fixed. For the given simulation time, the performance of the Quadcopter model with the optimal controller values are studied by analysing the 2-dimensional visualisation of the model. The controller is designed in such a way that even if there are any disturbances in the future, the model behaves well for the given set of inputs and the design parameters.

# INTRODUCTION

The aim of this project is to design a 2D simulation model for a Quadcopter and design a PID controller for the Quadcopter. When developing the simulation model, first a detailed theoretical description of the problem is studied followed by a paper calculation for the model, later followed by simulation analysis for designing a PID controller.

## Quadcopter

Quadcopter also known as Quad rotor Helicopter, Quad rotor is a multi-rotor helicopter that is lifted and propelled by four rotors. Quadcopter are classified as rotorcraft, as opposed to fixed-wing aircraft, because their lift is generated by a set of rotors (vertically oriented propellers).

Unlike most helicopters, Quadcopter uses two set of identical fixed pitched propellers: two clockwise and two anticlockwise. This use variation of RPM to control loft and torque. Control of the Quadcopter is achieved by altering the rotation rate of one or more rotor discs, thereby changing it torque load and thrust/lift characteristics.

A number of manned designs appeared in the 1920s. These vehicles were among the first successful heavier- than- air vertical take-off and landing (VTOL) vehicles [1]. However, early prototypes suffered from poor performance [1], and latter prototypes required too much pilot work load, due to poor stability augmentation and limited control authority.

More recently Quadcopter designs have become popular in unmanned aerial vehicle (UAV) research. These vehicles use an electronic control system and electronic sensors to stabilize the aircraft. With their small size and agile manoeuvrability, these Quadcopter can be flown indoors as well as outdoors. A typical Quadcopter is equipped with an inertial navigation unit (3 accelerometers, 3 gyroscopes and 3 magnetometers) for attitude determination, a barometer (outdoor) or an ultrasonic proximity sensor (indoor) for altitude measurements and optionally they come with a camera or GPS receiver.

## Indoor Quadcopter

Indoor Quadcopter cannot use GPS for absolute positioning and magnetometers provide noisy measurements due to disturbed local magnetic field. However they take benefit from absence of wind gusts, from relatively stable light conditions and their mission duration is usually shorter than the outdoor Quadcopter. There are already many companies producing Indoor Quadcopter, for example, Ascending Technologies GmBH [2]. The best example for an Indoor Quadcopter is UDI U839, produced by UDI RC. The UDI U839 is shown below.



Figure 1: UDI U839- An Indoor Quadcopter

## Outdoor Quadcopter

Outdoor Quadcopter are generally more durable, they take payload and can fly on longer missions than the Indoor Quadcopter. Absolute positioning is provided by GPS receiver. The best example for outdoor Quadcopter is Phantom 2 Vision+ [3]. The Phantom 2 Vision+ is shown in the figure below.



Figure 2: Phantom 2 Vision+ - An Outdoor Quadcopter

## Advantages of Quadcopter over comparably scaled Helicopters

There are several advantages to Quadcopter over comparable- scaled helicopters. First, Quadcopter do not require mechanical linkages to vary the rotor blade pitch angle as they spin. This simplifies the design and maintenance of the vehicle [4]. Secondly, the use of four rotors allows each individual rotor to have a smaller diameter than the equivalent helicopter rotor, allowing them to possess less kinetic energy during flight. This reduces the damage caused should the rotors hit anything. For small-scale UAVs this makes the vehicle safer for close interaction. Some small-scale Quadcopter have frames that enclose the rotors, permitting flights through more challenging environments, with lower risk of damaging the vehicle or its surroundings [5].

Due to their ease of both construction and control, Quadcopter aircraft are frequently used as amateur model aircraft projects.

## Uses of Quadcopter

Research Platform: Quadcopter are a useful tool for university researchers to test and evaluate new ideas in a number of different fields, including flight control theory, navigation, real time systems, and robotics. In recent year many universities have shown Quadcopter performing increasingly complex aerial manoeuvers.

Military Law Enforcement: Quadcopter unmanned aerial vehicles are used for surveillance and reconnaissance by military and law enforcement agencies, as well as search and rescue missions in urban environments. One such example is the Aeryon Scout, created by Canadian company Aeryon Labs [6], which is a small UAV that can quietly hover in place and use a camera to observe people and objects on the ground.

Commercial Use: The largest use of Quadcopter has been in the field of aerial imagery. Quadcopter UAVs are suitable for this job because of their autonomous nature and huge cost savings [7]. In December 2013, the Deutsche Post gathered international media attention with the project “Parcelcopter”, in which the company tested the shipment of medical products by drone- delivery. As Quadcopter are becoming less expensive media outlets and newspapers are using drones to capture photography of celebrities [8].

Investigating Purpose: Since Quadcopter is small in size and light in weight, they can get into places, where people cannot get into like caves, holes, tunnels, etc. In 2014, in Tamil Nadu, India Quadcopter was used to investigate the granite scam. Investigators used Quadcopter installed with camera and sensors to get in the tunnel and find the granite in it.

# SIMULINK

Simulink is a block diagram environment for multi domain simulation and Model Based Design. It supports system level design, simulation, automatic code generation and continuous test and verifications of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. It is integrated with Matlab, enabling us to incorporate Matlab algorithms into model and export simulation results into Matlab for further analysis.

The advantages of using Matlab Simulink over other programming software are [9]:

* A very large database of built-in algorithms for image processing and computer vision applications.
* Matlab allows us to test algorithms immediately without recompilation. We can type something at the command line or execute a section in the editor and immediately see the results, greatly facilitating algorithms development.
* The ability to auto- generate C code using Matlab Coder, for a large subset of image processing and mathematical functions, which we could then use in other environments, such as embedded systems or as a component in other software.

## Model Based Design:

Model- based design is a process that enables fast and cost effective development of dynamic systems; including control systems, signal processing and communications systems [10]. In Model- Based design, a system model is at the centre of the development process, from requirements development through design, implementation and testing. After model development, simulation shows whether the model works correctly.

## Modelling, Simulation and Analysis with Simulink:

With Simulink, we can move beyond idealized linear models to explore realistic nonlinear models, factoring in friction, air resistance, and other parameters that describe real - world phenomena. Simulink enables us to think of the development environment as a laboratory for modelling and analysing systems that would not be possible or practical otherwise. After we define the model, we can simulate its dynamic behaviour using a choice of mathematical integration methods either interactively in Simulink or by entering commands in the Matlab Command Window. Commands are particularly useful for running a batch of simulations. Using scopes and other display blocks, we can see the simulation results while a simulation runs. We can then change parameters and see what happens for the changes made. We can save simulation results in the Matlab workspace for post processing and visualization. From the various Analysis tools available, we can linearize the model for further use. Since Matlab and Simulink are integrated, we can simulate, analyse and revise our models in either Environment.

## System:

Identify the components of a system, determine the physical characteristics, and define dynamic behaviour with equations. We perform these steps outside of the Simulink software environment and before we begin building our model.

## Determining Modelling goals:

Before designing a model, we need to understand our goals and requirements. We need to ask ourselves these questions to help plan our model design:

* + - * What problems does the model help us solve?
      * What questions can it answer?
      * How accurately must it represent the system?

## Identifying System Components:

Once we understand our modelling requirements, we can begin to identify the components of the system.

* + - * Identify the components that correspond to structural parts of the systems. Creating a model that reflects the physical structure of a system.
      * Identify functional parts that we can independently model and test.
      * Describe the relationships between components, for example, data, energy, and force Transfer.

## Defining System Equations:

After we identify the components in a system, we can describe the system mathematically with equations. Derive the equations using scientific principles or from the input-output response of measured data. Many of the system equations fall into three categories:

* + - * For continuous systems, differential equations describe the rate of change for variables with the equations defined for all values of time.
      * For discrete systems, difference equations describe the rate of change for variables, but the equations are defined only at a specific time.
      * Equations without derivatives are algebraic equations. For example, the total current in a parallel circuit with two components is given by the algebraic equation. Most of the equations used in the thesis come under algebraic equation.

## Collect Parameter Data:

Firstly, create a list of equation variables and constant coefficient, and then determine the coefficient values from published sources or by performing experiments on the system. Then use the measured data from the system to define equation coefficient and parameters in our model.

* + - * Identify the parts that are measurable in a system
      * Measure physical characteristics or use published property values. Manufacturer data sheets are a good source for hardware values.

## Model System:

Build individual model components that implement the system equations, and define the interfaces for passing data between components.

## Model Top- Level Structure:

A model in Simulink is defined as a graphical representation of a system using blocks and connections between links. Once we finish defining a system, its components and equations, we can begin to build our model.

* + - * Use system equations to build a graphical model of a system with the Simulink editor.
      * If we place all the models in one level of a diagram, it will be difficult to read and understand if an error occurs or another scenario. One way to organize our model is to make use of Subsystems.
      * Identify the inputs and outputs connections between subsystems (for example feedback connection, etc.) The input/ output values change dynamically during a simulation.
      * Find out the constants for each components and the values that do not change unless, we change it.
      * The variables for each component and the values that change over time
      * The state variables that components have.

After we build a model component, we can simulate to validate the design.

* + - * Predict the expected output of the integrated model components.
      * Add blocks to approximate the actual inputs and control value.
      * Validate the model design by comparing the simulation output and our expected output.
      * If the result does not match our prediction, change our model to improve the accuracy of our prediction.

## Connecting Model components:

After we build and validate each model components, we can connect them into a complete model, simulate the model, and analyse the results.

* + - * Integrate the model components by first connecting two of them (for example, the plant and the controller). After validating the pair by simulation, continue connecting components until our model is complete.
      * We must take care of how each component we add, affects the other parts of the Model.

## Simulating Connected Components:

Validating our model determines if it accurately represents the physical characteristics of the modelled dynamic system.

* + - * Predict the expected simulation results and outputs of the sub systems.
      * Simulate the subsystems and compare the simulated results without expected results.

## Simulation:

Simulation is a process in which we validate and verify the model by comparing simulation results with:

* Data collected from a real system.
* Functionality describes in the mode requirements.

## Determine Simulation Goals:

Before we simulate a model, we need to understand our goals and requirements. Some of the possible Simulation goals are,

* + - * Understand input to output causality- I.e. for a given input set and nominal part values, look at how the input flow through the system to the output.
      * Verify mode- Compare the simulation results with collected data from the model system. Iteratively debug and improve design.
      * Optimize parameters- Change parameters and compare simulation runs.

## Collect Data:

Collect input and output data from an actual system. Use measured input data to drive the simulation. Use measured output data to compare with the simulation results from our model. We will use the measured input and output values to validate the model.

## Prepare model:

Preparing the model for simulation includes defining the external interfaces for input data and control signals, and output signals for viewing and recording simulation results.

## Set Parameters:

For the first simulation, use model parameters from the validated model. After comparing the simulation result with measured output data, change model parameters to more accurately represent the modelled systems.

## Run and Evaluate Simulation:

Simulate our model and verify that the simulation results match the measured data from the modelled system.

## Import data:

Simulink enables us to import data into our model. For large data sets, use a MATLAB MAT i.e. with an import block.

## Run Simulation:

Using measured input data, run a simulation and save results.

## Evaluate Result:

Evaluate the differences between simulated output and measured output data. Use the evaluation to verify the accuracy of our model and how well it represents the system behaviour. Decide if the accuracy of our model adequately represents the dynamic system behaviour. Decide if the accuracy of our model adequately represents the dynamic system we are modelling.

## Change Model:

Determine the changes to improve our model,

* + - * Parameters- some parameters were initially estimated and approximated. Optimize and update parameters.
      * Adding structure- Some parts or details of the system were not modelled. Add missing Details.

## Stability:

Stability in the general term is defined as the ability of the object to return to its equilibrium if disturbed. Static stability is defined as an object’s initial tendency upon displacement. An object with the initial tendency to return to its equilibrium position is said to have positive static stability. Dynamic Stability of a system or object is defined as the ability to return to a previously established steady motion, after being perturbed. Used, for instance, to refer to the ability of a walking robot to maintain its balance while moving. The static and Dynamic stability is illustrated in the following figure.

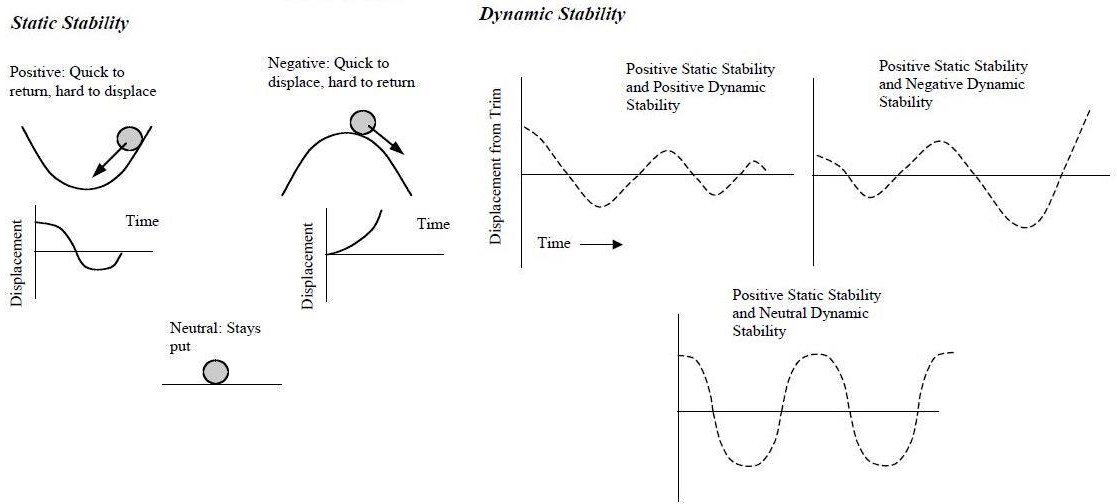


Figure 3: Static and Dynamic Stability

Period is time per cycle. Frequency, is inversely proportional to period, is cycles per unit of time. Amplitude is the difference between the crest or the trough and the original equilibrium condition. Damping is the force that decreases the amplitude of the oscillation with each cycle. The damping ratio is the time for one cycle divided by the total time it takes for the oscillation to subside. The higher the damping ratio, the more quickly the motion disappears. There are two modes of pitch oscillation: the heavily damped short period mode (damping ratio about 0.3 or greater), followed by the lightly damped, and more familiar, long period Phugoid mode.

Short Period mode is excited by a change in angle of attack. The change could be caused by a sudden gust or by a longitudinal displacement of the stick. The lightly damped, long period, or Phugoid, oscillation can take minutes to play out. But it doesn’t get to very often, unlike the short mode. During the Phugoid the aircraft maintains essentially a constant angle of attack.

# CONTROLLER

Proportional-integral-derivative controller is a control loop feedback mechanism widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set-point. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable. The PID controller algorithm involves three separate constant parameters and is accordingly: the proportional, the integral and the derivative values, denoted by P, I, D [11].

## SISO approach

A single-input and single-output (SISO) system is a simple single variable control system with one input and one output. As the linear model of the Quadcopter shows, it is possible to use SISO approach for controlling attitude components. SISO systems are typically less complex compared to MIMO systems [12]. Hence a SISO approach is advised for designing a PID controller for the system.

## PID Controller

PID (Proportional- Integral- Derivative) is a closed loop control system that tries to get the actual result closer to the desired result by adjusting the input. Quadcopter or multicopters use PID to achieve stability. Tuning of the PID controllers has been attracting interest for six decades. Numerous methods have been suggested so far try to accomplish the task by making use of different representations of the essential aspects of the process behaviour [13]. Among the well- known formulas are the Ziegler- Nicolas rule, the Cohen-Coon method, IAE, ITAE, and the internal model control. Control parameters are usually tuned so that the closed- loop system meets the following three objectives:

1. Stability and stability robustness, usually measured in frequency domain.
2. Transient response, including rise time, overshoot, and settling time.
3. Steady state accuracy [14].

The figure [15] below shows control block diagram that can be used for each one of 𝜙, 𝜃, 𝜓

components. As shown in the figure, one controller should be designed for each one Of 𝜙, 𝜃, 𝜓.

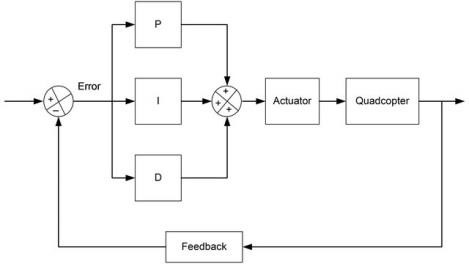


Figure 4: Simulink structure of PID Controller

Where,

Quadcopter is the plant for which the controller has to be designed. Here it is the linearized state space model.

The “PID” block is the PID controller for the system. There are 3 algorithms in a PID controller; they are P, I and D respectively. P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. These controller algorithms are translated into software code lines.

To have any kind of control over the Quadcopter or multicopters, we need to be able to measure the Quadcopter sensor output (for example the pitch angle), so we can estimate the error (how far we are from the desired pitch angle, e.g. horizontal, 0 degree). We can then apply the 3 control algorithms to the error, to get the next outputs for the motors aiming to correct the error.

First, let’s see how the PID controller works in a closed- loop system using the schematic shown above. The variable (e) represent the tracking error, the difference between the desired input value (R) and the actual output (Y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The signal (U) just past the controller is now equal to the proportional gain (Kp) ties the magnitude f the error plus the integral gain (Ki) times the integral of the error plus the derivative gain (Kd) times the derivative of the error.

## Effect of each parameter

The variation of each of these parameters alters the effectiveness of the stabilization. Generally there are 3 PID loops with their own “PID” coefficients, one per axis, so you will have to set P, I and D values for each axis (pitch 𝜃 , roll 𝜙 & yaw 𝜓 ). To a Quadcopter, these parameters can cause this behaviour.

Proportional Gain coefficient – Your Quadcopter can fly relatively stable without other parameters but this one. This coefficient determines which is more important, human control or the values measured by the gyroscopes. The higher the coefficient, the higher the Quadcopter seems more sensitive and reactive to angular change. If it is too low, the Quadcopter will appear sluggish and will be harder to keep steady. You might find the Quadcopter starts to oscillate with a high frequency when P gain is too high.

Integral Gain coefficient – This coefficient can increase the precision of the angular position. For example, when the Quadcopter is disturbed and its angle changes from 20 degree, in theory it remembers how much the angle has changed and will return 20 degrees. In practise if you make your Quadcopter go forward and the force it to stop, the Quadcopter will continue for some time to counteract the action. Without this term, the opposition does not last as long. This term is especially useful with irregular wind, and ground effect (turbulence from motors). However, when the I value get too high your Quadcopter might begin to have slow reaction and a decrease effect of the proportional gain as consequence, it will also start to oscillate like having high P gain, but with a lower frequency.

Derivative Gain coefficient – This coefficient allows the Quadcopter to reach more quickly the desired attitude. Some people call it the accelerator parameter because it amplifies the user input. It also decreases control action fast when the error is decreasing fast. In particle it will increase the reaction speed and in certain cases an increase the effect of the P gains.

## 3.2.3 How to tune Quadcopter PID Gains

I usually tune one parameter at a time, start with P, I and then D gain. We can also go back to fine tune the values I need. For P gain, I first start low and work my way up, until I notice it is producing oscillations. Fine tune it until you get to a point it is not sluggish & there is no oscillation. For I gain, again start low and increase slowly. Roll and Pitch our Quadcopter left and right, pay attention to the how long does it take to stop and stabilize. We want to get to a point where it stabilizes very quickly as we release the stick & it does not wander around for too low. We might also want to test it under windy condition to get a reliable I value.

For D gain, it can get into a complicated interaction with P and I values. When using D gain, we need to go back and fine tune P and I to keep the plant well stabilized. Quadcopter are symmetric, so we can set the same gain values for Pitch and Roll. The values for Yaw is not very important as those of Pitch and Roll so it is probably OK to set the same values as for Pitch/ Roll to start with (even it might not be the best). After our multi-copter is relatively stable, we can start to alter the Yaw gain. For non-symmetric multi-copter like, Hexa-copter, Tri-copter, we might want to fine tune the pitch, roll separately, after we might want to fine tune the Pitch and Roll separately, after we have some flight experience.

Consider, the following unity feedback system,

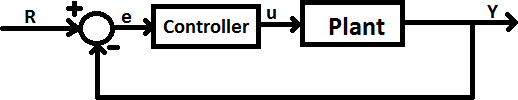


Figure 5: Controller-Plant Simulink Structure

Plant: A system to be controlled

Controller: Provides the excitation for the plant, designed to control the overall system behaviour. The following figure shows designed Simulink model of 2D Quadcopter,

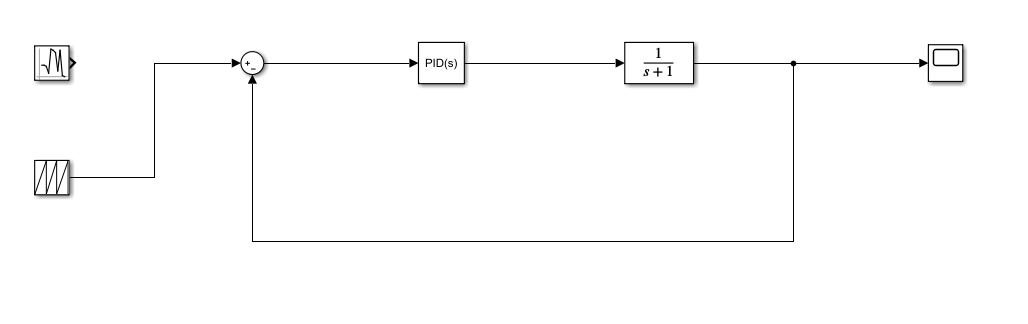


Figure 6: Designed Model of 2D Quadcopter

# SIMULATION

## Simulation:

Now that we have the complete equations of motion describing the dynamics of the system, we can create a simulation environment in which to test and view results of various inputs and controllers. I have used Matlab Simulink and script to create a simulation environment and test the controller for my Quadcopter.

The idea for creating the simulation is to define the time variable for the simulation and determining the iterations for the complete time duration. The initial simulation, velocity and the angular displacement is defined to zero state. Some disturbances are also defined in the angular velocity and the magnitude of the deviation is in radians per second. The input from the controller is obtained for the time variables. The linear and angular acceleration for the model; is obtained for the input obtained from controllers. The function for the all the physical forces and torques and defined separately in the simulation program as a separate function variable.

## Control:

The mathematical model of a Quadcopter is derived so that it is easier for developing a controller for the model. Since we can only control the voltage across the motors, the inputs to our system consist of the angular velocities of each rotor. Note that in our model, we can use the square of the angular velocities,

, and not the angular velocity, . For the notational simplicity, let us introduce the inputs. Let be the position of the Quadcopter in space, be the Quadcopter linear velocity, be the Roll, Pitch, Yaw angles, and be the angular velocity vector. (Note that all these are 3-vectors i.e. along, X, Y and Z axis) .

## PD Control:

First, we will try controlling the model using a PD Controller, with a component proportional to the error between our desired trajectory and the observed trajectory, and a component proportional to the derivative of the error. As the name suggests PD controller is a combination of proportional and a derivative controller the output (also called the actuating signal) is equals to the summation of proportional and derivative of the error signal.

## PID Control

PID control stands for proportional plus derivative plus integral control. PID control is a feedback mechanism which is used in control system. This type of control is also termed as three term control. By controlling the three parameters - proportional, integral and derivative we can achieve different control actions for specific work. PD controller holds advantages of simplicity and ease of implementation, but

they are often inadequate for controlling mechanical systems. Especially if there are noise and disturbances, PD controllers will often lead to steady state error. A PID control is a PD control with another term added, which is proportional to the integral of the process variable. Adding an integral term causes any remaining steady state error to build up and enact a change, so a PID controller should be able to track our trajectory (and stabilize the Quadcopter) with a significantly smaller steady state error. The equations remain the same as PD controller but with an integral term in error.

## Automated PID Tuning

Although PID control has the potential to perform very well, it turns out that the quality of the controller is highly dependent on the gain parameters. Tuning the parameters by hand may be quite difficult, as the ratios of the parameters is as important as the magnitudes of the parameters themselves; often, tuning parameters requires detailed knowledge of the systems and an understanding of the conditions in which the PID control will be used. That is in case of presence of unexpected disturbances or noise; the chosen parameters should be good enough to overcome the disturbances. Hence, we use automatic PID tuning to obtain an optimal set of parameters. The parameters we chose previously were tuned by hand for good performance, simply by running simulations with many possibly disturbance and parameter values and choosing something that worked reasonable well. This method is clearly suboptimal, not only because the resulting gains are not in any way guaranteed to be optimal or even close to optimal.

Ideally, we would be able to use an algorithm to analyse the system and find the “optimal” PID gains, for some reasonable definition of optimal. This problem has been studied, and many methods have been proposed. Many of these methods require detailed knowledge of the system being modelled, and some rely on properties of the system, such as stability or linearity. The method we will use for choosing PID parameters is a method known as extremum seeking.

The gradient descent method does, however, have many disadvantages. First, although it finds a local minimum, that minimum is only guaranteed to be a local minimum- there may be other minima which are better global minima. In order to avoid choosing suboptimal local minima in the cost function, we repeat our optimization several times, and choose the best result. We initialize our PID parameters randomly, so each time we run the optimization we will get a different result. In addition, instead of choosing disturbance and optimizing the response to that disturbance, we use random disturbance for all iteration and use the average response to compute costs and gradients. This ensures that our parameters are general and not optimized for a specific disturbance. In addition, we vary the step size and the number of disturbances to try per iteration, in order to increase the sensitivity of our results as our iteration continues. We stop iterations when we detect a steady state, which we do by computing a linear regression on the most recent costs and iterating until the slope is almost zero using a 99% confidence interval.

# CONCLUSION

The introduction about the Quadcopter is given followed by the advantages of the Quadcopter over comparable scaled helicopter and the application of Quadcopter in day to day life. We created a 2D model of the Quadcopter is using Matlab simulink for the equation of motion. This simulator was used to test and visualize Quadcopter control mechanics. I ignored the aerodynamically effects such as blade- flapping and the non-zero free stream velocity. Then, the non-linearized model is linearized for the given operating point. Later, the non-linearized and the linearized model are studied for a given set of step input. A detailed explanation about the PID control is given, with the PID plant structure, the transfer function and finally the tuning using a SISO tool approach. I began with a simple PD controller, even though the response of the model was good there was a significant steady- state error. In order to decrease the significant steady state error, I introduced the integral term to the controller to create the PID controller. I tested the PID controller with minor modifications to prevent integral wind up. I found out that the steady state error prevented in PID controller was much better that the PD controller when presented with the same disturbances and using the same Proportional and Derivative gains.

We found out that the tuning of PID controller was difficult, and would often lead to an unstable system for unknown disturbances. In order to avoid the difficulties of tuning the PID parameters and find the optimal set of gain values, I used the automatic PID tuning method in order to numerically estimate gradient of a cost function in PID- parameter space and iteratively choose a set of parameters to minimize the cost function. I found out that the controller designed using the Automatic PID tuning is performing much better than the one using manually tuned parameter even in case of presence of random disturbances.

**REFERENCES**

1. Leishman, J.G. (2000). Principles of Helicopter Aerosynamics. New Yorl, NY: Cambridge University Press.
2. [www.asctec.de](http://www.asctec.de/)
3. <http://www.dji.com/product/phantom-2-vision-plus>
4. Pounds, P.;Mahony, R,; Corke, P. (December 2006). "In the Proceedings of the Australasian Conference on Robotics and Automation". Auckland, New Zealand.
5. Hoffman, G,; Haung, H.; Waslander, S.L.; Tomlin, C.J. (20-23 August 2007). "In the Conference of the Anerican Institute of Aeronautics and Astronautics". Head Hilton, South California.
6. "Aeryon Labs Inc."
7. Illumin- The Quadcoptor's Coming of Age"
8. "Paparazzi Agency - We've Used Drones For A Long Time". [www.tmz.com](http://www.tmz.com/)
9. [http://www.mathworks.com/matlabcentral/answers/82408-the-advantages-of-matlab-over-other- programing-languges-for-image-processing](http://www.mathworks.com/matlabcentral/answers/82408-the-advantages-of-matlab-over-other-programing-languges-for-image-processing)
10. Simulink Getting Started Tutorial by Mathworks
11. <http://en.wikipedia.org/wiki/PID_controller>
12. <http://en.wikipedia.org/wiki/Single-input_single-output_system>
13. "Improving Disturbance Rejection of PID controllers by Means of the Magnitude Optimum Method",

D. Vrancic, S. Strmcnik, J. Kocijan and P. B. M. Oliveira

1. "On stabilizing PI Controller Ranges for Multi- variable Systems" pp. 620- 625 by, C. Lin, Q.G. Wang,

Y. He, G. Wen, X. Han and G. Z. H. Z. Li.

1. https://oscarliang.com/quadcopter-pid-explained-tuning/