AUTOPILOT SYSTEM

Sanket Sharma
dept. of Mechatronics Engineering
Chandighar University
Punjab, India
sharma.sanket272@gmail.com

Ankit Chaudhary
dept. of Mechatronics Engineering
Chandighar University
Punjab, India
ankitchaudhary.office@gmail.com

Ajit Singh
dept. of Mechatronics Engineering
Chandighar University
Punjab, India
ajitsohal2003@gmail.com

Kartikay
dept. of Mechatronics Engineering
Chandighar University
Punjab, India
chiranjeebkartik@gmail.com

Omm Jagga
dept. of Mechatronics Engineering
Chandighar University
Punjab, India
omjagga2002@gmail.com

***Abstract*—Cost is an important aspect in designing a target drone, however the poor performance of low cost IMU, GPS, and microcontrollers prevents the use of complex algorithms, such as ARS, or INS/GPS to estimate attitude angles. The autopilot consists of an altitude hold, roll hold, and path following controller. The altitude hold controller uses vertical speed output from a GPS to improve damping. We present an approach to understand dynamics and extrinsic physical effects of a quadrotor UAV.**

Keywords- autopilot, PD(PID), control system, Simulink, dynamics

I. Introduction

As drones are growing rapidly in popularity, their use in multiple section is increasing exponentially. Over the past few years, drones have become central to the functions of various businesses and governmental organizations and have managed to pierce through areas where certain industries were either stagnant or lagging behind. From quick deliveries at rush hour to scanning an unreachable military base, drones are proving to be extremely beneficial in places where man cannot reach or is unable to perform in a timely and efficient manner.
Drone is not just a body with a battery with propellers; it works in a much complex way as it seem to be. There are many intrinsic and extrinsic factors affecting the flight of drone.
To understand the very core of working of drone as compared to physical dynamic,
Applying and modeling full dynamics of drone is not a new idea and is not the focus of paper. Instead, contribution of this paper is a novel way to understand the core and basic dynamics of a quadrotor drone while replicating a defined trajectory in YZ plane. This way has enabled us to understand the very basic and important extrinsic and intrinsic factor affecting the drone flight.

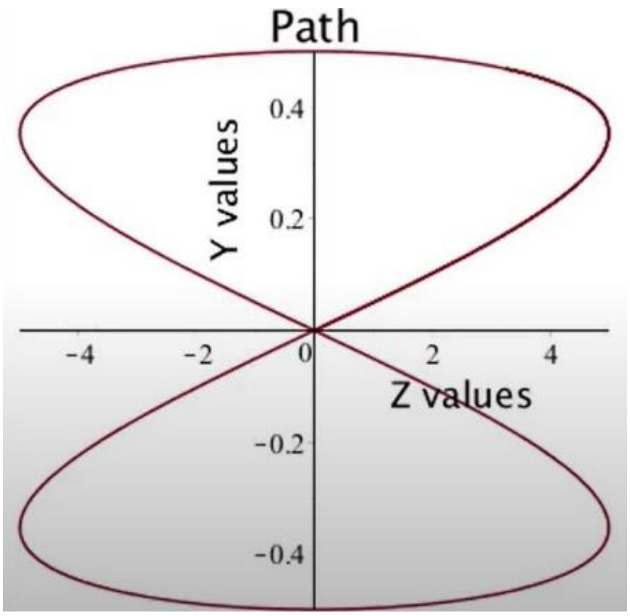
II. SYSTEM OVERVIEW

Automatic instruments like drones are used in many areas including military and civil. Recent application is food delivery using drones. Several industries (automotive, military, factories, space, etc.) use robots for dangerous and repetitive tasks.[1]

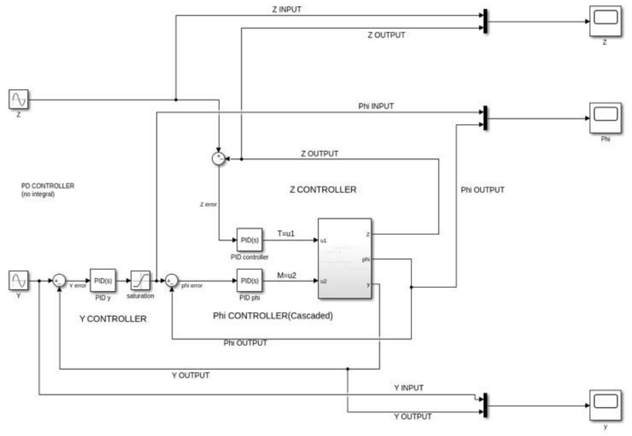
For such kind of tasks, we need AUTOPILOT SYSTEM which can effectively control the unmanned flight vehicle. An autopilot system helps increases efficiency and minimize risks of crashing. Flight trial cost and risks are minimized and the design cycles are greatly shortened. The feasibility and the effectiveness of the approach are verified through results from lab simulations and field trials.[2]

We have designed a flight controller for a 2d quadcopter in Simulink. We have modelled it in two dimensions, ZY plane and drone will be moving only in that plane. Cascade controller is designed for this task because the number of control variables are less than degree of freedom.

For our path we will be using a figure 8 the drone will be starting at (0,0) and moving in a figure 8 and coming back to (0,0).

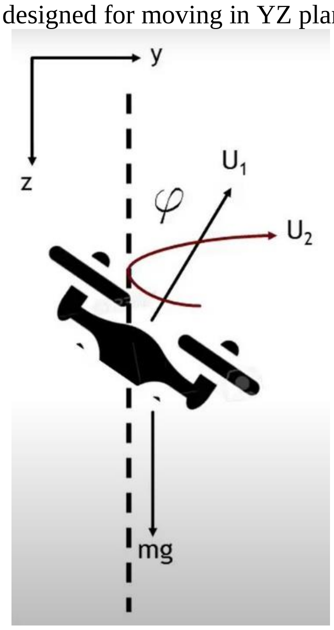


A comprehensive, production grade UAV flight control system is developed in Simulink, as presented



III. CONTROLL DESIGN

The control is designed for moving in YZ plane as shown



.

There are few equations of motion which governs the drone's motion, so to represent those in Simulink we will use different blocks for different purpose according to our need. Doing this will help us to generate the required code, which will be fed to actuators to carry out he motion. The controller has two inputs Y and Z.

There are 3 feedback loops in the controller. In the first feedback loop we use the error (difference between Z input and Z output) The Φ is the inner loop and Y is the outer loop, so there are two error feedback loops. Since it is a PD CONTROLLER, the KI value in all the PID BLOCKS are set to zero.

In PID BLOCK FOR Φ function KP & KD is very high because the inner loop dynamics needs to be faster than the outer loop, this is a fundamental concept of a cascade controller because the inner dynamics must adjust, much quicker with respect to the outer dynamics.

IV. PLANT MODEL

Our system combine two approaches, plant and feedback model, to design trajectory and control model of a quadrotor UAV. This combined approach will help the model design to be much efficient.

I our plant model approach, mathematical model for dynamics of drone with respect to Z and Y plane are constructed. This is done to find the dynamics displacement of drone in Z and Y axis in a particular time interval by double integrating the required equations of motion. Further in construction of model, it is linearized around hover point.

a. Equation of motions:

At the very core, some equations of motion are working to produce the required output for the trajectory and control design of drone. Acceleration in Z axis is defined as:

$$z`` = g - \frac{u_1}{m}.\cos(\Phi)$$

Acceleration in Y axis is defined as:

$$y`` = \frac{u_1}{m}.\sin(\Phi)$$

where,

$$\Phi`` = \frac{u2}{Ixx}$$

Here, Z = displacement in z axis, y being displacement in Y axis and Φ is angle of roll.

u_1 = force from propeller, u_2 = moment, m = mass, Ixx = moment of inertia

b. Linearization:

As the system is nonlinear because of sine and cosine functions, system is the linearized at hover point as it is most stable point. At hover point properties are: $u_2 \approx 0 = \Delta u_2$, $y = y_0$, $z = z_0$, $u_1 = mg$, $\Phi \approx 0 = \Delta\Phi$.

Here moment is approximately zero because roll angle at hover point is approximately zero. Acceleration in z and y axis is also zero because drone is stable at hover point. Hence $u_1 = mg$.

Therefore, equation of motion transforms to:

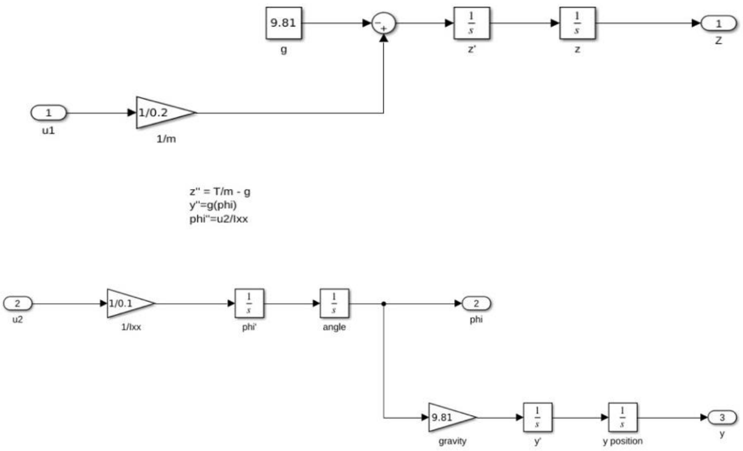
$$z`` = g - \frac{u_1}{m}.\cos(\Delta\Phi) = g - \frac{u_1}{m}$$

$$\Phi`` = \frac{u2}{Ixx}$$

$$y`` = \frac{mg}{m}.\sin(\Delta\Phi) = g\Phi, \sin(\Delta\Phi) = \Phi$$

(small angle approximation)

The input of plant model will be force from propeller, u_1 and moment, u_2. Output will be dynamic displacements in z and y axis i.e. z and y respectively and roll angle Φ. Model's control design is discussed in 'control design' section.



V. FEEDBACK SYSTEM

The open-loop configuration is not efficient to change the thrust, moment and roll angle of the quadrotor depending upon the error between desired co-ordinate and actual rotor position. Hence a PID feedback system is employed in quadrotor control system.

PID control block: A PID control respectively stands for proportional, integral and derivative control, and is the most commonly used control technique in industry. MATLAB's Simulink library provides a *PID controller block*, the block output is a weighted sum of the input signal, the integral of the input signal, and the derivative of the input signal. The weights are the proportional, integral, and derivative gain parameters.

Since there is no *steady-state error [4]* in the system, setting integral hyperparameter to zero convert the PID controller to *PD controller [5]*.

The difference between output from the plant and the trajectory input by the user is calculated and fed into the PD (PID) to calculate the required for the plant.

The input thrust for the plant is calculated by feeding the error of output Z from the plant and the Z input by the user to the PID.

The input moment for the plant is calculated by feeding the error of the cascaded feedback system of Y and Φ to the PID. In order to scale the y-coordinate error, a saturation block is used that bounds the output under upper limit and lower limit.

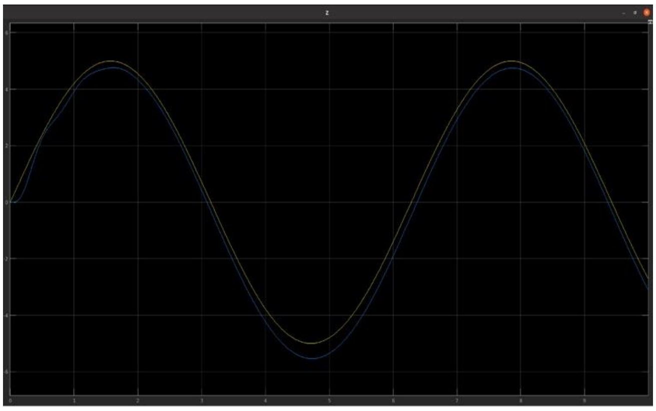
Saturation block: The Saturation block produces an output signal that is the value of the input signal bounded to the upper and lower saturation values. The upper and lower limits are specified by the parameters Upper limit and Lower limit [6].

VI. RESULTS

In this section, we present some results of our model computed using Simulink.

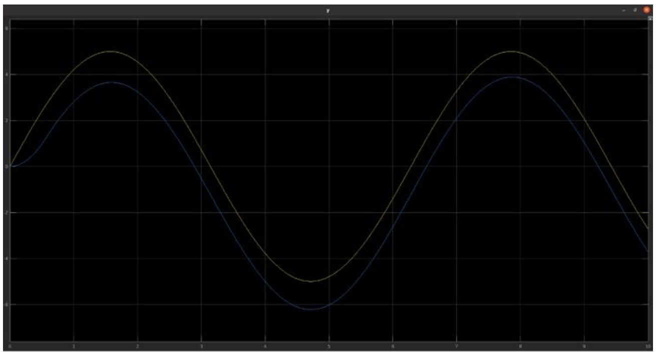
In all the result graphs presented below, 'yellow' depicts input trajectory data, and 'blue' depicts the output from the plant.

First we show result of Z-axis for the drone:



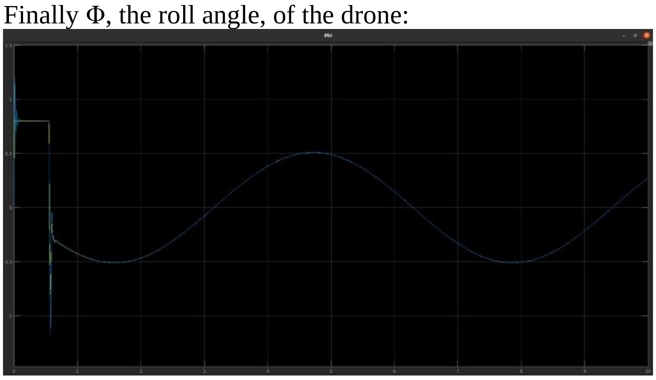
Result 1. Input trajectory vs Output from plant for Z plane

Then we show the Y-axis for the drone motion:



Result 2. Input trajectory vs Output from plant for Y plane

Finally Φ, the roll angle, of the drone:



Result 3. Input roll angle vs Output angle from plant

VII. CONCLUSIONS

In this paper we presented and experimentally tested a trajectory and control system designed a quadrotor UAV that uses a mathematical model at the core to replicate a quadrotor controller.

At the core we created a plant model in Simulink that models that dynamics of the drone in YZ plane, using two equations of motions. The model inputs the thrust from the rotor and the moment to output dynamics displacement of drone in YZ plane. The trajectory to be followed by the drone in input to the plant using a feedback system.

Error (difference of trajectory input Y and Z and the output of the plant Y Z and Φ) is fed back to the plant model as inputs.

We demonstrated a basic control and trajectory model for a quadrotor UAV.

ACKNOWLEDGMENT

This paper has been validated through experiments in Simulink.

We thank our team members who provided insight and expertise that greatly assisted this paper. We thank Mr Anurag Sharma for assistance and comments that greatly improved the manuscript.

REFERENCES

[1] Figueiredo, H. V., & Saotome, O. (2012, October 1). Simulation Platform for Quadricopter: Using Matlab/Simulink and X-Plane. IEEE Conference Publication | IEEE Xplore.

[2] Zuo, M. (2014, May). Model-Based Design of UAV Autopilot Software. Research Gate.

[3] VDEngineering. (2018, July 1). MATLAB & Simulink Tutorial: Quadrotor UAV Trajectory and Control Design (PID + Cascaded) [Video]. YouTube.

[4] Extras: Steady-State Error. (n.d.). Control Tutorials for MATLAB and Simulink.

[5] Figueiredo, H. V., & Saotome, O. (2012, October 1). Simulation Platform for Quadricopter: Using Matlab/Simulink and X-Plane. IEEE Conference Publication | IEEE Xplore.

[6] Proportional-Integral-Derivative (PID) Controllers - MATLAB & Simulink - MathWorks India. (n.d.). MATLAB.

[7] Wikipedia contributors. (2021, November 18). Control system. Wikipedia.