Autonomous Control of a Quadrotor UAV using Fuzzy Logic

Vijaykumar Sureshkumar Fnu
University of Cincinnati

Kelly Cohen
University of Cincinnati

Journal of Unmanned System Technology

Autonomous Control of a Quadrotor UAV using Fuzzy Logic

Vijaykumar Sureshkumar and Kelly Cohen
University of Cincinnati, Cincinnati, Ohio, United States

Abstract—UAV's are being increasingly used today than ever before in both military and civil applications. A certain level of autonomy is imperative to the future of UAV's. A quadrotor is a helicopter with four rotors, that make it more stable; but more complex to model and control. Characteristics that provide a clear advantage over other fixed wing UAV's are VTOL and hovering capabilities as well as a greater maneuverability. Fuzzy logic control has been chosen over conventional control methods as it can deal effectively with highly nonlinear systems, allows for imprecise data and is extremely modular. The objective of this research endeavor is to present the steps of designing, building and simulating an intelligent flight control module for a quadrotor UAV. Validation of the math model developed is discussed using actual flight data. Excellent attitude tracking is demonstrated for near hover flight regimes. System design is comprehensively dealt with. The responses are analyzed and future work involving hardware-in-the-loop simulations is proposed.

Keywords—Fuzzy logic, quadrotor, UAV, autonomous.

I. Introduction

THE field of aerial robotics has seen rapid development in both sensor and actuator technology. An emerging popular rotary wing UAV platform is the Quadrotor, which is gaining focus in realizing micro aerial vehicle concepts. It is a helicopter with four rotors, two spinning clockwise and two anticlockwise, thereby maintaining net zero moment. Its VTOL ability and excellent maneuverability make it a potentially indispensable tool in tackling natural disasters, surveillance, inspection of structures, geological applications etc.

Flying a rotary wing UAV using traditional radio receiver and transmitter (RX/TX) can be a daunting task especially in high stress situations. A certain level of autonomy is imperative to the future of such UAV's [1]. A Quadrotor is an under actuated system with six DOF and four control outputs. It is simple in design and construction as compared to a scaled single rotorcraft, but presents an interesting control challenge. Fuzzy logic control has been chosen to address this control problem, as it can effectively deal with highly nonlinear systems and provides inherent robustness. Modularity and adaptability are key advantages to the fuzzy logic approach. Genetic algorithms while providing good results in simulation are not suited for real time control applications. PID control is overwhelmingly used in typical cases but does not allow for modularity or adaptability in the simple sense, one controller does not fit all purposes. It should be noted here, that Fuzzy logic control in this effort is seen as a viable alternate to PID control, and as an intelligent control module but not incorporating artificial intelligence qualities. Nonlinear dynamic system identification for purposes of autonomous control can be ideally performed by neural networks; however this direction is not within the bounds of this research effort and hence has not been studied.

A Quadrotor was indigenously built utilizing the excellent resources of open source project, Aeroquad. In the course of this research endeavor, the Quadrotor went through multiple design iterations to overcome specific design issues, a quick overview of the builds has been provided. Propeller selection and other design considerations are discussed in detail along with data supporting the current design. A Mathematical model of the Quadrotor was developed in order to create a simulation environment where the controller could be validated and fine-tuned before testing commenced on the Quadrotor. The system overview, mathematical modeling, validation, controller development and simulation results are presented comprehensively in the following sections. Validation using actual flight data is presented along with the methodology for the same. Preliminary work concerning serial communications and signal transfer issues are discussed. Conclusions were drawn on the basis of these results and future work to resolve certain issues is proposed.

II. System

A comprehensive description of the system developed is provided here. Hardware that enables the wireless transfer of data between the Quadrotor and the ground station is described. Evolution of design is provided in a visual manner.

A. Description

The Quadrotor developed has four rotors symmetrically distributed around a central cabin that houses the electronics. The two cross arms have been constructed out of square 6061 Aluminum alloy tubes measuring ¾ inches x 24 inches. The 6 DOF IMU is mounted using vibration dampers on top plate to reduce sensor noise due to vibrations as far as possible. Motor mounts have been fashioned out of aluminum and carbon fiber and lock nuts have been used to secure them to the frame. This ensures that vibrations do not shake loose the actuator components. The main battery is integrated into the frame at the base position to ensure stability but not placed too low as this

Corresponding author: V. Sureshkumar (e-mail: sureshvr@mail.uc.edu).
This paper was submitted on November 29, 2013; revised on November 19, 2014; and accepted on December 15, 2014.

compromises the maneuverability of the craft [2]. It was decided to use a separate smaller flight battery to power the Arduino microcontroller and the sensor shield. Motor to motor distance is exactly 60 cm, and the total weight is 1.776 Kgs. It is built and flown as a '+' configuration. This final design was motivated by sluggish responses in preliminary simulations. A significant weight reduction, a robust, repeatable and low cost frame and smaller inertia values were successfully achieved. **Figure 1** below shows the system in its entirety. The complete component list is presented below.

TABLE I COMPONENT SPECIFICATION

Component	Specification	Weight
2 Aluminum square arms	¾" x ¾"x 24"	450 g
4 Landing gear	Aluminum + Carbon fiber	173 g
4 BLDC motors	950 Kv, 200 Watts	292 g
4 ESCs	Exceed RC proton 30A	164 g
4 fixed pitch propeller	APC SF 12x3.8 2CW, 2CCW	88 g
Main lithium polymer battery	Turnigy 11.1 V, 4000 mAh	353 g
Secondary battery	E-flite 7.4 V, 620 mAh	37 g
9 DOF IMU shield	ADXL 345 – triple axis accelerometer ITG 3200 – triple axis gyroscope HMC 5833L – triple axis magnetometer	40 g
Microcontroller	Arduino MEGA 2560	35 g
2 Xbee modules	1 Xbee Pro 900 MHz with Xbee shield(on board) 1 Xbee Pro 900 MHz with usb explorer dongle	11 g
Carbon fiber plates, wiring Standoffs, screws and lock nuts	Miscellaneous frame components	133 g
	Total Weight	1776 g

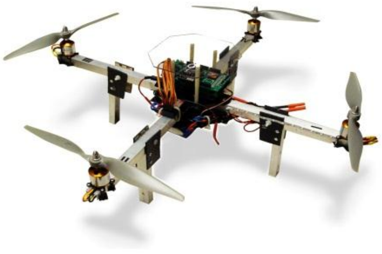

Figure 1 Picture of the Quadrotor

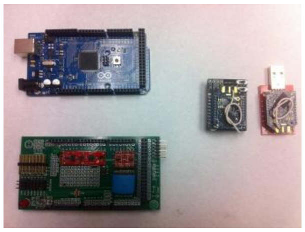

Figure 2 IMU shield and Arduino

Wireless communication

Figure 2 shows the IMU which is designed using a triple axis accelerometer ADXL 345, a triple axis gyroscope ITG 3200 and a triple axis magnetometer HMC 5833L. Two Xbee Pro 900 MHz modules have been used to facilitate wireless transfer of signals. This module is ideal for solutions where RF penetration and absolute transmission distance are paramount. A shield is used to communicate with the Arduino and it transmits data to another module that is connected to the ground station via an usb explorer dongle. Data is imported in real time into MATLAB environment for future hardware-in-the-loop simulations. This particular module offers the least interference to the 2.4 GHz band that is used by the RX/TX setup.

The open source flight software uses DCM [3] techniques to fuse sensor data into an estimation of the attitude of the craft. A 4th order low pass filter is used to filter the accelerometer and gyroscope data. The excellent functionality that the software provides has been utilized extensively for handling communications. The Arduino along with the IMU outputs filtered accelerometer and gyroscopic data as well as roll, pitch and heading angle estimates. We will propose in later sections how the control loops are to be evaluated.

B. Design Evolution

This research endeavor was borne out of work done towards a concept for an emergency roadable vehicle capable of VTOL. The concept took the form of a quadrotor, and a build of a scale model commenced. As the first build was completed, the scope of work involved in realizing the proposed strategy was enormous. The project focused on autonomous intelligent control and envisioned a future module for obstacle avoidance. Developing and implementing an indigenous controller would contribute significantly to the research already being carried out at MOST Aero labs at UC , while also demonstrating the efficacy of fuzzy logic control. This section details the design iterations that the Quadrotor went through before its final form and issues that were addressed.

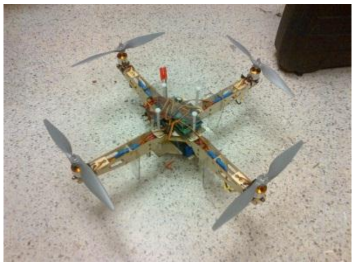

Figure 3 Build 1.0

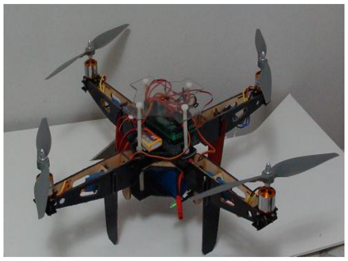

Figure 4 Build 1.1

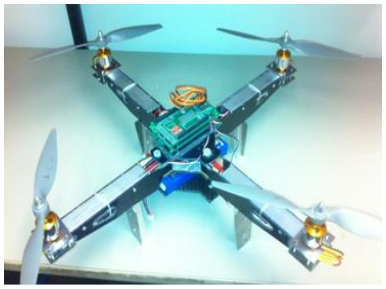

Figure 5 Build 1.2

The build 1.0 (see **Figure 3**) featured a plywood frame and a center to center distance of 0.46 m and 11” x 5.5” fixed pitch propellers. It was constructed purely to demonstrate the quadrotor concept and to allow investigation of design objectives. The salient points are good vibration absorption and excellent payload fraction; its inherent fragility did not permit a test flight.

Figure 4 shows the first design that achieved flight using a RX/TX setup. Carbon fiber arms were machined and overlaid on the wooden framework to reinforce it. The horizontal arms helped absorb vibrations, thus keeping sensor noise to a minimum. However the flex allowed by the arms led to unstable and undesirable flight characteristics.

This led to the build 1.2 (**Figure 5**) which incorporated sectioned aluminum square tubes to both stiffen the arms and produce more durable landing gear. Stability was achieved at the cost of increased weight. This prototype was the first to be modeled, however the responses obtained from the simulation showed an extremely sluggish behavior. Thrust to weight ratio went below the 2:1 threshold set from research into design concepts.

A complete overhaul of the system was proposed and the result was the quadrotor as shown in **Figure 1**. Increasing the span length improved stability as well as allowing the use of slightly bigger propellers. A 1” increase in propeller span increased thrust by over 20%. The ESC’s were incorporated into the central housing. Also a repeatable, reliable, low cost frame was developed which can be used as a starting point for all future projects. This iterative design process facilitated a basic understanding of the dynamics of a Quadrotor, the factors affecting it and design optimization.

C. Propeller selection and design considerations

There are several rules of thumb when it comes to propeller selection and evaluation. There are also practical considerations such as availability of clockwise and counter-clockwise propellers for multirotor UAV's in general. Price point and availability of data sheets for the selected propeller is also of prime importance. Reliability and repeatability was also considered.

The thrust to weight ratio mentioned above as 2:1 was a good starting point in that it ensures enough thrust exists for desired performance without the Quadrotor being too agile/difficult to fly. The flying weight was initially set to be around 1 kg with the plywood frame of Build 1.0 above. This eventually went up to 1.776 kg through the design evolution. The brushless motor selected was capable of producing a wide range of RPM with a low power draw and high efficiency at hover RPM. The University of Illinois at Urbana-Champaign has an extensive propeller database; this includes wind tunnel measurements for nearly 140 propellers used on small UAVs and model aircraft. The propellers were purchased off-the-shelf from retail outlets and were unmodified for those tests. Measurements include thrust and torque coefficient data over a range of advance ratios for specific RPMs.

This very useful resource was banked upon to find the right propeller according to the above constraints of motor, price point, availability etc. The Aeroquad open source project for multirotor UAV as well as several blogs of experienced radio control pilots was researched for starting points. For larger Quadrotor's that carry payloads, large propellers and low-kv motors tend to work better. These have more rotational momentum, and will more easily maintain stability.

Another online resource provides a digital calculator for the various flight parameters and how they change for different motor and propeller combinations. A comparison of the different propellers shortlisted along with pertinent information is given below.

Propeller 10x5 E & 10x7 E had very high pitch angles and did not meet the prerequisites above. APC 11x5.5 E & 12x3.8 SF were shortlisted and evaluated further, the poor flight times associated with the former as well as non-optimal power consumption led to the APC 12x3.8 SF as the propeller of choice. Its easy availability and reasonable price point made it an attractive option.

TABLE II PROPELLER CALCULATIONS

Propeller	Volts (V)	Amps (A)	Thrust (g)	Power (W)	Flight time (mins)
APC 10x5 E	12.31	14.20	839.73	174.80	4.6
APC 10x7 E	10.83	12.04	699.40	130.39	3.9
APC 10x4.7 SF	11.06	9.31	470.90	102.97	4.7
APC 11x5.5 E	10.78	13.86	791.53	149.41	3.6
APC 12x3.8 SF	12.12	17.91	1206.49	217.07	7.8

D. System modeling

The mathematical model presented was used to develop a simulation environment in MATLAB/Simulink to facilitate the development of proposed control strategy. Extensive simulations were performed to validate the flight controller and fine tune parameters of the fuzzy inference system. The quadrotor is a complex mechanical system; it collects numerous physical effects from the aerodynamics and mechanics domain [4]. It should be noted that ill-defined and complex phenomenon such as ground effect, gyroscopic effects, airflow interactions, propeller flapping etc were studied but not modeled due to their complexity. It is hoped that the inherent robustness fuzzy logic control embodies will tackle these shortcomings. Future development insists on a more comprehensive model that will lead to significant improvement in the results obtained.

Two experiments were setup to accurately quantify important model parameters such as thrust and torque generated with respect to the pulse width modulated output values of the actuators. These were favored over theoretical modeling to incorporate and take into account air interaction effects on thrust efficiency and friction losses. A torque stand setup (**Figure 6**) along with a high precision balance was used to generate Torque vs. PWM data, shown as a scatter plot in **Figure 7**.

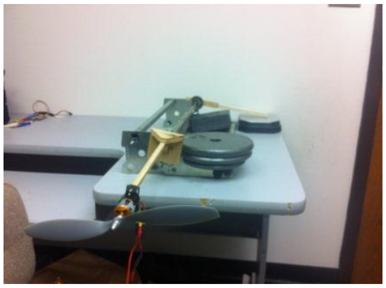

Figure 6 Torque and setup

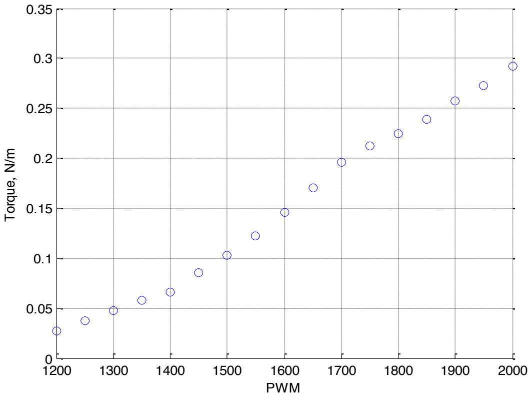

Figure 7 Torque vs PWM

A program was written in the Arduino IDE that sends incremental PWM values to all the motors with a delay built in to allow for recording of the scale readings. Characterizing the motor's in this way, while time consuming lends high confidence to the expected thrust and torques generated during future hardware-in-the-loop simulations.

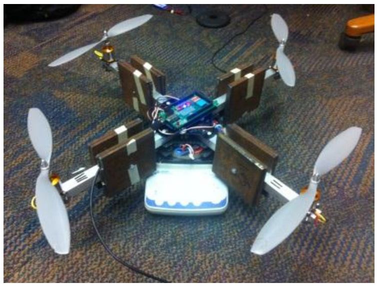

Figure 8 Experimental thrust setup

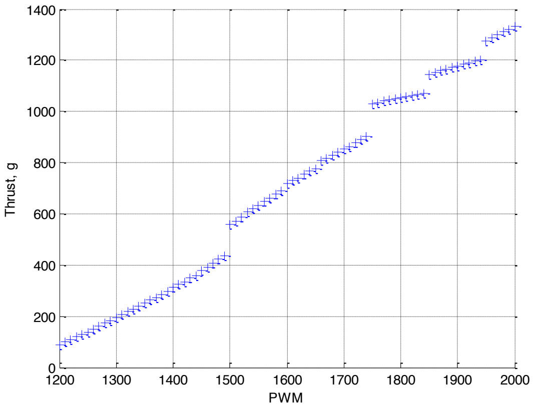

Figure 9 Thrust vs PWM

III. Mathematical Model

This section details out the development of a mathematical model of a quadrotor. The complete derivation of the kinematics and dynamics, along with an understanding of reference frames and coordinate systems, is provided. Forces and torques acting on the quadrotor and their effects are interpreted.

A. *Reference frames*

Axis systems provide a reference point or origin and a sense of positive displacement [5]. The use of several different reference systems is essential to make analysis simpler. The coordinate frames are transformed into one other through rotations and translations. Newton's equations of motion are given in the inertial frame as are GPS readings, future work like waypoint navigation etc. will also be specified in this frame. Sensors constituting the onboard IMU give readings with respect to the body frame. Forces and torques acting on the quadrotor are also evaluated in the body frame. The following reference frames shown in **Figure 10** are adopted:

1. The inertial frame, $F_i = (\vec{x}_{i,}\ \vec{y}_i, \vec{z}_i)$, is an Earth-fixed coordinate system with its origin located conveniently at the base station. It is the conventional North, East and Down (NED) frame.

2. The body frame, $F_b = (\vec{x}_{b,} \vec{y}_b, \vec{z}_b)$, has its origin at the COG of the quadrotor with its x-axis pointing forward along motor 1, y-axis pointing out to right along motor 2 and z-axis pointing out the belly.

3. The vehicle-carried NED frame, $F_v = (\vec{x}_{v,} \vec{y}_v, \vec{z}_v)$, has its axes aligned with the inertial NED frame but has its origin at the COG of the quadrotor.

Each frame can be obtained from another by performing translations and rotations. The vehicle frame can be transformed into the inertial frame by undergoing a pure translation. The body frame can be obtained from the vehicle frame by a series of rotations in a specific order. Let us define two more intermediate frames to illustrate this, the frame $F_{v\text{-}1}$ is obtained by rotating F_v positively about $\vec{z}_v$ by the yaw angle ψ, this means if roll and pitch angles are zero, it now coincides with the body frame. The frame F$_{v-2}$ is obtained by rotating this F$_{v-1}$ about the $\vec{y}_{v-1}$ by the pitch angle θ in a positive sense. Now if this frame is rotated about the $\vec{x}_{v-2}$ by the roll angle φ, we obtain the body frame. The rotation matrices are derived below.

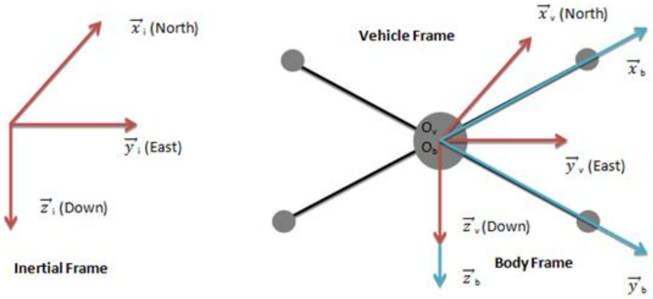

Figure 10 Reference frames

The transformation from F$_v$ to F$_{v-1}$ can be written as

$$F_{v-1} = R_v^{v-1}(\psi) F_v$$

where $$R_v^{v-1}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The transformation from F$_{v-1}$ to F$_{v-2}$ is given by

$$F_{v-2} = R_{v-1}^{v-2}(\theta) F_{v-1}$$

where $$R_{v-1}^{v-2}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

And finally the transformation from F$_{v-2}$ to F$_b$ is given by

$$F_b = R_{v-2}^{b}(\phi) F_{v-2}$$

where $$R_{v-2}^{b}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}$$

Thus the complete rotation matrix from the vehicle frame F$_v$ to the body frame F$_b$ is written as

$$\begin{aligned} R_v^b(\phi,\theta,\psi) &= R_{v-2}^{b}(\phi) R_{v-1}^{v-2}(\theta) R_v^{v-1}(\psi) \\ &= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \end{aligned}$$

where $c\theta \triangleq \cos\theta$ and $s\theta \triangleq \sin\theta$

B. Kinematics

The 6 DOF equations of motion (EOM) are more easily formulated in the body frame due to the fact that, the on board IMU measurements are in this frame, the control forces are given in this frame and the symmetry of the quadrotor simplifies equations [6]. The Euler angles are defined in the vehicle frame F$_v$ and its variations F$_{v-1}$ and F$_{v-2}$. The following vector notation will be used at all times.

$$S_i = \begin{bmatrix} x_n \\ y_e \\ z_d \end{bmatrix} \quad V_b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \omega_v = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

The velocity in the various frames of reference can be related using the rotation matrix derived above as

$$\begin{bmatrix} \dot{x}_n \\ \dot{y}_e \\ \dot{z}_d \end{bmatrix}_{F_i} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_{F_v} = R_b^v(\phi,\theta,\psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix}_{F_b} \tag{1}$$

where, $R_b^v = \left[R_v^b\right]^{-1} = \left[R_v^b\right]^T$

The angular rates p, q and r are defined in the body frame whereas the Euler angles are defined in the various vehicle frames, specifically the yaw angle ψ is defined in F$_v$, the pitch angle θ in the F$_{v-1}$ and the roll angle φ in the F$_{v-2}$. It is shown in [9 beard 2008] that

$$R_{v-2}^{b}\left(\dot{\phi}\right) = R_{v-1}^{v-2}\left(\dot{\theta}\right) = R_v^{v-1}\left(\dot{\psi}\right) = I \tag{2}$$

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= R_{v-2}^{b}\left(\dot{\phi}\right) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{v-2}^{b}(\phi) R_{v-1}^{v-2}\left(\dot{\theta}\right) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{v-2}^{b}(\phi) R_{v-1}^{v-2}(\theta) R_v^{v-1}\left(\dot{\psi}\right) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned}$$

Inverting this we get

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3}$$

C. Dynamics

Differentiation in a moving axis system is given by,

$$\frac{dS}{dt} = \dot{S} + \omega_b \times S \tag{4}$$

Where S is the position vector of a particle relative to the origin of the moving body frame and ω_b is the angular rate that the body frame is moving with respect to an inertial frame. Newton's laws of motion are only valid in inertial frames, thus we need the above relation to formulate our equations of motion in the body frame. Let F_{ext} be the force vector containing all external forces including gravity. Mass being constant, applying the above equation and the chain rule of differentiation, we can concisely write the linear EOM as,

$$F_{ext} = \frac{d\left(mV_b\right)}{dt} = m\dot{V}_b + \omega_b \times \left(mV_b\right) \tag{5}$$

Thus using our previously defined notation, this becomes

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{F_{ext}}{m} \tag{6}$$

Seckel [7] demonstrates how to develop the equations of angular motion. Let $M_{ext} = [L \ M \ N]^T$ be the moment vector containing all external moments developed, then the angular EOM can be written as,

$$M_{ext} = J\dot{\omega}_b + \omega_b \times \left(J\omega_b\right) \tag{7}$$

The quadrotor is completely symmetrical about all its axes; this conveniently renders the cross terms in the inertia matrix J to be zero. Thus $J_{xy} = J_{yz} = J_{xz} = 0$ and our inertia matrix J is given by,

$$J = \begin{pmatrix} j_x & 0 & 0 \\ 0 & j_y & 0 \\ 0 & 0 & j_z \end{pmatrix} \qquad J^{-1} = \begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix}$$

The moments of inertia are formulated according to [8] with the quadrotor modeled as a spherical dense centre of mass M_c and radius R_c and four point masses m_{mot} located at the arm length l.

$$J_x = \frac{2M_cR_c^2}{5} + 2l^2 m_{mot} = 0.0296Kgm^2$$

$$J_y = \frac{2M_cR_c^2}{5} + 2l^2 m_{mot} = 0.0296Kgm^2$$

$$J_z = \frac{2M_cR_c^2}{5} + 4l^2 m_{mot} = 0.0566Kgm^2$$

Thus the angular EOM can be written as,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{L}{J_x} \\ \frac{M}{J_y} \\ \frac{N}{J_z} \end{bmatrix} \tag{8}$$

D. Forces and Moments

The forces and moments developed are due to gravity and the thrust and torques developed by the four propellers. **Figure 11** shows the top view of the quadrotor and the forces developed by each motor. **Figure 12** shows the forces and torques acting on the quadrotor and positive sense of rotation for roll, pitch and yaw.

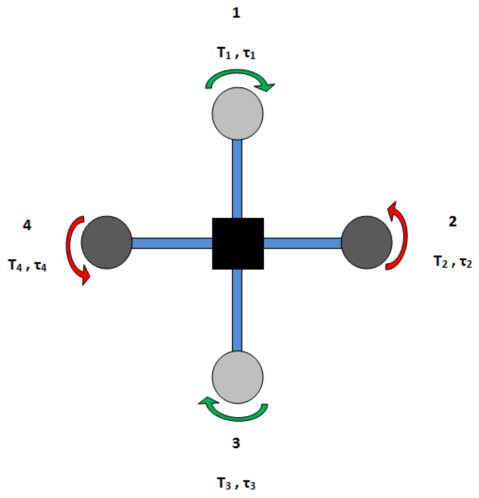

Figure 11 Top view of the quadrotor

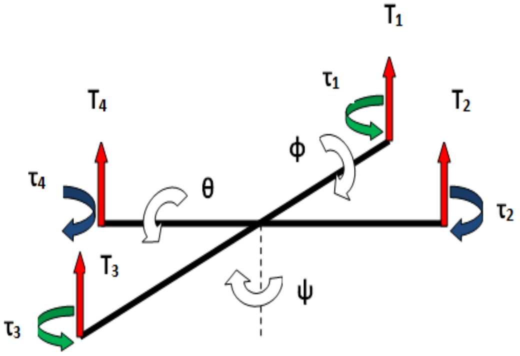

Figure 12 Force and moment definitions

The total force acting on the quadrotor is,

$$T = T_1 + T_2 + T_3 + T_4$$

The rolling torque L is produced by the thrust difference between motors 2 and 4 as,

$$L = l\left(F_4 - F_2\right)$$

The pitching torque M is produced by the thrust difference between motors 1 and 3 as,

$$M = l\left(F_1 - F_3\right)$$

The yawing torque N is produced by the total difference in clockwise and counterclockwise torques generated by all four motors as,

$$N = \tau_2 + \tau_4 - \tau_1 - \tau_3$$

F$_{ext}$ thus has only a f$_z$ component i.e. –T, but the gravitational force also needs to be given in the body frame,

$$\begin{aligned} F_{ext} &= R_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \\ &= \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \end{aligned} \qquad (9)$$

Thus equations (1)-(9) become,

$$\begin{bmatrix} \dot{x}_n \\ \dot{y}_e \\ -\dot{z}_d \end{bmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (10)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -g\sin\theta \\ g\cos\theta\sin\phi \\ g\cos\theta\cos\phi \end{bmatrix} + \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \qquad (11)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad (12)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{J_y - J_z}{J_x} qr \\ \dfrac{J_z - J_x}{J_y} pr \\ \dfrac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \dfrac{L}{J_x} \\ \dfrac{M}{J_y} \\ \dfrac{N}{J_z} \end{bmatrix} \qquad (13)$$

Equations (10)-(13) provide the complete 6 DOF EOM in the body frame of reference.

IV. Fuzzy Flight Controller

The intelligent control strategy proposed is presented in detail in this section. The Quadrotor was flown with a traditional TX-RX, utilizing the open source flight software available, to gain better insight into the dynamics of such an interesting UAV. The open loop model was run multiple times: dynamic responses were observed and tracked to achieve the heuristic understanding required before delving into the Fuzzy logic control process. The limits for the pitch and roll angles were derived via flight tests. Wireless telemetry also allowed the observation of typical acceleration and angular velocity values which helped in range setting of the FLC's (Fuzzy logic controller).

A. *Designing with Fuzzy Logic*

Fuzzy logic is a misnomer. Developed by Lotfi.A.Zadeh (1965), it is firmly grounded in mathematical theory [9]. FLC is a digital control methodology that emulates human decision making by allowing partial truths i.e. multivalued logic. It allows for the imprecision inherent in real world scenarios. Studying Fuzzy logic as a graduate student at the University of Cincinnati under Dr. Kelly Cohen, I was fascinated by the immense potential it offered. Completing key projects such as pitch control of an F-4 fighter jet, classic inverted pendulum balance problem, etc. provided a strong foundation in FLC. The output of a fuzzy system is smooth and continuous thereby lending itself to the control of continuously variable systems such as electric motors. In general, FLC is used when the system to be controlled is nonlinear, time variant and ill defined.

B. *Development*

Fuzzy logic manipulates inputs using the heuristically developed rule base and converts them into outputs. An overview of this process is shown in **Figure 13** below.

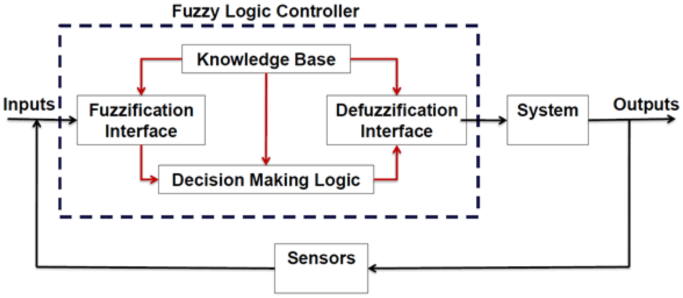

Figure 13 Control process overview

The flight controller must perform two separate tasks simultaneously: Control the Quadrotor's position as well as stabilize its attitude. The mathematical model illuminates the fact that X and Y motions are coupled with pitching and rolling motions respectively. Thus in all six states namely X-Coordinate, Y-Coordinate, Z-Coordinate, heading angle (ψ), roll angle (φ) and pitch angle (θ) have to controlled simultaneously. Accordingly Six FLC's have been designed to control each state; the FLC X and FLC Y are cascaded with FLC θ and FLC φ respectively, whereas FLC Z and FLC ψ are stand-alone controllers. Each FLC has two inputs namely error and error rate and one output which is the Δ PWM value.

MATLAB's Fuzzy Logic toolbox was used to develop the controllers, which use the Mamdani-type inference method and the centroid method for defuzzification.

A predefined bias "offset" is used to counteract the weight of the quadrotor. Translation in the +X direction is achieved by a pitch down i.e. nose down movement, hence PWM_1 = offset – uX and PWM_3 = offset + uX, while the other motors are unaffected i.e. PWM_2 = PWM_4 = offset. This keeps the net lift constant and the net torque zeroed. Similarly ±Y translation is achieved by rolling right/left respectively, altering the speeds of motors 2 and 4, and keeping that of 1 and 3 constant. Z translation is achieved by increasing/decreasing the speeds of all the motors by the same amount i.e. uZ. Control of the heading angle requires an unbalanced torque, this is manipulated by either increasing speeds of the clockwise motors (1, 3) and decreasing that of the counterclockwise motors (2, 4) by the same amount, or vice versa, thus producing an anticlockwise or clockwise net moment. The desired roll and pitch angles are always fed as zero so as to maintain the stability of the craft, when a X or Y translation is desired, the FLC X or FLC Y anticipate the desired pitch and roll angles continuously and the FLC θ or FLC φ achieve these angles by manipulating the speeds of the motors as above. The roll and pitch angles anticipated have set limits and are chosen empirically in such a way so as to always stabilize the attitude of the craft. It should be noted that in this research effort, priority has been given to achieving quick stable responses and not to aggressive maneuvering.

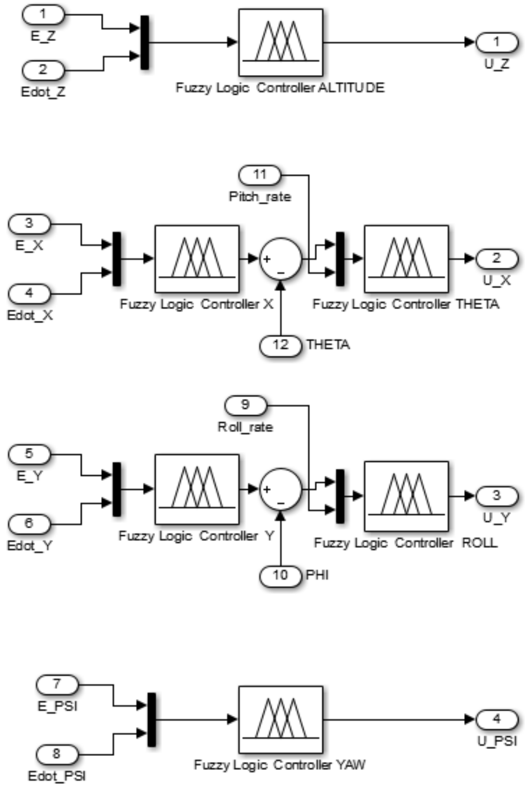

Figure 14 Control scheme

The control scheme presented in the **Figure 14** above results directly from an understanding gained during initial flight tests, the logic described above and from the interpretation of coupling of motions by studying the formulated mathematical model.

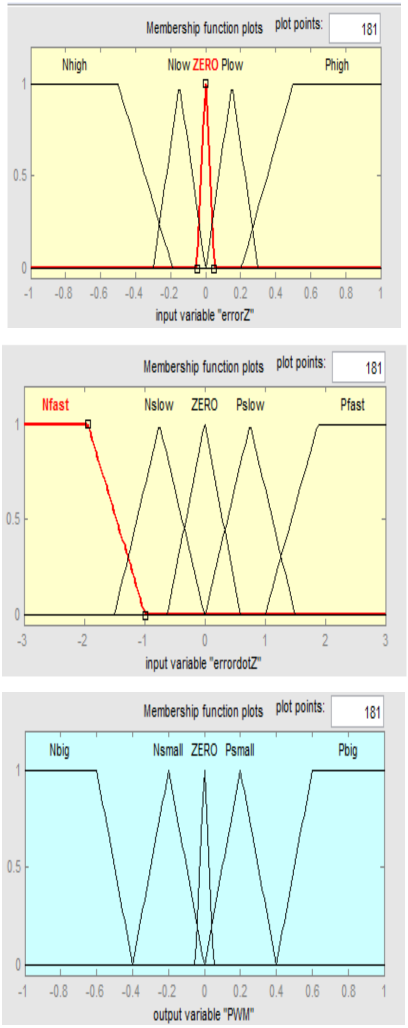

Figure 15 Membership functions: (a) Error; (b) Error rate; and (c) Output

C. Membership functions and rule base

For the FLC's related to position, the error is normalized to the interval [-1 1], the error rate is confined to the range [-3 3] and the output is also normalized to the range [-1 1]. For example, in the case of FLC Z, error is e = (desired_Z) – (actual_Z) and error rate is given by $\dot{e} = -\dot{Z}$ and output is uZ. The physical significance of confining the error rate range is to restrict the velocity in NED directions to ± 3 m/s, this was set empirically and it ensures a fast yet stable response with very low offshoot. For the sake of modularity, the error rates are fed as the translational/rotational rates directly and the rule base is developed accordingly. A combination of triangular and trapezoidal shapes was used to construct the membership functions. Identical membership functions are used for all inputs and outputs. The placement and relative placement of membership functions on the universe of discourse was far more influential on the result than their respective shapes. As a consequence of this, more time was spent on range setting and tweaking the functions rather than selection of shapes. The linguistic variables used for error are negative/positive high/low and zero, for error rate negative/positive fast/slow and zero, for output negative/positive big/small and zero. The **Figure 15** below shows the membership functions developed for error, error rate and output.

Rules are developed as if-then statements based on heuristics and experience. For example, IF errorZ is “Nhigh” and errordotZ is “Pfast” THEN PWM is “Nbig”. The heuristic interpretation of this rule is as follows: if actual altitude is much higher than desired, thus error is negative high, and the error rate i.e velocity in Z direction is positive fast meaning a high upward velocity, then the output is negative big, i.e. a sharp decrease in PWM value for each motor thereby causing quick deceleration. A total of 25 rules have been developed for each FLC. The complete rule base is presented in the table below using the linguistic variables previously defined.

TABLE III RULE BASE

E Ė	Nhigh	Nlow	Zero	Plow	Pfast
Nfast	Zero	Psmall	Psmall	Pbig	Pbig
Nslow	Nsmall	Zero	Psmall	Psmall	Pbig
Zero	Nbig	Nsmall	Zero	Psmall	Pbig
Pslow	Nbig	Nsmall	Nsmall	Zero	Psmall
Pfast	Nbig	Nbig	Nbig	Nsmall	Zero

V. SIMULATION RESULTS

The closed loop controller and the quadrotor simulator have been implemented in MATLAB Simulink [10]. The input to the quadrotor block are the four motor PWM values, it outputs the linear and angular accelerations that are twice integrated to obtain linear and angular velocities and positions. The model can effectively be broken into four major blocks. The Pre-processor receives the user defined inputs of altitude, X coordinate, Y coordinate and heading angle and calculates, normalizes and outputs the errors to the controller block. The controller block contains the six FLC’s and takes as its inputs the errors and the error rates of all the states and outputs the control variables Uz, uX, uY and uPSI. The aggregator block fuses all the control actions into individual PWM motor values as shown below. The EOM are embedded in the quadrotor block of the simulator.

$$PWM_1 = offset + uZ - uX + uPSI$$

$$PWM_2 = offset + uZ - uY - uPSI$$

$$PWM_3 = offset + uZ + uX + uPSI$$

$$PWM\,4 = offset + uZ + uY - uPSI$$

A large number of open loop simulations were performed initially to better understand the quantitative effect of changing PWM values. Each controller loop was successively closed and the parameters were fine-tuned in the end wherein all six FLC’S were running in tandem. The controller has been developed such that it can effectively handle multiple coupled inputs. The purpose of these extensive simulations is to prove the efficacy of the proposed controller prior to future hardware-in-the-loop simulations.

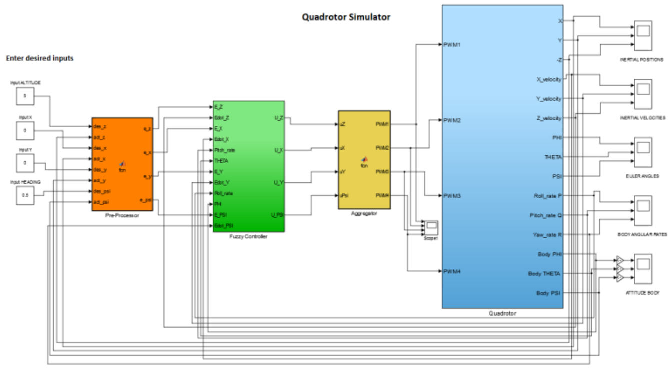

Figure 16 MATLAB Simulink quadrotor model

An altitude perturbation of 10 m is shown in **Figure 17**. It can be seen that the roll and pitch angles are within 2-3 degrees and are stabilized fairly quickly. The system reaches its intended position within 6 seconds.

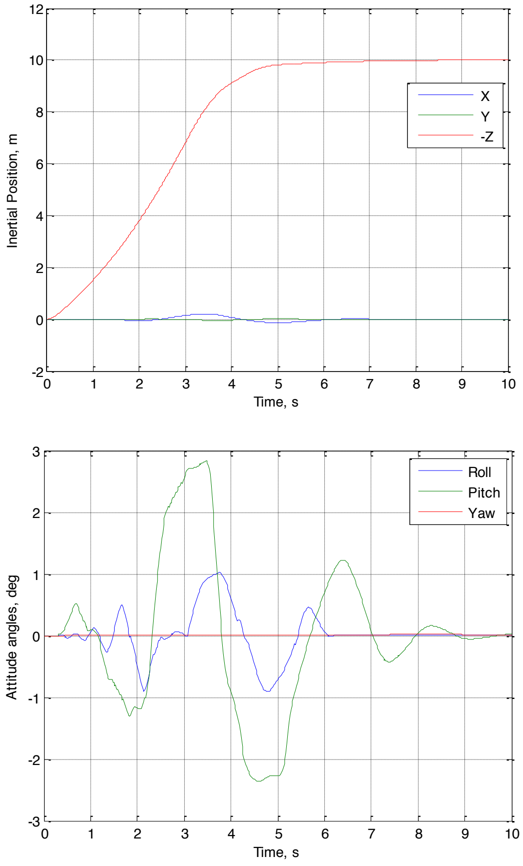

Figure 17 Altitude simulation from ground to 10 m

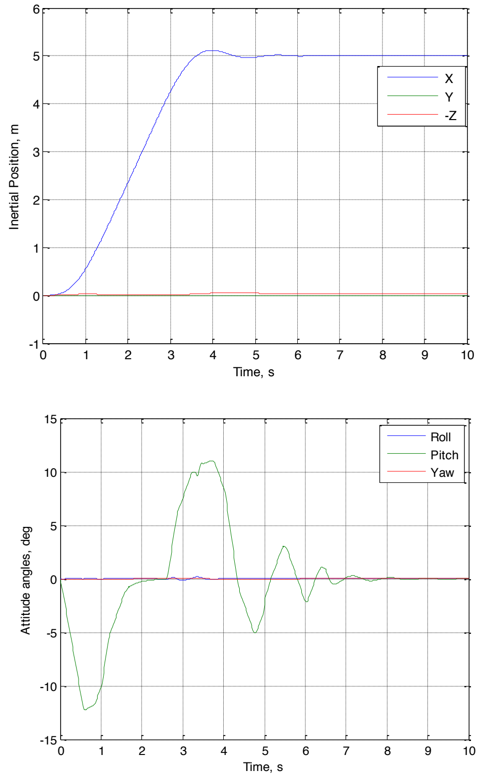

Figure 18 X position simulation from 0 to +5 m

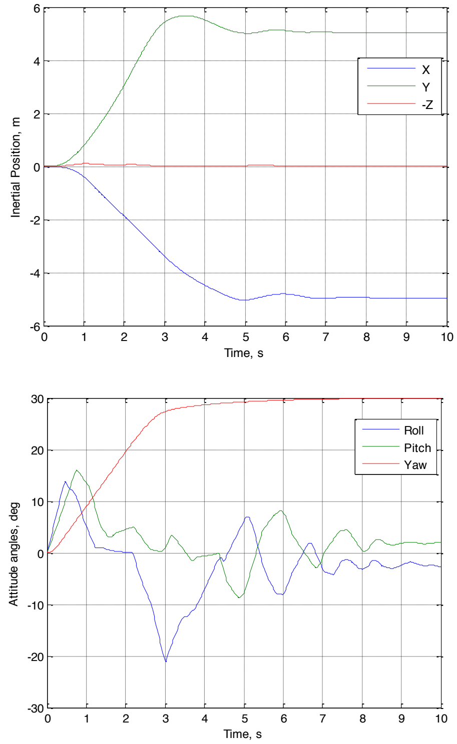

Figure 19 Coupled motion simulation of X= -5 m, Y= +5 m, and ψ = 30°

Figure 18 shows the response to an X coordinate perturbation of 5m and the body angle plot. A quick stable response is obtained while not producing unwanted motion in other planes. The controller is intended to be utilized such we can give successive positional and heading commands in order to carry out a desired trajectory in real time i.e. for example rising to an altitude of 10 m and then traversing a square trajectory of side 5 m. Future work will include a GUI in MATLAB where top level commands may be sent to the quadrotor. **Figure 19** shows a coupled X and Y perturbation along with a simultaneous heading angle perturbation of 30 degrees.

VI. Experimental Validation

The validation of the math model developed is of prime importance to this research endeavor. It is clear that the simulation model must be validated with actual flight data to lend high confidence to the results obtained for the fuzzy logic controller developed. A specific methodology was derived in order to go about the validation process in a comprehensive manner.

With implementation efforts for the fuzzy logic controller still in early stages, it was decided to conduct the validation with the already functional PID based stock controller which was part of the open source Aeroquad project. Code was written in the Arduino IDE environment to allow for the logging of the controlled motor outputs at 50 Hz for post processing. This development took significant investment of time to achieve clean coherent data that could be imported into MATLAB & Simulink as an input to the Quadrotor Simulink model.

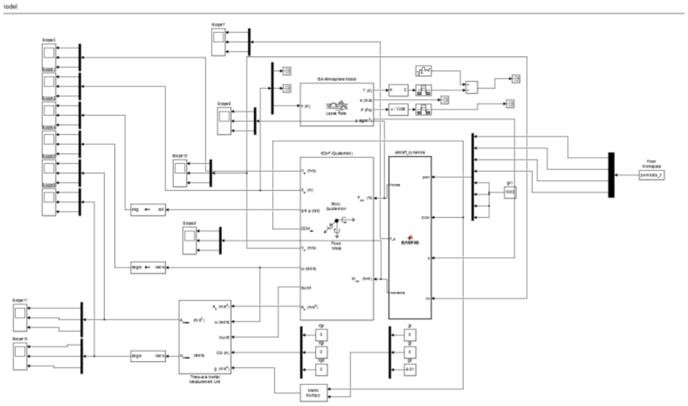

Figure 20 Validation model in Simulink

The validation model presented below in **Figure 20** shows a MATLAB timeseries data block that feeds the logged motor PWM signals into the quadrotor dynamic model. The mask over the Quadrotor dynamics's block from **Figure 16** (Blue block 'Quadrotor') has been lifted in the above figure in the interest of clarity and supporting future researchers. The Validation model consists of five distinct blocks, namely the Timeseries data block mentioned above, the flight dynamics block that converts the motor PWM inputs into forces and moments in the body frame, the 6 DOF quaternion block that preexists in Simulink, the three axis inertial measurement unit block that mimics the IMU onboard for accelerometer and gyroscopic data and lastly the atmosphere model block that incorporates environmental effects. The model is an outdoor simulation model that takes into account air density, temperature and pressure. Modeling of wind effects was not done primarily to keep complexities out of the model and ensure flight test data is free from external forces as far as possible.

An experimental flight of about one minute was conducted for validation purposes and the flight envelope consisted of non-aggressive maneuvers and stable hovering with pure altitude perturbations. Roll and pitch i.e. movements from the X-Y plane were performed separately and not overlapped in order to identify and correct the model for discrepancies.

It should be mentioned here that fairly accurate results have been obtained while keeping in mind certain factors such as the data logging could only be accomplished at 50 Hz for attitude as well as motor output simultaneously, but the actual controller updates the motor commands at 100 Hz, thus causing loss of significant number of data points. This is the reason why the validation for now cannot be performed for aggressive flight conditions. Also the PID controller employed cannot be simulated accurately as the internal workings remain unidentifiable. Thus while it would make more sense to simulate using the pilot's joystick inputs i.e. throttle, aileron, elevator and rudder, this is not possible at this moment in the development cycle. It is hoped that future validation work would follow the above thought process.

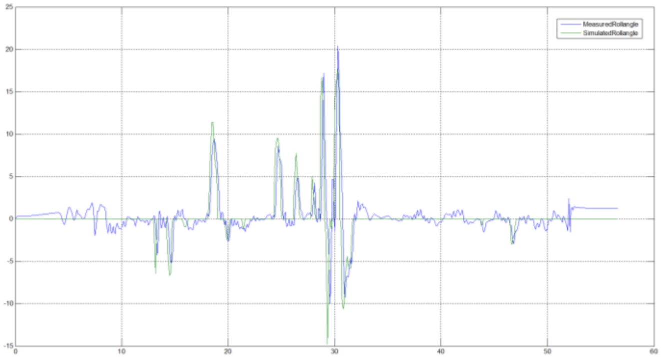

Figure 21 Measured roll angle (phi_flight) vs simulated roll angle (phi_sim)

Figure 21 shows the measured and simulated roll angles in degrees vs. time in seconds. It is seen quite clearly that the simulation follows both the trend and the magnitude of the flight roll angles. The small roll angles do not show as much coherence as larger angles simply because the IMU data is quite jittery and noise inclusive and the moments produced under a particular value are neglected by the simulator for accuracy reasons. However, the simulator captures the flight dynamics to a very high extent. **Figure 22** below shows the measured pitch angle vs. the simulated pitch angle.

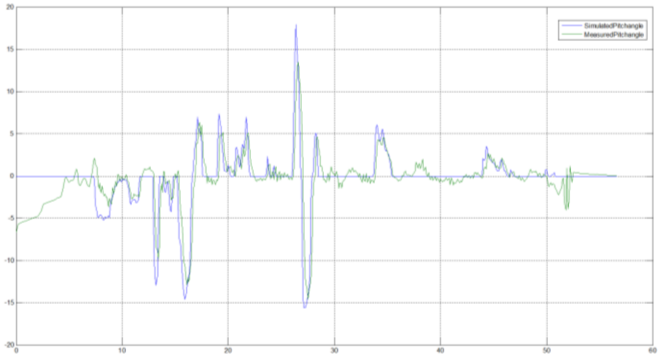

Figure 22 Measured pitch angle (theta_flight) vs simulated pitch angle (theta_sim)

Similar to the roll angle tracking, it is observed that a significantly high coherence exists between the actual flight attitude data and the simulated body angles. **Figures 23** and **24** show plots for the rotor forces that are developed for the above flight as well as moments generated in the body frame of reference.

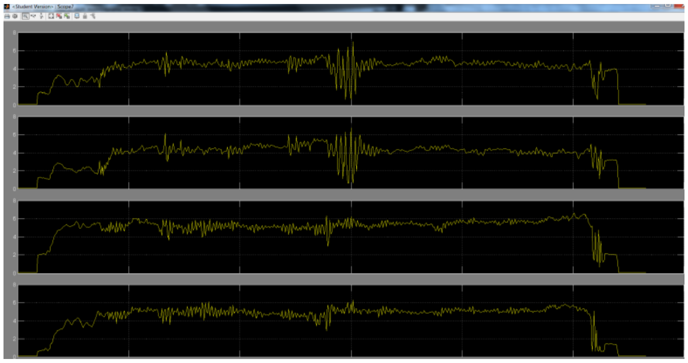

Figure 23 Plot of forces developed by each rotor for the validation flight example

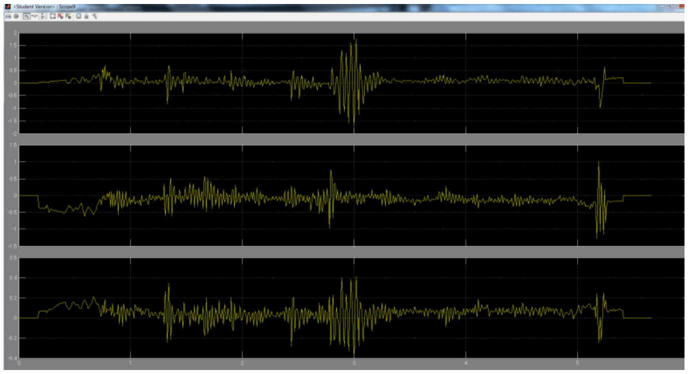

Figure 24 Plot of moments in each axis for the validation flight example

The rotor forces developed are between magnitudes of 0 – 7 N which are typically expected, the hover rotor force for each rotor is experimentally seen to be 4.75 N. Some considerations were made in order to account for the lower frequency of data logging that proved invaluable in cutting off low degree oscillatory behavior. It should be noted here that comparing simulations to flight test data and extracting meaningful results is a very time intensive and recursive effort. The plot of the moments developed in each axis shows highly oscillatory behavior due to the fact that the roll and pitch angles computed form the IMU data by the flight controller are rarely stable, even non zero when not in actual flight and thus the controller always produces corrective action. This results in the actual PWM motor outputs to vary a lot even in stationary hover conditions. To account for this, a cutoff moment value is set so as to only capture slightly larger oscillations, in the range of 4-40 degrees in the roll and pitch axes. This is why the simulator is able to appreciably capture the real dynamics of the system. Moments in the yaw axis were developed due to the varying motor outputs but the offset set for this axis more or less zeroed out the final effects and this is in accordance to what was observed during the actual flight test.

VII. Conclusion

This paper deals with the mathematical modeling and control of a quadrotor UAV. An intelligent fuzzy flight controller has been proposed, designed and built using MATLAB's fuzzy logic toolbox. A simulation environment was developed using the EOM derived and simulations show the mathematical model to be realistic in modeling position and orientation. Responses prove the effectiveness of the six FLC's in controlling position as well as achieving stability.

The presented mathematical model consists only of the basic structures of the quadrotor's dynamics. Several phenomena were not modeled due to their complexity and associated uncertainty. It is hoped to be proved experimentally that the inherent robustness fuzzy logic control embodies will deal with these shortcomings.

Experimental validation is carried out with actual flight data thus lending high confidence to the math model developed. It is seen that the tracking of the attitude data is particularly encouraging; there are of course several considerations with respect to the simulations that have been made. The high coherence demonstrated brings significance in this research effort and the simulation results for the fuzzy flight controller. It is worth mentioning here in the interest of redundancy that the simulator gives good results for near hover flight regimes and low velocities, the inherent robustness that FLC demonstrates should account for controllability across the entire flight envelope.

Future work is being directed towards achieving a completely autonomous flight in controlled indoor conditions. Wireless communication using the Xbee modules and incorporation of data packet loss redundancies has been successfully achieved. A more comprehensive simulation environment including wind disturbances and sensor noise inclusion is being developed. The most challenging obstacle is motion and attitude estimation using the IMU measurements. Real time quadrotor-in-the-loop simulations will allow comparison with the simulation results obtained and would bring significant benefits to this research endeavor

References

[1] Sabo, Chelsea, and Cohen, Kelly, "Fuzzy Logic Unmanned Air Vehicle Motion Planning," *Advances in Fuzzy Systems*, Vol. 2012, No. 13, Jan. 2012. CrossRef

[2] Bresciani, Tommaso. *Modelling, identification and control of a quadrotor helicopter*. Department of Automatic Control, Lund University, 2008.

[3] Premerlani, William, and Paul Bizard. "Direction cosine matrix imu: Theory."*DIY DRONE: USA* (2009): 13-15.

[4] Bouabdallah, Samir, Andre Noth, and Roland Siegwart. "PID vs LQ control techniques applied to an indoor micro quadrotor." *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 3. IEEE, 2004.

[5] Dreier, E. M., *Introduction to Helicopter and Tiltrotor Simulation*, AIAA Education Series, AIAA, VA 20191, 2007, pp. 27.

[6] Bouabdallah, Samir, and Roland Siegwart. "Full control of a quadrotor."*Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*. IEEE, 2007.

[7] Seckel, Edward. *Stability and control of airplanes and helicopters*. Academic Press, New York, 1964.

[8] Raza, Syed Ali, and Wail Gueaieb. "Intelligent Flight Control of an Autonomous Quadrotor." *InTech* (2010).

[9] Correspondent, Kevin Self. "Designing with fuzzy logic." *IEEE Spectrum, Nov* (1990): 42-44.

[10] MATLAB 8.0 and Simulink R2012b, The MathWorks, Inc., Natick, Massachusetts, United States.