Modeling, Simulation and Flight Testing of an Autonomous Quadrotor

Rahul Goel[a], Sapan M. Shah[b], Nitin K. Gupta[c,1], N. Ananthkrishnan[c,d]

[a]Department of Aeronautics and Astronautics, MIT, USA
[b]Whirlybird Electronics Pvt. Ltd., Mumbai
[c]IDeA Research and Development Pvt. Ltd., Pune
[d]Coral Digital Technologies Pvt. Ltd., Bangalore

Abstract: Rotary-wing Unmanned Mini Aerial Vehicles (RUMAV) represent a useful class of flying robots because of their strong abilities of VTOL, high maneuverability and controllability, especially in enclosed areas. In this paper, we present the development of an autonomous four-rotor RUMAV, called Quadrotor. Starting with modeling, simulation, and control design, this paper presents the results from flight experiments conducted on a flying platform. A classical control approach (PID) is used to design the control law. Once the control algorithm is validated using simulations and 3D visualization, it is implemented on hardware and experiments on a test-rig and in free flight are conducted.

Key words: Quadrotor, automation, dynamics, modeling, simulation, control, autopilot

NOMENCLATURE

A	rotation matrix
A_p	area of propeller
B	body fixed frame
D	drag moment
F_{wI}	forces due to translational velocity and wind
G	gear ratio
h	altitude of vehicle
I	inertial frame
I_{xx}, I_{yy}, I_{zz}	quadrotor moments of inertia about body x, y, z axes, respectively
I_{ct}	inertial counter torque
J	moment of inertia matrix
J_p	moment of inertia of single propeller
k_d	aerodynamic drag moment coefficient
k_i	current constant of motor
k_r	friction coefficients due to rotational velocities
k_s, k_u	friction coefficients due to translational velocities
k_t	aerodynamic thrust coefficient
k_v	speed constant of motor
L	distance of center of propeller from origin
M	matrix relating Euler time derivatives with body angular rates
M_f	friction torque
M_g	gyroscopic moments
m	mass of quadrotor vehicle assembly
P	pitch of propeller blade
R	resistance of motor
R_p	radius of propeller
T	thrust force
V	voltage applied to motor
w	wind velocity
$\dot{X}_i'$, $\dot{Y}_i'$	forward and sideward velocity in horizontal plane
$\dot{X}_i$, $\dot{Y}_i$	velocities in inertial frame
α	angular speed of propeller
ϕ, θ, ψ	Euler angles
ϕ_b, θ_b	base Euler angles commanded
τ	torque
τ_t	disturbance torque
τ_m	motor torque
ω	angular rates of quadrotor in body frame

Subscripts

$1, 2, 3, 4$	propeller number
b	coordinates in body frame
c	commanded value
d	desired value
i	coordinates in inertial frame
o	obtained value

Email addresses: rahulg@mit.edu (Rahul Goel), sapan2118@gmail.com (Sapan M. Shah), nitin@idearesearch.in (Nitin K. Gupta), akn@idearesearch.in (N. Ananthkrishnan)
URL: http://www.idearesearch.in (Nitin K. Gupta)
[1]Corresponding author

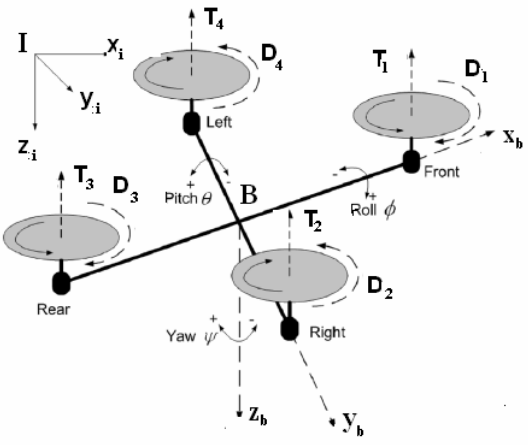

Figure 1: Quadrotor Axis System

1. INTRODUCTION

Unmanned Mini Aerial Vehicles (UMAVs) have found potential applications in military purposes like surveillance, border patrolling, mine detection, crowd control, aerial delivery of payload, civilian purposes like disaster management during floods, earthquakes, fire, commercial missions like aerial photography, television and cinema shootings, and research and development program which need flying vehicles to perform various experiments. The UMAVs are expected to become much more common soon because of their potential in preventing pilot exposure to danger. In this paper we discuss a kind of Rotary-wing Unmanned Mini Aerial Vehicle (RUMAV), called Quadrotor. Quadrotor is a system consisting of four independent propellers attached at each corner of a cross frame (Fig. 1). It has vertical take-off and landing (VTOL) capabilities which gives it higher maneuverability and hovering capabilities.

The concept of Quadrotor was first experimented in 1907 by Breguet and Richet [1]. Their large, heavy, manned vehicle is reported to have lifted only over a small height and for short duration. Since then, there has been tremendous research and development in the field of rotorcraft, but the Quadrotor concept did not get much attention. However, over the last few years, many groups [2, 3, 4, 5, 6, 7] are working in order to exploit the potential advantages of Quadrotors as RUMAVs of future. This paper presents a complete system solution to development of a Quadrotor starting from modeling, simulation, controller design to implementation on hardware and conducting flight tests.

As shown in Fig. 1, one pair of opposite propellers of Quadrotor rotates clockwise, whereas the other pair rotates anticlockwise. This way it is able to avoid the yaw drift due to reactive torques. This configuration also offers the advantage of lateral motion without changing the pitch of the propeller blades. Fixed pitch simplifies rotor mechanics and reduces the gyroscopic effects. Control of Quadrotor is achieved by commanding different speeds to different propellers, which in turn produces differential aerodynamic forces and moments. For hovering, all four propellers rotate at same speed. For vertical motion, the speed of all four propellers is increased or decreased by the same amount, simultaneously. In order to pitch and move laterally in that direction, speed of propellers 3 and 1 is changed conversely. Similarly, for roll and corresponding lateral motion, speed of propellers 2 and 4 is changed conversely. To produce yaw, the speed of one pair of two oppositely placed propellers is increased while the speed of the other pair is decreased by the same amount. This way, overall thrust produced is same, but differential drag moment creates yawing motion. In spite of four actuators, the Quadrotor is still an under-actuated system.

The rest of the paper is organized as follows: Section 2 discusses the dynamic modeling. Section 3 presents the control law design and simulation results. Section 4 illustrates the integrated model with 3D visualization. Finally, hardware implementation and flight test results are presented in Section 5.

2. QUADROTOR DYNAMIC MODELING

Let us consider an inertial frame, I and a body frame, B as shown in Fig. 1. Using Euler angle parameterization, the orientation of vehicle in space is given by rotation matrix A (given below) from frame B to I. The dynamic model is derived under the following assumptions:

1. Structure is rigid and symmetrical.
2. The center of mass of vehicle and origin of B coincide.
3. The propellers are rigid in plane. Pitch is fixed.

Euler time derivatives are related to body angular rates as:

$$\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T = M^{-1} \begin{bmatrix} \omega_{x_i} & \omega_{y_i} & \omega_{z_i} \end{bmatrix}^T = M^{-1} A \begin{bmatrix} \omega_{x_b} & \omega_{y_b} & \omega_{z_b} \end{bmatrix}^T \quad (1)$$

where, $M = \begin{pmatrix} \frac{C\psi}{C\theta} & \frac{S\psi}{C\theta} & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$

and, $A = \begin{pmatrix} C\psi C\theta & C\psi S\theta S\phi - S\psi C\phi & C\psi S\theta C\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{pmatrix}$

where, $C\phi$, $S\phi$, etc. are $\cos\phi$, $\sin\phi$, etc.

Since we are only concerned about the velocity of centre of mass located at origin of B, we can directly get body frame velocities from inertial frame velocities, using the transformation matrix as:

$$\begin{bmatrix} \dot{x}_b & \dot{y}_b & \dot{z}_b \end{bmatrix}^T = A^{-1} \begin{bmatrix} \dot{x}_i & \dot{y}_i & \dot{z}_i \end{bmatrix}^T \quad (2)$$

2.1. Force Equations

Aerodynamic force (thrust) of a propeller can be shown proportional to square of its rotational speed using momentum theory [8]. It is modeled as:

$$T_i = C_1 \left(\frac{1 - 2\pi LCS}{P\alpha_i} + 2\pi \frac{\dot{z}_b - w_{z_b}}{P\alpha_i} \right) \quad (3)$$

where, $C_1 = k_t \rho A_p \alpha_i^2 R_p^2$

$C = 1$ if $i = 1$ or 4, or $C = -1$ if $i = 2$ or 3, and $S = \omega_{yb}$ if $i = 1$ or 3, or $S = \omega_{xb}$ if $i = 2$ or 4.

Forces due to translational velocity of quadrotor and wind disturbances are modeled as:

$$F_{wI} = A \begin{bmatrix} k_s (w_{x_b} - \dot{x}_b) & k_s \left(w_{y_b} - \dot{y}_b\right) & k_u (w_{z_b} - \dot{z}_b) \end{bmatrix}^T \quad (4)$$

Hence, linear momentum balance in inertial frame gives:

$$\begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \\ \ddot{z}_i \end{bmatrix} = -\begin{bmatrix} \omega_{x_b} \\ \omega_{y_b} \\ \omega_{z_b} \end{bmatrix} \times \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} + g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{F_{wI}}{m} - \frac{T_1 + T_2 + T_3 + T_4}{m} A \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (5)$$

2.2. Moment Equations

Aerodynamic drag moment of a propeller can be shown to be proportional to square of its rotational speed using momentum theory [8]. It is modeled as:

$$D_i = C_2 \left(1 - \frac{2\pi LCS}{P\alpha_i} + 2\pi \frac{\dot{z}_b - w_{z_b}}{P\alpha_i}\right) \quad (6)$$

where, $C_2 = k_d \rho A_p \alpha_i^2 R_p^3$

Inertial counter torque, which is the reaction torque produced by a change in rotational speed of propellers is modeled as:

$$I_{ct} = J_p \left(-\dot{\alpha}_1 + \dot{\alpha}_2 - \dot{\alpha}_3 + \dot{\alpha}_4\right) \quad (7)$$

Friction torque due to rotational motion is modeled as [4]:

$$M_f = k_r \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \quad (8)$$

Disturbance torque due to uncontrollable factors (wind etc.) is modeled as:

$$\tau_d = \begin{bmatrix} \tau_{x_b} & \tau_{y_b} & \tau_{z_b} \end{bmatrix}^T \quad (9)$$

Gyroscopic moments, caused by combination of rotations of four propellers and vehicle frame are modeled as:

$$M_g = J_p \begin{bmatrix} \dot{\theta}\alpha & \dot{\phi}\alpha & 0 \end{bmatrix}^T \quad (10)$$

where, $\alpha = -\alpha_1 + \alpha_2 - \alpha_3 + \alpha_4$

Hence, angular momentum balance in body frame gives:

$$\begin{bmatrix} \dot{\omega}_{x_b} \\ \dot{\omega}_{y_b} \\ \dot{\omega}_{z_b} \end{bmatrix} = -J^{-1} \omega \times J \begin{bmatrix} \omega_{x_b} \\ \omega_{y_b} \\ \omega_{z_b} \end{bmatrix} - J^{-1} \left(M_f + \tau_d + M_g\right) + J^{-1} \begin{bmatrix} L(T_4 - T_2) \\ L(T_1 - T_3) \\ D_1 - D_2 + D_3 - D_4 + I_{ct} \end{bmatrix} \quad (11)$$

where, $\omega\times = \begin{bmatrix} 0 & -\omega_{z_b} & \omega_{y_b} \\ \omega_{z_b} & 0 & -\omega_{x_b} \\ -\omega_{y_b} & \omega_{x_b} & 0 \end{bmatrix}$

Table 1: Quadrotor Parameters [5]

Design Variable	Value	Units
m	4.493	Kg
L	0.38	m
G	80/12	-
J_p	1.46E-3	Kg/m^2
I_{xx}, I_{yy}	0.177	Kg/m^2
I_{zz}	0.334	Kg/m^2
R_p	0.228	m
P	0.152	m
V	5	Volts
R	0.3	Ohms
k_t	0.008	-
k_d	0.0013	-
k_i	3.87E-3	Nm/Ohms
k_v	0.0004	Volts/RPM
k_r	0.35	Nms/rad
k_s, k_u	1	Ns/m

2.3. Motor Dynamics

A standard DC motor with negligible inductance is modeled as:

$$\begin{aligned} \tau_{m_i} &= k_i \left(V_i - \frac{k_v \alpha_i}{G}\right)/R \\ \dot{\alpha}_i &= \frac{G\tau_{m_i} - D_i}{J_p} \end{aligned} \quad (12)$$

Table 1 gives the parameters of Quadrotor considered.

3. CONTROLLER DESIGN AND SIMULATION

3.1. Controller

Controller design for the system involves two loops: inner and outer. In the inner loop, four parameters: θ, ϕ, ψ and h are independently controlled by suitably adjusting the speed of propellers. For controlling θ, speed of propellers 1 and 3 is adjusted. For controlling ϕ, speed of propellers 2 and 4 is modified. For controlling ψ and h, speed of all the four propellers is suitably adjusted. Four separate PIDs are built corresponding to each parameter. Based on the error signals, the inner loop PIDs command differential voltages in order to reach the set point.

In the outer loop, forward and sideward velocities ($\dot{X}_i'$ and $\dot{Y}_i'$) are controlled (Eq. 13). The velocities commanded are in a frame which is obtained by rotating the inertial frame by ψ. Since a joystick is used for navigating the Quadrotor it is found more practical to command these velocities. Based on simulation results, look up tables are formulated, to find base θ and ϕ required to fly at a particular velocity. These base angles (θ_b and ϕ_b), act as set points for inner loop controllers of θ and ϕ. The outer loop PIDs command additional pitch ($\Delta\theta$) and roll ($\Delta\phi$) angles based on error in velocities, in order to reach the set point.

Table 2: PID Gains Table

Control	Output	K_p	K_i	K_d
θ	$+V_1$, $-V_3$	0.05	0	0.02
ϕ	$+V_4$, $-V_2$	0.05	0	0.02
ψ	$+V_1$, $-V_2$, $+V_3$, $-V_4$	0.005	0	0.004
h	$+V_1$, $+V_2$, $+V_3$, $+V_4$	0.01	0.0007	0.01
$\dot{X}_i'$	$-\Delta\theta$	-1.5	-0.06	0
$\dot{Y}_i'$	$\Delta\phi$	1.5	0.06	0

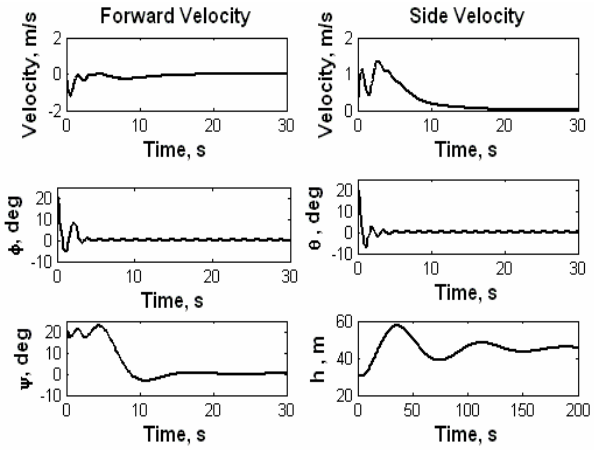

Figure 2: Starting from $\phi = \theta = \psi = 18°$, and $h = 30m$, the system stabilizes at zero Euler angles, and attains $h = 45m$

$$\begin{aligned} \dot{X}_i' &= \dot{X}_i \cos\psi + \dot{Y}_i \sin\psi \\ \dot{Y}_i' &= -\dot{X}_i \sin\psi + \dot{Y}_i \cos\psi \end{aligned} \quad (13)$$

PID gains obtained are presented in Table 2.

3.2. Simulation

The model developed above is next simulated using *Simulink* for verification and in order to tune the controller parameters. First, the parameters of inner loop are tuned. In this simulation (Fig. 2), the task of the controller was to stabilize the orientation angles at zero and attain a height of 45 meters. Initial conditions were: $h = 30m$, $\phi = \theta = \psi = 18°$, $w = \tau = \omega = 0$. Initial rotational speed of propellers is calculated from trim condition (hover condition). A fixed step size, Runge-Kutta solver, with 'auto' step sizing is used for all simulations.

After the performance of inner loop controllers is found satisfactory, simulations are carried out to tune the parameters of outer loop controllers. In the simulation results shown in Fig. 3, the task of the controller is to obtain a forward velocity $(\dot{X}_i') = 10m/s$ and $h = 50m$. For this case, the same initial conditions are used as in previous simulation, with the only difference that initial Euler angles are kept at zero. Being a symmetric body, controller for sideward velocity $(\dot{Y}_i')$ is same as that for forward velocity. As can be seen from Figs. 2 and 3, altitude controller is sluggish (note the different time scale of h vs. Time subplot in Figs. 2 and 3).

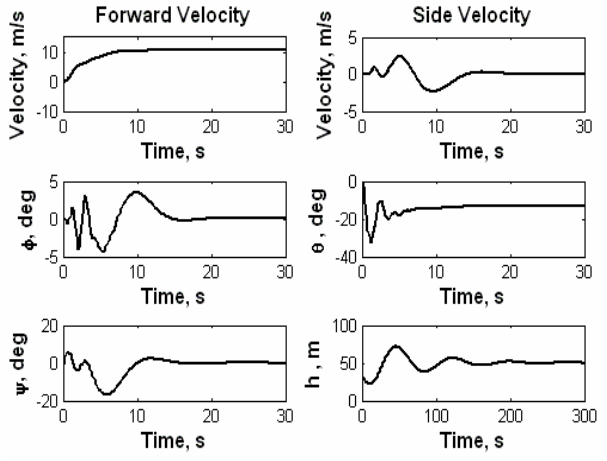

Figure 3: Starting from $\dot{X}_i' = 0$, and $h = 30m$, the system stabilizes at forward velocity $(\dot{X}_i')$ of $10m/s$, and $h = 50m$

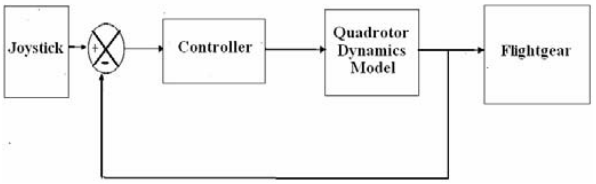

Figure 4: Block diagram representation of integrated model

4. INTEGRATED MODEL

For 3D visualization, *Flightgear* [9], an open source flight simulator under GNU license is used. Interfacing Flightgear with Matlab essentially requires sending the state vectors from Matlab/Simulink to Flightgear. A preconfigured interface block with Aerospace Blockset of Matlab is used for this purpose. For better speed and performance, Matlab and Flightgear are run on separate PCs. We use the inbuilt model of a helicopter, Eurocopter Bo105, for 3D visualization of Quadrotor in Flightgear. Figure 4 shows the block diagram of hardware integrated model. The implementation of controller is shown in Fig. 5. Here, *base values* block sets θ_b and ϕ_b required to fly at desired $\dot{X}_i'$ and $\dot{Y}_i'$. PIDs of *Outer Loop* block commands $\Delta\theta$ and $\Delta\phi$. PIDs of *Inner Loop* block commands the four voltages. *Saturation and Rate Limits* block constrains the voltages commanded. The resulting voltages then go as input to the Quadrotor dynamic model.

The leftmost blocks in Figs. 4 and 5 represents joystick interface. A standard force feedback joystick is used for setting the desired values of $\dot{X}_i'$, $\dot{Y}_i'$, ψ and h. This way we can give the inputs through joystick in real time and observe the visualization in Flightgear. Figure 6 shows a snapshot taken during the simulation in Flightgear when the model is initialized at $h = 1000m$. For visualization in Flightgear, the initial

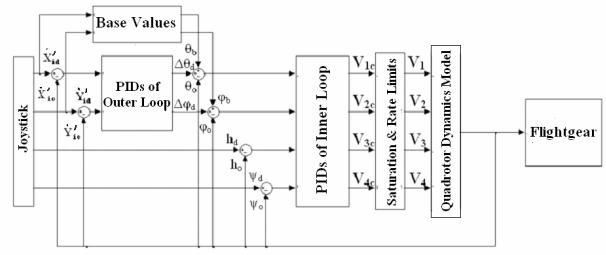

Figure 5: Controller design for Quadrotor

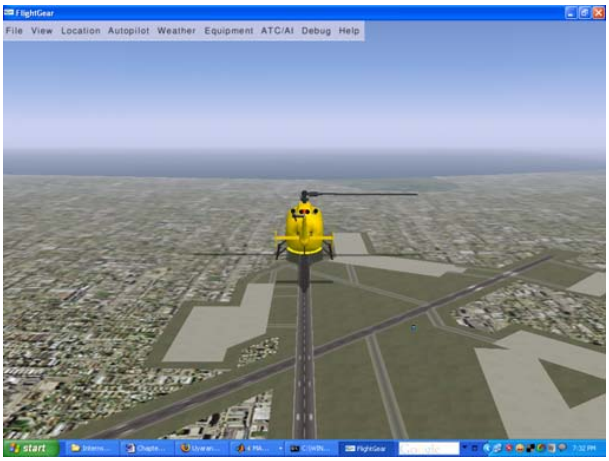

Figure 6: Snapshot of simulation in *Flightgear*

frame coordinates are converted into latitude, longitude and height at the place where the visualization is required. The other inputs required to be given to Flightgear for visualization are the three Euler angles.

5. EXPERIMENTAL SETUP AND RESULTS

Once the base studies on modeling and control design for a quadrotor platform are completed, the next task is to conduct experiments and implement the control algorithm on a flying platform. Towards this objective a quadrotor model was constructed and it was interfaced with an autopilot developed at Coral Digital Technologies, Bangalore. The first set of experiments are conducted on a 3-DOF test-rig, allowing only rotations, in order to test the attitude stabilization capability of the vehicle. Once sufficient confidence is gained in attitude stabilization, free flights are conducted and autonomous hover is demonstrated.

The quadrotor is designed using X-UFO (commercially available model) as the baseline. To improve the lifting capability of X-UFO, the brushed motors are replaced with brushless motors. The final version, capable of lifting autopilot and suitable battery pack, weighs 320 grams (including battery, autopilot), and has an additional 40 grams payload capability. The length and width of this prototype are 65 cm, each.

5.1. Hardware Design

The autopilot uses a single 16 bit 24HJ series PIC microcontroller for all the computations, communication, and

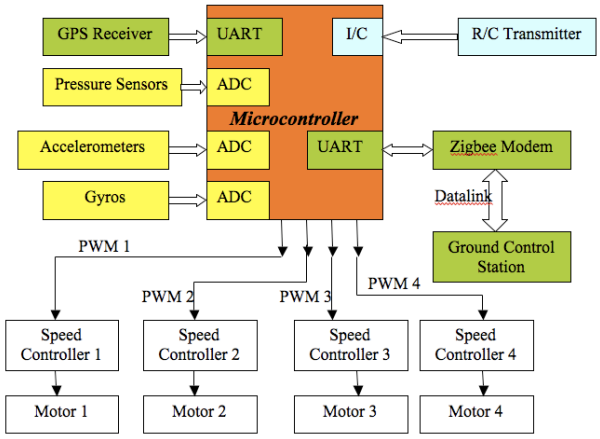

Figure 7: Autopilot architecture

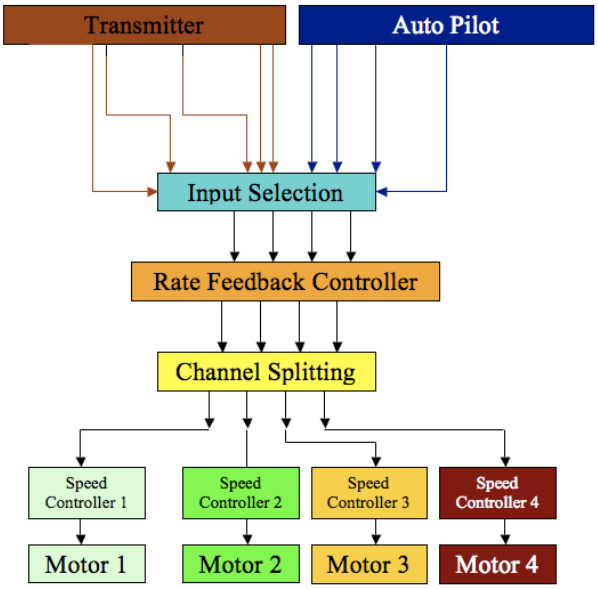

Figure 8: Flow diagram of autopilot algorithm in manual mode

switching between manual and auto modes. The navigation algorithm used three gyros and two accelerometers to estimate orientation. Altitude and velocity are estimated using pressure sensors, and position is obtained using the GPS. A Zigbee modem is used for communication with the *ground control station* during flight. An onboard SD card records several parameters at 50Hz during flight for post flight analysis of flight data. The autopilot hardware architecture along with key interfaces and communication protocols used is illustrated in Fig. 7.

The ground control station (GCS) is capable of updating way-points and PID gains during flight. GCS receives data from the autopilot in order to monitor vehicle trajectory and other key parameters during flight.

The autopilot design incorporates switching between auto and manual model for safety during flight tests. The flow diagram of the autopilot algorithm in manual mode is shown in Fig. 8. The rate feedback loop is part of the vehicle dynam-

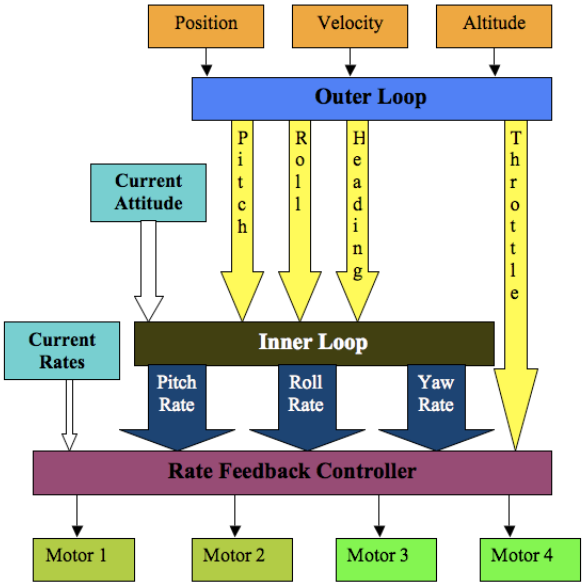

Figure 9: Schematic of controller in autonomous mode

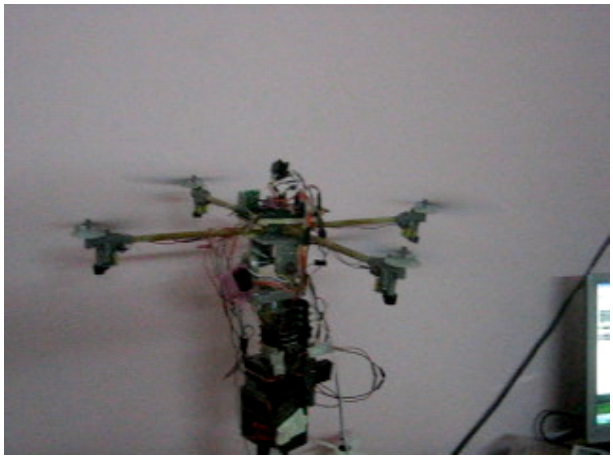

Figure 10: Quadrotor mounted on a 3-DOF test rig

ics, i.e., even in the manual mode, a rate feedback controller continues to work. It would be impossible to fly a quadrotor manually if such a rate feedback is not provided to assist the pilot. Such an architecture also allows to selectively assign any of the four inputs (thrust, roll, pitch, yaw) to manual or auto modes. The outputs of the autopilot correspond to elevator, aileron, rudder, and throttle inputs. These are converted into motor RPM commands by the *channel splitting* block. The schematic of autonomous controller is shown in Fig. 9.

5.2. Indoor Attitude Stabilization Results

The vehicle with autopilot is mounted on a 3-DOF test rig as shown in Fig. 10. The autopilot (mounted on the vehicle) is connected to the PC to monitor attitudes and control action in real-time. The yaw degree of freedom is restricted, and controller performance in pitch and roll is evaluated.

Figs. 11 and 12 show the test rig results. In this case, the experiment is started with autopilot in manual mode, and is

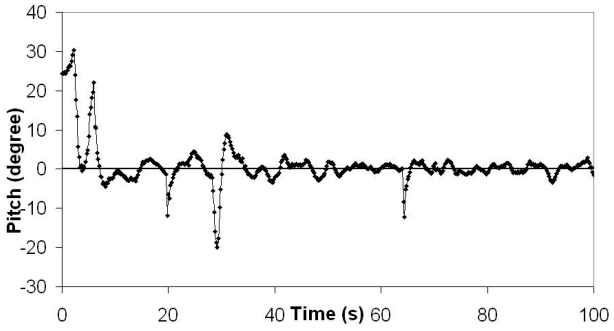

Figure 11: Pitch angle of the quadrotor on 3-DOF test rig

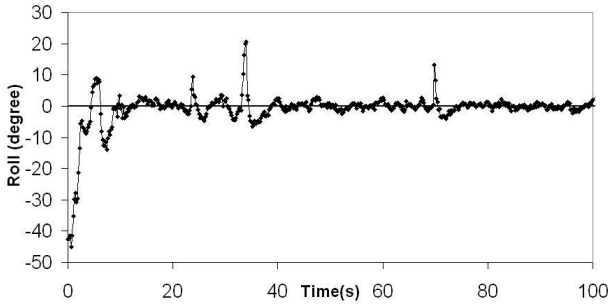

Figure 12: Roll angle of the quadrotor on 3-DOF test rig

switched to auto mode at $t = 19s$. During the entire experiment, the desired roll and pitch are kept at zero, and a constant thrust is maintained. The spikes in Figs. 11 and 12 represent the manual disturbance imparted to the system to test the controller robustness. The controller is observed to perform well and reject disturbances suitably in pitch and roll.

5.3. Free Flight Results

The free flight experiments are conducted with the aim of achieving autonomous hover (attitude stabilization and control). During these experiments, only the thrust controlled in manual mode by the pilot. The results for attitude stabilization and control are shown in Figs. 13 and 14. The dotted lines represent the commanded value and the solid lines represent the measured attitude of the vehicle. A satisfactory performance in attitude stabilization is obtained, and the vehicle is able to keep itself afloat in hover with no active pilot inputs. Slow drift in position is observed which could be eliminated once the outer control loop is closed using position feedback.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented an approach for modeling the quadrotor RUMAV, simulation and 3D visualization, controller design, and its implementation on hardware. The controller is shown to work well for attitude stabilization and control during a free flight experiment.

The next stage of work will involve closing the outer control loop, and testing the way-point navigation on this platform.

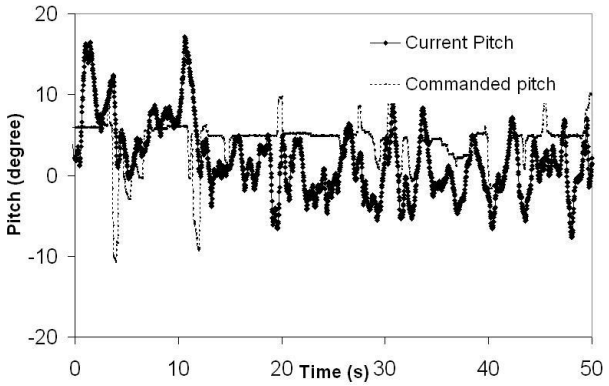

Figure 13: Pitch angle of the quadrotor during free flight experiment

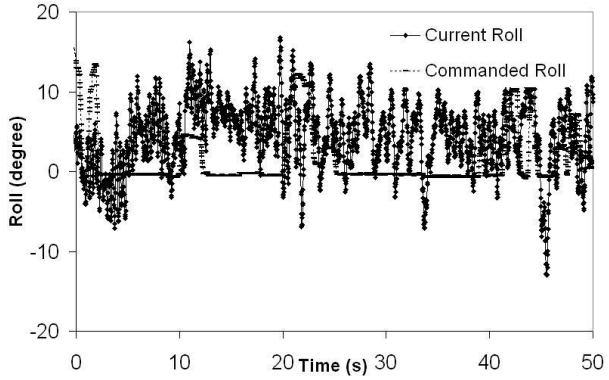

Figure 14: Roll angle of the quadrotor during free flight experiment

ACKNOWLEDGMENTS

The authors would like to acknowledge the contribution by Cdr. V.S. Renganathan and Mr. Srikanth towards development of autopilot hardware, and Mr. Shyam for helping with the flight tests.

REFERENCES

[1] J. G. Leishman. The Breguet-Richet Quad Rotor Helicopter of 1907, *http://www.glue.umd.edu/~leishman/Aero*, accessed Jan. 2009.

[2] S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro Quadrotor, *International Conference on Robotics and Automation,* New Orleans, USA, 2004.

[3] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic Modeling and Configuration Stabilization for an X4-Flyer, *15th Triennial World Congress of International Federation of Automatic Control,* Barcelona, Spain, 2002.

[4] R. Mahony, E. Altug, and J. P. Ostrowski. Control of a Quadrotor Helicopter using Visual Feedback, *Proceedings of 2002 IEEE Conference on Robotics and Automation,* Washington DC, pp. 72-77, 2002.

[5] E. B. Nice. Design of a four rotor hovering vehicle, MS thesis, Cornell University, 2004.

[6] A. Tayebi, and S. McGilvray. Attitude Stabilization of a VTOL Quadrotor Aircraft, IEEE Transactions on Control Systems Technology, Vol. 14, pp. 562-571, 2006.

[7] C. J. Tomlin, J. S. Jang, S. L. Waslander, and G. M. Hoffmann. Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning, *IEEE International Conference on Intelligent Robots and Systems,* Alberta, Canada, pp. 468-473, 2005.

[8] A. R. S. Bramwell, G. Done, and D. Balmford. *Bramwells Helicopter Dynamics,* 2nd ed., Butterworth Heinemann, Oxford, UK, 2001.

[9] *http://www.flightgear.org,* accessed Jan. 2009.

Rahul Goel Rahul Goel completed his B. Tech. in Aerospace Engineering from IIT Bombay, and is currently pursuing his Masters at Aero-Astro Department of MIT, USA. His current research interests include human space flight, human factors engineering, space systems design, dynamics, modeling and simulation, Finite Element Analysis.

Sapan Shah has a postgraduate degree from IIT Bombay, and is currently working for Whirlybrid Electronics Pvt. Ltd. at Mumbai. He has a very keen interest in UAVs/MAVs, automation and embedded systems.

Nitin Kumar Gupta holds post graduate degrees from IIT Bombay and University of Maryland, USA. He is currently serving as the Director and CEO at IDeA Research, a commercial R&D lab based in Pune. His research interests include dynamics, modeling, simulation, controls, and embedded systems.

Dr. Ananthkrishnan has served on the faculty at IIT Bombay and CalTech, USA. After spending over 15 years in academia, he continues to actively contribute to research as a member of the board of directors at IDeA Research, Tetrahedrix Engineering and Coral Digital Technologies. His research interests include nonlinear dynamics, aerodynamics, control systems, and systems design.