Autonomous Attitude Stabilization of a Quadrotor

Santosh D.R[1], Hemanth Kumar[2], Bharath.M.K[3], S.M.Vaitheeswaran[4], T.V Rama Murthy[5]
[1] Student (M.Tech, Reva ITM), Bangalore, Karnataka, 560064
[2, 3] Project Assistant, NAL, CSIR, Bangalore, Karnataka, 560017
[4] Principle Scientist, ALD, NAL, CSIR, Bangalore, Karnataka, 560017 (corresponding author)
[5] Senior Professor Reva ITM, Bangalore, Karnataka, 560064

Abstract
Quadrotor are four rotor helicopters that fly using a pair of rotors spinning in opposite directions. It is an under actuated system: it uses four different types of controllers: pitch, roll, yaw and altitude to control six degrees of freedom. To obtain the control there is translational rotational coupling. Thus the system is highly non-linear. The aim of the present work is stabilize the quad rotor using a Proportional, Integral and Derivative Controller in real time autonomously. The controller calculates the proximity of the input to the set value, the rate at which the input is moving to the set point and the duration for which the input is away from the set point. These form respectively the proportional, derivative and integral components of the controller and they are weighted and tuned to combine to obtain the motor offset value.

Nomenclature

MAV = Micro Air Vehicle
PID = Proportional Integral Derivative
YPR = Yaw, Pitch, Roll

I. Introduction

A quadrotor, or quadrotor helicopter, is an aircraft that becomes airborne due to the lift force provided by four rotors usually mounted in cross configuration, hence its name. They differ from conventional helicopters where all the four rotors work together to produce an upward thrust. The movement of the quadrotor is obtained by varying the speed of each rotor. For example, to roll or pitch, one rotor's thrust is decreased and the opposite rotor’s thrust is increased by the same amount. This causes the quadrotor to tilt. The force vector is split into a horizontal and vertical component. The quadrotor begins to move in a direction opposite to the newly created horizontal component and since the force vector is split, it becomes smaller in the vertical direction and the quadrotor begins to fall. Therefore some kind of compensation is required to compensate and this is done by increasing the thrust. The quadrotor flies by using inertial sensors. Since the autopilot cannot hover the aircraft for a long time, some kind of external system like the pilot in the loop is essential. This is obtained for example through external radio control or some form of vision system. The quadrotor uses four different types of controllers: pitch, roll, yaw and altitude to control six degrees of freedom. Thus it is an under actuated system. To obtain the control there is translational rotational coupling. Thus the system is highly non-linear.

The aim of the present work is stabilize the quad rotor using a Proportional, Integral and Derivative controller. Each of the controllers for example to pitch obtains the pitch angle and the set point value. To stabilize or hover the set point would be taken to zero, the state where the quadrotor will be perfectly level with respect to ground. To move, the set point is moved up or down to obtain the tilt in the desired direction. In real time, the controller calculates the proximity of the input to the set value, the rate at which the input is moving to the set point and the duration for which the input is away from the set point. These form respectively the proportional, derivative and integral components of the controller and they are weighted and tuned to combine to obtain the motor offset value. For autonomous control, these weights are estimated in an autonomous fashion obtained by minimizing the input value and desired value through an optimization process.

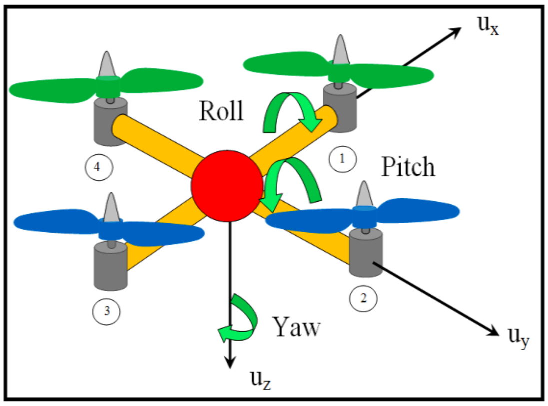

Fig 1 Yaw, pitch and roll rotations of a quadrotor

A. Quadrotor dynamics

Aerodynamic forces and moments are derived using a combination of momentum and blade element theory (Prouty, 1995; Castillo *et al*., 2005). A quad rotor has four motors with propellers. A voltage applied to each motor results in a net torque being applied to the rotor shaft, $\boldsymbol{Q}_i$, which results in a thrust, $\boldsymbol{Ti}$. If the rotor disk is moving, there is a difference in relative velocity between the blade and air when moving through the forward and backward sweeps, which results in a net moment about the roll axis, $\boldsymbol{R}_i$.

Forward velocity also causes a drag force on the rotor that acts opposite to the direction of travel, $\boldsymbol{D}_i$. The thrust can be defined in terms of aerodynamic coefficients C_T as

$$\mathrm{T} = \frac{1}{2}\rho \mathrm{A} C_{\mathrm{T}} r^2 \omega^2 \tag{1}$$

Where A is the area of the rotor blade, ρ is the density of air, r is the radius of the blade and ω is the angular velocity of the propeller as shown in Eq (1).

The body frame and the inertial frame of the quadrotor is shown in Fig 2.

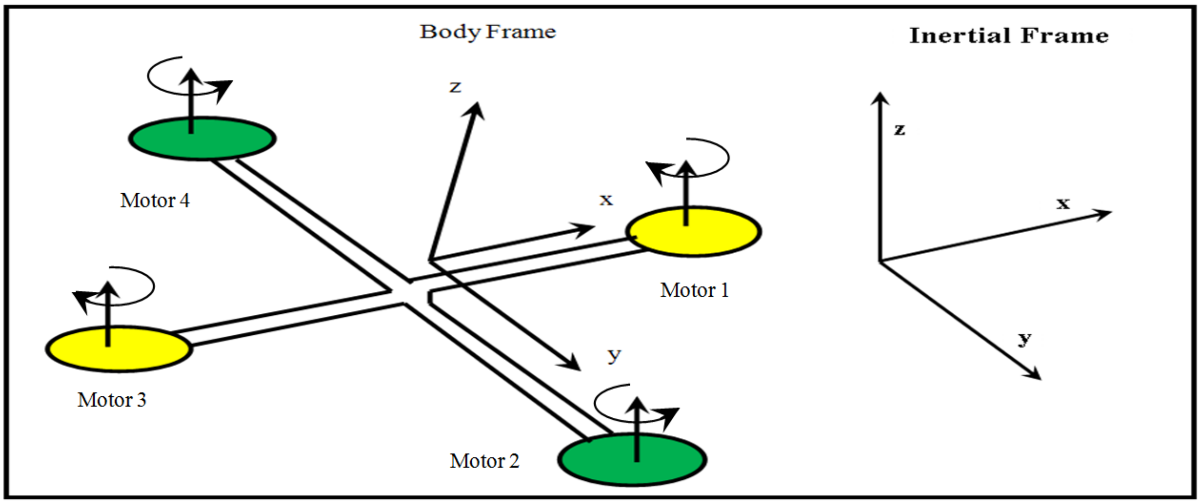

Fig 2 Quadrotor Body and Inertial Frame

The control inputs to quadrotor are defined by the following

Vertical force input is given by: $u_1 = \mathrm{K}_{\mathrm{T}} \sum_{i=1}^{4} \omega_i^2$

Moment of roll input is given by: $u_2 = \mathrm{K}_{\mathrm{T}} \left(\omega_4^2 - \omega_2^2\right)$

Moment of pitch input is given by: $u_3 = \mathrm{K}_{\mathrm{T}} \left(\omega_1^2 - \omega_3^2\right)$

Moment of yaw input is given by: $u_4 = \mathrm{K}_D \left(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2\right)$ (2)

B. Quadrotor Kinematics

The kinematics of a quadrotor is defined in the inertial and body frames. The position and velocity of the quadrotor is related in inertial frame by x=$(x, y, z)^T$ and $\dot{x} = (\dot{x}, \dot{y}, \dot{z})^T$ respectively and the Euler angles θ= $(\varphi, \theta, \psi)^T$ as the roll, pitch and yaw angles in the body frame. The angular velocity vector ω is not equal to the angular velocity $\dot{\theta}$ in the body frame as ω is a pointing velocity vector whereas $\dot{\theta}$ is derivative of Euler angles. The angular velocity vector ω is given by in Eq (3).

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\varphi & c_\theta s_\varphi \\ 0 & -s_\varphi & c_\theta c_\varphi \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3}$$

The orientation of the body frame in quadrotor is related with the inertial frame by a rotation matrix, derived by ZYX Euler angle representation as shown in Eq (4).

$$R = \begin{bmatrix} c_\theta c_\psi - c_\theta s_\varphi s_\psi & -c_\psi s_\varphi - c_\varphi c_\theta s_\psi & s_\theta s_\psi \\ c_\theta c_\psi s_\varphi + c_\varphi s_\psi & c_\varphi c_\theta c_\psi - s_\varphi s_\psi & -c_\psi s_\theta \\ s_\varphi s_\theta & -c_\varphi s_\theta & c_\theta \end{bmatrix} \tag{4}$$

Where c_θ and s_θ represents cosin and sin functions.

C. Equations of Motion for a quadrotor

The motion of the quadrotor is derived by the acceleration of the quadrotor. The acceleration of the quadrotor is obtained by thrust, gravity, and linear friction in inertial frame. The thrust vector of a quadrotor is obtained by the voltage applied to the motors, which is in the body frame. This thrust vector is mapped to the inertial frame by using the rotational matrix R. The linear motion of the quadrotor in the body frame is given in Eq (5).

$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RT_B + F_D \tag{5}$$

Where x is the position of quadrotor,
m is the mass of the quadrotor,
g is acceleration due to gravity,
F_D is drag force and
T_B is thrust vector

The linear equations of motion in Eq 4 are to be mapped to inertial frame. The rotational equations of motion are obtained from body frame of the quadrotor and it is expressed with respect to quadrotor center instead of inertial center in body frame. The rotational equations of the quadrotor are derived from Euler angles and given by in Eq (6).

$$I\dot{\omega} + \omega \times (I\omega) = \tau \tag{6}$$

Where, ω is the angular velocity vector. I is inertia matrix and τ is external torque vector.
The above equation can be written as,

$$\dot{\omega}=\begin{bmatrix}\dot{\omega}_x\\ \dot{\omega}_y\\ \dot{\omega}_z\end{bmatrix}=I^{-1}\left(\tau-\omega\times(I\omega)\right) \tag{7}$$

The inertia matrix is given by

$$I=\begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix} \tag{8}$$

The body frame rotational equations of motion is given by

$$\dot{\omega}=\begin{bmatrix}\tau_\phi I_{xx}{}^{-1}\\ \tau_\theta I_{yy}{}^{-1}\\ \tau_\psi I_{zz}{}^{-1}\end{bmatrix}-\begin{bmatrix}\frac{I_{yy}-I_{zz}}{I_{xx}}\omega_y\omega_z\\ \frac{I_{zz}-I_{xx}}{I_{yy}}\omega_x\omega_z\\ \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_x\omega_y\end{bmatrix} \tag{9}$$

II. Quadrotor Control

Controlling of a quadrotor for the stabilization is a hard task in the field of quadrotor design. Control of quadrotor helicopters is achieved by varying the thrust of two sets of counter-rotating rotor pairs. Attitude is controlled through differential actuation of opposing rotors, with yaw controlled using the difference in reaction torques between the pitch and roll rotor pairs. The control system deals with the controlling the thrust, angular velocity of the quadrotor to make the quadrotor stable. In this paper the stabilization is done using a PID control and it is extended to an automatic tuning of gains with PID control.

D. PID Control

For the controlling of a quadrotor a feedback looping system is used. In this paper a proportional integral and derivative control loop is implemented. In this a proportional component, an integral component and a derivative component is used for the error reduction and to control the quadrotor. Quadrotor will only have a gyro, so the angle derivatives $\dot{\phi}, \dot{\theta},$ and $\dot{\psi}$ in the controller are considered. These measured values will give the derivative of error, and their integral will provide the actual error. To stabilize the quadrotor in a horizontal position, the desired velocities and angles will all be zero. Torques are related to angular velocities by $\tau = I\ddot{\theta}$, so set the torques proportional to the output of our controller, with *t* = Iu(t).

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -I_{xx}\left(K_d \dot{\phi} + K_p \int_0^T \dot{\phi}\, dt\right) + K_i \int_0^T \int_0^T \phi dtdt \\ -I_{yy}\left(K_d \dot{\theta} + K_p \int_0^T \dot{\theta}\, dt + K_i \int_0^T \int_0^T \dot{\theta}\, dtdt\right) \\ -I_{zz}\left(K_d \dot{\theta} + K_p \int_0^T \dot{\psi}\, dt + K_i \int_0^T \int_0^T \dot{\psi}\, dtdt\right) \end{bmatrix} \tag{10}$$

Thus, the relationship between torque and our inputs are derived, therefore τ_B is given by

$$\tau_B = \begin{bmatrix} Lk(\gamma_1 - \gamma_3) \\ Lk(\gamma_2 - \gamma_4) \\ b(\gamma_1 - \gamma_2 + \gamma_3 - \gamma_4) \end{bmatrix} = \begin{bmatrix} -I_{xx}\left(K_d \dot{\phi} + K_p \int_0^T \dot{\phi}\, dt + K_i \int_0^T \int_0^T \phi dtdt\right) \\ -I_{yy}\left(K_d \dot{\theta} + K_p \int_0^T \dot{\theta}\, dt + K_i \int_0^T \int_0^T \dot{\theta}\, dtdt\right) \\ -I_{zz}\left(K_d \dot{\theta} + K_p \int_0^T \dot{\psi}\, dt + K_i \int_0^T \int_0^T \dot{\psi}\, dtdt\right) \end{bmatrix} \tag{11}$$

This gives us a set of three equations with four unknowns. This can constrain by enforcing the constraint that our inputs must keep the quadrotor aloft:

$$T = mg \tag{12}$$

The equation (12) ignores the fact that the thrust will not be pointed directly up. This will limit the applicability of our controller, but should not cause major problems for small deviations from stability. If the current angle accurately is determined, then it can be compensated. If the gyro is precise enough, integrate the values obtained from the gyro to get the angles θ and Φ. Now calculate the thrust necessary to keep the quadrotor aloft by projecting the thrust mg onto the inertial z axis. We find that

$$T_{proj} = mg \cos\theta \cos\phi \tag{13}$$

Therefore, with a precise angle measurement, the thrust will be equal to

$$T = \frac{mg}{\cos\theta \cos\phi} \tag{14}$$

in which case the component of the thrust pointing along the positive z axis will be equal to mg. The thrust is proportional to a weighted sum of the inputs given by Eq (15),

$$T = \frac{mg}{\cos\theta\cos\phi} = k\sum\gamma_i \Rightarrow \sum\gamma_i = \frac{mg}{k\cos\theta\cos\phi} \qquad (15)$$

With this extra constraint, set four linear equations with four unknown's γ_i. Solve for each γ_i, and obtain the following input values from Eq (16):

$$\gamma_1 = \frac{mg}{4k\cos\theta\cos\phi} - \frac{2be_\phi I_{xx} + e_\psi I_{zz}kL}{4bkL}$$
$$\gamma_2 = \frac{mg}{4k\cos\theta\cos\phi} + \frac{e_\psi I_{zz}}{4b} - \frac{e_\theta I_{yy}}{2kL}$$
$$\gamma_3 = \frac{mg}{4k\cos\theta\cos\phi} - \frac{-2be_\phi I_{xx} + e_\psi I_{zz}kL}{4bkL}$$
$$\gamma_4 = \frac{mg}{4k\cos\theta\cos\phi} + \frac{e_\psi I_{zz}}{4b} + \frac{e_\theta I_{yy}}{2kL} \qquad (16)$$

E. Auto Tuning of PID Gains

Even though the performance of the PID control is well, the performance is mainly dependent on the gain parameters that are set in the PID loop. Tuning the gain parameter by hand is quite difficult task as the ratio of the gain parameters is equally important as the magnitudes of the gain parameters. Often the tuning of gain parameters requires detailed knowledge of the system and an understanding of the conditions in which the PID control will be used. The gain parameters were initially tuned by hand for good performance by running simulations till quadrotor attains the stability. This method is a suboptimal, as the gain tuning by hand is difficult and even after tuning it is no way guaranteed to be optimal or even close to optimal.

The auto-tuning of gains overcomes this problem. The gain parameters are considered as a vector $\vec{\theta} = (K_p, K_i, K_d)$ which minimizes cost function J($\vec{\theta}$). The cost function is given by Eq (17).

$$J\left(\vec{\theta}\right) = \frac{1}{t_f - t_0}\int_{t_0}^{t_f} e\left(t,\vec{\theta}\right)^2 dt \qquad (17)$$

Where $e\left(t,\vec{\theta}\right)$ is the error in following reference trajectory with some initial disturbance. Now calculate the gradient of the cost function. This process is done by certain iteration to improve the parameter vector by defining a parameter update rule.

$$\vec{\theta}(k+1) = \vec{\theta}(k) - \alpha\nabla J\left(\vec{\theta}\right) \qquad (18)$$

Where, $\nabla J\left(\vec{\theta}\right)$ is gradient function $\vec{\theta}(k)$ is parameter vector after k iteration and α step size which indicates the adjustment of parameter vector at each step of iteration. As $k \rightarrow \infty$ the cost function approaches to the minimum value of PID gain parameter.

III. Experimental Results

A Matlab simulation of quadrotor stabilization is implemented. Initially the quadrotor is made to maintain at a height of 5m. The quadrotor is resembled by four rotors attached to the arms and a control system at center. At this height the quadrotor has a certain amount of thrust and the quadrotor is in an unstable state due to the various factors acting on the rotors like air, gravity etc. These initial thrusts of the quadrotor are estimated and are indicated by the lines on the rotors as shown in Fig 3. The PID controller estimates the error in the attitude and gradually reduces the error by repeated number iterations. As the attitude components are been compensated, the variation in the orientation leads the quadrotor to have the motion to certain distance and then stabilize. Fig 4 shows the gradual reduction in the attitude variation and moving towards x axis direction. Fig 5 shows a completely stabilized portion of the quadrotor where the minimum amount of thrust is applied to keep the quadrotor at the given height.

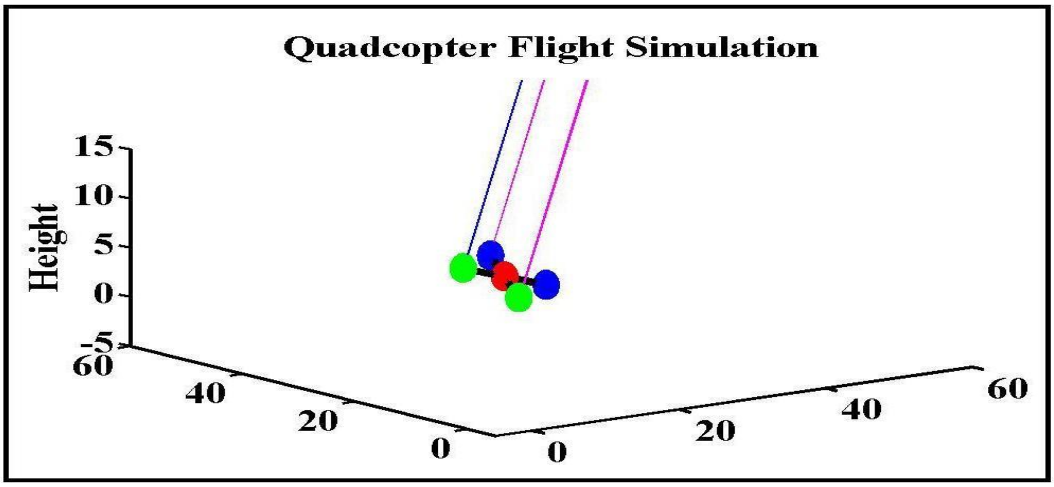

Fig 3 Initial position of a Quadrotor

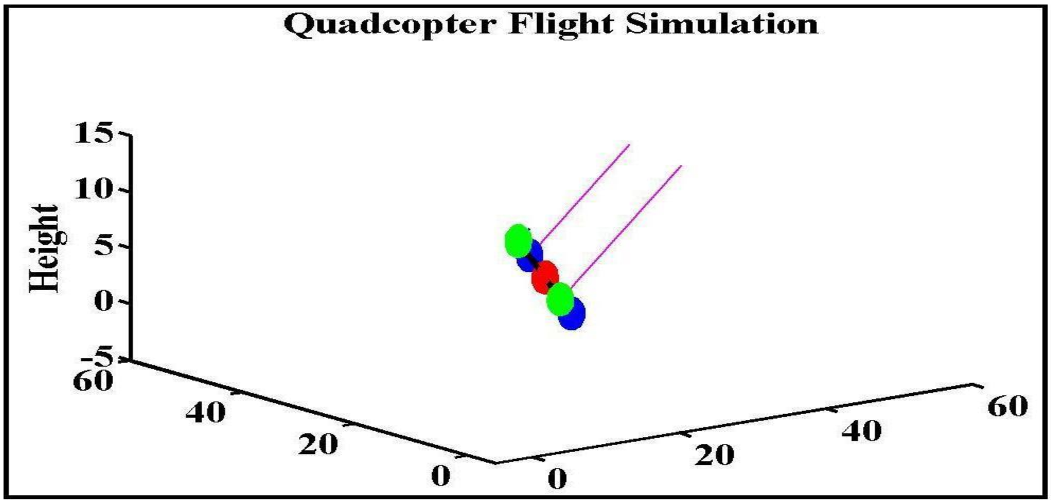

Fig 4 Stabilization using PID control

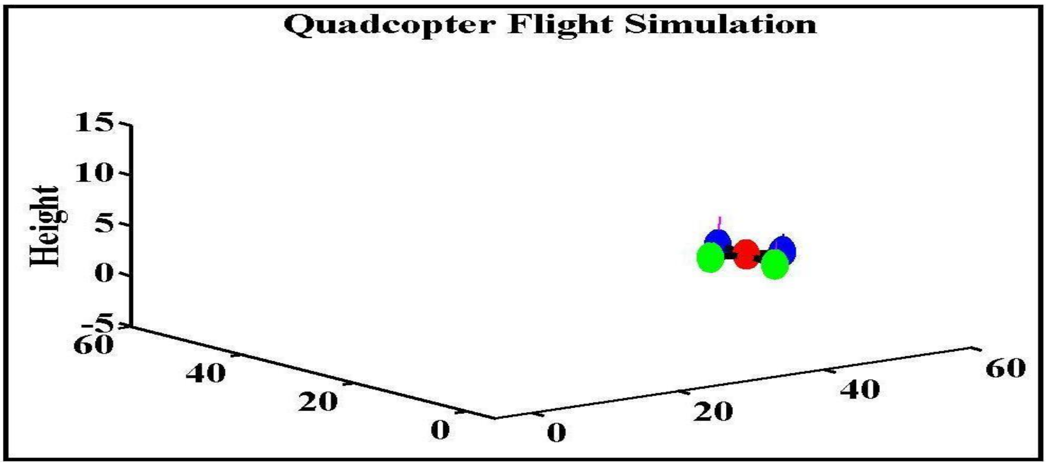

Fig 5 Stabilized quadrotor

The attitude of a quadrotor is described by three parameters namely Yaw, Pitch and Roll. These three parameter are to be controlled to fly a quadrotor in the space. The pitch (G), roll (R) and yaw (B) parameters for the above described quadrotor are been plotted by plotting the angular velocities and angular displacements of the quadrotor. In the first step the gains for the quadrotor are set manually and the YPR graph for a PID control is plotted as shown in Fig 6. The attitude variation and its correction can be viewed by plotting the angular velocity and angular displacement with respect to time. The angular displacement plot indicates the stabilization of the quadrotor in 4 sec. This is followed by an auto tuning process of the gains and YPR graph is plotted as in Fig 7. To verify the results of the YPR values of quadrotor using a PID controller with auto tuning and without auto tuning are plotted separately as shown in Fig 8. Fig 8 clearly indicates that the YPR components of a PID control using manually tuned gains require more time to stabilize as compared to the YPR components of PID control using auto tuned gains.

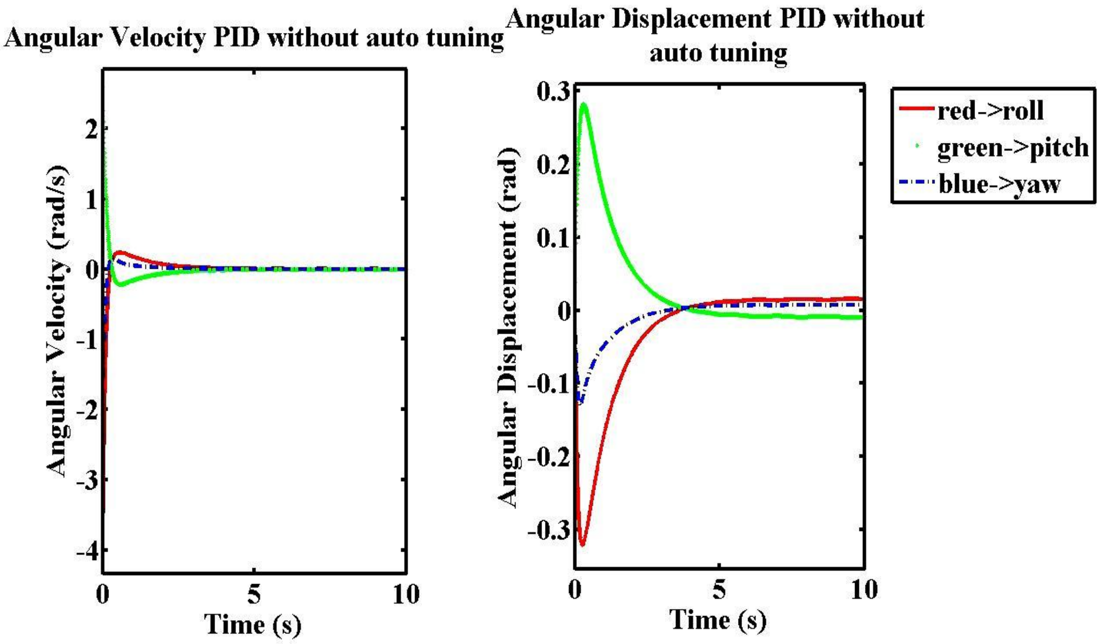

Fig 6 Angular velocity and displacement of a PID control without auto tuning

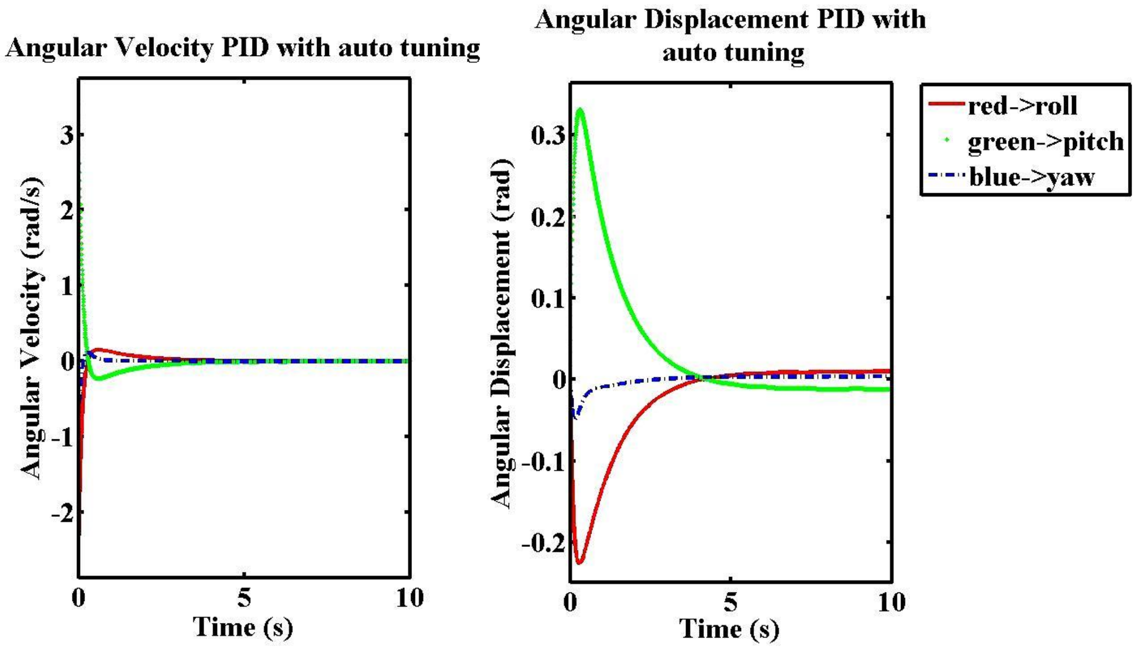

Fig 7 Angular velocity and displacement of a PID control with auto tuning

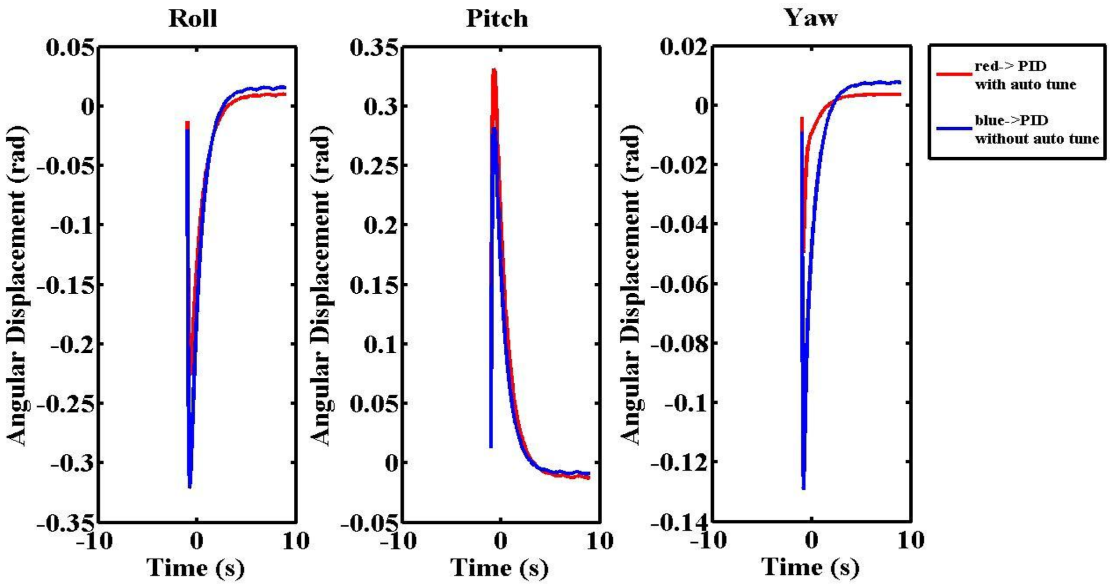

Fig 8 Comparison of YPR separately

IV.Conclusions

The present paper describes the implementation of an auto tuning PID control simulation for a quadrotor for autonomous stabilization. The Matlab simulation result indicates the stabilization of a quadrotor using a PID controller. In this simulation result PID control is been implemented with an auto tuning of gains. The pitch, roll and yaw variations are shown and are minimized to zero. The error reduction in PID control using auto tuning is faster and more accurate than manual tuning of the gains. Hence in real time implementation auto tuning is more preferable than manual tuning.

Acknowledgments

The authors SMV, MKB and HKE, NAL gratefully acknowledge the funding received from AR&DB, GoI under the NPMICAV program to carry out the above work. The work is carried out at National Aerospace Laboratories, CSIR, Bengaluru, Karnataka, INDIA.

References

[1] M. P. Vitus, V. Pradeep, G. Hoffmann, S. L. Waslander, and C. J. Tomlin, "*Tunnel-MILP: Path planning with sequential convex polytopes*," in the Proceedings of the AIAA Guidance, Navigation, and Control Conference, (Honolulu, Hawaii, USA), 2008.

[2] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "*Flight test and simulation results for an autonomous aerobatic helicopter*," in Proceedings of the 2008 AIAA Guidance, Navigation, and Control Conference, (Honolulu, Hawaii), 2008.

[3] Luukkonen, Teppo. " *Modelling and control of quadcopter*" . Aalto University, School of Science.August 22, 2011.

[4] Killingsworth, Nick; Krstic, Miroslav. "*PID Tuning Using Extremum Seeking: Online, Model-Free Performance Optimization*". IEEE Control Systems Magazine. February 2006.

[5] Huang, Hoffman, Waslander, and Tomlin. " *Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering*". Robotics and Automation. May 2009.

[6] David Lara, Anand Sanchez, Rogelio Lozano, P. Castillo "*Real-Time Embedded Control System for VTOL Aircrafts: Application to stabilize a quad-rotor helicopter*". Proceedings of the 2006 IEEE International Conference on Control Applications Munich, Germany, October 4-6, 2006.

[7] Rahul Goel, Sapan M. Shah, Nitin K. Gupta, N. Ananthkrishnan. "*Modeling, Simulation and Flight Testing of an Autonomous Quadrotor".* Proceedings of ICEAE 2009.

[8] Ashfaq Ahmad MIAN, Dao-bo WANG. "*Dynamic modeling and nonlinear control strategy for an underactuated quad rotor rotorcraft* ". Journal of Zhejiang University SCIENCE A ISSN 1673-565X; ISSN 1862-1775.