Attitude Stability of Quadcopter Using Classic Control with Angular Acceleration

Mohamed A.Abdelkhalek, Mohamed S.El-Demerdash,
Ahmed A.El-Tahan
Aerospace Engineering Department
Cairo University
Cairo, Egypt
Mohamed.Elmezain93@eng-st.cu.edu.eg ,
Mohamed.demerdash.1993@gmail.com, Ahmedtahan1994@gmail.com

Tarek N.Dief
Department of Earth System Science and Engineering
Kyushu University
Fukuoka, Japan
Tarek.na3em@riam.kyushu-u.ac.jp

***Abstract*— This paper presents the dynamics and control system for a linearized model of Quadcopter. The control part gives implementation of a new control action which consider the angular acceleration plus PD controller PD-A and comparing it with regular PD controller. The PD-A controller is aimed to enhance the response of a system than the regular PD. In order to show the effectiveness of the PD-A over the PD, theoretical and experimental studies were carried out based on one degree of freedom quadcopter model. MATLAB Simulink software is used in the simulation to generate the theoretical results. The comparison between the theoretical and experimental results revealed a significant stability improvement for the proposed control methodology in comparison with the conventional PD methodology.**

Keywords*— Acceleration; Dynamic Modeling; PD control; Quadcopter; Simulink; Unmanned Aerial Vehicle.

Notations and Symbols

X, Y, Z:	The inertial frame axes.
x_b, y_b, z_b:	Body axes.
$\hat{\imath}, \hat{\jmath}, \hat{k}$:	Unit vectors in direction of x_b, y_b, z_b respectively.
p, q, r :	Angular velocity components in body axes.
u, v, w:	Linear velocity components in body axes.
U, V, W:	Linear velocity components in inertial frame.
ϕ:	Roll angle.
θ:	Pitch angle.
ψ:	Yaw angle.
$\vec{\omega}$:	Angular velocity in the body axes $(\vec{\omega} = p\hat{\imath} + q\hat{\jmath} + r\hat{k})$
$\vec{V}$:	Linear velocity in the body axes $(\vec{V} = u\hat{\imath} + v\hat{\jmath} + w\hat{k})$
R:	Rotation transformation matrix.
Ω_i:	Angular velocity of the i^{th} rotor.
Ω_{nom}:	Nominal angular velocity of the rotor.
PWM_i:	Pulse Width Modulation signal for i^{th} motor.
PWM_{nom}:	Nominal Pulse Width Modulation signal for i^{th} motor.
b:	Motor thrust constant.
d:	Motor torque constant.
$\vec{T}_i$:	Thrust force of the i^{th} motor.
m:	Mass of the quadcopter.
l:	Length between the motor center and C.G.
g:	Acceleration due to gravity ($g = 9.81\ m/s^2$)
I:	Inertial tensor of the quadcopter about body axes.

$\delta_{col}, \delta_{long}, \delta_{late}, \delta_{pend}$: Control actions for altitude, pitch angle, roll angle and yaw angle respectively.

Subscripts "b" and "I" denote body axes and Inertial frame axes respectively.

I. Introduction

Quadcopter is one of the most famous drones, it is a VTOL (Vertical Take-Off and Landing) aircraft which could be used in many fields like military surveillance, agriculture services, civil fields which require high level of safety with low cost, etc. Moreover, it is effective in performing unmanned dangerous tasks. Owing to its simplicity in design and manufacturing, it presents a good platform for applying different control techniques [1]. Linear and nonlinear control algorithms, such as PID [2]-[3], Linear Quadratic Gaussian (LQG) [4], backstepping [5]-[6], sliding mode control [7], nonlinear robust control [8]-[9] and adaptive control [9]-[10], are used to design the quadrotor controller. The main objectives for the design of this controller are the quadrotor stability and its tracking to a desired trajectory. The challenge is how to control the system and perform the mission while there are uncertainties in the control system such as external disturbances, sensor error, actuator degradation and time delays [11].

In this paper, a new controller methodology is designed theoretically and tested experimentally on one degree of freedom (roll angle) of an indoor quadcopter. This methodology implies PD controller for roll angle error plus a derivative action for rate of change of the roll angle error in order to enhance the system performance characteristics.

To prove that, theoretical studies were firstly executed and followed by some experiments. The results will be discussed in details.

II. Mathematical Model

The quadcopter is a nonlinear, multivariable and under actuated system of 6 DOFs and only 4 actuators. In this work, the mathematical model equations of motion are derived using a full quadcopter with body axes as shown in Fig. 1. It is assumed that the quadcopter and its components are rigid bodies, body mass is concentrated at the center of gravity and the body axes are the principle axes for the quadcopter [12]. The aerodynamic forces and gyroscopic effects are neglected in this study.

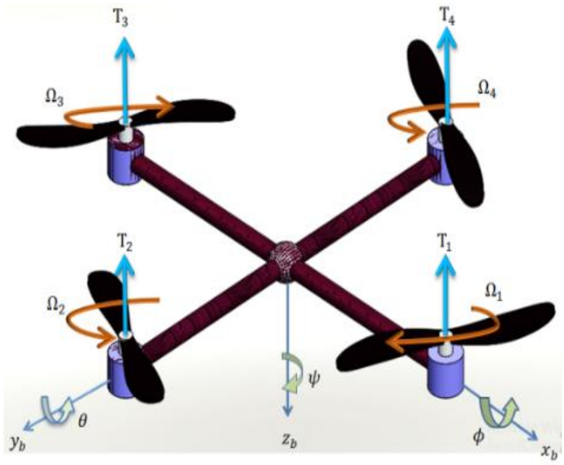

Fig. 1. Full quadcopter with body axes.

The rotation transformation matrix ZYX is used to transform the vector components from body axes to inertial frame of reference and vice versa as following [13]-[14]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [R]_{3\times 3} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \qquad \textbf{(1)}$$

Where,

$$[R] = \begin{bmatrix} cos\theta cos\psi & -sin\psi cos\phi + cos\psi sin\theta sin\phi & sin\psi sin\phi + cos\psi sin\theta cos\phi \\ sin\psi cos\theta & cos\psi cos\phi + sin\psi sin\theta sin\phi & -cos\psi sin\phi + sin\psi sin\theta sin\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix}$$

Since, $[R]$ is an orthogonal matrix, then $[R]^{-1} = [R]^T$.

A. Equations of motion "Using Newton's second law"

First: Translational equations of motion

$$\vec{\mathrm{F}} = \mathrm{m}\vec{\mathrm{a}}_{\mathrm{b}} \qquad \textbf{(2)}$$

Where $\vec{F}$ is the force vector in body axes, m is the body mass and $\vec{a}_b$ is the acceleration vector in body axes.

The force vector $\vec{F}$ can be derived as the following:
The thrust force vector in body axes for each motor is

$$\vec{T}_i = b\Omega_i^2(-\hat{z}_b) = -b\Omega_i^2\hat{z}_b$$

then, the total thrust force vector in body axes is given by

$$\vec{T} = -b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)\hat{z}_b \qquad \textbf{(3)}$$

The gravitational force vector in inertial frame axes is given by

$$\vec{F_g} = mg\hat{Z} \qquad \textbf{(4)}$$

Transforming the gravitational force (4) to body axes using (1):

$$\vec{F}_{g_b} = mg\begin{bmatrix} -sin\theta \\ cos\theta sin\phi \\ cos\theta cos\phi \end{bmatrix} = \qquad \textbf{(5)}$$
$$= -mgsin\theta\hat{x}_b + mgcos\theta sin\phi\hat{y}_b + mgcos\theta cos\phi\hat{z}_b$$

then, (2) can be rewritten as following:

$$\vec{F} = \mathrm{m}\vec{\mathrm{a}}_{\mathrm{b}} = \vec{T} + \vec{F}_{g_b} = -b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)\hat{z}_b + mg\begin{bmatrix} -sin\theta \\ cos\theta sin\phi \\ cos\theta cos\phi \end{bmatrix}$$

then, $\vec{a}_b = \frac{\vec{F}}{m} = \begin{bmatrix} -gsin\theta \\ gcos\theta sin\phi \\ -\frac{U_1}{m} + gcos\theta cos\phi \end{bmatrix}$,

where $U_1 = b\sum_{i=1}^{i=4}\Omega_i^2$

using $\vec{a}_b$ to calculate the acceleration vector in inertial frame of reference $\vec{a}_I$ as described in [13]:

$$\vec{a}_I = \vec{a}_b + \vec{\omega} \times \vec{V} \qquad \textbf{(6)}$$

Where $\vec{\omega}$ is the resultant angular velocity vector due to the three rotations ϕ, θ and ψ given by:

$$\vec{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & cos\theta sin\phi \\ 0 & -sin\phi & cos\theta cos\phi \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \qquad \textbf{(7)}$$

Then, (6) can be rewritten as:

$$\vec{a}_I = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} -gsin\theta + qw - vr \\ gcos\theta sin\phi - pw + ur \\ -\frac{U_1}{m} + gcos\theta cos\phi + pv - qu \end{bmatrix} \qquad \textbf{(8)}$$

Equation (6) is also used to get the rate of change of Euler angles as the following:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & tan\theta sin\phi & tan\theta cos\phi \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi/cos\theta & cos\phi/cos\theta \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad \textbf{(9)}$$

Second: Rotational equations of motion

$$\vec{M}_b = I\dot{\vec{\omega}} + \vec{\omega} \times (I\vec{\omega}) \qquad \textbf{(10)}$$

Where, $\vec{M}_b$ is the moment vector in body axes.

The moment vector $\vec{M}_b$ can be derived as the following:
Torque results from differential Ωs**:**

$$\vec{\tau}_\phi = l(T_4 - T_2)\hat{x}_b = lb(\Omega_4^2 - \Omega_2^2)\hat{x}_b$$
$$\vec{\tau}_\theta = l(T_1 - T_3)\hat{x}_b = lb(\Omega_1^2 - \Omega_3^2)\hat{x}_b$$

Torque results from drag reaction on blades:

$$\vec{\tau}_i = d\Omega_i^2(\pm\hat{z}_b)$$

then, $$\vec{\tau}_{\psi} = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)\hat{z}_b$$

then,
$$\vec{M}_b = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_1^2 - \Omega_3^2) \\ d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}$$

since, $I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$, $\vec{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$ and $\dot{\vec{\omega}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$

Then, (10) can be rewritten as following:

$$\vec{M}_b = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} - (I_{yy} - I_{zz})qr \\ I_{yy}\dot{q} - (I_{zz} - I_{xx})pr \\ I_{zz}\dot{r} - (I_{xx} - I_{yy})pq \end{bmatrix} =$$

$$= \begin{bmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_1^2 - \Omega_3^2) \\ d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}$$

Since, $\dot{\vec{\omega}} = I^{-1} \sum \vec{M}_G - I^{-1}[\vec{\omega} \times (I\vec{\omega})]$
then,

$$\dot{\vec{\omega}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{(I_{yy} - I_{zz})qr + U_2}{I_{xx}} \\ \dfrac{(I_{zz} - I_{xx})pr + U_3}{I_{yy}} \\ \dfrac{(I_{xx} - I_{yy})pq + U_4}{I_{zz}} \end{bmatrix} \qquad \textbf{(11)}$$

Linearizing (8), (9) and (11) using small disturbance theory [14] to get the linear open loop transfer function. Linearization is made at hovering condition where

$$\phi_o = \theta_o = \psi_o = \dot{\phi} = \dot{\theta} = \dot{\psi} = u_o = v_o = w_o = p_o = \\ = q_o = r_o = 0$$

Then, the linear open loop transfer functions are:

$$\left.\begin{array}{l} \dfrac{\phi}{\delta_{lat}} = \dfrac{2lb\Omega_{nom}}{I_{xx}} * \dfrac{1}{s^2} \\ \dfrac{\theta}{\delta_{long}} = \dfrac{2lb\Omega_{nom}}{I_{yy}} * \dfrac{1}{s^2} \\ \dfrac{\psi}{\delta_{pend}} = \dfrac{2d\Omega_{nom}}{I_{zz}} * \dfrac{1}{s^2} \\ \dfrac{Z}{\delta_{col}} = \dfrac{-2b\Omega_{nom}}{m} * \dfrac{1}{s^2} \end{array}\right] \qquad \textbf{(12)}$$

III. EXPERIMENTAL SETUP

The experimental setup is designed and constructed as shown in Fig. 2.

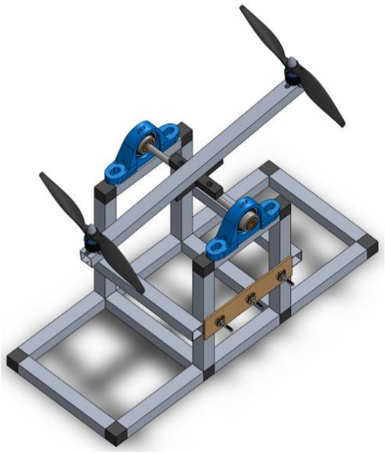

Fig. 2. Half-quadcopter experimental setup layout

It is consists of three main subsystems: half-Quadcopter, sensor and microcontroller. The half-Quadcopter setup forms the physical system while the sensor provides the attitude information to the control system. The microcontroller combines the electronic signals with the hardware to achieve stability.

There are four main components used in the system: brushless motors, Electronic Speed Controllers (ESC)'s, microcontroller and sensor -Inertial Measurement Unit (IMU) module mpu6050-. Motors are acting as the actuators of the system, while IMU module is used to measure the roll angle(ϕ) and the roll angle rate $(\dot{\phi})$, while the angular acceleration is obtained analytically using finite difference method. The microcontroller is the brain which processes, analyzes, and computes the suitable control action for the data provided by the IMU.

For implementation, it is required to transform the control action into angular velocity for each motor according to allocation method [14] as given:

$$\Omega_2 = \Omega_{nom} - 0.5\delta_{lat} \qquad \textbf{(13)}$$

$$\Omega_4 = \Omega_{nom} + 0.5\delta_{lat} \qquad \textbf{(14)}$$

Where, δ_{lat} is the control action for the roll angle (ϕ).

Control outputs from the microcontroller are in the form of Pulse Width Modulation (PWM) signal. These signals are carried through the ESC (which connects between the motor, the controller and the power supply source) into the motors (see Fig. 3).

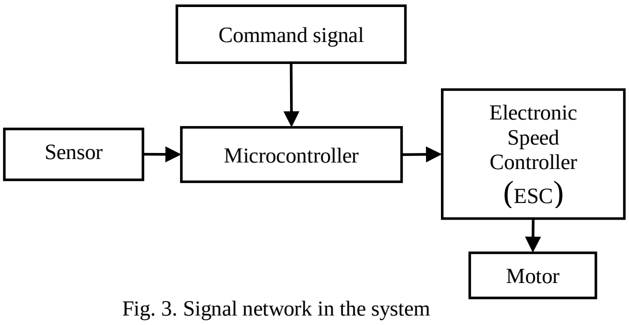

Fig. 3. Signal network in the system

It is clear that the angular velocity cannot be controlled directly in contrast with PWM signal, thus a relation between the PWM signal and the motor thrust force is required, so it's assumed that $T_i = b * \Omega_{\mathrm{i}}^2 = B * PWM_i$, where PWM is in (μ sec.) and B is the motor constant that relates the motor thrust force with PWM and its value is obtained experimentally (see Table 1).

Then, (12),(13) and (14) can be rewritten respectively as:

$$\frac{\phi}{\delta_{lat}} = \frac{lB}{I_{xx}} * \frac{1}{s^2} \tag{15}$$

$$PWM_2 = PWM_{nom} - 0.5\delta_{lat}$$

$$PWM_4 = PWM_{nom} + 0.5\delta_{lat}$$

Where, PWM_{nom} is estimated for our model as shown in Table I.

For fine tuning process, three potentiometers were used to change three gains k_p, k_d and k_a.

TABLE I: Half-Quadcopter model physical parameters

Parameter	Description	Value	Units
$\boldsymbol{PWM_{nom}}$	Nominal Pulse Width Modulation.	200	(μs)
$\boldsymbol{l}$	Length between the motor center and the model center of gravity.	25	(cm)
$\boldsymbol{B}$	Proportionality factor between motor thrust force and PWM.	1.066×10^{-2}	$(\frac{N}{\mu s})$
$\boldsymbol{I_{xx}}$	Roll inertia	1.634×10^{-2}	$(kg.m^2)$

IV. CONTROLLER DESIGN

The controller is designed and implemented in the system in order to minimize the error between the required output and the actual system output. The PD and PD-A control algorithms have been considered and implemented in literature to control the attitude of the model considered.

The controller is firstly designed using the linear model (see Fig. 4), then it is applied and the actual system.

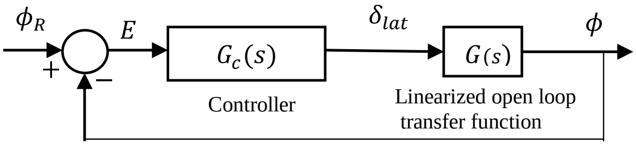

Fig. 4. Linear model scheme

$G(s)$ is the linearized transfer function of the model considered, then by using data in Table I, (15) can be rewritten as:

$$G(s) = \frac{\phi}{\delta_{lat}} = \frac{lB}{I_{xx}} * \frac{1}{s^2} = \frac{0.1517}{s^2}$$

It is observed from Fig.4 that the error signal is given by

$$e = \phi_R - \phi, \ \dot{e} = \dot{\phi}_R - \dot{\phi}, \ \ddot{e} = \ddot{\phi}_R - \ddot{\phi}$$

Since it is required to have zero roll angle: $\phi_R = 0$.

Then,

$$e = -\phi, \ \dot{e} = -\dot{\phi}, \ \ddot{e} = -\ddot{\phi} \tag{16}$$

Notice that the actions applied to the system are based on dealing with error signal in radians.

PD and PD-A controllers.

The system to be controlled is a one dimensional rotational system. The controller is based on how to control the roll angle (ϕ) to reach zero degree with the horizontal with the least possible overshoot and the least possible time, in addition to damping out external disturbances so as the system stay as close as possible to zero angle. Two different ways considered to achieve the mentioned goals, the first way is to apply a PD controller on the error signal ($\phi_R - \phi$) and the other way is to apply a PD-A controller on the error signal ($\phi_R - \phi$). The proportional action is related directly to the error and it is used to decrease the steady state error. The derivative action is used to control the rate of change of the actuating error and provide a correcting action before the system reach a significant error. The accelerator action is used to control the rate of the change of the error rate of change (error acceleration).

First, PD Controller is applied on the linearized system [15] (see Fig. 5).

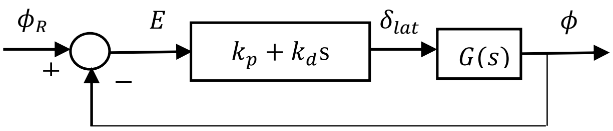

Fig. 5. PD controller on the linear model

Control action $\delta_{lat}(t)$ is

$$\delta_{lat}(t) = k_P e(t) + k_d \dot{e}(t) \tag{17}$$

Assuming zero initial conditions (i.e. $\phi(0) = \dot{\phi}(0) = \ddot{\phi}(0) = 0$), the Laplace transformation of (17) will be:

$$\delta_{lat}(s) = k_P E(s) + k_d s E(s) \qquad \textbf{(18)}$$

Using (16), (18) becomes:

$$\delta_{lat}(s) = -k_P \phi(s) - k_d s \phi(s)$$

If we used gains $k_P = 300\ and\ k_d = 777$, then the controller transfer function will be:

$$G_{c_1}(s) = \frac{\delta_{lat}(s)}{E(s)} = k_P + k_d S = 777(S + 0.3861) \qquad \textbf{(19)}$$

In order to implement the controller in the actual system we need to transform the controller equation (19) into pulse transfer function using simple method ($s = \frac{1-z^{-1}}{T_s}$) with sample time $T_s = 0.01$. seconds, then controller becomes [16]:

$$G_{c_1}(z) = \frac{\delta_{lat}(z)}{E(z)} = 77700(-z^{-1} + 1.003861) \qquad \textbf{(20)}$$

The difference equation for (20) is given by:

$$\delta_{lat}(nT_s) = 77700(-e(n-1)T_s + 1.003861e(nT_s)) \qquad \textbf{(21)}$$

Where, $n = 0,1,2, \ldots$

Second, PD-A Controller is applied on the linearized system (see Fig. 6).

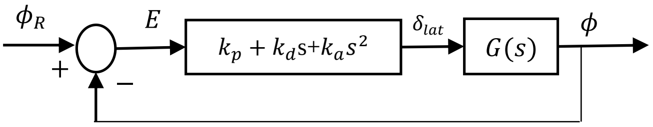

Fig. 6. PD-A controller in the linear model

The control action $\delta_{lat}(t)$ is

$$\delta_{lat}(t) = k_P e(t) + k_d \dot{e}(t) + k_a \ddot{e}(t) \qquad \textbf{(22)}$$

Applying the same sequence as PD on (22) with gains $k_P = 300, k_d = 777\ and\ K_a = 60$, the pulse transfer function of PD-A controller will be:

$$G_{c_2}(z) = \frac{\delta_{lat}(z)}{E(z)} = 6 * 10^5(z^{-2} - 2.1295z^{-1} + 1.13) \qquad \textbf{(23)}$$

The difference equation of (23) is given by:

$$\delta_{lat}(nT_s) = 6 * 10^5(e(n-2)T_s - 2.1295e(n-1)T_s + 1.13e(nT_s)) \qquad \textbf{(24)}$$

Where, $n = 0,1,2, \ldots$

The effects of gains $k_P, k_d\ and\ k_a$ are showed in Table II. (Data in the table obtained from theoretical analysis).

Table II: The effects of $k_P, k_d\ and\ k_a$ independently.

Closed Loop Response	**Rise Time**	**Peak Overshoot**	**Settling Time**	**Steady State error**
Increase k_P	Decrease	Increase	Small Change	Decrease
Increase k_d	Small Change	Decrease	Decrease	Small Change
Increase k_a	Increase	Decrease	Increase	Decrease

To clarify the theoretical difference between PD and PD-A controllers, a unit step response of the linearized system using PD controller with $k_P = 300\ and\ k_d = 10$ (where PD controller is obtained when $k_a = 0$) and PD-A controller with gains $k_P = 300, k_d = 10$ and $k_a = 20, 60\ and\ 100$ is shown in Fig. 7.

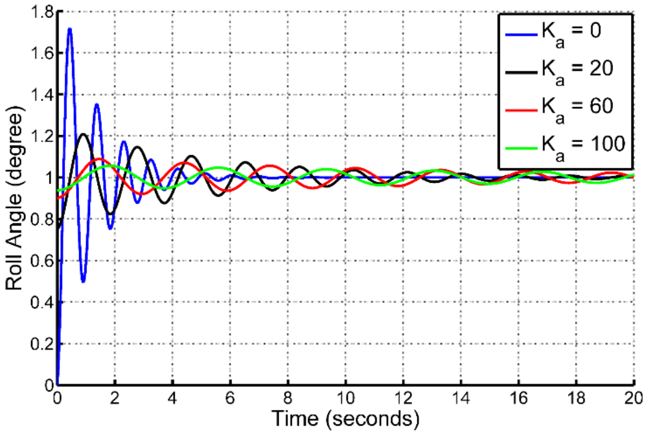

Fig. 7. Unit step response for the linearized model using PD & PD-A controllers

From Fig .7 it is noticed that adding k_a increases the system settling time, but on the other hand it decreases the overshoot and decreases the output frequency.

V. RESULTS AND DISCUSSION

A. Software Simulation

A nonlinear model simulation has been made by means of the simulink toolbox in Matlab as shown in Fig. 8. Applying unit step input signal for latitude (lat.) -while zero input for collateral, longitudinal and pendulum- on the two cases of using PD or PD-A controller in (19) and (23) gave results with the same behavior of the experimental results except that it was much faster than the actual system response due to the fact that the system lag was not taken into consideration.

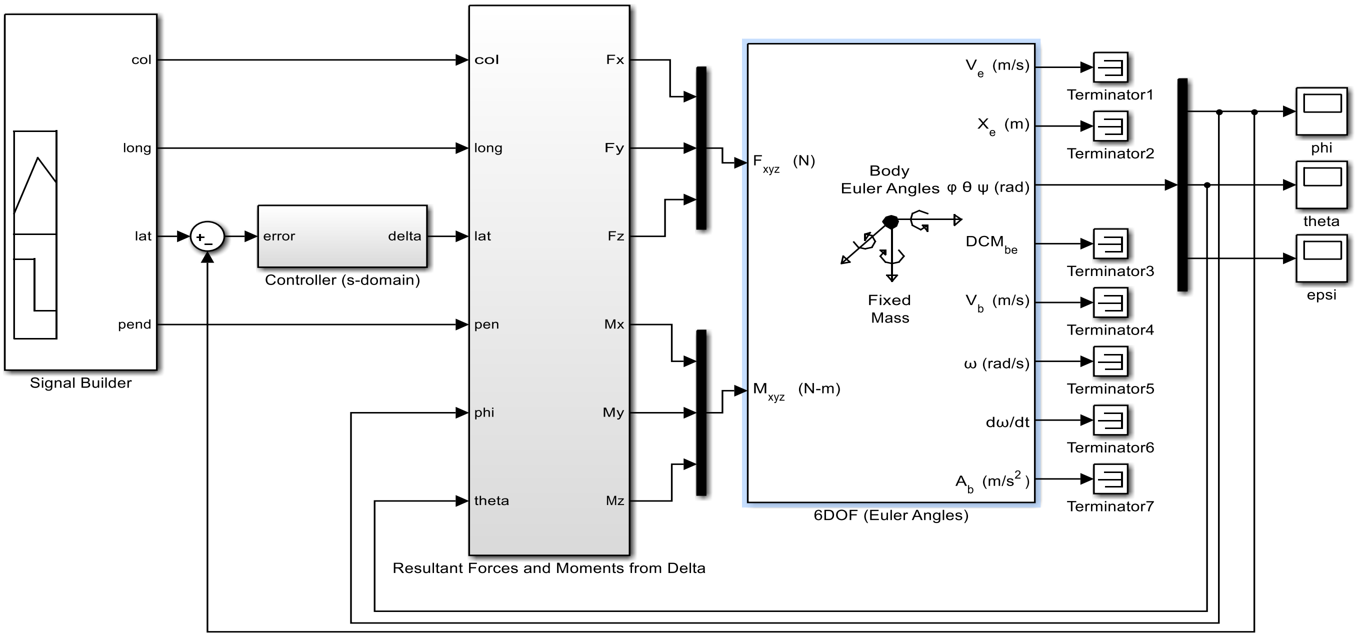

Fig. 8. Simulink nonlinear model

B. Hardware Implementation

Implementing (19) and (23) in the microcontroller, we could get the responses shown in Fig. 9.

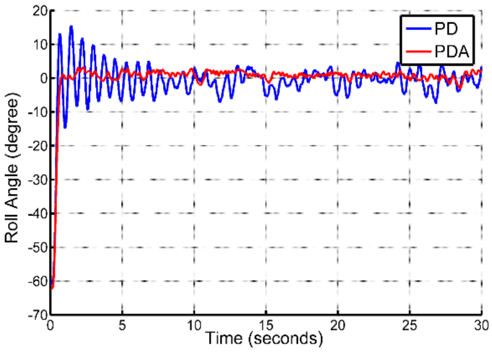

Fig. 9: Actual step response using PD & PD-A controllers

It is noticed from Fig.7 that the theoretical PD has a fast response, while PD-A decreased the system speed. Thus, theoretically speaking, using PD-A considering the system speed is not recommended, but this is not the real case because the actual system is subjected to disturbances with high frequencies coming from various uncontrolled sources. And that is where PD-A becomes more beneficial if it is compared with PD as shown in Fig. 9. As it is noticed, PD cannot handle the disturbances efficiently as compared to the PD-A which although it decreases the system speed, it decreases the overshoot and decreases the frequency and amplitude of fluctuations in the steady state.

VI. Conclusion and Future Work

In this paper, it is shown that if PD controller is used on a system and the system still has relatively high frequency fluctuations, PD-A will be preferably to be used as it showed theoretically and experimentally that it decreases the frequency and amplitude of fluctuations greatly although it decreases the system speed. So PD-A is very useful in damping the high frequency disturbances that may present in the system effectively, which may form an alternative to filters in some cases. But if the PD could guarantee the desired transient and steady state characteristics, the use of PD-A will be a bad choice as it will decrease the system speed.

Future work: Using an integrator, comparing the PID-A with PD-A responses and applying these techniques on a full quadcopter. Also improving the simulink model to represent simulated response time to be closer to reality.

References

[1] A.Azzam and Xinhua Wang in: *Quad Rotor Arial Robot Dynamic Modeling and Configuration Stabilization*, (2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics).

[2] Theerasak S., Pined L., Wonlop C. and Itthisek N., “Path Tracking of UAV Using Self-Tuning PID Controller Based on Fuzzy Logic”, SICE Annual Conference, Taipei, Taiwan, August 2010.

[3] Paul E. I. Pounds and Aaron M. Dollar “Stability of Helicopters in Compliant Contact Under PD-PID Control”, IEEE Transactions on Robotics, Vol. 30, No. 6, December 2014.

[4] Wei W., Hao M., Min X., Liguo W. and Xuefei Y., ” Attitude and Altitude Controller Design for Quad-Rotor Type MAVs”, Mathematical Problems in Engineering, Vo. 2013.

[5] Hongtao Z., Xiaohui Q. and Hairui D., “An Adaptive Block Backstepping Controller for Attitude Stabilization of a Quadrotor Helicopter”, WSEAS Transactions on Systems and Control, Vol. 8, No. 2, April 2013.

[6] An H., Li J., Wang J. and Ma H., "Backstepping-Based Inverse Optimal Attitude Control of Quadrotor", Int. J Adv Robotic Sy, Vol. 10, 2013.

[7] En-Hui Z., Jing-Jing X. and Ji-Liang L., "Second Order Sliding Mode Control for a Quadrotor UAV", ISA Transactions, Vol. 53, 2014.

[8] Bo Z., Bin X., Yao Z. and Xu Z., "Nonlinear Robust Sliding Mode Control of a Quadrotor Unmanned Aerial Vehicle Based on Immersion and Invariance Method", Int. J. Robust Nonlinear Control, 2014.

[9] Bara J. E. and Aydin Y., "Robust Nonlinear Composite Adaptive Control of Quadrotor", International Journal of Digital Information and Wireless Communications (IGDIWC) Vol. 4, No. 2, 2014.

[10] Byung-Yoon L., Hac-In L.. and Min-Jea T., "Analysis of Adaptive Control Using On-line Neural Networks for a Quadrotor UAV", 13th International Conference on Control, Automotion and Systems, Gwangju, Korea, Oct. 2013.

[11] Zachary T. D., Anuradha M. A. and Eugene L., "Adaptive Control of Quadrotor UAVs: A Design Trade Study With Flight Evaluations", IEEE Transactions on Control System Technology Vol. 21, No. 4, July 2013.

[12] A. Bousbaine, M. H. Wu and G. T. Poyi in: *MODELLING AND SIMULATION OF A QUAD ROTOR HELICOPTER.*

[13] R. C. HIBBELER in: Engineering Mechanics – Dynamics, Twelfth Edition.

[14] Tarek N.Deif, Ayman H.Kassem, Gamal M.EL-baioumi: *Attitude Stabilization of Indoor Quad rotor Using classic control*, (International Review of Aerospace Engineering (I.RE.AS.E),Vol.7,N2 June 2014).

[15] Bernard Tat Meng Leong, Sew Ming Low, Melanie Po-Leen Ooi: *Low Cost Microcontroller-based Hover Control Design of a Quadcopter, (*International, Symposium on Robotics and Intelligent Sensors 2012).

[16] Roland S. and Burns "Advanced Control Engineering", 2001- Chapter 10.