International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 21 (2017) pp. 11598-11605
© Research India Publications. http://www.ripublication.com

A Simulation Study for Practical Control of a Quadrotor

Jeongho Noh* and Yongkyu Song**

**Graduate student, Ph.D. program, ** Ph.D., Professor*
Department of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang City, South Korea.
Orcid ID: 0000-0001-7956-8464
***Corresponding author*

Abstract

In this paper a thorough simulation study is performed and analyzed for practical control of a quadrotor. After a systematic construction of a control system, linear and nonlinear techniques are compared to see their performances on position control of a quadrotor. In the simulation heading-hold and heading-varying commands are tried and the feedback effects of Euler angular rates and body rates are also examined. Finally an effective technique for long distance guidance is proposed.

INTRODUCCTION

Recently more and more quadrotors (or multicopters) are adopted to various applications. To perform safe and satisfactory missions a good flight controller should be designed. First a brief survey on quadrotor control is described and then quadrotor dynamics and control structure are discussed. After the controller is set various situations are simulated and analyzed. In the literature several techniques such as linear, nonlinear, learning-based, hybrid schemes are used to control quadrotors. As linear techniques PID or PD controller is most used and Linear Quadratic or H_∞ schemes often used. Those linear techniques, in fact, are good enough for most of quadrotor control [1].

Bouabdallahet al. showed that both PID and Linear Quadratic techniques work well on the control of indoor quadrotor under some disturbances[1,2]. Li et al. used H_∞ and Model Predictive Control to stabilize attitude and position control under weak wind[3]. Since quadrotor dynamics are inherently nonlinear, several nonlinear techniques are tried. Bouabdallah and Siegwart showed that backstepping and sliding-mode techniques can be used to control a quadrotor indoor under some disturbances[4]. Fang et al. combined backstepping with an adaptive controller to deal with model uncertainty and disturbance, reducing outputs' overshooting, delay, and steady-state errors[5]. Feedback linearization techniques first linearize the nonlinear parts of the quadrotor dynamics and adopt linear control for the linearized system. Kendoul et al. showed that feedback linearization is able to control a quadrotor in several flight tests[6]. As a learning-based technique, Efe used neural networks to simplify PID controller, thus reducing computation time[7].

As seen in the above a flight controller based-on a single-technique such as PID does not show a good performance other than a given flight condition.

In this paper thus a thorough simulation study is performed and analyzed for practical control of a quadrotor. After a systematic construction of a control system, linear and nonlinear techniques are compared to see their performances on position control of a quadrotor. In the simulations heading-hold and heading-varying commands are tried and the feedback effects of Euler angular rates and body rates are also examined. Finally an effective technique for long distance guidance is proposed.

QUADROTOR DYNAMICS

In this section simple quadrotor dynamics are introduced using conventional Euler angles [8]. First we consider 4 frames shown in Fig. 1. By rotating the Earth-fixed frame(X_f, Y_f, Z_f, unit vectors $\overrightarrow{\iota_f}, \overrightarrow{\jmath_f}, \overrightarrow{z_f}$) through yaw angle ψ about Z_f-axis, one gets 1-frame (X_1, Y_1, Z_1, unit vectors $\overrightarrow{\iota_1}, \overrightarrow{\jmath_1}, \overrightarrow{z_1}$), and by rotating 1-frame through pitch angle θ about Y_1-axis, one gets 2-frame (X_2, Y_2, Z_2, unit vectors $\overrightarrow{\iota_2}, \overrightarrow{\jmath_2}, \overrightarrow{z_2}$), and finally by rotating 2-frame through roll angle ϕ about X_2-axis, one gets 3-frame (X_3, Y_3, Z_3, unit vectors $\overrightarrow{\iota_3}, \overrightarrow{\jmath_3}, \overrightarrow{z_3}$), which is equal to the body-fixed frame (X_b, Y_b, Z_b, unit vectors $\overrightarrow{\iota_b}, \overrightarrow{\jmath_b}, \overrightarrow{z_b}$).

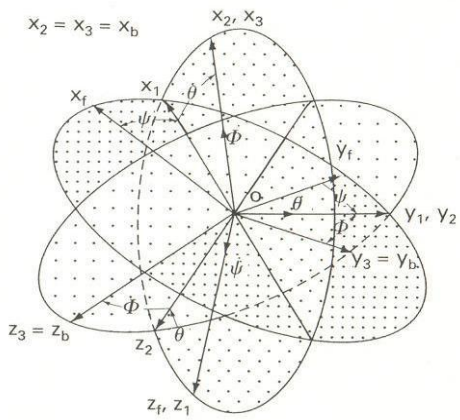

Figure 1: Relationships between frames by Euler angles

To derive quadrotor dynamics let's denote the forces and torques by each motor as F_i, τ_i as shown in Fig. 2. Then the forces driven by motors and gravity are gives as

$$\text{Thrust} \qquad \vec{F} = (-F)\overrightarrow{k_b} \tag{2.1}$$

$$\text{Gravity} \quad m\vec{g} = mg\overrightarrow{k_f} \tag{2.2}$$

where $F = F_f + F_r + F_b + F_l$. Therefore the translational equations of motion are given as

$$m\ddot{\vec{r}} = (-F)\overrightarrow{k_b} + mg\overrightarrow{k_f} \tag{2.3}$$

where $\vec{r} = x_f\overrightarrow{i_f} + y_f\overrightarrow{i_f} + z_f\overrightarrow{k_f}$ is the position vector from the origin of the Earth-fixed frame to the center of mass of the quadrotor.

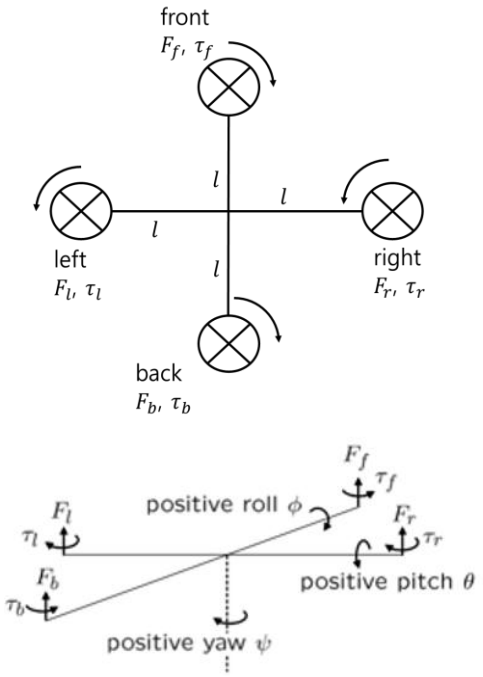

Figure 2: Forces and torques by motors

Then the equations are resolved as follows

$$\begin{aligned} m\ddot{x}_f &= (-F)(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi) \\ m\ddot{y}_f &= (-F)(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi) \\ m\ddot{z}_f &= mg + (-F)\cos\theta\cos\phi \end{aligned} \tag{2.4}$$

These equations can be expressed as follows using the rotation between the Earth-fixed frame and 1-frame.

$$\begin{bmatrix} m\ddot{x}_f \\ m\ddot{y}_f \\ m\ddot{z}_f \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -F\sin\theta\cos\phi \\ F\sin\phi \\ mg + (-F)\cos\theta\cos\phi \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m\ddot{x}_1 \\ m\ddot{y}_1 \\ m\ddot{z}_1 \end{bmatrix} \tag{2.5}$$

These equations can be used in giving heading commands.

If the body-fixed frame for a quadrotor is a principal axes system then the rotational equations of motion are given as follows

$$\begin{aligned} \dot{p} &= \frac{(I_y - I_z)}{I_x} qr + \frac{1}{I_x}\tau_\phi \\ \dot{q} &= \frac{(I_z - I_x)}{I_y} rp + \frac{1}{I_y}\tau_\theta \\ \dot{r} &= \frac{(I_x - I_y)}{I_z} pr + \frac{1}{I_z}\tau_\psi \end{aligned} \tag{2.6}$$

Where

$$\text{Rolling torque} \qquad \tau_\phi = l(F_l - F_r) \tag{2.7}$$

$$\text{Pitching torque} \qquad \tau_\theta = l(F_f - F_b) \tag{2.8}$$

$$\text{Yawing torque} \qquad \tau_\psi = \tau_r + \tau_l - \tau_f - \tau_b \tag{2.9}$$

And the relationship between the angular velocities in the body frame (p, q, r) and the Euler rates are given as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.10}$$

NONLINEAR CONTROL OF A QUADROTOR

For the control of a quadrotor we first consider the attitude control and the position control is then designed based on it. For the attitude control let's assume that the Euler angles and their rates are small then one can assume that the following holds.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.11}$$

Then the rotational equations of motion can be expressed as

$$\begin{aligned} \ddot{\phi} &= \frac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} + \frac{1}{I_x}\tau_\phi \\ \ddot{\theta} &= \frac{(I_z - I_x)}{I_y}\dot{\psi}\dot{\phi} + \frac{1}{I_y}\tau_\theta \\ \ddot{\psi} &= \frac{(I_x - I_y)}{I_z}\dot{\phi}\dot{\theta} + \frac{1}{I_z}\tau_\psi \end{aligned} \tag{2.12}$$

For the attitude control one can use a stable PD controller as follows

$$\begin{aligned} \ddot{\phi} &= \frac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} + \frac{1}{I_x}\tau_\phi = -k_{d1}\dot{\phi} - k_{p1}(\phi - \phi_d) \\ \ddot{\theta} &= \frac{(I_z - I_x)}{I_y}\dot{\psi}\dot{\phi} + \frac{1}{I_y}\tau_\theta = -k_{d2}\dot{\theta} - k_{p2}(\theta - \theta_d) \\ \ddot{\psi} &= \frac{(I_x - I_y)}{I_z}\dot{\phi}\dot{\theta} + \frac{1}{I_z}\tau_\psi = -k_{d3}\dot{\psi} - k_{p3}(\psi - \psi_d) \end{aligned} \tag{2.13}$$

Since (p, q, r) are available from the gyros one can still use them to eliminate the nonlinear terms as follows

$$\tau_\phi \equiv I_x\{-k_{d1}\dot{\phi} - k_{p1}(\phi - \phi_d)\} - (I_y - I_z)qr$$
$$\tau_\theta \equiv I_y\{-k_{d2}\dot{\theta} - k_{p2}(\theta - \theta_d)\} - (I_z - I_x)rp \qquad (2.14)$$
$$\tau_\psi \equiv I_z\{-k_{d2}\dot{\psi} - k_{p3}(\psi - \psi_d)\} - (I_x - I_y)pq$$

Or one can easily feedback (p, q, r) if the maneuver is not so big as follows

$$\tau_\phi \equiv I_x\{-k_{d1}p - k_{p1}(\phi - \phi_d)\} - (I_y - I_z)qr$$
$$\tau_\theta \equiv I_y\{-k_{d2}q - k_{p2}(\theta - \theta_d)\} - (I_z - I_x)rp \qquad (2.15)$$
$$\tau_\psi \equiv I_z\{-k_{d2}r - k_{p3}(\psi - \psi_d)\} - (I_x - I_y)pq$$

For position control we first assume that the heading is held in the North direction, i.e., $\psi = 0$. Then the equation (2.5) is given as follows

$$\ddot{x}_f = (-\tfrac{\mathrm{F}}{\mathrm{m}})(\sin\theta\cos\phi)$$
$$\ddot{y}_f = \left(\tfrac{F}{m}\right)\sin\phi \qquad (2.16)$$
$$\ddot{z}_f = \mathrm{g} - \tfrac{\mathrm{F}}{\mathrm{m}}\cos\theta\cos\phi$$

Now the position control is performed by PD controller as follows

$$\ddot{x}_f = (-\tfrac{F}{m})(\sin\theta\cos\phi) \equiv -k_{d4}\dot{x}_f - k_{p4}(x_f - x_d) \equiv u_x$$
$$\ddot{y}_f = \left(\tfrac{F}{m}\right)\sin\phi \equiv -k_{d5}\,\dot{y}_f - k_{p5}(y_f - y_d) \equiv u_y \qquad (2.17)$$
$$\ddot{z}_f = g - \frac{F}{m}\cos\theta\cos\phi \equiv -k_{d6}\dot{z}_f - k_{p6}(z_f - z_d) \equiv u_z$$

That is, feedback linearization is accomplished by

$$\frac{F}{m} = \frac{g - u_z}{\cos\phi\cos\theta} \qquad (2.18)$$
$$u_y = \frac{g - u_z}{\cos\theta}\tan\phi \qquad (2.19)$$
$$u_x = (u_z - g)\tan\theta \qquad (2.20)$$

Thus the pitch and roll commands for position control can be given by

$$\phi_d = \tan^{-1}\left(\frac{u_y\cos\theta}{g - u_z}\right),\ \theta_d = \tan^{-1}\left(\frac{u_x}{u_z - g}\right) \qquad (2.21)$$

Then the systematic controller block diagram is given as follows

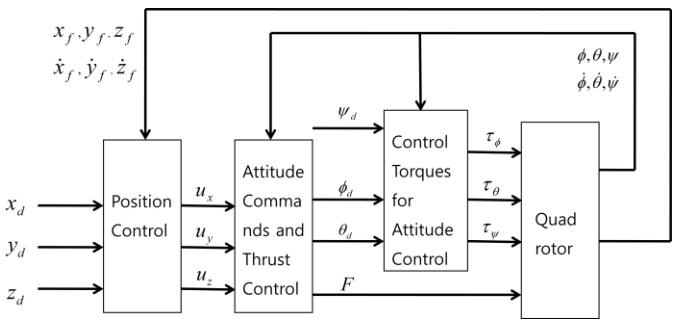

Figure 3: Structure for multicopter control

In case the heading command is not held zero but varys it can be handled as follows. After rearranging the equation (2.4)

$$\begin{bmatrix}\ddot{x}_f\\ \ddot{y}_f\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}-\frac{F}{m}\sin\theta\cos\phi\\ \frac{F}{m}\sin\phi\end{bmatrix} \qquad (2.22)$$

use PD control (2.17) as follows

$$\begin{bmatrix}\ddot{x}_f\\ \ddot{y}_f\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}-\frac{F}{m}\sin\theta\cos\phi\\ \frac{F}{m}\sin\phi\end{bmatrix} = \begin{bmatrix}u_x\\ u_y\end{bmatrix}. \qquad (2.23)$$

Or

$$\begin{bmatrix}-\frac{F}{m}\sin\theta\cos\phi\\ \frac{F}{m}\sin\phi\end{bmatrix} = \begin{bmatrix}\cos\psi & \sin\psi\\ -\sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}u_x\\ u_y\end{bmatrix}$$
$$= \begin{bmatrix}u_x\cos\psi + u_y\sin\psi\\ -u_x\sin\psi + u_y\cos\psi\end{bmatrix} \equiv \begin{bmatrix}u_x'\\ u_y'\end{bmatrix} \qquad (2.24)$$

Thus the pitch and roll commands can be computed as follows

$$\phi_d = \tan^{-1}\left(\frac{u_y'\cos\theta}{g - u_z}\right),\ \theta_d = \tan^{-1}\left(\frac{u_x'}{u_z - g}\right). \qquad (2.25)$$

This technique can be used for arbitrary heading commands.

SIMULATION OF ATTITUDE CONTROL

In order to see the performance of attitude controller we adopt PD controller of the equation (2.14) implementing MATLAB Simulink as follows (Fig. 4.). For the simulation the parameters in [9] are used.

Parameter	Values
m	2.3536 [kg]
I_x	0.1676 $[\mathrm{kg}\cdot m^2]$
I_y	0.1676 $[\mathrm{kg}\cdot m^2]$
I_z	0.2794 $[\mathrm{kg}\cdot m^2]$

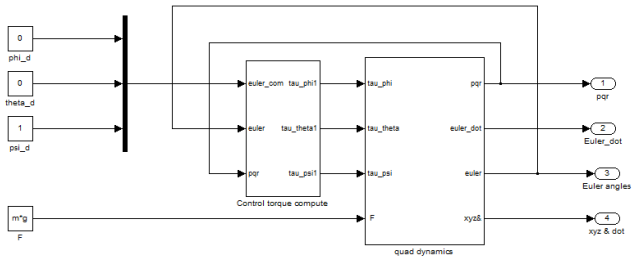

Figure 4: Attitude control of multicopter

In the simulation two sets of control gains were tried as follows.

Gains	Simulation1	Simulation 2
$k_{p1},\ k_{p2},\ k_{p3}$	100	16
$k_{d1},\ k_{d2},\ k_{d3}$	16	4.8

Pitch and roll responses in Fig. 5. are good enough and yaw response is also good though a little overshoot. With a set of big gains the response times are within 0.5 seconds, which seem fast enough for the translational maneuver of a quadrotor.

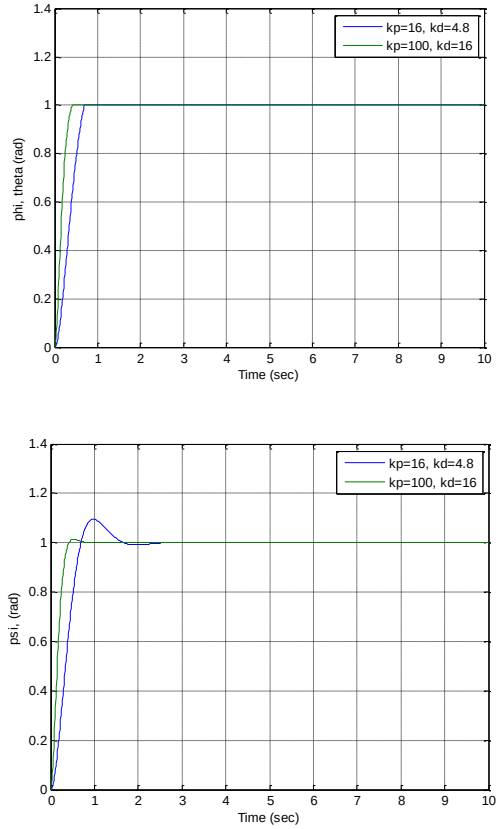

Figure 5: Simulation results of attitude control

POSITION CONTROL WITH NONLINEAR SCHEME

For the position control we need to implement the equations (2.17) and (2.21), which was done using MATLAT Simulink as in the following figure.

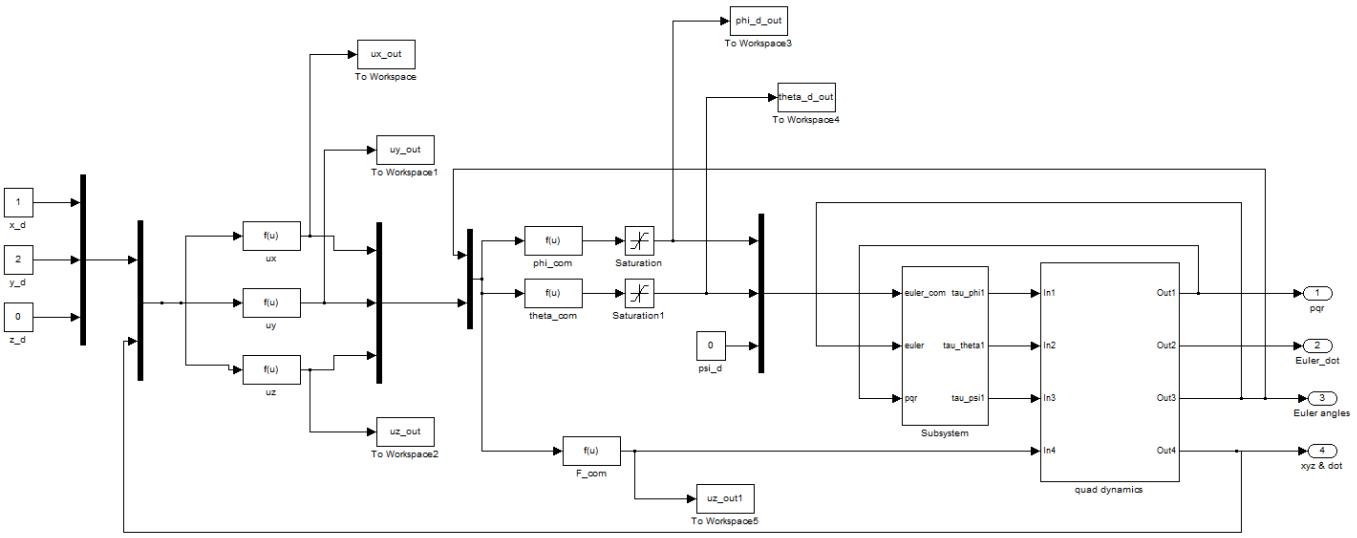

Figure 6. Position control block-diagram for multicopter

In the simulation (Fig. 7.) the altitude and heading commands were set to zero and $x_d = 1\,m, y_d = 2\,m$. The position commands were accomplished in 1 second and the heading is kept zero. The next simulation was done with the heading command of first view to the target, i.e., $\psi_{com} = \text{atan}(2,1)$. In this case the equations (2.18), (2.24), and (2.25) are implemented because the heading is not held zero. The simulation (Fig.8.) shows that the heading catches up the command quickly and the pitching and rolling works together with the heading.

LINEAR CONTROL

In this section we try linear control techniques. First we linearize the quadrotor dynamics (Eqn. (2.12)) assuming that the heading is kept zero and the trim is held for $\phi_0 = 0, \theta_0 = 0$. Then the rotational dynamics are linearized as follows

$$\Delta\ddot{\phi} = \frac{1}{I_x}\tau_\phi$$
$$\Delta\ddot{\theta} = \frac{1}{I_y}\tau_\theta \qquad (2.26)$$
$$\Delta\ddot{\psi} = \frac{1}{I_z}\tau_\psi$$

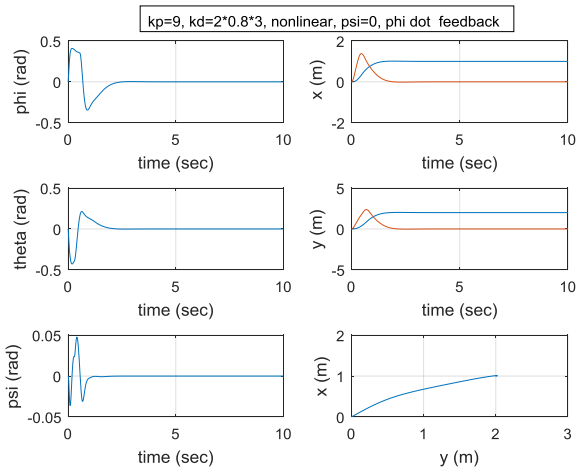

Figure 7: Position control with nonlinear scheme ($\psi_{com} = 0$, Euler rates feedback)

And the translational equations of motion are linearized as follows

$$\Delta\ddot{x}_f = -g\Delta\theta$$
$$\Delta\ddot{y}_f = g\Delta\phi \qquad (2.27)$$
$$\Delta\ddot{z}_f = -\frac{\Delta F}{m}$$

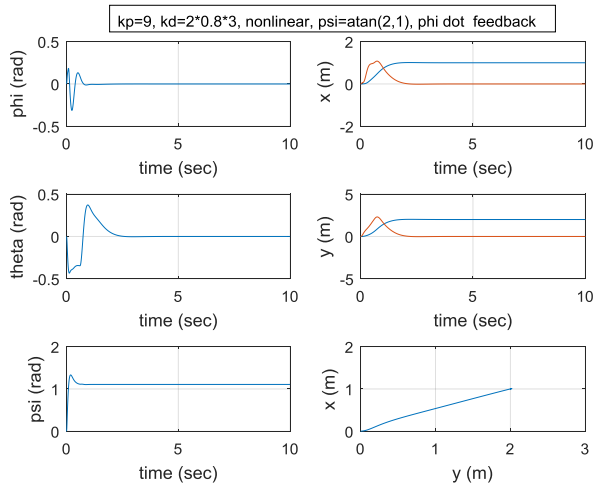

Figure 8: Position control with nonlinear scheme ($\psi_{com} = \text{atan}(2,1)$, Euler rates feedback)

Then if we use PD control as in eqn. (2.17)

$$\Delta\ddot{x}_f = -g\Delta\theta \equiv -k_{d4}\Delta\dot{x}_f - k_{p4}(\Delta x_f - \Delta x_d) \equiv \Delta u_x$$
$$\Delta\ddot{y}_f = g\Delta\phi \equiv -k_{d5}\Delta\dot{y}_f - k_{p5}(\Delta y_f - \Delta y_d) \equiv \Delta u_y \qquad (2.28)$$
$$\Delta\ddot{z}_f = -\frac{\Delta F}{m} \equiv -k_{d6}\Delta\dot{z}_f - k_{p6}(\Delta z_f - \Delta z_d) \equiv \Delta u_z$$

the attitude angle and thrust commands are given as

$$\Delta\theta_d = -\frac{\Delta u_x}{g}$$
$$\Delta\phi_d = \frac{\Delta u_y}{g} \qquad (2.29)$$
$$\Delta F_d = -m\Delta u_z$$

In case the heading is not kept zero one may use the equation (2.22) assuming that the variables are small as follows

$$\begin{bmatrix}\Delta\ddot{x}_f \\ \Delta\ddot{y}_f\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi \\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}-\frac{F_0}{m}\Delta\theta \\ \frac{F_0}{m}\Delta\phi\end{bmatrix}$$
$$= \begin{bmatrix}\cos\psi & -\sin\psi \\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}-g\Delta\theta \\ g\Delta\phi\end{bmatrix} \equiv \begin{bmatrix}\Delta u_x \\ \Delta u_y\end{bmatrix} \qquad (2.30)$$

That is,

$$\begin{bmatrix}-g\Delta\theta \\ g\Delta\phi\end{bmatrix} = \begin{bmatrix}\cos\psi & \sin\psi \\ -\sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}\Delta u_x \\ \Delta u_y\end{bmatrix} \qquad (2.31)$$

Therefore the attitude commands are given as

$$\Delta\theta_d = \frac{\Delta u_x \cos\psi + \Delta u_y \sin\psi}{-g}$$
$$\Delta\phi_d = \frac{-\Delta u_x \sin\psi + \Delta u_y \cos\psi}{g} \qquad (2.32)$$

The simulation results with linear control are similar to those

with nonlinear one as seen in Fig. 9.

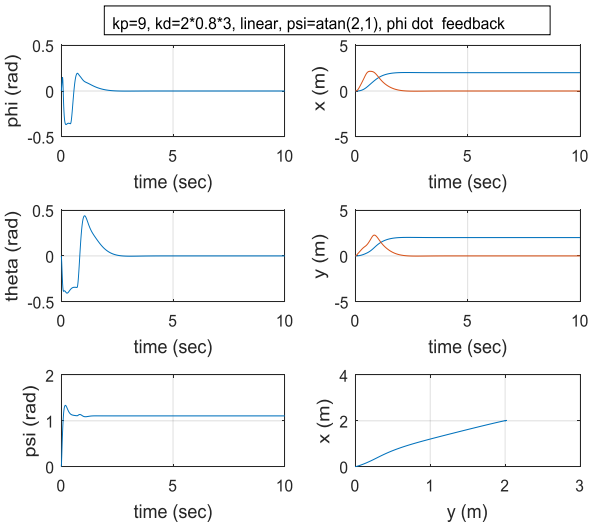

Figure 8: Position control with linear scheme (ψ_{com} = atan(2,1), Euler rates feedback)

The next simulation (Fig. 10) shows the effects of p,q,r feedback rather than Euler rates $\dot{\phi}, \dot{\theta}, \dot{\psi}$ and some overshoots are shown because in this case the exact 2nd feedback system is not made.

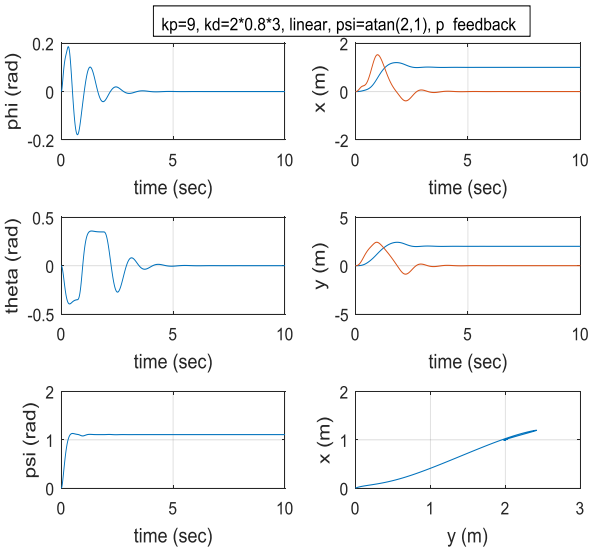

Figure 90. Position control with linear scheme (ψ_{com} = atan(2,1), Angular rates feedback)

Another simulation is tried to see the effect of varying heading command which is given to direct the target all time (ψ_{com} = atan(y, x)). With this heading command the final approach to the target seems unstable due to the sensitivity of the heading computation near the target.

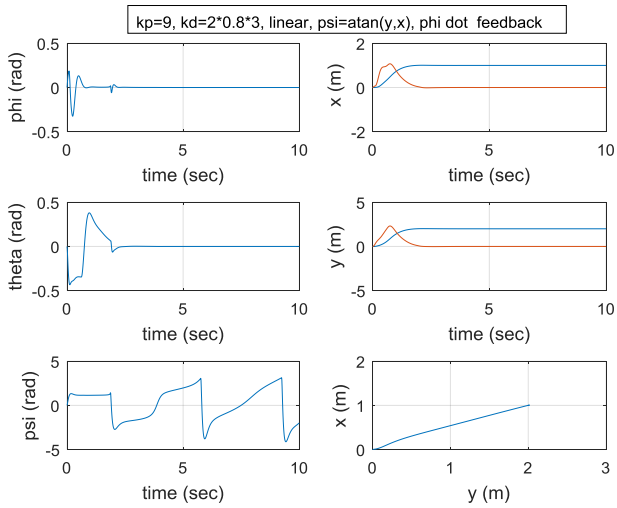

Figure 101: Position control with linear scheme (ψ_{com} = atan(y, x), Euler rates feedback)

POSITION CONTROL VIA SPEED CONTROL

If we try to send the quadrotor farther away (for example, $x_d = 20\,m, y_d = 30\,m$), big overshoots results in with previous gains (Fig. 12.) because the closed-loop system works as a lightly damped system due to the limits of the attitude commands (our case $|\phi_d| \leq 20°, |\theta_d| \leq 20°$)

If the gains are tuned again as in Fig. 13., a good tracking can be made. However, if the target is changed into a far distant position (for example, $x_d = 100\,m, y_d = 200\,m$), then big overshoots come again even with the tuned gains. Thus another gain tuning should be tried. And on can see that it is not easy to tune a proper set of gains for each target position. Therefore it is necessary to find a consistent way to be applied to any target distance. For this purpose a position control via speed control is proposed. That is, a speed control is used until the quadrotor reaches the target within a prespecified radius and then the previously used PD control is applied to precisely send the quadrotor to the target. That is, using the equation (2.27), one can let the feedback keep the speed 10 m/sec, for example, as follows in case $\psi = 0$.

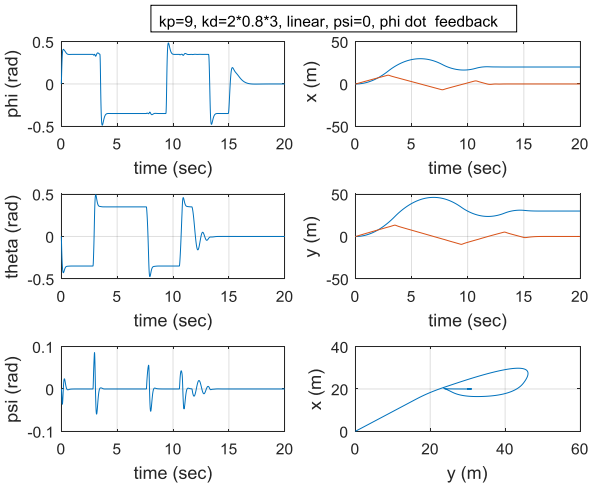

Figure 112: Position control with big overshoots ($(\psi_{com} = 0$, Euler rates feedback)

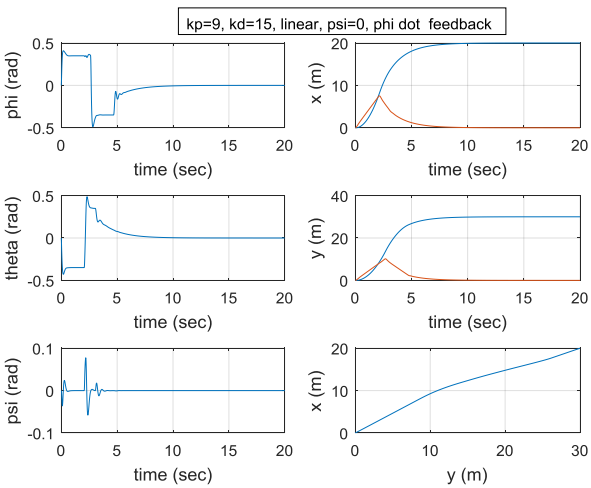

Figure 123: Position control with gains tuned. ($(\psi_{com} = 0$, Euler rates feedback)

$$\begin{bmatrix}\Delta\ddot{x}_f\\ \Delta\ddot{y}_f\end{bmatrix} = \begin{bmatrix}-g\Delta\theta_d\\ g\Delta\phi_d\end{bmatrix} = \begin{bmatrix}-k(\Delta\dot{x}_f - 10)\\ 0\end{bmatrix} \tag{2.33}$$

In case the heading is not kept zero one may use the coordinate transformation between the Earth frame and the 1-frame, that is,

$$\begin{bmatrix}\ddot{x}_f\\ \ddot{y}_f\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}-\frac{F}{m}\sin\theta cos\phi\\ \frac{F}{m}\sin\phi\end{bmatrix}$$

$$= \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}\ddot{x}_1\\ \ddot{y}_1\end{bmatrix} \tag{2.34}$$

Or using the following linearized one

$$\begin{bmatrix}\Delta\ddot{x}_f\\ \Delta\ddot{y}_f\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}-g\Delta\theta\\ g\Delta\phi\end{bmatrix}$$

$$= \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}\Delta\ddot{x}_1\\ \Delta\ddot{y}_1\end{bmatrix} \tag{2.35}$$

We can use the following feedback

$$\begin{bmatrix}\Delta\ddot{x}_1\\ \Delta\ddot{y}_1\end{bmatrix} = \begin{bmatrix}-g\Delta\theta_d\\ g\Delta\phi_d\end{bmatrix} = \begin{bmatrix}-k(\Delta\dot{x}_1 - 10)\\ 0\end{bmatrix}. \tag{2.36}$$

Thus once the flying gets started the quadrotor turns the heading to the target line, i.e.,

$$\psi_c = tan^{-1}(\frac{x_d - x_f}{y_d - y_f}).$$

and then the speed control (2.36) is operated until the quadrotor reaches near the target and then PD controller used. Fig. 14 shows this simulation results.

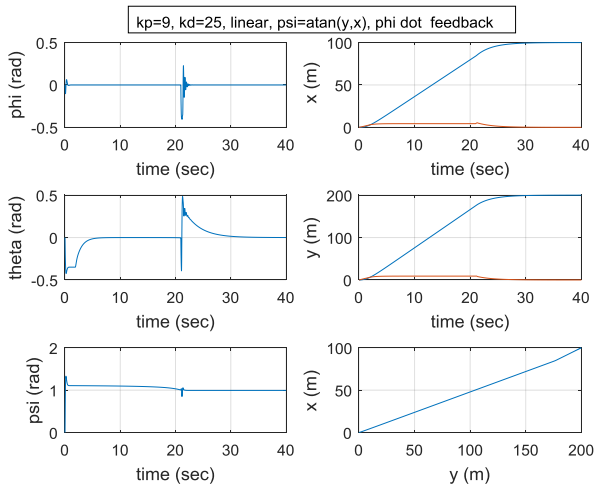

Figure 14: Position control via speed control ($\psi_{com} = \mathrm{atan}(y, x)$)

One can still use different speed command schedule according to the remaining distance.

CONCLUSION

In this work various situations were simulated for quadrotor control. First the simulations show that nonlinear and linear control schemes are similar in performance. As far as the heading command is concerned constant heading commands, whether it is zero or not, are not a problem but real-time varying heading command to the target seems to bring the instability near the target due to the computational sensitivity. For the attitude control it seems better to feedback the Euler rates rather than the body rates. Finally in order to send the quadrotor to a target with arbitrary distance the proposed control strategy (position control via speed control) seems working well whereas the gains should be changed in the gain tuning method.

ACKNOWLEDGEMENT

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) through Advanced Production Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA)(grant number : 315014-3)

REFERNNCES

[1] Paul Edward Ian Pounds "Design, construction and control of a large quadrotor micro air vehicle," Phd. thesis, Australian National University, 2007.

[2] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," In Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, volume 3, pages 2451–2456 vol.3, 2004. doi: 10.1109/IROS.2004.1389776.

[3] Jun Li and Yuntang Li, "Dynamic analysis and PID control for a quadrotor, " In Mechatronics and Automation (ICMA), 2011 International Conference on, pages 573–578, 2011. doi: 10.1109/ICMA.2011.5985724.

[4] S. Bouabdallah and R. Siegwart. "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 2247– 2252, 2005. doi: 10.1109/ROBOT.2005.1570447.

[5] Zheng Fang and Weinan Gao, "Adaptive backstepping control of an indoor microquadrotor," Research Journal of Applied Sciences, 4, 2012.

[6] Farid Kendoul, Zhenyu Yu, and Kenzo Nonami, "Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicle," Journal of Field Robotics, 27(3):311–334, 2010.

[7] M.O. Efe, "Neural network assisted computationally simple pid control of a quadrotor uav," Industrial Informatics, IEEE Transactions on, 7(2):354–361, 2011. ISSN 1551-3203. doi: 10.1109/TII.2011.2123906.115

[8] R. C. Nelson, Flight Stability and Automatic Control, 2nd Ed., McGraw-Hill, 1998.

[9] A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. Control Theory & Applications, IET, 3(3):303–314, 2009.

[10] C. Balas, *Modeling and Linear Control of a Quadrotor,* MSc Thesis, Cranfield University, 2007