Real Time Simulation of Autonomous Quadrotor

Idris E. Putro, G. B. Kim, Gilar B. Raharja, K.J. Yoon

Department of Aerospace Information System, Konkuk University.
#1 Hwayang-dong Kwangjin-gu, 143-701 Seoul. Korea
**Corresponding: Gyou-Beom Kim, (phone: +82-2-447-7091, e-mail: dodol2k@hotmail.com)*

Abstract. The use of micro UAVs has been widely well known. An interesting type to be developed is quadrotor UAV. A quadrotor is a small, agile four-rotor system which is suitable for surveillance, remote inspection and disaster support missions. This kind of vehicle has nonlinear behavior, yet it is not as complex as an unmanned small scale helicopter. In this research, nonlinear quadrotor modeling and linearization of this model were done. These dynamic system and linear model were carried out to develop an autonomous control system.

This paper focuses on the development of a quadrotor control system which was validated through real time simulation for further flight test application. LQR Optimal control was chosen to accomplish the trajectory following mission. This control design was applied both on the quadrotor linear model and nonlinear model based on Matlab/Simulink Environment. Real time simulation was developed to validate the control system. xPC Target was used for prototyping, testing and deploying real-time system using single board computer (SBC) PC-104. This configuration required 2 host PCs and 2 SBCs acting as target PC. The host PCs were used to control the target PCs while these target PCs were ran simultaneously in real-time. Target PCs were divided into Plant Platform and Controller Platform. Plant platform was used to represent nonlinear or linear model of quadrotor, while control algorithm was compiled to the other systems as controller platform. Final Simulation was generated through standalone target boot mode at which two target PCs were connected to each other in sending and receiving data for real-time simulation. This real-time simulation was expected to resemble the environment of a real flight test.

http://www.imavs.org/

http://www.imavs.org/

1. INTRODUCTION

Now days, the developments of unmanned aerial vehicles and flying robots have rapidly increased. The wide ranges application of these vehicles such as surveillance, disaster prevention and defense or security support mission made them became the important subjects to be developed. One of the suitable aerial vehicles for those purposes is Quadrotor. The small quadrotor has capability to hover, and vertical take-off and landing (VTOL) with having less complexity in vehicle dynamics compared with small helicopter. However, the autonomous control system development of small quadrotor is still challenging as well as small halicopter due to its nonlinear behavior and inherently unstable.

Designing and testing of control system cannot be done by directly doing flight test. It will risk the vehicles due to an expensive onboard system and spent a lot of time and effort to understand the quadrotor system behavior. Hence, understanding its flight behavior for designing control system become important steps to do before flight test will be done. Flight and control simulation are the best ways to design and test the control system associated with the quadrotor behavior. Using simulation, we can simulate every flight condition prior to applied our control system and then analyzed or revised it so that the precision controller can be develop more accurate. Once, we ensure that the controller is robust enough, then we can implement it as an embedded system for the flight test.

Developing embedded controller must be consider carefully since the embedded system will be applied in real situation on the flight test. The responses of the embedded controller must be able to suffice deterministic time and meet the real-time environment. Hence the Flight and control simulation should be set into real-time simulation.

There have been a lot of method to develop real-time simulation for testing flight controller. Here, we present real-time simulation based on xPC target for our controller and implemented it to our autonomous quadrotor. The quadrotor model either nonlinear or linear will be run simultaneously while the controller will response and try to stabilize the quadrotor behavior. All the real-time simulation based on xPC target were carried out under

Matlab/Simulink Environment by using Real Time Workshop (RTW).

2. MAV CONFIGURATION

The quadrotor is designed to meet the requirements of a versatile platform for our research-lab and equipped with necessary hardware for the MAV 2010 Competition. The selection of sensor-types is a trade-off between precision, weight, and cost, thus the added sensors are as the following table:

Sensors	Manucfatures/ Brands	Specifications
Processor	ATMega 2560	- 16 MHz - 8 bit
AHRS	XA3300	- < 100 Hz - 0.2^0static error
Ultrasonic	SensCom MiniA	- 50KHz electrostatic - 0.15 to 6.10 m distance range
Infrared	6P2Y0-A710	- ADC output - 70 to 450 cm distance range

Table 1: *Sensors specifications*

Radio control was used to excite commands through transmitter and received by radio receiver on quadrotor vehicle. Then the receiver generates input signal for rotor as actuator system to lead quadrotor move in translation and angular motions.

Internal-loop controller was also added to stabilize the angular behavior. This internal-loop was known as stability augmentation system (SAS) and using PID method to keep stability of quadrotor dynamics in attitude performances.

The main specifications of entire quadrotor vehicle are listed in the table below:

Height	9.5	cm
Length	38	cm
Weight	0.499	kg
Blade Span	10	cm
Payload Capacity	0.15	kg

Table 2: *Quadrotor specifications*

The payload capacity is suited for carrying the sensory and others hardware equipment required for autonomous flight

3. REALTIME HARDWARE SETUP

Real-time simulation will be configured through xPC Target, and the quadrotor model will be carried out using real time workshop (RTW) to meet real-time requirement.

This hardware setup consist of 2 host PCs and 2 target PCs, serial cable connection with cross-configuration for target PCs to communicate each others. The host PCs will be connected to target PCs through 100Mbps TCP/IP connection for downloading executable files into target computer using xPC target in MATLAB. This high-speed connection allows to use xPC target explorer in the host computer as a console for monitoring and controlling the target computer during the process of the Real time simulation.

http://www.imavs.org/

http://www.imavs.org/

Firstly, we configured the xPC target using DOSloader target boot mode which will allow us to control the target PCs through host PCs during simulation. Model parameters and others flight conditions can be changed easily during simulation process without starting simulation from the beginning. However, this mode have not give us real flight simulation since the controller model and quadrotor model still being controlled by host PCs which are run under windows operating system.

Secondly, the real simulation will be carried out under Standalone target boot mode. This configuration will mimic the embedded hardware for real flight test situation. In standalone target boot mode, the quadrotor model and controller model should be compiled to generate the same model for real-time execution. Those compiled model cannot be controlled by host PCs during simulation since those model were starting automatically after the target PCs booting-up their system. In this step, both target PCs are running independently and communicate each other using serial cable connection. The quadrotor model will be represented by model target PC and controller model will be represented by controller target PC. Those target PCs are interacting actively and simultaneously in real-time as well as the embedded controller will response the quadrotor dynamics in real flight.

The host PCs and target PCs arrangements are shown on the following figures:

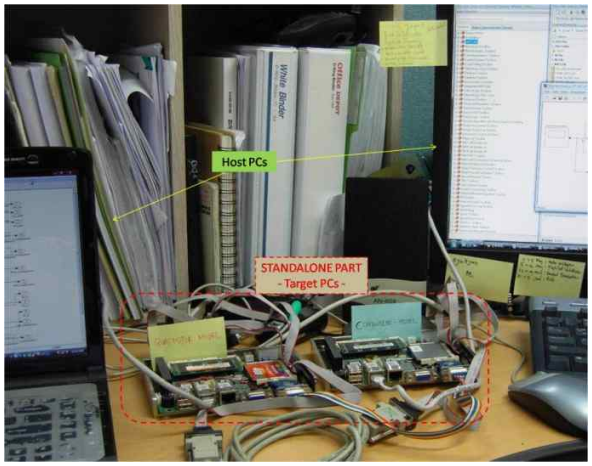

Figure 1: *Host and Target PC Arrangement*

Host PCs

Basically the host PCs is working under Windows operating system to do Matlab/Simulink software. The host PC will run quadrotor model and controller model under Matlab/Simulink and generate compiled files for target PCs to be controlled during simulation.

This host PCs will be connected to target PCs through 100Mbps TCP/IP connection for downloading executable files into target computer using xPC target in Matlab. This high-speed connection allows using xPC target explorer in the host computer as a console for monitoring and controlling the target computer during the process of the

real time simulation. The host PCs also will be used to display and analyze all of the flight simulation data whether from the quadrotor model or controller model.

Figure 2: *Host computer runs simulink model of vehicle*

Acer 4935 Laptop is used as a host computer to run the quadrotor model, meanwhile desktop computer is used to run controller model. The specification of those host PCs can bee seen as below :

Specifications	Acer 4935	Desktop Computer
Processor	Intel Core 2 Duo 2.0 GHz	Intel Pentium IV 2.4 GHz
Memory	DDR2 SDRAM 2 GB	DDR SDRAM 1 GB
Graphic	512 MB dynamics memory	256 MB
Display	14.1 inches wide bright	15 inches
Communication port	USB 2.0 and UDP port	USB 2.0, UDP port and serial port

Table 3: *Specification of Host Computer*

Communication between host PCs and target PCs are connected using UDP port. This high-speed connection will establish xPC target spy in host PC while the simulation is running.

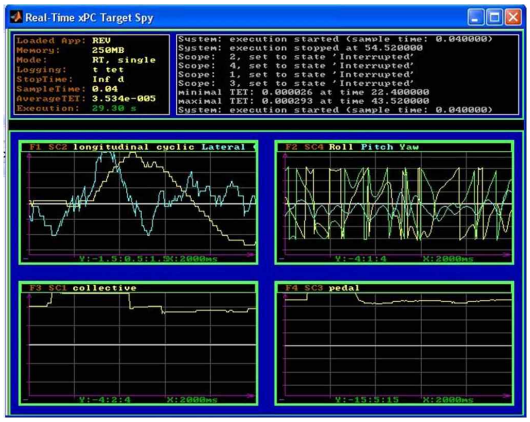

Figure 3: *xPC target spy runs nonlinear model of quadrotor*

xPC target spy is target screen spy in host PC while the target PC was running. This command is commonly used to display the target PC simulation in host PC thus we can monitor the simulation of target PC on the host PC and directly controlled it from host PC.

Target PCs

The target PCs are the computer which are running the executable files representing quadrotor model and controller model and will be run simultaneously as a real-time kernel application on free-DOS operating system. Target PCs can be controlled

http://www.imavs.org/

http://www.imavs.org/

through host PCs under DOSloader target boot mode and also can be run automatically under standalone target boot mode. In this research, we have done both of those configurations where the DOSloader mode was using to tune the quadrotor model and controller parameters, then those optimized model were used to generate compiled model for running under standalone mode. The last configuration of controller model on target computer resemble the embedded controller system that can be used for real flight test.

The target PCs were using a 3.5" single board computer (SBC) that powered by Intel Celeron 600MHz and 512MB DDR memory (SBC: WAFER-8522). The main specifications of target PC can be seen as the following table:

CPU	On-board Intel® Celeron®M 600MHz 512K cache
Memory	DDR226 512 MHz
I/O	5x RS-232 Dual Realtek RTL8110SC Ethernet 4x USB 1x IDE 1x PS/2 for KB/MS 1x TPM v.1.2 module on-board
Dimensions	146 mm x 102 mm
Weight	GW: 700 g and NW: 175 g

Table 4: *Specification of Target Computer (SBC-Single Board Computer)*

The SBC performance allows the process running in high-frequency as in desktop PC under a small compact package that makes possible to used as flight controller computer in the future. The integrated VGA on SBC is used to display the running behavior of each model in an LCD monitor; this design sustains observation and analysis the simulation process in standalone mode, so that host computers can be eliminated for controller model to simplify the system after all of configuration has been fixed previously by using DOSloader mode.

Figure 4: *Single Board Computer WAFER-8522*

SBC quadrotor model is connected to SBC controller model by two channels of RS-232 at 115,200 bps, 8 data bits and 1 stop bits. Output data from quadrotor model was sent through RS-232 com-1 port and will be received as input data by SBC controller model at com-1 port. Reversely, command data from SBC controller model was sent through RS-232 com-2 port and received as

input data at com-2 port by SBC quadrotor model. Since the SBC has six RS-232 ports, using these two channels separately can give advantages to avoid serial communication problems such as CPU overload that might be occurred when using only one serial port alternately

4. QUADROTOR DYNAMICS

The real-time simulation of autonomous quadrotor is using small quadrotor flying robot which has been developed in our laboratory. The nonlinear model of this quadrotor has also been derived and designed into Matlab/Simulink environment to support real time simulation.

The quadrotor is modeled as four rotors system with cross configuration. Each rotors system are consisted of propeller connected to the motor. All rotors system are placed symmetrically and its propellers is set to be parallel from opposite ones in term of rotation axes. The front and rear propeller rotate counter-clockwise, while the left and right ones turn clockwise.

The translation motions are provided by increasing (or decreasing) the propeller speeds. If these propeller speeds are regulated to the same amount, it leads to a vertical forces which raise or lower the altitude of quadrotor. The rotation motions are provided by increasing (or decreasing) the cross-coupled propeller speeds. It leads to a torque with respect to the certain axis (xB and yB) which make the quadrotor turn roll or pitch. The overall vertical thrust is the same as in hovering, hence this command lead only to a angular acceleration such as rolling, and pitching. Yawing motion is provided separately, by increasing (or decreasing) the front-rear propeller's speed and by decreasing (or increasing) the left-right couple in the same time. It leads to a torque with respect to the zB axis which makes the quadrotor turn yaw.

Starting from Newton-Euler formulation, the translation and angular motions of the quadrotor dynamics can be derived as [4]:

$$u = \frac{1}{m}(a\psi s\theta a\varphi \quad + s\psi s\varphi\ ).\sum_{i=1}^{4} T_i \tag{1a}$$

$$v = \frac{1}{m}(s\psi s\theta a\varphi \quad - a\psi s\varphi\ ).\sum_{i=1}^{4} T_i \tag{1b}$$

$$w = g - \left(\frac{1}{m} a\theta\ .a\varphi\ .\sum_{i=1}^{4} T_i\right) \tag{1c}$$

$$\dot{p} = \dot{\theta}.\frac{\dot{\psi}(I_{yy} - I_{zz})}{I_{xx}} + \frac{l}{I_{xx}}(T_4 - T_2) \tag{2a}$$

$$\dot{q} = \dot{\varphi}.\frac{\dot{\psi}(I_{zz} - I_{xx})}{I_{yy}} + \frac{l}{I_{yy}}(T_3 - T_1) \tag{2b}$$

$$\dot{r} = \dot{\varphi}.\frac{\dot{\theta}(I_{xx} - I_{yy})}{I_{zz}} + \frac{l}{I_{zz}}(\tau_1 - \tau_2 + \tau_3 - \tau_4) \tag{2c}$$

http://www.imavs.org/

http://www.imavs.org/

The translation motions are primary influenced by Thrust profiles provided by four rotor system and also attitude performance of quadrotor flight. On the other side, the rotational motions are influenced by Inertial moments of the vehicles and also hub-forces which lead the moment for vehicle motion.

5. CONTROL SYSTM DESIGN

Optimal Control is used to improve the attitude stabilization. Optimal control based on linear quadratic regulator (LQR) is developed and tested to keep the angular angle of the quadrotor stable performance when small disturbances occur. This can be done by setting the angular rate errors tend to zero when disturbance applied thus the angle still can be kept after steady state.

The LQR method try to find the control inputs that can minimize the chosen output by minimized Q and R weighting matrices [5][6].

Let subject to the system dynamics can be gives as the following:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (3)$$

The state-feedback law u=-Kx minimize the quadratic cost function:

$$J(u) = \int_0^\infty (x^T Q x + u^T R u + 2x^T N u)dt \quad (4)$$

By maintaining minimize quadratic cost function thus the state vectors xt is regulated as close as the origin without an excessive expenditure of control effort [16].

Furthermore the optimal gain matrix can be found by solving the matrix differential of Riccati equation:

$$\dot{K}(t) = K(t)A(t) + K(t)B(t)R^{-1}(t)B^T(t)K(t) - Q(t) - A^T(t)K(t) \quad (5)$$

Then the optimal gain matrix can be obtain from the following equation:

$$u^*(t) = -R^{-1}(t)B^T(t)K(t)x(t) \quad (6a)$$

$$u^*(t) = -K_{opt}(t)x(t) \quad (6b)$$

$$K_{opt}(t) \triangleq -R^{-1}(t)B^T(t)K(t) \quad (7)$$

6. REAL TIME SIMULATION

Real time simulation was done by simulate both quadrotor model and controller model of target PCs. This model were obtained from xPC target compiling process. Plant target pc which is represented quadrotor linear model were connected to the controller target pc through serial cable connection on RS232 port. The simulink block diagram of those target PCs are shown as below:

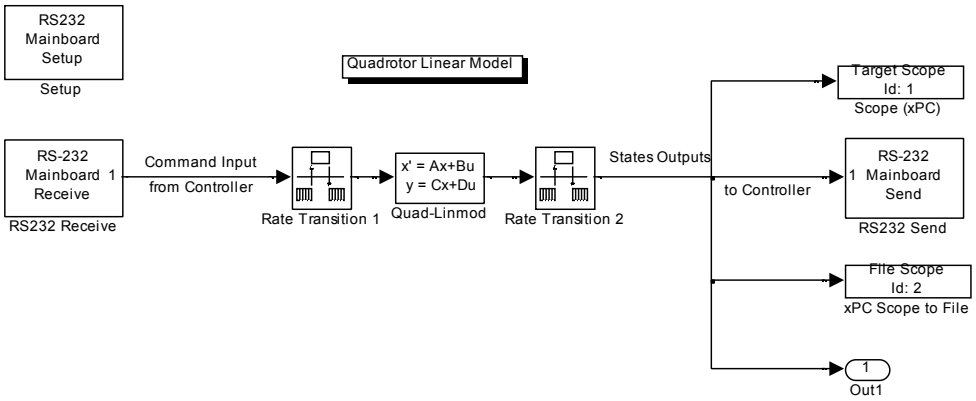

Figure 5: *Simulink block of Quadrotor Linear Model*

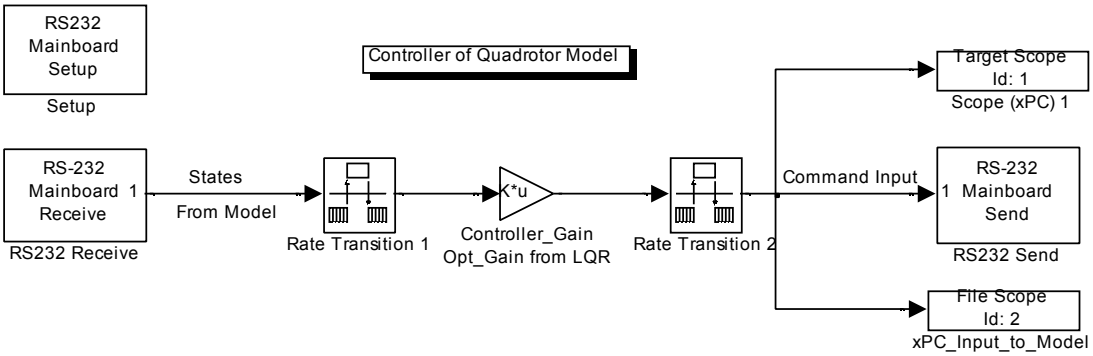

Figure 6: *Simulink block of Quadrotor LQR Controller Model*

Plant target and Controller target PCs above were compiled using C compiler on Real Time Workshop under Matlab/Simulink Environment and generated executable files to be run on single board computer. These executable files running on SBC-target computer are called target PCs.

Simulation result under real time environment can be seen as the following figure:

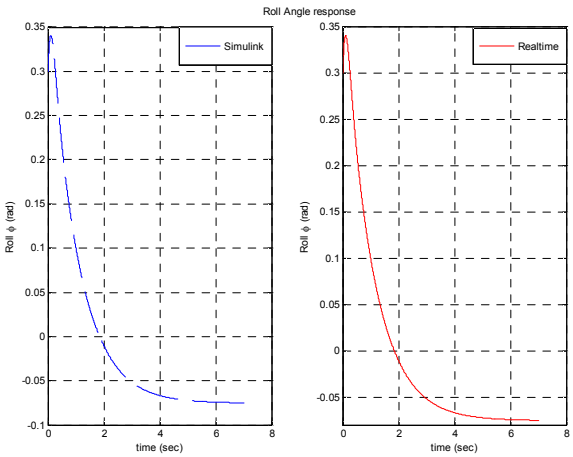

Figure 7: *Comparison of simulink and target pc simulation: Roll angle response*

http://www.imavs.org/

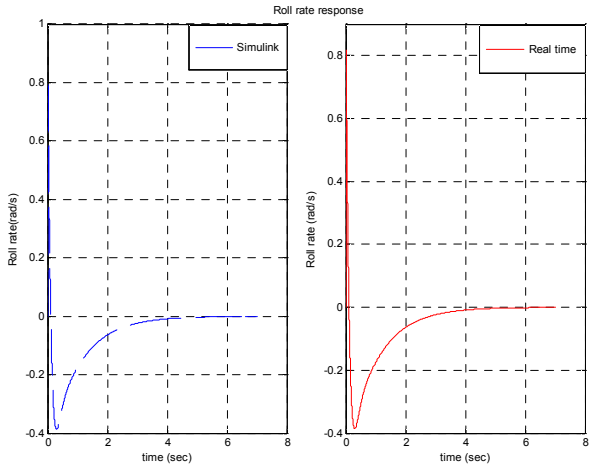

Figure 8: *Comparison of simulink and target pc simulation: Roll rate response*

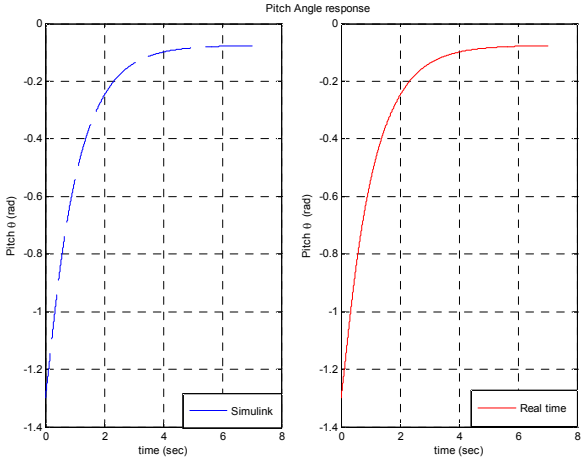

Figure 9: *Comparison of simulink and target pc simulation: Pitch angle response*

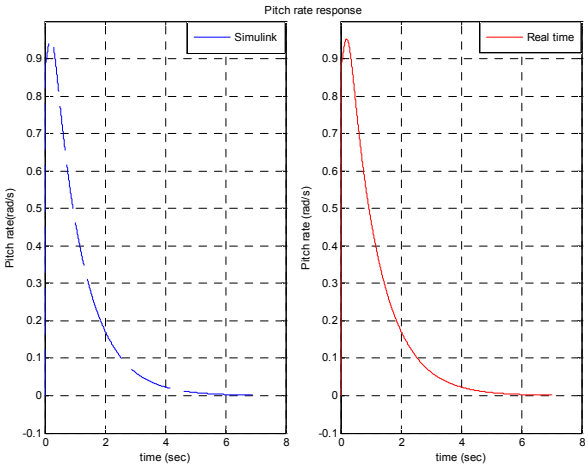

Figure 10: *Comparison of simulink and target pc simulation: Pitch rate response*

Those figure above show the attitude response on lateral mode. There are a very small different response time between simulink result and realtime result. The difference value only within 0.001 sec since the plant model was very simple.

From realtime simulation also can be seen that the controller is able to work properly. The controller can maintain the roll rate close to zero ($p \approx 0\, rad\ /s$) and also keep the roll angle to $\varphi = -0.074\, rad$ for hovering flight.

The others simulation results for Longitudinal and directional mode in hovering condition are depicted from figure below:

http://www.imavs.org/

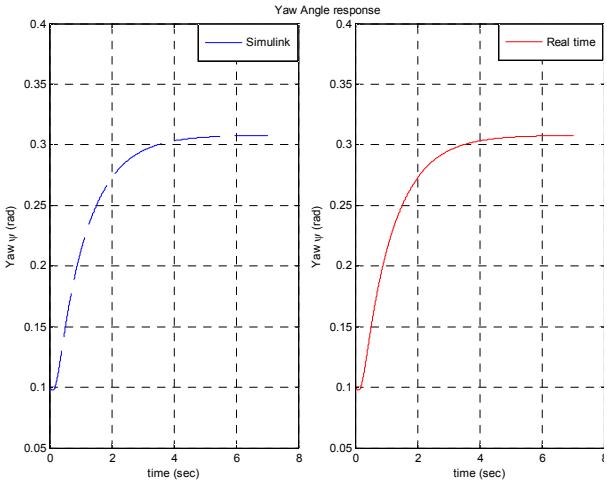

Figure 11: *Comparison of simulink and target pc simulation: Yaw angle response*

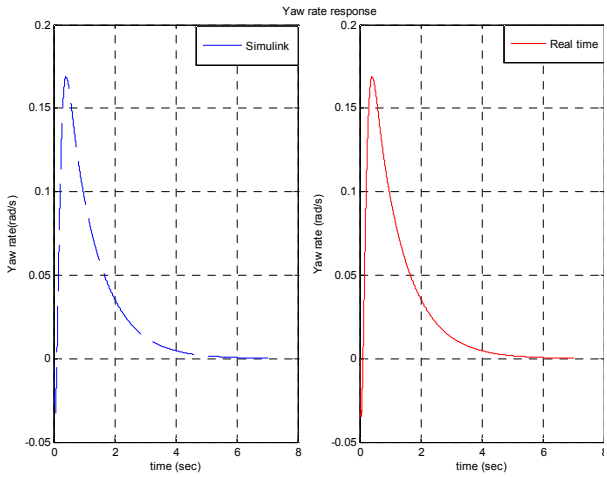

Figure 12: *Comparison of simulink and target pc simulation: Yaw rate response*

Figure 9 to 12 show the same behavior with lateral mode. The controller in real time situation is able to keep pitch and yaw angle near the operating points of hovering flight, $\theta = -0.0767\ rad$ and $\psi = 0.3077\ rad$. Meanwhile the model also can maintain the rate attitude tend to zero, $q = 0\ rad/s$ and $r = 0\ rad/s$

7. CONCLUSION

Rea-ltime Simulation of Quadrotor has been done. Quadrotor model and controller model are able to simulate simultaneously in real-time based on xPC target. It was demonstrated that Plant target PC and Controller target PC were successfully run under standalone mode and maintain its operating points for hovering flight condition. The small difference of operating points are reasonable since the controller has only handled the data from quadrotor linear model in attitude performances.

Further development can be planned by adding actuator system to this real-time hardware setup, thus hardware in the loop simulation can be developed for ground test simulation to support flight test preparation.

http://www.imavs.org/

ACKNOWLEDGMENT

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-C1090-1031-0003)

REFERENCES

[1]Pieter J. Mosterman, Sameer Prabhu, Andrew Dowd, John Glass, Tom Erkkinen, John Kluza, and Rohit Shenoy, Embedded Real-Time Control via MATLAB, Simulink, and xPC Target. Handbook of Networked and Embedded Control Systems, page 419-446, Birkhäuser Publisher, Boston, 2005

[2]I. Hassanzadeh, S. Mobayen, and G. Sedaghat, Design and Implementation of a Controller for Magnetic Levitation System Using Genetic Algorithms, Journal of Applied Sciences 8 (24), page 4644-4649, 2008

[3]Yunpeng Cao, Wanqing Teng, and Huijie Zhang, Dynamic Modeling and Hardware-in-the-Loop Simulation Testing for LPG Engine, Proceedings of the 2007 IEEE Int'l Conference on Mechatronics and Automation, August 5-8, 2007, Harbin, China.

[4]S. Bouabdallah, P. Murrieri, R. Siegwart, "Design and Control of an Indoor Micro Quadrotor", Proceeding of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, April 2004

[5]Katsuhiko Ogata, Matlab for Control Engineers, Pearson Prentice Hall, page 395-398, USA, 2008

[6]The MathWorks™, Matlab Help, Control System Toolbox/Function Reference/LQR Design

http://www.imavs.org/