

Development of Real-Time Flight Simulator for Quadrotor

by Joga Setiawan

Submission date: 09-Jul-2020 11:35AM (UTC+0700)

Submission ID: 1355276092

File name: Development_of_real-time_flight_simulator_for_quadrotor.pdf (952.04K)

Word count: 3057

Character count: 15478

Development of Real-Time Flight Simulator for Quadrotor

Joga D. Setiawan¹, Yuhanes D. Setiawan¹, Mochammad Ariyanto¹,
 Agus Mukhtar¹ and Agus Budiyo²
¹Laboratory of Control System and Robotics
 Faculty of Engineering, Diponegoro University, Indonesia
²Smart Robot Center
 Aerospace Engineering Department, Konkuk University, South Korea
 Email: jdharma@yahoo.com, yuhanes_dedy@yahoo.com

Abstract—The purpose of this research is to build a real-time flight simulator for quadrotor systems and verify if the simulator runs in real time. The simulator was constructed by writing the mathematical models of quadrotor dynamic in MATLAB/Simulink. Using xPC Target software environment, three PCs were set up to work in sync to make the simulation run in real time. This paper shows several simulation results of the quadrotor dynamic in three flight directions. These results validate the stable performance of PD controller being used. At the same time it can be revealed that the simulation runs in real time by observing the percentage processing usage of the computer as function of simulation time. The simulator can be very useful for researchers to understand the character of the quadrotor system being developed and for pilots to complete a mission using the quadrotor system.

I. INTRODUCTION

The most widely used of UAV is rotary wing UAV, because it can be used for aerial monitoring, aerial surveillance, aerial photograph, aerial object interaction/aerial grasping [1], [2], and also it can fly aggressively, as well as vertically take-off landing (VTOL). Nowadays one of the most developed rotary wing UAVs is quadrotor. Real-time flight simulator is one of the tools used for designing quadrotor systems. The benefits of the simulator are to reduce risk of accident, design development time, cost, and number

Manuscript received September 22, 2012. This work was supported in part by Research Grant of National Innovation System, Ministry of Research and Technology, Republic of Indonesia, under research contract Grant 38.6/UN7.5/PG/2012.

Yuhanes D. Setiawan, Mochammad Ariyanto, and Agus Mukhtar are with Control Systems and Robotics Laboratory, Mechanical Engineering Department, Diponegoro University, Indonesia (+628179565980, e-mail: yuhanes_dedy@yahoo.com, ari_janto5@yahoo.co.id, agus_mukhtar@yahoo.co.id).

Joga D. Setiawan is Faculty Member at Mechanical Engineering Department, Diponegoro University, Indonesia (e-mail: jdharma@yahoo.com).

Agus Budiyo is Faculty Member at Smart Robot Center, Aerospace Engineering Department, Konkuk University, South Korea (e-mail: agus@konkuk.ac.kr).

of flight test.

Konkuk University [3] has developed real-time simulation of quadrotor. The real-time simulation was done in MATLAB/Simulink using xPC Target in which two host PCs and two PC targets were used. However the development of real-time simulation of quadrotor to be used for flight simulator has not been done yet. The closest flight simulator that has been developed is flight simulator for a small scale helicopter and fixed wing. Georgia Institute of Technology [4] and [5] has developed a real-time flight simulator for a small scale helicopter. The flight simulator tools can provide real-time display of all flight data, including plotting, logging and modifying flight data. In reference [6], a realistic simulation based on MATLAB/Simulink with flight-gear for fixed wing UAV has been developed. The real-time simulator is equipped with hardware in the loop.

Control Systems and Robotics Lab, Diponegoro University, has developed a quadrotor equipped with gripper for aerial object interaction/aerial grasping, as shown in Fig. 1. This paper deals with development of real-time flight simulator for quadrotor to facilitate pilot training purpose.



Fig. 1. Quadrotor prototype equipped with gripper at Control Systems and Robotics Lab., Diponegoro University.

A 6-DOF nonlinear model of quadrotor dynamic including actuator dynamics, sensor gain from IMU (Inertial Measurement Unit), forces, and moments

caused by four propellers is developed in MATLAB/Simulink environments. The PD control system is utilized to control altitude rate and attitude at hover flight. PD control is proven capable of controlling the attitude of quadrotor as in [7]-[9].

II. QUADROTOR DYNAMICS AND CONTROL

A. Quadrotor Dynamics

Quadrotor consists of a rigid cross frame equipped with four brushless DC motors as shown in Fig. 2. The front and the rear propeller rotate in clockwise while the right and the left rotate in counter clockwise. Vertical motion is achieved by increasing or decreasing total thrust of quadrotor while maintaining an equal individual thrust $[U_i]$ for going up and going down respectively.

Roll motion is accomplished by increasing (or decreasing) the left propeller speed and by decreasing (or increasing) the right one $[U_2]$. It leads to torque which makes the quadrotor turn in the x_B axis. The total vertical thrust has to be the same with trust in hovering, such that this command only leads to roll angle acceleration and makes the quadrotor move right or left.

Pitch motion is accomplished by increasing (or decreasing) the front propeller speed and by decreasing (or increasing) the rear one $[U_3]$. It leads to torque which makes the quadrotor turn in the y_B axis. The total vertical thrust has to be the same with trust in hovering, such that this command only leads to pitch angle acceleration and makes the quadrotor move forward or backward.

Yaw motion is accomplished by increasing (or decreasing) the front and the rear propeller speed and by decreasing (or increasing) the left and the right propeller speed $[U_i]$. It leads to torque which makes the quadrotor turn in the z_B axis. The total vertical thrust has to be the same with trust in hovering, such that this command only leads to acceleration of yaw angle [5].

This work assumes that:

- The quadrotor structure is rigid and symmetrical.
- The center of mass and the center of geometry coincide.
- The propellers are rigid.
- Thrust is proportional to the square of the propellers speed.
- The four brushless DC motors have the same properties.

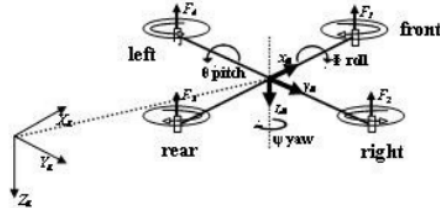


Fig. 2. Configuration of quadrotor.

Two frames as shown in Fig. 2 can be considered as earth inertial fixed frame (E-frame) and body-fixed frame (B-frame). Using the first Euler axioma and Newton's second law, the dynamics of a generic 6-DOF of rigid body can be written as follows.

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad (1)$$

Where m is the mass of the body, $I_{3 \times 3}$ is a 3 by 3 diagonal inertia matrix, \dot{V}^B is the linear acceleration vector with respect to the body fixed frame, $\dot{\omega}^B$ is the angular acceleration vector with respect to the earth inertial frame, F^B is the force vector from the quadrotor with respect to the body fixed frame and τ^B is the torque vector from quadrotor with respect to the body fixed frame.

Equation of motion of quadrotor is influenced by 3 contributors. The first contributor is the gravitational vector. The gyroscopic effects produced by propeller rotation are taken into account to be the second contributor, since two of them are rotating clockwise and the other two are rotating counter clockwise. The third contributor takes account of the forces and torques directly produced by the main movement inputs, u_i where $i=1$ to 4. Equation (2) represents the quadrotor dynamics respect to the body fixed coordinate of quadrotor.

$$\begin{aligned} \dot{u} &= (vr - wq) + g \sin \theta \\ \dot{v} &= (wp - ur) - g \cos \theta \sin \theta \\ \dot{w} &= (uq - vp) - g \cos \theta \cos \theta + \frac{U_1}{m} \\ \dot{p} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega + \frac{U_2}{I_{XX}} \\ \dot{q} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega + \frac{U_3}{I_{YY}} \\ \dot{r} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \end{aligned} \quad (2)$$

Where, J_{TP} is the total rotational moment of inertia around the propeller axis.

The control of quadrotor in earth inertial fixed frame can be determined using (3),

$$\begin{aligned} \ddot{X} &= (\sin \psi \sin \theta + \cos \psi \sin \theta \cos \theta) \frac{U_1}{m} \\ \ddot{Y} &= (-\cos \psi \sin \theta + \sin \psi \sin \theta \cos \theta) \frac{U_1}{m} \\ \ddot{Z} &= -g + (\cos \theta \cos \theta) \frac{U_1}{m} \\ \dot{p} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega + \frac{U_2}{I_{XX}} \\ \dot{q} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega + \frac{U_3}{I_{YY}} \\ \dot{r} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \end{aligned} \quad (3)$$

While equations of angular velocity as input of propeller are given in (4):

$$\begin{aligned} U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= lb(-\Omega_2^2 + \Omega_4^2) \\ U_3 &= lb(-\Omega_1^2 + \Omega_3^2) \\ U_4 &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{aligned} \quad (4)$$

Detailed development of propeller thrust, forces and moments can be found in [8].

B. PD Control

PD control design is used to stabilize the quadrotor at hover flight condition, as shown in Fig. 3. PD control is used for four subsystems; roll, pitch, yaw and throttle for vertical rate.

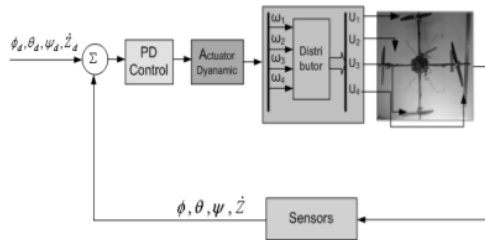


Fig. 3. Basic structure of PD control in real-time flight simulator for quadrotor.

1) Roll control of quadrotor is used to stabilize the roll angle (ϕ) of quadrotor. Fig. 4 shows the block diagram of the PD control for roll angle.

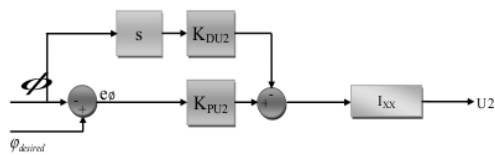


Fig. 4. Block diagram of roll control.

2) Pitch control of quadrotor is used to stabilize the pitch angle (θ) of quadrotor. Fig. 5 shows the block diagram of the PD control for pitch angle.

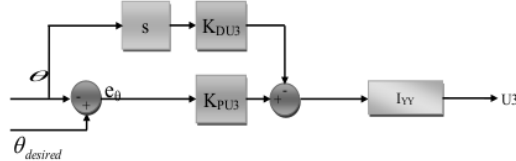


Fig.5. Block diagram of pitch control.

3) Yaw control of quadrotor is used to stabilize the yaw angle (ψ) of quadrotor. Fig. 6 shows the block diagram of the PD control for yaw angle.

4) Throttle control of quadrotor is used to control the linear vertical velocity (\dot{z}). Fig. 7 shows the block diagram of the PD control for throttle.

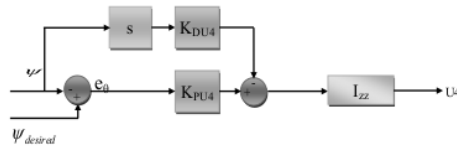


Fig. 6. Block diagram of yaw control.

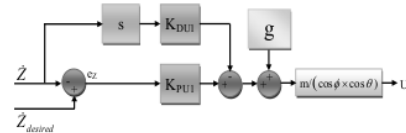


Fig.7. Block diagram of throttle control.

Ziegler-Nichols method was used to obtain the best PD gains, K_P and K_D . The resulted gains are

$$\begin{aligned} K_{PU1} &= 6; K_{DU1} = 0.375 \\ K_{PU2} &= 12; K_{DU2} = 2.177 \\ K_{PU3} &= 12; K_{DU3} = 2.177 \\ K_{PU4} &= 1.8; K_{DU4} = 0.844 \end{aligned}$$

III. HARDWARE AND SET-UP

In this research, xPC Target with MATLAB is used to make quadrotor flight simulator run in real time. It is important, because the real behavior of the model can be understood from this simulation prior to applying it to the hardware. Whether the model behaves well or not can be checked and if there is something wrong, it can be immediately corrected.

This research uses three computers in which one of them is used as the host computer, another as the target computer, and the rest as the support computer which is used to display plot and provide sound of simulation. Set-up of the hardware is shown in Fig. 8. Specifications of those computers are as follows:

- Host Computer: Intel® Core™2 Duo CPU, E4600@2.40 GHz, 2.39 GHz, 2.00 GB of RAM.
- Target Computer: Intel®, Pentium® 4 CPU 2.93 GHz, 2.93 GHz, 0.99 GB of RAM.
- Support Computer: Intel®, Pentium® 4 CPU 2.93 GHz, 2.93 GHz, 0.99 GB of RAM.

However, delay is still found in the real-time simulation because of limitation of computer facility. The delay happens at 3D visualization which uses Virtual Reality. In this research, LAN / Ethernet connection is used to link one computer to the others. This connection is chosen because, according to MATLAB Help, LAN connection is able to transfer data faster than the other connection, Serial RS-232.

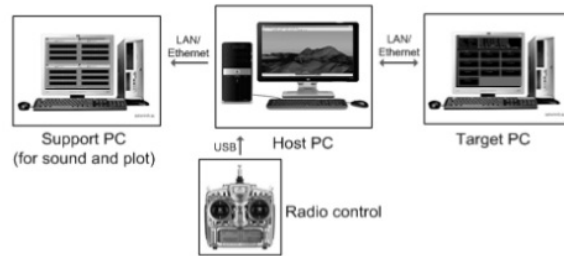


Fig. 8. Hardware set-up.

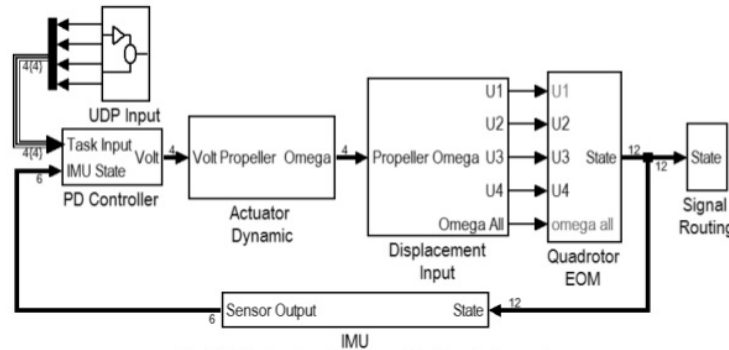


Fig. 9. Plant and controller model of quadrotor system.

IV. REAL-TIME FLIGHT SIMULATOR

The quadrotor model described in section II is separated into four parts in the implementation of real-time simulator using xPC Target. The four separated models are shown in Fig. 9 - 12. The plant and controller model is shown in Fig. 9 whereas the input model which is run at host computer is shown in Fig. 10. The input data from joystick hardware is sent to the plant and controller model which is run at target computer such that simulation can be done in real time. The detailed model of actuator dynamic can be found in [8]. Then output from plant and controller model is sent to virtual reality model and scope model which are run at host computer and support computer respectively. Virtual reality model in Fig. 11 enables 3D visualization of the simulation which is shown in Fig. 12. The scope model in Fig. 13 shows the numerical values of the simulation such as position and velocity.

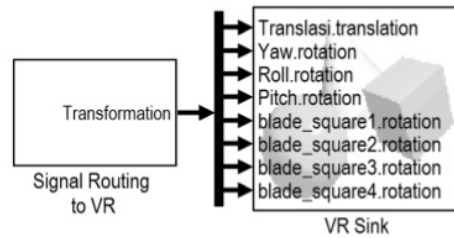


Fig. 11. Virtual reality model of quadrotor system.

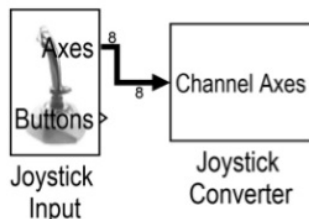


Fig. 10. Input model of quadrotor system.



Fig. 12. 3D visualization of the simulation.

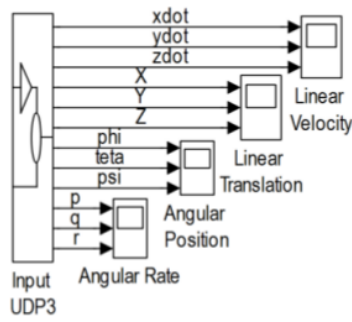


Fig. 13. Scope model of quadrotor system.

After building the real-time simulator, verification for the simulation model was performed by testing three flight directions: longitudinal, lateral and vertical. These tests observe the velocity and acceleration in each direction and also analyze test force and its joystick angle in inertial fixed frame which are shown in Fig. 14, 15 and 16 for testing in x axis, y axis and z axis respectively. It can be seen from the longitudinal and lateral flight test that the test forces are directly proportional to joystick angle, whereas in vertical flight test, it is not. In vertical flight test, the test force always goes back to zero even if the input from external hardware is non-zero value. It happens because the model is a closed-loop system which has a compensator that makes the response of vertical direction in quadrotor to be zero.

Results from these tests are velocity and acceleration in each direction which are shown in Fig. 17, 18, and 19 for flight rate test in longitudinal, lateral, and vertical direction respectively. It can be seen from Fig. 17 and 18 that the velocity will increase proportionally to the given test force. At the end of the test, both the test forces are made zero, but the velocities end in around 380 m/s for longitudinal test and 500 m/s for lateral test. It happens because the model does not consider wind force which resists movement of quadrotor. Unlike these result, vertical flight test gives different result as shown in Fig. 19, i.e. velocity of quadrotor in z axis ends in zero. It happens because the model considers force of gravity which makes the climbing rate of the model become zero when test force is made zero.

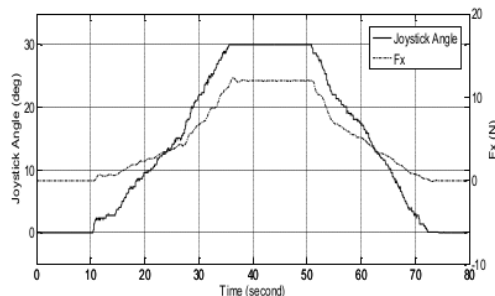


Fig. 14. Test force in x axis (F_x) and its joystick angle in inertial fixed frame.

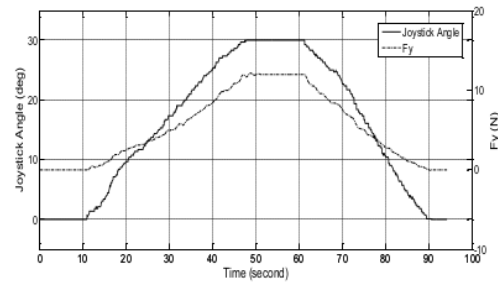


Fig. 15. Test force in y axis (F_y) and its joystick angle in inertial fixed frame.

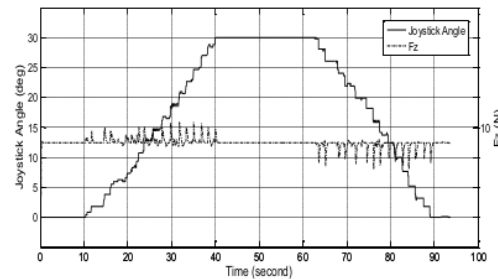


Fig. 16. Test force in z axis (F_z) and its joystick angle in inertial fixed frame.

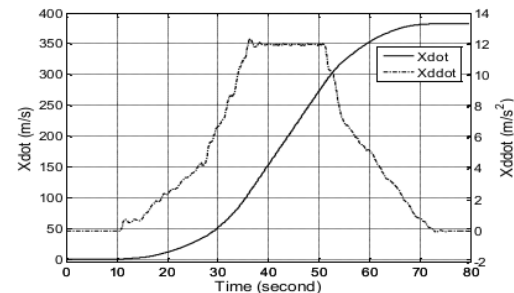


Fig. 17. Longitudinal velocity and longitudinal acceleration of quadrotor in inertial fixed frame.

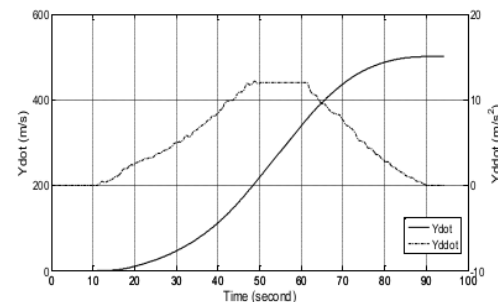


Fig. 18. Lateral velocity and lateral acceleration of quadrotor in inertial fixed frame.

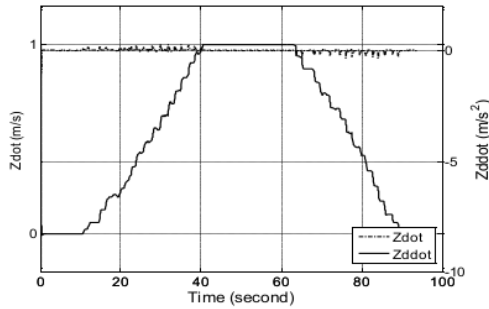


Fig. 19. Vertical velocity and vertical acceleration of quadrotor in inertial fixed frame.

Verification to check whether the simulation runs in real time or not was conducted by evaluating the percentage of processing usage of the computer as function of simulation time. If the processing is not more than 100%, the simulation runs in real time. This method has been used by Dantas [10]. The percentage of processing usage of the computer (CPU_{Time}) can be found by using the equation:

$$CPU_{Time} = [1 - (T_{TS} - T_{TET}(t)/T_{TS})] \times 100 \% \quad (5)$$

where " T_{TS} " is the sampling period of the simulation software and " $T_{TET}(t)$ " is the Task Execution Time (TET) obtained in a determined time " t ". This simulation used 0.01 second as the sampling period and the result of verification is shown in Fig. 20.

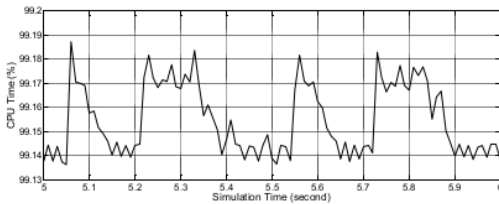


Fig. 20. Simulator computer percentage of use (process time).

It can be observed that the processing is well below 100%, thereby it can be stated that the simulation runs in real time.

V. CONCLUSION

This work presented a methodology for developing a real-time flight simulator for quadrotor system. By evaluating the dynamic of the quadrotor in three flight directions, it can be seen that the PD controller make the closed system stable.

It was also confirmed that simulator run in real time. The simulator enables engineers and pilots to predict behavior of model without conducting practical test. It is very beneficial to reduce cost and time in design process. For future study, Hardware-In-the-Loop (HIL) simulation will be used in this research to verify real-time performance of quadrotor system hardware.

ACKNOWLEDGMENT

Gratitude is given to Dr. Eng. Munadi as the research team member.

REFERENCES

- [1] Mellinger, D., Q. Lindsey, M. Shomin and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *proceeding of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2668-2673.
- [2] Paul E. I. Pounds, Daniel R. Bersak, and Aaron M. Dollar, "Grasping from the air: hovering capture and load stability," in *proceeding of IEEE International Conference on Robotics and Automation*, pp. 2491-2498, 2011.
- [3] Idris E. Putro, G. B. Kim, Gilar B. Raharja, and K. J. Yoon, "Real-time simulation of autonomous quadrotor," in *proceedings of International Micro Air Vehicle Conference and Flight Competition*, IMAV2010, Braunschweig, Germany, July 6-9, 2010.
- [4] Christian Munzinger, "Development of a real-time flight simulator for an experimental model helicopter," Diploma Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, United States of America, 1998.
- [5] Eric N. Johnson and Sumit Mishra, "Flight simulation for the development of an experimental UAV," in *proceedings of AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2002.
- [6] Dongwon Jung and Panagiotis Tsiotras, "Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle," American Institute of Aeronautics and Astronautics, 2007.
- [7] W. Yiting, "Development and implementation of a control system for a quadrotor UAV," Master Thesis, Department of Electrical and Computer Engineering, University of Applied Science Ravensburg-Weingarten, Germany, 2009.
- [8] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," Master Thesis, Department of Automatic Control, Lund University, Sweden, 2008.
- [9] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. Dissertation, Ecole Polytechnique Federal Lausanne, Switzerland, 2007.
- [10] Dantas, J. L. D. and de Barros, E. A., "A real-time simulator for AUV development," *ABCMS Symposium Series in Mechatronics*, vol.4, pp. 538 – 549, 2010.

Development of Real-Time Flight Simulator for Quadrotor

ORIGINALITY REPORT

3%

SIMILARITY INDEX

%

INTERNET SOURCES

3%

PUBLICATIONS

1%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

< 1%

★ International Journal of Intelligent Unmanned
Systems, Volume 4, Issue 3 (2016)

Publication

Exclude quotes On

Exclude matches Off

Exclude bibliography On