City Research Online

City, University of London Institutional Repository

Citation: Ruiz, J. A. D. and Aouf, N. ORCID: 0000-0001-9291-4077 (2017). Real-Time Setup with PD and Backstepping Control for a Pelican Quadrotor. In: Gusikhin, O. and Madani, K. (Eds.), Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics. (pp. 676-681). SciTePress. ISBN 978-989-758-263-9

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: http://openaccess.city.ac.uk/22089/

Link to published version:

Copyright and reuse: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

City Research Online:	http://openaccess.city.ac.uk/	publications@city.ac.uk

Real-Time Setup with PD and Backstepping Control for a Pelican Quadrotor

Jose Alejandro Dena Ruiz[1], Nabil Aouf[2]
[1,2]*Centre for Electronic Warfare, Information and Cyber, Defence Academy of the United Kingdom*
Cranfield University, Shrivenham, SN6 8LA United Kingdom
{*j.a.dena-ruiz, n.aouf*}*@cranfield.ac.uk*

Keywords: PD control, Backstepping control, Kalman Filter, Optitrack, ROS.

Abstract: In this paper, a real-time setup and an implementation of a Proportional Derivative (PD) controller for orientation and comparison between PD and BackStepping (BS) controllers for linear positioning are presented using a Pelican quadrotor from Ascending Technologies (AscTec). An onboard Inertial Measurement Unit (IMU) was used for orientation control and Optitrack Vision Tracking System for linear positioning control. A linear Kalman filter was implemented for linear velocity estimation. The software and hardware integration was achieved with the help of the Robot Operating System (ROS). Simulations and experiments with this drone platform are achieved in order to implement different controller algorithms and analyse them in order to achieve better aircraft performance.

1 INTRODUCTION

The studies in quadrotor control designing have been increasing rapidly in recent years. Linear controllers design for quadrotors have been achieved in several work, like the Linear Quadratic Regulator(LQR) and a Proportional Integral Derivative (PID) (Khatoon et al., 2014) (Reyes-Valeria et al., 2013). Nonlinear control design has also been achieved with different techniques, like Backstepping (Das et al., 2009), Sliding Mode (Runcharoon and Srichatrapimuk, 2013) and Feedback Linearisation (Saif, 2009). (Castillo et al., 2005) compared the performance of a nonlinear control algorithm with a LQR control law. Results show the unstable response of a linear controller applied to a nonlinear system, while the nonlinear controller shows stable response. (Gomes et al., 2016) used an AR.Drone quadrotor and a Vicon motion capture system to track a moving target with a Proportional Derivative (PD) controller for linear positioning. (Mashood et al., 2016) showed experimental results of two AR.Drone following a squared path using VICON system and MATLAB/SIMULINK for feedback and control implementation. This was possible with AR Drone Simulink Development Kit (ARDSDK). (Campbell et al., 2012) showed the design and implementation of a quadrotor aircraft autopilot, allowing the UAV to take off, transit from one location to another and land at a desired location. The position data is obtained from an Optitrack system and a PID control technique was used to achieve the desired response for (X,Y,ϕ,θ) and PD controller for (Z,ψ), the integration of the motion system and the controllers were developed on MATLAB/SIMULINK.

An improved PID controller was implemented by (Zheng et al., 2016) showing the comparison against the traditional PID and Backstepping (BS). Simulation and experimental results show stability and tracking performance, using a motion capture system and MATLAB/SIMULINK as the feedback measurement and the platform for control implementation respectively.

(Corona-Sanchez and Rodriguez-Cortes, 2013) presented outdoor and indoor experimental validation of a nonlinear controller for a quadrotor vehicle. Also showed a real-time programming strategy, parameter identification technique and a nonlinear gain tuning procedure.

A lot of related work has been done so far. Nevertheless the novelty of this paper is to show the use of our particular drone platform along with the Robot Operating System (ROS) and the Optitrack system for real time experimental scenarios. The result of these powerful components is to obtain an stabilised drone, in one hand running an orientation PD controller at 1000 Hz in the embedded processor and in the other hand for position control, a linear PD or a nonlinear

Backsepping controllers are presented to demonstrate the efficiency of ROS and optitrack at 120 Hz. This way the setup remains open for the development and implementation of a new control techniques.

2 THE QUADROTOR DYNAMIC MODEL

The model is derived based on some assumptions, in order to simplify the dynamics of the complex system to be suitable for simulation and control design, Figure 1. A symmetric and rigid structure, rigid propellers and ground effect has been ignored. The full Model derivation can be found on (Bouabdallah and Siegwart, 2005).

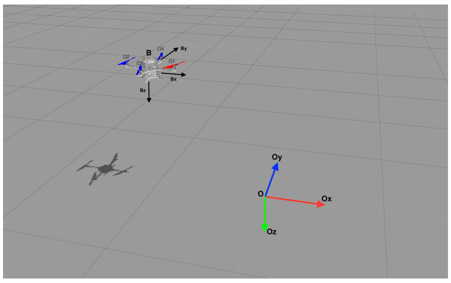

Figure 1: Quadrotor AscTec Pelican and inertial frame on Gazebo Simulator.

Using Newton's laws of Mechanics and Euler's Dynamics equation, the model consists of six equations for the system dynamics and four equations describing the inputs of the system:

$$\ddot{X} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\frac{1}{m}U_1 \tag{1}$$

$$\ddot{Y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\frac{1}{m}U_1 \tag{2}$$

$$\ddot{Z} = (\cos\phi \cos\theta)\frac{1}{m}U_1 - g \tag{3}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}(\frac{I_{yy} - I_{zz}}{I_{xx}}) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega + \frac{U_2}{I_{xx}} \tag{4}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}(\frac{I_{zz} - I_{xx}}{I_{yy}}) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega + \frac{U_3}{I_{yy}} \tag{5}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}(\frac{I_{xx} - I_{yy}}{I_{zz}}) + \frac{U_4}{I_{zz}} \tag{6}$$

where equation 1, 2 and 3 describe the linear acceleration of the vehicle in the direction of O_x, O_y and O_z axes using the North, East, Down (N, E, D) convention respectively, while equations 4, 5 and 6 represent the angular accelerations of the vehicle about the same axes respectively. l corresponds to the arm length holding the propeller, ϕ, θ and ψ, represent the Euler angles about the body axes B_x, B_y and B_z respectively. I_{xx}, I_{yy} and I_{zz}, are the inertial components about the x-axis, y-axis and z-axis respectively. $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the translational velocity components along the main axes. $U_i \equiv 1,2,3,4$ represents the system inputs.

$$U_1 = \sum T_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{7}$$

$$U_2 = (-T_2 + T_4) = bl(-\Omega_2^2 + \Omega_4^2) \tag{8}$$

$$U_3 = (T_1 - T_3) = bl(\Omega_1^2 - \Omega_3^2) \tag{9}$$

$$U_4 = (-1)^i \sum M_{D_i} = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{10}$$

$$\Omega = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \tag{11}$$

where equation 7 represents the total thrust, 8 the pitch moment, 9 the roll moment and 10 perform the yaw moment. b is the thrust factor in hover and d is the drag factor in hover. Increasing or decreasing of the speed of the four propellers together will be responsible for the altitude (z-axis) change in position and velocity while varying the speed of one pair of propellers (Ω_3 and Ω_1) will cause the aircraft to tilt about the y-axis which is denoted as pitch angle θ. Similarly varying the speed of the propellers pair (Ω_4 and Ω_2) will cause the aircraft to tilt about x-axis which is denoted as roll angle ϕ. Finally the vector summation of the reaction moment produced by the rotation of the pair (Ω_3 and Ω_1) and the reaction moment produced by the rotation of the pair (Ω_4 and Ω_2) will cause the quadrotor to spin about its axis (z-axis), which is denoted as yaw angle ψ.

3 QUADROTOR CONTROLLERS

In this section, a PD and Backstepping controller were implemented for linear positioning, while only PD controllers for orientation.

A set of PD controllers have been chosen for the UAV attitude based on the good response in simulations and experiments, fast response and less than five percentage error in steady state. The Euler angles are used as feedback signals to the proposed controller in order to achieve the desired orientation. Traditional implementation can be defined as:

$$U = K_p e(t) + K_d \frac{de}{dt} \tag{12}$$

Where K_p is the proportional gain, K_d the derivative gain, $e(t)$ is the difference between the current controlled value and the desired value. In our case, three PD individual controllers are implemented:

$$U_2 = K_{p\phi}(d_\phi - \phi) + K_{d\phi}(d_{\dot\phi} - \dot\phi) \tag{13}$$

$$U_3 = K_{p\theta}(d_\theta - \theta) + K_{d\theta}(d_{\dot\theta} - \dot\theta) \tag{14}$$

$$U_4 = K_{p\psi}(d_\psi - \psi) + K_{d\psi}(d_{\dot\psi} - \dot\psi) \tag{15}$$

Where 13, 14 and 15 represent the torques for each axis due to the rotor's thrust, τ_ϕ, τ_θ and τ_ψ respectively. $K_{p\phi}$, $K_{p\theta}$ and $K_{p\psi}$ are the proportional gains for each controller and $K_{d\phi}$, $K_{d\theta}$ and $K_{d\psi}$ the derivative ones. d_ϕ, d_θ and d_ψ are the desired *pitch*, *roll* and *yaw*, ϕ, θ and ψ correspond to the current attitude and position values.

Considering that desired position d_x, d_y, d_z are known and x, y, z can be measured, the equations 1, 2 and 3 can be solved in order to determine the control variables:

$$U_1 = \frac{1}{\cos\phi\cos\theta}(r_3 + mg) \tag{16}$$

$$d_\phi = \arcsin(\frac{m}{U_1}(r_1 \sin\psi - r_2 \cos\psi)) \tag{17}$$

$$d_\theta = \arcsin(\frac{m}{U_1}(r_1 \cos\psi + r_2 \sin\psi)) \tag{18}$$

Where 16 is the force to control the vertical displacement z, 17 and 18 are the desired angles to position the drone at the desired x and y on the cartesian plane respectively. r_1 and r_2 represent the output control signal of the desired control technique that will be mapped to the desired angles, g is the acceleration due to the gravity force and m is the aircraft mass.

Equations 19-21 represent the PD controllers for the linear position of the UAV.

$$r_1 = K_{px}(d_x - x) + K_{dx}(d_{\dot x} - \dot x) \tag{19}$$

$$r_2 = K_{py}(d_y - y) + K_{dy}(d_{\dot y} - \dot y) \tag{20}$$

$$r_3 = K_{pz}(d_z - z) + K_{dz}(d_{\dot z} - \dot z) \tag{21}$$

Using the Backstepping approach, one can synthesise the control law forcing the system to follow the desired trajectory. Writing the model equations for linear position in space-state:

$$x_1 = x \tag{22}$$

$$x_2 = \dot{x_1} = \dot x \tag{23}$$

$$x_3 = y \tag{24}$$

$$x_4 = \dot{x_3} = \dot y \tag{25}$$

$$x_5 = z \tag{26}$$

$$x_6 = \dot{x_5} = \dot z \tag{27}$$

First, we consider the tracking-error for x_1 position:

$$z_1 = x_{1d} - x_1 \tag{28}$$

then, by considering the lyapunov function z_1 positive definite and it's time derivative negative semidefinite:

$$V(z_1) = \frac{1}{2}z_1^2 \tag{29}$$

$$\dot V(z_1) = z_1(\dot{x}_{1d} - \dot{x_2}) \tag{30}$$

The stabilisation of z_1 can be obtained by introducing the virtual control input x_2:

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \tag{31}$$

$$with: \alpha_1 > 0 \tag{32}$$

$$\therefore \dot V(z_1) = -\alpha_1 z_1^2 \tag{33}$$

Now, we have to make $x_2 = \dot{x}_{1d} + \alpha_1 z_1$ creating the variable z_2:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{34}$$

For the second step, we consider the augmented Lyapunov function:

$$V(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2) \tag{35}$$

$$\tag{36}$$

and it's time derivative is:

$$\dot V(z_1 z_2) = z_2 r_1 - z_2(\ddot{X}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2 \tag{37}$$

Extracting r_1 and making $\ddot{X}_1 d = 0$ is satisfied that $\dot V(z_1 z_2) < 0$, now adding the term $\alpha_2 z_2$ with $\alpha_2 > 0$ to stabilise z_1 and we apply the same procedure for the y and z axis:

$$r_1 = z_1 - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2 \tag{38}$$

$$r_2 = z_3 - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4 \tag{39}$$

$$r_3 = z_5 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6 \tag{40}$$

with:

$$z_3 = x_{3d} - x_3 \tag{41}$$

$$z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3 \tag{42}$$

$$z_5 = x_{5d} - x_5 \tag{43}$$

$$z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5 \tag{44}$$

4 THE PELICAN PLATFORM

The Pelican quadrotor from AscTec was chosen as the prototype to develop the experiments, it's physical architecture made of carbon fibre and well distributed sensors and components make it a reliable platform for research purposes. This Quadrotor offers plenty of space and various interfaces for individual components and payloads. The total quadrotor weight including sensors and battery is 1.63 Kg. Figure 2.

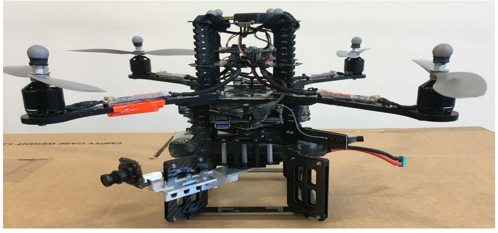

Figure 2: Quadrotor AscTec Pelican with Autopilot and Mastermind.

The basic electronic components are the AscTec Autopilot, Motor controllers and a single board computer, the Mastermind. The AscTec Autopilot also known as Flight-Control-Unit (FCU) contains the High Level Processor and the Low Level Processor (HLP and LLP) running at 1 Khz which are in charge of the aircraft control. The overall architecture is presented on Figure 3, the LLP handles sensor data processing, data fusion as well as an stable attitude control algorithm. The HLP is open for user purposes, to implement control algorithms, sensor fusion, etc.

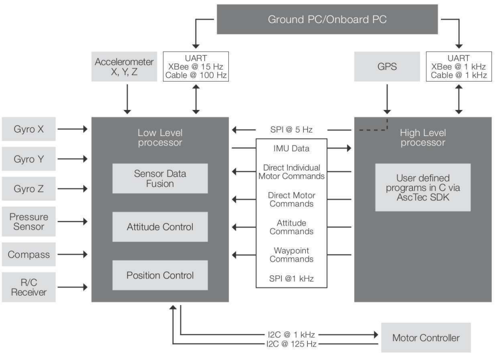
Ground PC/Onboard PC
Accelerometer X, Y, Z
UART XBee @ 15 Hz Cable @ 100 Hz
GPS
UART XBee @ 1 kHz Cable @ 1 kHz
Gyro X
Gyro Y
Gyro Z
Pressure Sensor
Compass
R/C Receiver
Low Level processor
Sensor Data Fusion
Attitude Control
Position Control
SPI @ 5 Hz
IMU Data
Direct Individual Motor Commands
Direct Motor Commands
Attitude Commands
Waypoint Commands
SPI @1 kHz
High Level processor
User defined programs in C via AscTec SDK
I2C @ 1 kHz
I2C @ 125 Hz
Motor Controller

Figure 3: Autopilot electronic architecture.

The AscTec Mastermind is an onboard processor board, which can be carried by the AscTec Pelican, it can be used like a ground PC, but it is mounted on the flight system. In comparison to its weight and size, it offers an extremely high processing power, high data rates and a great variety of standard PC interfaces to connect several kind of hardware devices. Features like a Dual Core Atom, a Core 2 Duo, or a Core i7, Firewire, WIFI and hardware serial ports are supported.

5 THE QUADROTOR CONTROL SYSTEM

The complete Quadrotor control system is presented on figure 4. The full system is composed by the Pelican from AscTec, which is the platform to control. The PD orientation controllers obtained on equations 13, 14 and 15 were implemented in the HLP on Pelican, these controllers are using the angles provided by the LLP as the feedback measurements. In the AscTec Mastermind, a ROS application is interacting with the HLP through the serial port, this application is sending the new desired angles and receiving the current states of the control variables and gives the user the possibility of a real-time gain tuning. Figure 4 shows the full control system overview.

Once the orientation controller is working, a second ROS application is running parallel on the AscTec Mastermind, this application is performing the linear position control algorithms for the Drone, allowing the user to choose between the PD or BS controllers obtained in section 3, create trajectories and test new different control techniques. An Optitrack Motion Capture System was used to retrieve the actual Drone position with a precision up to 2 mm. A setup of 6 cameras connected to a 12-port POE switch along with a host computer, Optitrack Motive application runs and streams the current position of a rigid body at 120 Hz to the AscTec Mastermind through a wifi router. This second application also allows the user to select between a PD or Backstepping controller for linear position as well as real-time gain tuning.

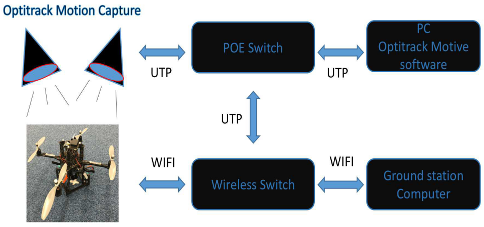
Optitrack Motion Capture
UTP
POE Switch
UTP
PC Optitrack Motive software
UTP
WIFI
Wireless Switch
WIFI
Ground station Computer

Figure 4: Quadrotor Control System overview.

6 EXPERIMENTAL RESULTS

The real experiments were done using the AscTec Pelican as a UAV, for orientation control angles and angular velocities were obtained from the onboard autopilot IMU, whereas to retrieve the X, Y and Z position the optitrack system was used. A ROS package was used to publish into ROS network the position of the rigid body and the use of a linear Kalman filter for velocity estimation. A two metres squared trajectory was chosen as the drone trajectory and altitude of one metre.

Velocity estimation was achieved using the linear Kalman filter based on (Kim, 2010). Optitrack system provides a high accuracy on ground truth at 120 Hz (dt=0.0083), taking this parameters into account and the time-varying for the process noise matrix Q, the values of the matrices resulted as follow:

$$A = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{45}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{46}$$

$$R = \begin{bmatrix} 4X10^{-6} & 0 & 0 \\ 0 & 4X10^{-6} & 0 \\ 0 & 0 & 4X10^{-6} \end{bmatrix} \tag{47}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & dt^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & dt^2 \end{bmatrix} \tag{48}$$

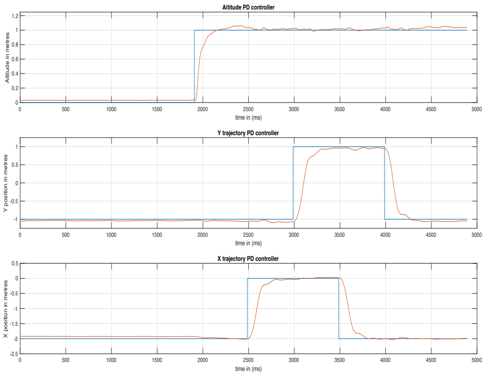

Figure 5: Real XYZ displacement with a PD controllers

Figures 5-7 show the results in real-time using the PD controllers and Figures 8-10 show the results for the BS controllers.

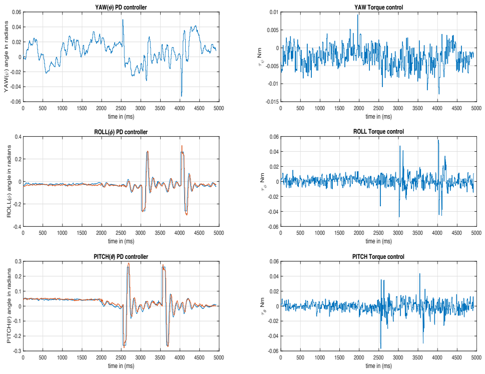

Figure 6: Real Torques and desired angles ROLL, PITCH and YAW with a PD controllers

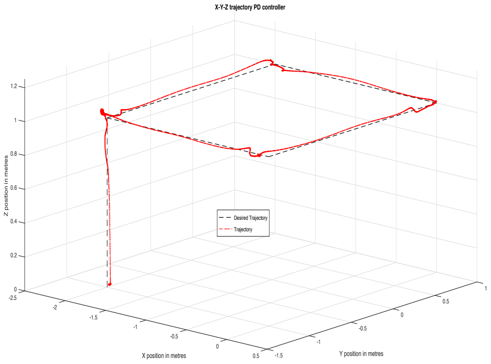

Figure 7: Real 3D trajectories using a PD controllers

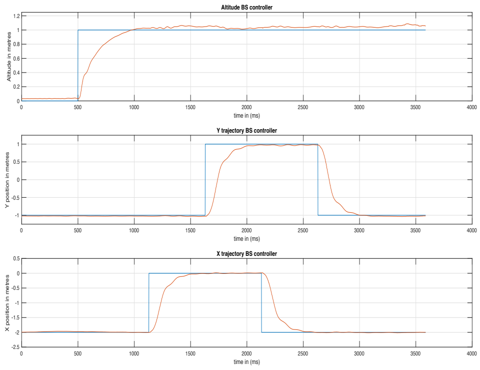

Figure 8: Real XYZ displacement with a BS controllers

7 CONCLUSIONS

This paper presents the development of a quadrotor control system for orientation and position control, using the onboard IMU and Optitrack system to feedback current signals. On one hand the implementation of the algorithms for orientation control was possible using the HLP from the autopilot, the program was developed on C and the execution rate was 1 kHz. On the other hand for positioning control the control al-

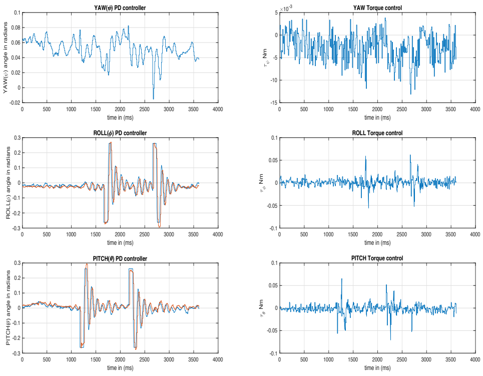

Figure 9: Real Torques and desired angles ROLL, PITCH and YAW with a BS controllers

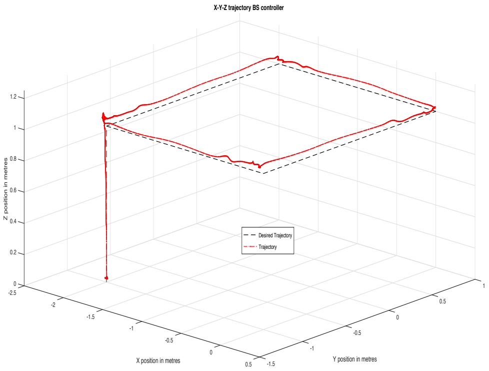

Figure 10: Real 3D trajectories using a BS controllers

gorithms were developed under C++ and ROS, which allow us to create a network where the ground station, robots and sensors can exchange information at 120 Hz. The Drone performance was tested using an square trajectory with 2 metres side, which was enough to move the desired roll and pitch angles and see their control response with the PD controller. The desired angles was reached within a short period of time, for instance 0.25 rad in 100 ms on the experimental results for roll angle as figure 6 shows. Regarding positioning control, a linear PD controller and a non linear Backstepping controller were tested, getting a better response for X and Y position with the Backstepping controller in figure 8 than the PD in figure 5, whereas for the altitude Z the PD controller had a quicker response compared with BS as shown in figures 7 and 10.

REFERENCES

Bouabdallah, S. and Siegwart, R. (2005). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *Proceedings - IEEE International Conference on Robotics and Automation*, 2005(April):2247–2252.

Campbell, J., Hamilton, J., Iskandarani, M., and Givigi, S. (2012). A systems approach for the development of a quadrotor aircraft. In *SysCon 2012 - 2012 IEEE International Systems Conference, Proceedings*, pages 110–116.

Castillo, P., Lozano, R., and Dzul, A. (2005). Stabilization of a mini rotorcraft with four rotors: Experimental implementation of linear and nonlinear control laws. *IEEE Control Systems Magazine*, 25(6):45–55.

Corona-Sanchez, J. J. and Rodriguez-Cortes, H. (2013). Experimental real-time validation of an attitude nonlinear controller for the quadrotor vehicle. In *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*, pages 453–460.

Das, A., Lewis, F., and Subbarao, K. (2009). Backstepping approach for controlling a quadrotor using lagrange form dynamics. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 56(1-2):127–151.

Gomes, L. L., Leal, L., and Oliveira (2016). Unmanned quadcopter control using a motion capture system. *IEEE Latin America Transactions*, 14(8):3606–3613.

Khatoon, S., Gupta, D., and Das, L. K. (2014). Pid & lqr control for a quadrotor: Modeling and simulation. *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, pages 796–802.

Kim, P. (2010). *Kalman Filter for Beginners with MATLAB Examples*. A-JIN, Republic of Korea, 1 edition.

Mashood, A., Mohammed, M., Abdulwahab, M., Abdulwahab, S., and Noura, H. (2016). A hardware setup for formation flight of uavs using motion tracking system. In *ISMA 2015 - 10th International Symposium on Mechatronics and its Applications*.

Reyes-Valeria, E., Enriquez-Caldera, R., Camacho-Lara, S., and Guichard, J. (2013). Lqr control for a quadrotor using unit quaternions: Modeling and simulation bt - electronics, communications and computing (conielecomp), 2013 international conference on. pages 172–178.

Runcharoon, K. and Srichatrapimuk, V. (2013). Sliding mode control of quadrotor. *2013 International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, (1):552–557.

Saif, A. A. H. (2009). Quadrotor control using feedback linearization with dynamic extension. *2009 6th International Symposium on Mechatronics and its Applications, ISMA 2009*, (1):25–27.

Zheng, H., Zeng, Q., Chen, W., Zhu, H., and Chen, C. (2016). Improved pid control algorithm for quadrotor based on mcs. In *Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference*, number 1, pages 1780–1783, Nanjin, China.