2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Low Cost X4 Platform to Study Control Algorithms

D. Lara [1], G. Romero [1], D. Flores[1], L. Ramirez [1], A. Zuñiga [1], C. Pegard [2], R. Abdelhamid[2] and E. Alcorta [3]

***Abstract*— In this paper, we present the design of an autopilot embedded system applied to control a four rotor mini helicopter, called X4-flyer. The main goal is to achieve a stable flight, controlling the vehicle attitude; this is done using two control strategies: a linear PD control and nonlinear nested saturations control. The onboard system uses a simplified architecture and the main processor has a structured program to handle the data acquisition, and calculates the control to send actuators correction to get a desired set point. Real time experiments, in a pivoted platform, show that the autopilot is a reliable low cost platform.**

I. INTRODUCTION

The design and development of Unmanned Aerial Vehicles (UAV) began during the First and Second World Wars, and were used in recognition missions. Nowadays, UAV is defined as an aerial robot which is capable of flying autonomously or semi-autonomously with similar capabilities of a conventional aircraft [1]. These robots have inside an autopilot system to control the flight and establish a bidirectional link with a ground station; in this manner the pilot in land can take decisions about the course and objectives of the flight. The interest on using UAVs for different applications either for military (surveillance) or civil purposes has been increased [2], but the UAV's control problem becomes more complex, each time, due to theirs applications. Then many research groups in universities have intensified their activity in this field, taking the challenge to develop control strategies, and test platforms, which in many cases are expensive. Then, in this paper we describe the main aspects to design and implement a low cost four rotor rotorcraft UAV (X4) test platform. We claim that this platform is low cost in the sense that with cheaper components it is possible to build it, as compared with professional UAV development systems that are too expensive for educational budget. The X4 helicopter is a vertical takeoff and landing vehicle (VTOL) able to move in the 3D space with the ability to fly in a stationary way [3]. It has four electric motors (see figure 1), where each one is attached to a rigid cross frame. The front and rear rotors rotate counter-clockwise while the left and right rotors rotate clockwise, to cancel the gyroscopic effects and aerodynamic torques in stationary trimmed flight. Vertical motion is controlled by the sum of the thrusts of each motor. Longitudinal/ Lateral motion is achieved by controlling the differential speed between (front and rear)/(right and left) motors, tilting the X4-Flyer around the (y-axis)/(x-axis) to generate the pitch/roll angle. The yaw angle is obtained having two sets of rotors rotating in opposite direction, increasing or decreasing the speed of the front and rear motors while decreasing or increasing the speed of the lateral motors, keeping the total thrust constant, unchanging the altitude.

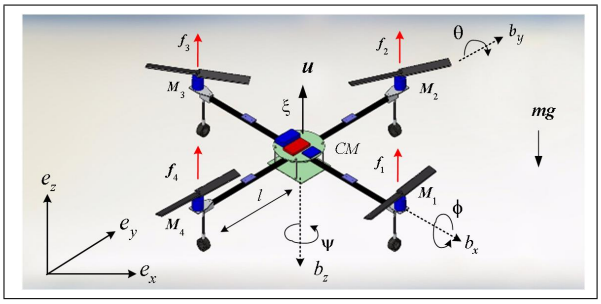

Fig. 1. Forces and moments schema for the four-rotor rotorcraft.

Many research works have studied this type of vehicles; for example, in [4] a dynamical model that incorporates the airframe, motor's dynamics, aerodynamics and gyroscopic effects is presented and a control law based on the Lyapunov approach is proposed. In [5] backsteeping and sliding mode techniques are applied to an indoor micro quadrotor, with results validated in a PC-based bench test platform. The attitude stabilization problem is treated in [6] using a quaternion based model and a Lyapunov function for a PD controller, the results were presented both in simulation as experimental using an embedded control system. A nested saturations control technique can be found in [3], such technique was implemented in a PC based platform with a satisfactory performance. In [7], the authors proposed an observer for estimate the attitude and the horizontal velocity of a VTOL aircraft. Although there are many publications dealing with control techniques, a few of publications treat with the design of a low cost and simplified UAV architecture. Therefore, we present a methodology to embed on board an efficient hardware and software system, including suitable sensors, to build a didactic platform that allows to validate control algorithms. This paper is organized as follows: the architecture used to stabilize the X4-Flyer is given in section 2. In section 3, we present the Quad-rotor model. Section 4 describes how a a control law is applied. Experimental results are shown in Section 5. Finally, conclusions are given in Section 6.

[1]UAMRR-Universidad Autónoma de Tamaulipas, Carr. Reynosa-San Fdo. y Canal Rodhe Col. Arcoiris, Reynosa Tamps, 88779, México. dlara@uat.edu.mx

[2]Laboratoire MIS, Université de Picardie Jules Verne 33 rue S. Leu, 80039 Amiens, France. claude.pegard@u-picardie.fr

[3]Universidad Autónoma de Nuevo Leon, Ciudad Universitaria, Nuevo Leon, México. efrain.alcortagr@uanl.edu.mx

II. X4 ARCHITECTURE DESCRIPTION

In this section we describe the hardware and software architecture used to build the onboard system, called autopilot, that allows to the quad-rotor fly in autonomous way. This onboard electronic system consists of two microcontrollers, orientation and position sensors, such as the the inertial measurement unit (IMU), and ultrasonic sensors. The system also include a wireless modem.

A. Architecture

The autopilot system uses as main processor the low cost microprocessor MBED (NXP LPC1768), which has an execution speed of 96MHz, and an architecture of 32-bit, with 32KB of RAM and 512KB of FLASH memory. It is powered using 5V USB or 4.5-9V supply applied to VIN pin and includes a built-in USB programming interface that is as simple as using a USB Flash Drive. This processor came in a module with a wide variety of devices such as an integrated USB bus, three serial ports, one CAN bus, one Ethernet ports, six PWM outputs, general purpose ports, and six analog input channels. In addition the modules have a small size and are supplied in a 40-pin DIP package (54x26mm). MBED microprocessor is programmed using a C-language development system, with a Developer Website Lightweight Online Compiler. Figure 2 shows the autopilot architecture block diagram whose connections are indicated in table I. The autopilot consists of a simple master-slave configuration connected by serial port. Both microprocessor work in parallel, and each one does a group of specific tasks at the same time. The master processor gets the attitude and angular rate from the inertial sensors (gyroscopes, accelerometers and magnetometers). Through the serial port, obtain the position data (x, y, z) from the slave microcontroller; with this data calculates the control law to send the correction using PWM output to each motor driver. In addition sends and receives periodically data information to and from the ground station using a serial wireless modem. The slave processor (BASIC STAMP 2SX) obtains the position data $(x, y, z\ )$ reading the ultrasonic sensors, filtering such data to eliminate noise.

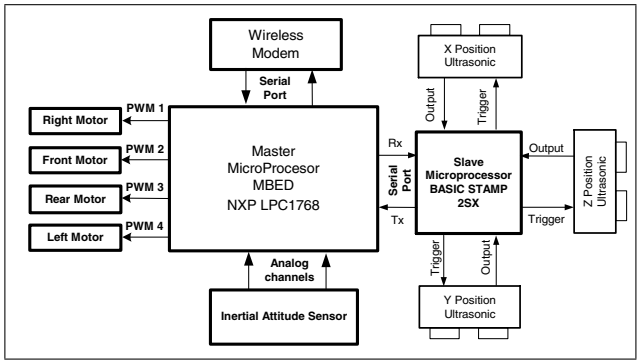

Fig. 2. Architecture scheme using MBED processor as main processor

TABLE I
MBED PIN CONNECTIONS FOR AUTOPILOT.

Sensor or Actuator	MBED Port	Description
Accelerometers X, Y Z	P18, P19, P20	Analog input
Gyroscopes X, Y, Z	P15, P16, P17	Analog input
Compas sensor, MOSI, MISO, SCK	P11, P12, P13	Serial
Slave processor Tx, Rx	P9,P10	Serial
Modem TX, RX	P28, P27	Serial
Motors RIGHT-LEFT-FRONT-REAR	P21-P24	PWM

B. Operating principle of the inertial measurement unit (IMU)

The IMU consists of three type of sensors: accelerometers, gyroscopes and magnetometers; each sensor measures the angles (pitch, roll and yaw, known as Euler Angles) and angular rates around a respective axis; figure 3 shows the schematic diagram for the Euler angles calculation. The accelerometers measure acceleration on each axis, and can be used as a tilt sensors when the component of gravity is measured. Then, combining three accelerometers it is possible to estimate the pitch and roll angles, nevertheless, the linear acceleration and vibration can cause a not accurate reading. The gyroscopes are a second type of sensor used to measure the angular rate around each axis and are not affected by acceleration and vibration, as it measures the speed of each axis, but this sensor has a accumulative drift error over time, when its output is integrated. The IMU has three gyroscopes orthogonally arranged. The magnetometer measures the magnetic field of the earth and is used to obtain the angular position relative to magnetic north. The main microprocessor uses the analog to digital converter (ADC) to read the signal from the sensors to calculate the angular orientation in real time, this is done as follows:

- First, the signal data from accelerometers and gyroscopes is filtered using low pass digital filter.
- With data from accelerometers, an estimation for tilt angles is done: pitch, $\phi_{acc} = \arctan(a_y/a_z)$, and roll, $\theta_{acc} = \arctan\left(a_x/\sqrt{a_y^2 + a_z^2}\right)$.
- Using the gyroscopes signal data in kinematics Euler equation, the angular rate in terms of the inertial frame of reference is calculated.
- A First order complementary filter data fusion algorithm is used to calculate attitude and heading.

The signal from each sensor must be free of noise and interference; then, the implementation of digital filters plays an important role in the design of an inertial measurement unit. To obtain the yaw angle, the outputs X_m, Y_m and Z_m of three axis magnetometer are used in a rotation matrix, (in function of pitch and roll angles) to calculate the components X_h and Y_h as follows:

$$X_h = X_m \cos\phi + Y_m \sin\phi \sin\theta - Z_m \sin\phi \sin\theta \quad (1)$$

$$Y_h = Y_m \cos\theta + Z_m \sin\theta \quad (2)$$

From this magnetic field components projections in the horizontal plane, a first estimation of the yaw angle is done. The gyroscope on the z axis serves to obtain the angular yaw

rate, and using again the Euler kinematics, the angular yaw rate expressed in the inertial reference frame is obtained. The estimated yaw angle and yaw rate are used in the complementary filter to calculate the yaw angle.

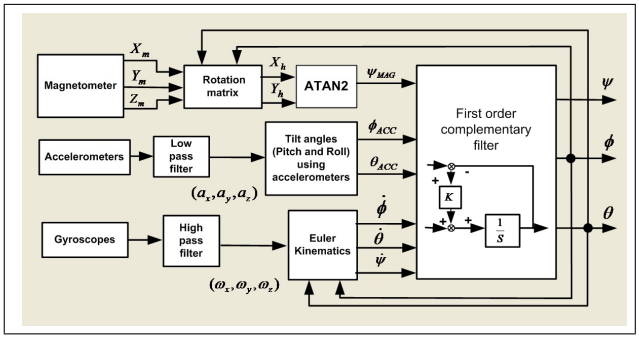

Fig. 3. Algorithm to calculate the Euler angles.

The IMU uses the Combo Razor 6DOF contains inertial sensors such as a three-axis accelerometer (ADXL335) and three gyroscopes, the two axis gyro LPR530AL, for the angular velocity around the x and y axes, and the single axis gyro LY530ALH, for the angular velocity about the axis z. The MicroMag 3-Axis Magnetometer SEN-00244 is used to calculate the yaw angle.

C. Software structure

The system software handles several functions to read sensors and calculate the control law that stabilizes the aerial vehicle. This software is a state machine, which is represented by the functional scheme shown in figure 4, and the program is executed in four layers.

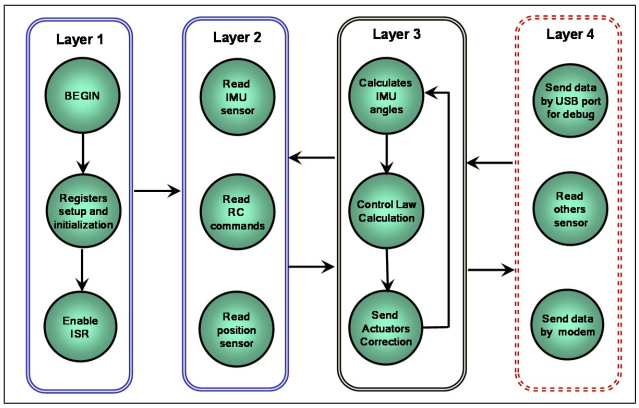

Fig. 4. Autopilot software state machine.

The first layer is executed only one time, here the program sets and initializes all registers and variables used during the program execution. Also the interrupt service routines (ISR) are enable. The second layer is dedicated to read all the IMU sensors, position sensor (slave microprocessor) and Remote control commands; this is done reading ADC channels and serial ports by interrupt routines. The main loop execution begins in the third layer, where the main tasks to be executed are: a) calculation of the euler angles and angular rates, b) compute the control law for stabilization and position and c) send correction to the motors using the PWM output, where the speed for each motor is set as follows:

$$Velocity\ front/rear\ motor = throttle + yaw\ control \pm pitch\ control$$

$$Velocity\ right/left\ motor = throttle - yaw\ control \pm roll\ control$$

Fourth layer establishes the communication with the ground station, one way is by wireless modem, the other is using USB programming port, this is useful for debugging.

D. Ground station

The ground station (GS), consists of four elements interrelated each one with a specific function. *The operator* pilots, supervises and makes decisions of the flight when is presented a situation not covered by the control algorithm. *The software* or informatics application is the interface man-vehicle that displays flight parameters, allows to visualize the environment to pilot the vehicle and depending on the application, it can control the aerial robot remotely; the software must be highly reliable and is developed using software engineering. *The hardware* is the computer equipment peripherals on which the ground station software resides. *The link* consists of the wireless transmitter-receiver equipment and antennas used to establish a bidirectional communication between the aerial robot and the GS. The design of the GS consists of two parts: the interface that corresponds to the application screen on which are shown virtually the main flight instruments, and the link that refers to the communication devices used by the aerial robot and the GS. Figure 5 shows a simple graphical interface that consists of three sections. The first section allows the configuration of communication port; it serves to establish the parameters for the corresponding communication channel, for instance the baud rate. The second section contains the instruments indicators that displays the vehicle attitude. Other indicators included are the speed meter, altimeter, and battery level. The third section can serve to graphic the position of the vehicle, if the robot flies indoor, for example in a testing room, then only a coordinate axis x and/or y respect to time is displayed. If the vehicle flies outdoor, then the geodetic position is showed using local maps.

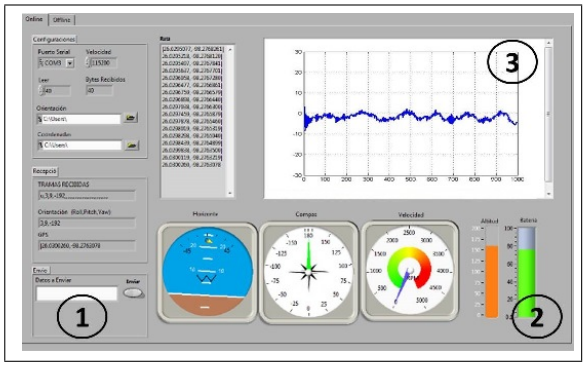

Fig. 5. Simplified ground station.

III. THE QUAD-ROTOR MODEL

This model is basically obtained representing the aerial vehicle as a solid body in the 3D space, subject to one force and 3 moments [9]. The quad-rotor dynamics, depicted on figure 1, is modeled by the following equations [3]

$$\begin{aligned} m\ddot{x} &= -u\sin\theta && (3)\\ m\ddot{y} &= u\cos\theta sin\phi && (4)\\ m\ddot{z} &= u\cos\theta\cos\phi - mg && (5)\\ \ddot{\psi} &= \tau_\psi/I_z && (6)\\ \ddot{\theta} &= \tau_\theta/I_y && (7)\\ \ddot{\phi} &= \tau_\phi/I_x && (8) \end{aligned}$$

where x and y are the coordinates in the horizontal plane, and z is the vertical position, ψ is the yaw angle around the z-axis, θ is the pitch angle around the y-axis, and ϕ is the roll angle around the x-axis. u is the thrust directed out the bottom of the aircraft and τ_ψ, τ_θ and τ_ϕ are the yawing, pitching and rolling moments respectively. Considering only the attitude dynamics, the following matrix transfer function that includes time delay and actuator dynamics was obtained in [8].

$$\Phi(s) = G(s, e^{-\tau s})u(s) \quad (9)$$

where:

$$G(s,e^{-\tau s}) = \begin{pmatrix} \frac{k_1 e^{-\tau s}}{s^2(s+a_\phi)} & 0 & 0\\ 0 & \frac{k_2 e^{-\tau s}}{s^2(s+a_\theta)} & 0\\ 0 & 0 & \frac{k_3 e^{-\tau s}}{s^2(s+a_\psi)} \end{pmatrix} \quad (10)$$

and $u(s) = \begin{bmatrix} v_\psi(s) & v_\theta(s) & v_\phi(s) \end{bmatrix}^T$, represent the input, $k_1 = lk_\phi/I_x$, $k_2 = lk_\theta/I_y$, $k_3 = lk_\psi/I_z$. Those parameters were computed for our X4-flyer and have the following values: $k_1 = k_2 = 33.23$, $k_3 = 16.95$, $a_\phi = a_\theta = a_\psi = 4.1$. The time delay has been computed to be $\tau = 0.017$. This mathematical model is used to synthesize the control law to stabilize the closed-loop system.

IV. APPLICATION OF CONTROL ALGORITHMS

A. Altitude and yaw control

The controllers that are used here come from the paper [3]. Let us first recall the main lines of the controller synthesis. In order to obtain a desired linear behavior for the vertical position, the following control strategy is proposed.

$$u = (r_1 + mg)/\cos\theta cos\phi \quad (11)$$

where $r_1 = -a_1\dot{z} - a_2(z - z_d)$, with a_1 and a_2 as positive constants and z_d is the desired altitude. Similarly, the yaw angular position can be controlled by using

$$\tau_\psi = -a_3\dot{\psi} - a_4\psi \quad (12)$$

Introducing (11)-(12) into (3)-(6) and provided that $\cos\theta\cos\phi \neq 0$, we obtain

$$\begin{aligned} m\ddot{x} &= -(r_1 + mg)\frac{\tan\theta}{\cos\phi} && (13)\\ m\ddot{y} &= (r_1 + mg)\tan\phi && (14)\\ m\ddot{z} &= -a_1\dot{z} - a_2(z - z_d) && (15)\\ \ddot{\psi} &= -a_3\dot{\psi} - a_4(\psi - \psi_d) && (16) \end{aligned}$$

It has been proved in [3] that by choosing a_i for $i = 1, 2, 3$ suitably, we can ensure a stable well-damped response in the vertical and yaw control axes. Then, from (15) and (16), it follows that $\psi \to \psi_d$ and $z \to z_d$.

B. Control of lateral position and roll

Assuming that $r_1 \to 0$ for a time T large enough. Then, (13) and (14) reduce to

$$\begin{aligned} \ddot{x} &= -g\frac{\tan\theta}{\cos\phi} && (17)\\ \ddot{y} &= g\tan\phi && (18) \end{aligned}$$

Consider the subsystem given by (8) and (18), in order to simplify the analysis, we will impose a very small upper bound on $|\phi|$, such that $\tan\phi \to \phi$, because is arbitrarily small. Therefore, (8) and (18) reduce to

$$\begin{aligned} \ddot{y} &= g\phi && (19)\\ \ddot{\phi} &= \tau_\phi && (20) \end{aligned}$$

which represent four integrators in cascade. Then the controller is given by

$$\begin{aligned} \tau_\phi &= -\sigma_{\phi_1}(\dot{\phi} + \sigma_{\phi_2}(\phi + \dot{\phi} + \sigma_{\phi_3}(2\phi + \dot{\phi} + \frac{\dot{y}}{g} + && (21)\\ &\quad \sigma_{\phi_4}(\dot{\phi} + 3\phi + 3\frac{\dot{y}}{g} + \frac{y}{g})))) && (22) \end{aligned}$$

where σ_s is a saturation function with upper bound $s > 0$. This control law indeed, comes from the technique developed in [10] based on nested saturations control. Using this technique, it has been proved in [3] that $\phi, \dot{\phi}, y$ and $\dot{y}$ converge to zero.

C. Control of forward position and pitch

From (19), we obtain $\phi \to 0$, then the (x, θ) subsystem is given by

$$\begin{aligned} \ddot{x} &= -g\tan\theta, && (23)\\ \ddot{\theta} &= \tau_\theta && (24) \end{aligned}$$

As before, we assume that $|\theta|$ has a very small bound such that $\tan\theta \approx \theta$. Therefore, (23) reduces to

$$\ddot{x} = -g\theta, \quad (25)$$

Similarly, using the procedure proposed in the previous subsection we obtain

$$\begin{aligned} \tau_\theta &= -\sigma_{\theta_1}(\dot{\theta} + \sigma_{\theta_2}(\theta + \dot{\theta} + \sigma_{\theta_3}(2\theta + \dot{\theta} - \frac{\dot{x}}{g} + && (26)\\ &\quad \sigma_{\theta_4}(\dot{\theta} + 3\theta - 3\frac{\dot{x}}{g} - \frac{x}{g})))) && (27) \end{aligned}$$

D. PD control

In order to compare the performance of the control law proposed we have implemented a standard PD control, which is given by

$$u = -k_{p_1}(z - z_d) - k_{d_1}\dot{z} + mg \quad (28)$$

$$\tau_\psi = -k_{p_2}\psi - k_{d_2}\dot{\psi} \quad (29)$$

$$\tau_\phi = -k_{p_3}\phi - k_{p_4}y - k_{d_3}\dot{\phi} - k_{d_4}\dot{y} \quad (30)$$

$$\tau_\theta = -k_{p_5}\theta + k_{p_6}x - k_{d_5}\dot{\theta} + k_{d_6}\dot{x} \quad (31)$$

where k_{pi} and k_{di} for $i = 1, \ldots, 6$ are positive constants.

TABLE II

PARAMETER VALUES USED IN BOTH THE NESTED SATURATIONS AND PD CONTROLLERS

Control parameter	Value	Control parameter	Value
a_1	1.5	k_{p_1}	1.2
a_2	1.2	k_{p_2}	7.2
a_3	131.04	k_{p_3}	9
a_4	7.2	k_{p_4}	2
ϕ_1	640	k_{p_5}	9
ϕ_2	320	k_{p_6}	2
ϕ_3	160	k_{d_1}	1.5
ϕ_4	80	k_{d_2}	131.04
θ_1	640	k_{d_3}	163.8
θ_2	320	k_{d_4}	0.333
θ_3	160	k_{d_5}	163.8
θ_4	80	k_{d_6}	0.333

V. EXPERIMENTAL RESULTS

This section presents the experimental results obtained with the X4-flyer platform using the autopilot embedded system, shown in figure 6, which is an inexpensive architecture and can be done by someone with a minimal experience in circuit electronic design.

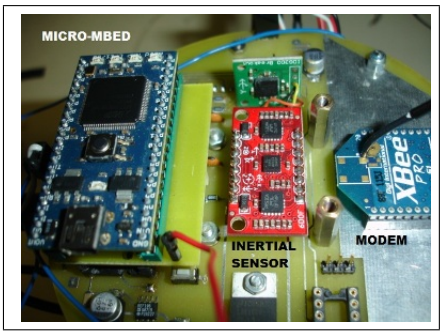

Fig. 6. Low cost autopilot circuit

The experiments were used here to validate in safe mode only the vehicle angular position or attitude, but allows validate also linear position. The vehicle was attached to a pivot mechanism, which allows it rotate, and move in forward-backward and up-down directions freely, as depicted in figure 7. This means, that the mechanism serves to validate attitude stabilization and position, which imitates the vehicle ordinary fly. The link between the autopilot and the GS computer was done using the microprocessor USB programming port, sending data every $5ms$ and stored for later visualization. We compare the performance of two control algorithms given in the above section. The responses for the pitch angle are shown in graphs appearing in figure 8, these response was obtained during a test of 125 seconds.

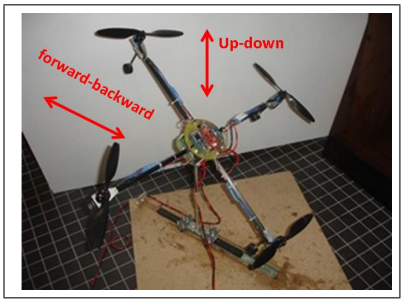

Fig. 7. scheme of experimental setup.

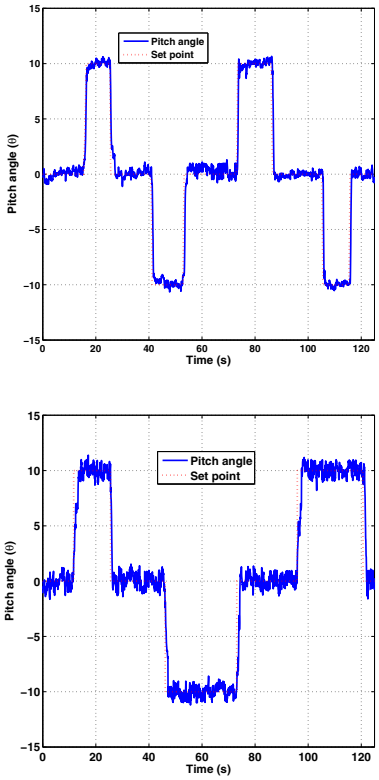

Fig. 8. Pitch angle response: using PD and nested saturation control

In similar way, the response for roll angle are given in figure 9, and for yaw angle in figure 10. The set point was changed manually, using the remote control joystick, but the only allowed values for this set point were 0, 10 and -10 degrees for pitch and roll; this to evaluate the step response. The reference set point for yaw angle, was established to zero for all tests. From the figure we can see that the response is almost similar for both controllers , but in the PD control exist more oscillation and noise than in the nested saturations control algorithm. Parameters adjustment for the controllers are given in tables II.

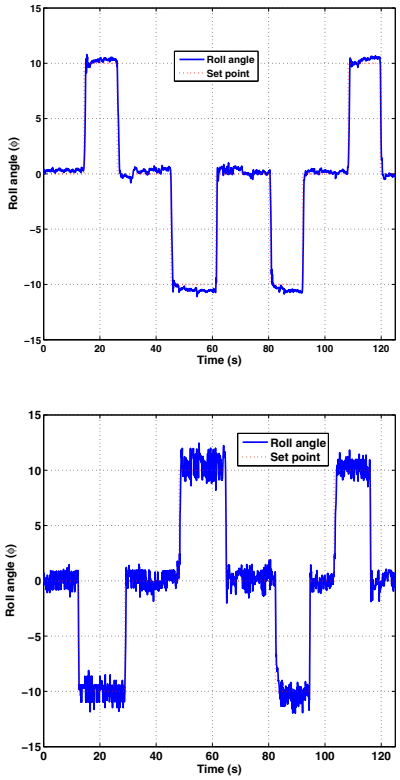

Fig. 9. Roll angle response: Left, nested saturation control and right, PD control

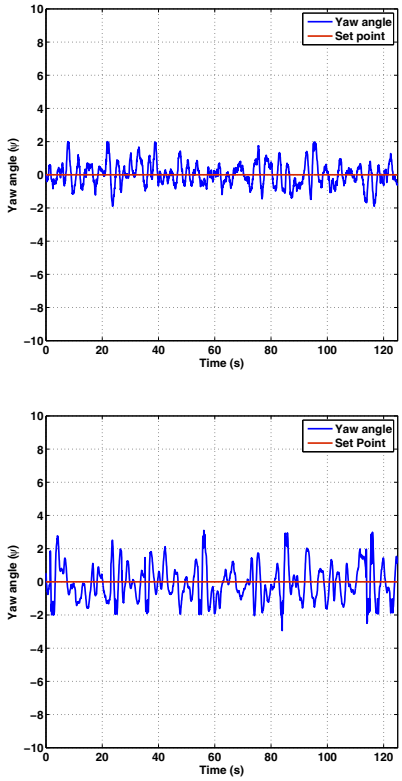

Fig. 10. Yaw angle response: Left, nested saturation control and right, PD control

VI. CONCLUSIONS

In this document, we presented the implementation of two control algorithms in a real time embedded system, in order to compare their performance. The objective of this work was to develop a low cost architecture for the autopilot, using an easy development system. This can allow to validate several control algorithms in real time. It is worthy to mention that in a low cost microprocessor embed sophisticated control algorithms need more programming efforts, but for educational use where a simple control law is used, this can be a good solution. In attitude plots we can see that although the PD controller has almost same behavior as compared with the Nested saturations controller, when it is used to stabilize, an significant oscillation is present. Oscillation in both controllers is because we have used low cost inertial sensors. Nevertheless for practical results are still satisfactory in our indoor platform.

ACKNOWLEDGMENT

Authors wish to thank to the Mexican Council of Science and Technology (CONACYT), the Autonomous University of Tamaulipas and the Autonomous University of Nuevo Leon for the support in this research project.

REFERENCES

[1] K. P. Valavanis, Advances in unmanned aerial vehicles, state of the art and the road to autonomy, Springer (2007).

[2] R. Austin, *Unmanned aircraft systems: UAVS design, development and deployment*, Wiley (2010).

[3] P. Castillo, A. Dzul, and R. Lozano, Real-time stabilization and tracking of a four rotor mini rotorcraft. *IEEE Transactions on Control Systems Technology*, vol. 12, pp. 510–516, (2004).

[4] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, Dynamic modelling and configuration stabilization for an x4-flyer, in *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, (2002).

[5] S. Bouabdallah and R. Siegwart, Backstepping and sliding-mode techniques applied to an indoor micro quadrotor, in *IEEE International Conference on Robotics and Automation, ICRA2005*, Barcelona, Spain, April (2005).

[6] A. Tayebi and S. McGilvray, "Attitude stabilization of a four-rotor aerial robot," in *Proceedings of the 43th IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[7] A. Sánchez, I. Fantoni, R. Lozano, and J. D. L. Morales, "Observer-based control of a PVTOL aircraft," in *16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[8] D. Lara, G. Romero, A. Sanchez, R. Lozano, A. Guerrero, Robustness margin for attitude control of a four rotor mini-rotorcraft: Case of study, *Mechatronics*, Vol 20, pp. 143-152, (2010).

[9] H. Goldstein, *Classical Mechanics*. Addison-Wesley (1980).

[10] A. R. Teel, "Global stabilization and restricted tracking for multiple integrators with bounded controls," *Systems & Control Letters*, vol. 18, pp. 165–171, 1992.