Proceedings of the 2006 IEEE
International Conference on Control Applications
Munich, Germany, October 4-6, 2006

Real-Time Embedded Control System for VTOL Aircrafts: Application to stabilize a quad-rotor helicopter

David Lara, Anand Sanchez, Rogelio Lozano, P. Castillo

Abstract— **In this paper, we present the design of an autopilot embedded control system for VTOL aircrafts using low cost sensors. The embedded control system uses parallel processing architecture. In addition, multitasking software is used to implement the data acquisition, control law computation, and correction output to get the desired set point. The control law can be easily tuning to improve the performance of the vehicle. We evaluate the performance of this platform in a quad-rotor helicopter. The main goal is to achieve the stationary flight using two control strategies, a linear PD control and nonlinear nested saturations control. Real time experiments show that the autopilot is a platform relievable with low cost components.**

I. INTRODUCTION

Recently, the use of Unmanned Aerial Vehicles (UAVs) has grown up for different applications either for military or civil purposes such as forest fire detection, oil pipeline inspection, high voltage lines failure detection and surveillance [6]. A flight-control system represents one of the most difficult applications for embedded software, due to the catastrophic consequences of even the smallest errors. In addition, the robustness of the dynamic models and stability of the algorithms strategies for the flying vehicles are still questionable, since they have not been applied to applications as critical as a flight-control systems. An autonomous aircraft, which in itself is a class of robot, needs dynamic reconfigurability for its new generation of advanced flight control systems.

The goal of designing embedded systems is to get a good separation of the system's functionality into hardware and software as well as an efficient and correct implementation of the hardware/software components and their integration into a complete system. Beside system functionality the implementation has to meet all non-functional requirements. Typically, the design and implementation phases start with a uniform, abstract, and expressive system model.

It is clear that no single sensor can provide completely accurate vehicle position information. Therefore, multisensor integration is required in order to provide the on-vehicle system with complementary, sometimes redundant information for its location and navigation task. Many fusion technologies have been developed to fuse the complementary information from different sources into one representation format. The information to be combined may come from multiple sensors during a single period of time or from a single sensor over an extended period of time. Integrated multisensor systems have the potential to provide high levels of accuracy and fault tolerance. Multisensor integration and fusion provide a system with additional benefits. These may include robust operational performance, extended spatial coverage, extended temporal coverage, an increased degree of confidence, improved detection performance, enhanced spatial resolution, improved reliability of systems operation, increased dimensionality, full utilization of resources, and reduced ambiguity.

In this work we present a real-time embedded control system for VTOL aircrafts. Our objective is to develop a platform to stabilize flying vehicles which are able to make vertical take-off and landing. In order to test the embedded system we use a quad-rotor aircraft. This helicopter has the same aerodynamic properties that the conventional helicopters.

Many research works have consider this type of configuration previously. For example, [5] presented a dynamical model that incorporates the airframe, motor's dynamics, aerodynamics and gyroscopic effects. They proposed a control law based on the Lyapunov approach that guarantees global convergence of the state to the origin. In [1] backstepping and sliding mode techniques are applied to an indoor micro quad-rotor. Their results are validated in a PC-based bench test platform. The Attitude stabilization problem is treated in [8] using a quaternion based model and a new Lyapunov function for a PD controller. In that paper, the authors present their results in simulation and in experimentally using an embedded control system. However, in the later, the quad-rotor is only allowed to rotate freely around a fixed pivot. Other sophisticated control technique based in nested saturations can be found in [2]. In this publication the algorithms are implemented in a PC based platform and the performance is satisfactory. However the position and orientation measuring device is limited to work in radius of 1 meter. In [7] the authors faced this problem and proposed an observer for estimate the attitude and the horizontal velocity of a PVTOL aircraft. This is an essential research topic, which could resolve the high cost and availability sensors problems. In this work the main objective is to embed on board an efficient hardware and software system with suitable sensors that allow the aircraft to fly in safe way keeping a small area loading and the less as possible weight, optimizing in this way the power management.

This paper is organized as follows. The architecture and software description used for the stabilization of X4-Flyer are given in Section II. In Section III, we describe the dynamic

This work was supported by CONACYT, PROMEP and SEP in Mexico, also by SFERE and CNRS in France. D. Lara, A. Sanchez, R. Lozano, and P. Castillo are with Heudiasyc Laboratory, Centre de Recherches de Royallieu , B.P. 20529, Compiègne Cedex 60205, France {dlara, asanchez, rlozano, castillo}@hds.utc.fr

model of the VTOL aircraft. Section IV describes the control law design and the PD control used. Experimental results are shown in Section V. Finally, conclusions are given in Section VI.

II. Embedded control system

In this section we describe the embedded control system that allows to stabilize the VTOL aircraft. An embedded control system generally consists of three elements: sensors, actuators, and a computer. The control computer interacts with the continuous dynamics of the plant via the sensors and actuators. For each aircraft, we assume that there is only one computer system and its major function is to compute and generate control commands for the actuators that are based on sensor measurement.

The onboard electronic system consists of two micro-controllers, the inertial measurement unit sensor, wireless modem, and three ultrasonic sensors.

A. Architecture

Our autopilot system uses as main core an 8-bit micro-processor RABBIT RCM3400 module running at 29.4 MHz, with 512K bytes of flash memory, 512K bytes of SRAM, eight single-ended or four differential analog inputs, 6 serial ports (A to F), 4 PWM outputs, two input capture for pulse width measurement. In addition the module has a small size (29 mm $\times$ 34 mm $\times$ 8 mm). RCM3400 modules from $Z-World$ Inc. are supported by an innovative *C*–language development system, the *Dynamic C* compiler, that provides libraries for string manipulation and math functions, allowing an easy development and is able to handle multitasking software.

Figure 1 shows the autopilot architecture block diagram, in which we can see the RCM3400 and the BS2SX connected using a serial RS232 port D, they work in parallel, so each one does a group of specific tasks at the same time. RCM3400 tasks are:

1) Width pulse measurement of each pulse for the train of pulses incoming from the Receiver FUTABA throughput the input capture pulse pin G1.
2) To get the attitude data (angular position and angular rate) from the IMU 3DM-G which has internally a combination of three gyros, three accelerometers and three magnetometers, we use the serial port D of the micro-controller.
3) To get the position data (x,y,z) from the BS2SX.
4) to calculate the control law.
5) to send the correction using PWM output to each one of the four FETS to drive the current of the DC motors.

In addition it sends periodically data information to the ground station using a wireless modem connected to the serial port E. BS2SX tasks are: obtain the x,y,z position information reading the ultrasonic sensors, filtering the data to eliminate noise and send the values to the RCM3400.

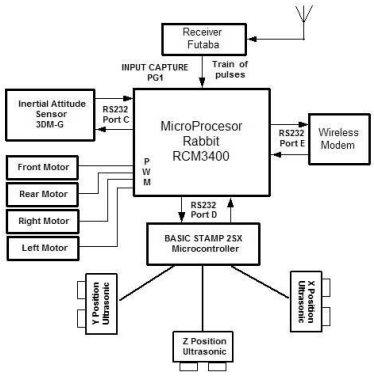

Fig. 1. Schema of the embedded control system.

B. Software Strategy: Parallel processing and Multitasking

As mentioned before, Dynamic C compiler for the RCM3400 has the ability to handle multitasking routines. This property allows to the main loop program treat several tasks "virtually in parallel" using the directive co-state of the dynamic C. In reality, a single processor can only execute one instruction at a time. When an application has multiple tasks to perform, multitasking software can usually take advantage of natural delays in each task to increase the overall performance of the system. Each task can do some of its work while the other tasks are waiting for an event, or for something to do. In this way, the tasks execute almost in parallel, we take advantage and exploit this feature in our application in the way that is described as follow. Functional software state machine block diagram is shown in Figure 2.

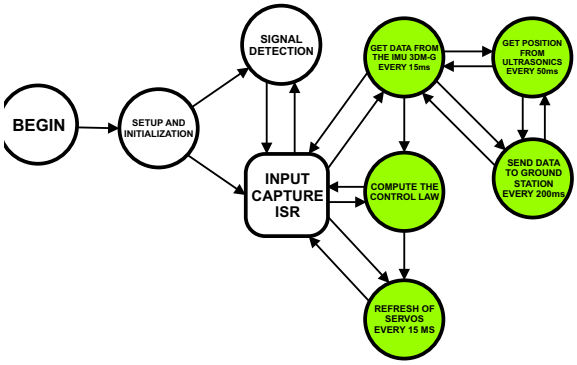

Fig. 2. Loop operations of the embedded system.

The program for the autopilot begin with the setup and initialization of the registers and variables that will be used for the program execution, then, an input capture interrupt service routine (ISR) is enable. The ISR, triggered by falling edge, consists in detect and measure the pulses incoming from the receiver FUTABA. If the signal is valid, then a periodical train of seven pulses must exist. They are described in the Table I.

Before the program enter in the main loop, it checks for valid signal incoming from the receiver, verifying the length

of the pulse 0, if no signal is present the autopilot indicates it flashing LEDS and waits for valid signal. For safety strategy, when valid signal is recognized, the program remain waiting until the remote control power lever is set to its minimal value. Once the program exits of safety loop, the main loop execution begin. The loop consists in five co-states.

Pulse	Name	Description
0	Flank	Used for ISR synchronization
1	Yaw	Yaw set point
2	Pitch	Offset adjustment for pitch
3	Power	Value of the thrust of the motors
4	Roll	Offset adjustment for roll
5	Auto-Manual	Switch for Auto-manual mode
6	Yaw-Trim	Yaw Fine adjustment

TABLE I

FUTABA PULSE DEFINITION

The first co-state reads the angular position and angular rate from the Inertial Measurement Unit (IMU) 3DM-G every 16ms, the communication with both devices is done using the Microstrain protocol. Once the attitude data is obtained, second co-state takes place immediately to compute the control law to stabilize the vehicle. Third co-state reloads the motors speed using the *PWM* output, calculated according with the following relation[1],

$$\begin{aligned} PWM_{M_1} &= u + \tau_\theta + \tau_\psi \\ PWM_{M_2} &= u - \tau_\phi - \tau_\psi \\ PWM_{M_3} &= u - \tau_\theta + \tau_\psi \\ PWM_{M_4} &= u + \tau_\phi - \tau_\psi \end{aligned}$$

where u is the throttle input, τ_ψ, τ_θ and τ_ϕ are the control inputs for the yaw, pitch and roll moments, respectively. M_i is the motor i, from $i = 1..4$, see Figure 3.

Fourth co-state, consists in wait for obtain the data from the microcontroller BS2SX which is working in parallel with the RCM3400. x, y, z position data is sent every 50 ms by the BS2SX, then "wait for" means that the main loop program don't stop polling for this data, every time that data arrive, an interruption occurs. Fifth co-state is the task with less priority and is done every 200 ms, consists in establish communication with the ground station sending the data for results visualization by modem.

C. Position measurement using Ultrasonic Sensors

A critical problem encountered when implementing control laws on vehicle-applications is knowing the relative distance from the vehicle to an object. We have decided to use the SRF04 ultrasonic range finder in order to estimate the x, y, z position. The main reasons for chose the SFR04 sensor are: it is relatively an inexpensive proximity sensor and is easy to configure. However, they are limited in resolution and the size of object they can detect.

III. DYNAMIC MODEL

The aerial vehicle used in this work is an electric quad-rotor aircraft, where each motor is attached to a rigid cross frame as it can be appreciated in Figure 3. It is a vertical takeoff and landing vehicle (VTOL) able to move omnidirectionally with the ability to fly in a stationary way.

In the quad-rotor helicopter the front and rear rotors rotate counter-clockwise while the left and right rotors rotate clockwise, canceling gyroscopic effects and aerodynamic torques in stationary trimmed flight. Vertical motion is controlled by the collective throttle input, i.e., the sum of the thrusts of each motor. Forward/backward motion is achieved by controlling the differential speed of the front and rear motors. This causes the quad-rotor to tilt around the y-axis generating a pitch angle. The left/right motion of the vehicle is achieved by controlling the differential speed of the right and left motors, tilting around the x-axis and producing a roll angle. Finally, yaw movement is obtained by taking advantage of having two sets of rotors rotating in opposite direction. Thus a yaw angular displacement is obtained by increasing or decreasing the speed of the front and rear motors while decreasing or increasing the speed of the lateral motors. This is done keeping the total thrust constant so that the altitude remains unchanged.

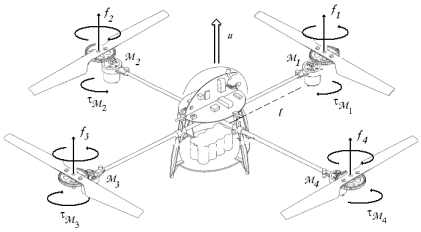

Fig. 3. Schema of the four-rotor rotorcraft.

The dynamic model of this aircraft is basically obtained representing the quad-rotor as a solid body evolving in 3D and subject to one force and 3 moments [4].

The quad-rotor dynamics, depicted on Figure 3, are modeled by the following equations [2]

$$m\ddot{x} = -u\sin\theta \quad (1)$$
$$m\ddot{y} = u\cos\theta sin\phi \quad (2)$$
$$m\ddot{z} = u\cos\theta\cos\phi - mg \quad (3)$$
$$\ddot{\psi} = \tau_\psi \quad (4)$$
$$\ddot{\theta} = \tau_\theta \quad (5)$$
$$\ddot{\phi} = \tau_\phi \quad (6)$$

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 moments (yawing moment, pitching moment and rolling moment).

[1]this case is for the quad-rotor aircraft. We can change easily this relation for another vehicle.

IV. CONTROL STRATEGIES

In this section, we present two control strategies to stabilize the quad-rotor helicopter. First, we present a non linear control algorithm, obtained using the Lyapunov analysis and the saturation functions. The second algorithm is a classical linear control PD.

A. Nonlinear control strategy

This controller comes from the paper [2]. Let us first recall the main lines of the controller synthesis.

1) Altitude and Yaw Control: In order to obtain a desired linear behavior for the vertical position, we propose the following control strategy

$$u = \frac{r_1 + mg}{\cos\theta cos\phi} \tag{7}$$

where r_1 is a stable polynomial, define as

$$r_1 = -a_1\dot{z} - a_2(z - z_d) \tag{8}$$

where a_1 and a_2 are positive constants and z_d is the desired altitude. Similarly, the yaw angular position can be controlled by using

$$\tau_\psi = -a_3\dot{\psi} - a_4\psi \tag{9}$$

Introducing (7)-(9) into (1)-(4) and provided that $\cos\theta\cos\phi \neq 0$, we obtain

$$\begin{aligned} m\ddot{x} &= -(r_1 + mg)\frac{\tan\theta}{\cos\phi} && (10)\\ m\ddot{y} &= (r_1 + mg)\tan\phi && (11)\\ m\ddot{z} &= -a_1\dot{z} - a_2(z - z_d) && (12)\\ \ddot{\psi} &= -a_3\dot{\psi} - a_4(\psi - \psi_d) && (13) \end{aligned}$$

It has been proved in [2] that by choosing a_i for $i = 1,2,3,4$ suitably we can ensure a stable well-damped response in the vertical and yaw control axes. Then, From (12) and (13) it follows that $\psi \to \psi_d$ and $z \to z_d$.

2) Control of lateral position and roll angle: From (8) and (12) we have $r_1 \to 0$ for a time T large enough. Therefore, (10) and 11) reduce to

$$\begin{aligned} \ddot{x} &= -g\frac{\tan\theta}{\cos\phi} && (14)\\ \ddot{y} &= g\tan\phi && (15) \end{aligned}$$

Consider the subsystem given by (6) and (15). In order to simplify the analysis we will impose a very small upper bound on $|\phi|$ such that $\tan\phi - \phi$ is arbitrarily small. Therefore, (6) and (15) reduce to

$$\begin{aligned} \ddot{y} &= g\phi && (16)\\ \ddot{\phi} &= \tau_\phi && (17) \end{aligned}$$

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

$$\tau_\phi = -\sigma_{\phi_1}\left(\dot{\phi} + \sigma_{\phi_2}\left(\phi + \dot{\phi} + \sigma_{\phi_3}\left(2\phi + \dot{\phi} + \frac{\dot{y}}{g} + \sigma_{\phi_4}\left(\dot{\phi} + 3\phi + 3\frac{\dot{y}}{g} + \frac{y}{g}\right)\right)\right)\right) \tag{18}$$

where σ_s is a saturation function with upper bound constant $s > 0$. This control law indeed, comes from the technique developed in [9] based on nested saturations control. Using this technique, it has been proved in [2] that $\phi, \dot{\phi}, y$ and $\dot{y}$ converge to zero.

3) Control of forward position and pitch angle: From (16)-(18), we obtain $\phi \to 0$, then the (x,θ) subsystem is given by

$$\begin{aligned} \ddot{x} &= -g\tan\theta, && (19)\\ \ddot{\theta} &= \tau_\theta && (20) \end{aligned}$$

As before, we assume that $|\theta|$ has a very small bound such that $\tan\theta \approx \theta$. Therefore, (19) reduces to

$$\ddot{x} = -g\theta, \tag{21}$$

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

$$\tau_\theta = -\sigma_{\theta_1}\left(\dot{\theta} + \sigma_{\theta_2}\left(\theta + \dot{\theta} + \sigma_{\theta_3}\left(2\theta + \dot{\theta} - \frac{\dot{x}}{g} + \sigma_{\theta_4}\left(\dot{\theta} + 3\theta - 3\frac{\dot{x}}{g} - \frac{x}{g}\right)\right)\right)\right) \tag{22}$$

B. Linear control strategy

In order to obtain the linear model from (1)-(6), we define the following state equations

$$[x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}]^T = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \psi\ \dot{\psi}\ \theta\ \dot{\theta}\ \phi\ \dot{\phi}]^T$$

and we also define the control inputs as

$$[u_1\ u_2\ u_3\ u_4]^T = [u - mg\ \ \tau_\psi\ \ \tau_\theta\ \ \tau_\phi]^T$$

Then, the dynamic equations yield

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{-u_1}{m}\sin x_9 - g\sin x_9 \\ x_4 \\ \frac{u_1}{m}\cos x_9\sin x_{11} + g\cos x_9\sin x_{11} \\ x_6 \\ \frac{u_1}{m}\cos x_9\cos x_{11} + g\cos x_9\cos x_{11} - g \\ x_8 \\ u_2 \\ x_{10} \\ u_3 \\ x_{12} \\ u_4 \end{bmatrix}$$

or

$$\dot{x} = f(x,u)$$

Let $x = 0$ be an equilibrium point with $f(0,0) = 0$. Then, the linearization by Taylor series about the origin produces the following linear system

$$\dot{x} = Ax + B\bar{u} \tag{23}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T,$$

$$\bar{u} = [u_1 \ u_2 \ u_3 \ u_4]^T$$

System (23) is controllable, since the controllability matrix, $\mathscr{C} = [B \ \ AB \ \ A^2B \ \ A^3B]$, is of full rank with controllability index $\mu = 4$.

To stabilize system (23) we propose the standard PD control as follows

$$u_1 = -k_{p_1}(z - z_d) - k_{d_1}\dot{z} \quad (24)$$

$$u_2 = -k_{p_2}\psi - k_{d_2}\dot{\psi} \quad (25)$$

$$u_3 = -k_{p_5}\theta + k_{p_6}x - k_{d_5}\dot{\theta} + k_{d_6}\dot{x} \quad (26)$$

$$u_4 = -k_{p_3}\phi - k_{p_4}y - k_{d_3}\dot{\phi} - k_{d_4}\dot{y} \quad (27)$$

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

In order to compare the performance of the embedded control system we apply the previous linear control strategy to stabilize system (1)-(6), see next section.

V. Experimental Results

In this section, we show the experimental results obtained with our platform that consists basically in the VTOL aircraft (in this case the quad-rotor) with the autopilot embedded system, which communicates with a PC ground station by wireless modem. Data information is send by the autopilot every 200 ms and it is stored in the PC for later visualization. Helicopter mechanical structure used for our test was built by Draganfly, we have replaced only the main control circuit board by the our autopilot design.

Note that, the quad-rotor helicopter can be controlled manually using the remote control Futaba. To test our real-time embedded control system, we have applied the two control strategies given in section IV. Figures 5-10 show the performance of every controller when applied to the quad-rotor system. The control objective is to make the helicopter hover at an altitude of 30 cm during 180 s. Then, the desired values are: (x, y, z)=(0, 0, 30) in cm and (ψ, θ, ϕ)=(0,0,0)°. To measure the orientation of the aircraft we use a commercial IMU sensor (Microstrain), composed by accelerometers, gyrometers and magnetometers. To measure the position of the aircraft we use the ultrasonic sensors, this sensor could be changed by a camera or a GPS.

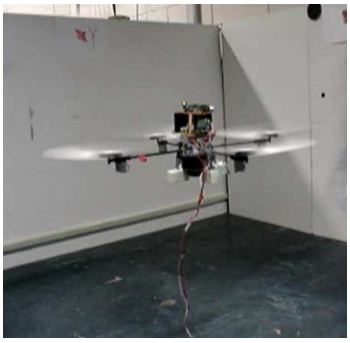

Fig. 4. Real time embedded control system for a quad-rotor aircraft.

The control inputs implemented in the program are: u is the throttle input or 'gas', τ_θ is used to control the x position and the pitch angle (θ), τ_ϕ is used to control the y position and the roll angle (ϕ), and τ_ψ controls the yaw angle (ψ). Parameters adjustment for PD and Nested Saturations controllers are given in Table II. These parameters were tuned in manual form, until obtaining a better response of the system. We use the following approximation to estimate the linear velocity in the control law: $\dot{q}_t \approx \frac{q_t - q_{t-T}}{T}$, where q is a given variable and T is the sampling period. In order to obtain a good estimate of this velocity, pass bass numerical filters have been used.

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
T	50 ms		

TABLE II

Parameter values used in the non linear and PD controllers

Figures 5 - 7 show the orientation of the quad-rotor helicopter. Note that the response is almost similar when using the two control strategies. This is because the helicopter is close to zero, and it is well know that when the system is close to the origin a linear controller is sufficient to stabilize it. Nevertheless, our interest is not to compare two control strategies, because [3] have done it. The main goal is to test two different control strategies in the proposed embedded control system, to observe the performance of the platform and to show the advantage of use embedded systems.

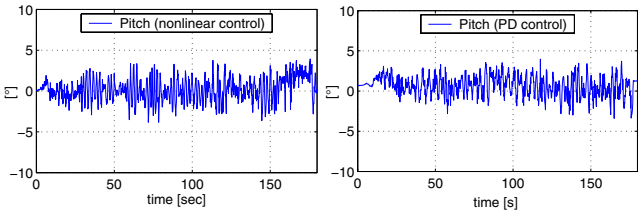

Fig. 5. Pitch angle.

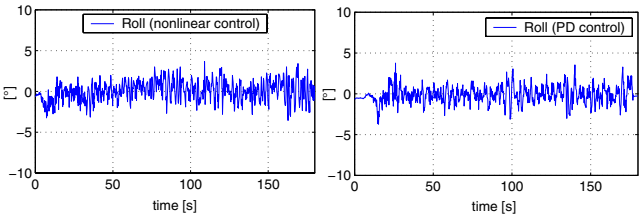

Fig. 6. Roll angle.

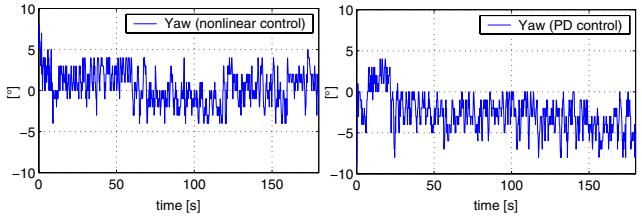

Fig. 7. Yaw angle.

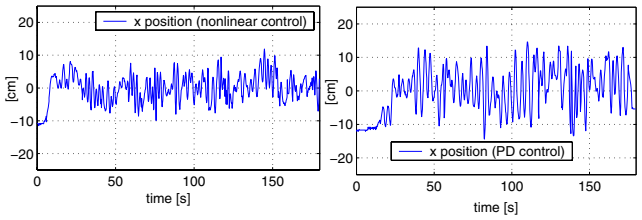

Fig. 8. x position.

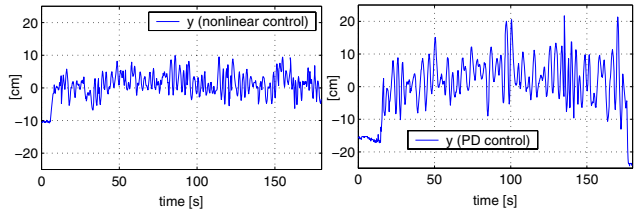

Fig. 9. y position.

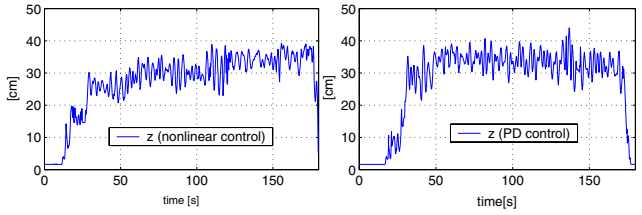

Fig. 10. z position.

Figures 8 to 10 show the x, y and z position of the aircraft. Note that, when the nonlinear controller was applied, the error position is minor with respect to the linear controller. In addition, the nonlinear controller is more robust than the linear controller when some perturbations are applied.

Figure 4 shows the real control application of the embedded control system. In this experience we use the nonlinear control strategy. To measure the position of the vehicle we use the ultrasound sensors, note that, the quad-rotor keeps a desired distance (position) even though a person or an object comes near to aircraft, see Figure 4.

Further information and videos are available at the web site: http://www.hds.utc.fr/∼asanchez

VI. Conclusions

In this paper, we presented a real time embedded control system for the VTOLs aircraft. In order to compare the performance of the platform, we have made some test using a quad-rotor helicopter and two control strategies. One of these is a conventional linear control, and the other is a nonlinear control law. The performance of both strategies were close when the operational points are close to zero. The nonlinear controller is most robust than the linear controller subject to aggressive perturbations in the orientation and to the noise originated from sensors.

Experimental results have showed that the embedded control system performs well. The experience was made indoor the laboratory, but we also made some experiences outdoor, that is, in real conditions. In outdoor experience we didn't use the ultrasound sensors and the closed-loop system was made only to stabilize the orientation. Next work, we will use a GPS sensor to measure the position of the aircraft.

References

[1] 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.

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

[3] P. Castillo, R. Lozano, and A. Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, 25(6), 2005.

[4] H. Goldstein. *Classical Mechanics*. Addison-Wesley Series in Physics. Addison-Wesley, U.S.A., second edition, 1980.

[5] 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.

[6] K. Kaaniche, D. Lara, C. Pegard, A. Dupuis, and P. Vasseur. Autonomous aerial surveillance of road traffic. In *Proceedings of the 20th Bristol UAV systems Conference*, Bristol, U.K., 2005.

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

[8] 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.

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