Kinematic and Dynamic Modeling and Control of a 3-Rotor Aircraft

Philippe Rongier, Erwann Lavarec, François Pierrot

Kinematic and Dynamic Modeling and Control of a 3-Rotor Aircraft

Philippe Rongier and Erwann Lavarec
WANY Robotics
R&D Department
101 place Pierre Duhem, 34961 Montpellier Cedex 2, France
prongier@wanyrobotics.com, elavarec@wanyrobotics.com

François Pierrot, IEEE senior member
LIRMM - CNRS
Robotics Department
161 rue Ada, 34392 Montpellier Cedex 5, France
pierrot@lirmm.fr

***Abstract*—This paper deals with the design of a controller and its implementation in a mini-rotorcraft toy with 3 rotors. A new original low-cost tilt angle sensor is introduced and kinematic and dynamic models are developed and implemented with the aim of developing an autopilot mode to stabilize and maintain hovering flight on demand. The result has to fit into a low-cost microcontroller and run in real-time.**

Index Terms: 3-rotor aircraft, hovering flight, Planar Vertical Take Off and Landing (PVTOL), aircraft control and dynamics, low-cost tilt angle sensor

I. INTRODUCTION

The Automatic control of flying machines has been a challenge to researchers for many years [1].
This paper shows the design and implementation of a real time command for maintaining the hovering flight of a 3-rotor toy aircraft [2]. Rotorcraft are one of the most complex types of flying machine [3]. Blackhawk Vectron®[1] [4] is a gyro-helicopter based toy. This 3-rotor aircraft uses a new concept in hovering flight since the entire body rotates, providing a gyroscopic effect. Despite this effect, the stability of the aircraft is not sufficient to maintain hovering flight. The toy isn't easily manoeuverable and often crashes to the ground, so it has been decided to improve the auto-pilot mode. Thus, a new low-cost tilt angle sensor has been designed and integrated into the aircraft. With a kinematic and dynamic model of the aircraft [5] and sensor data, a real time command is computed to fit into a low-cost microcontroller. One of the main difficulties in simplifying the models is fitting them into a tiny, low computing power embedded microcontroller. Most vertical take-off and landing aircraft rely on expensive and delicate gyro stabilization and/or accelerometer systems to remain stable in hovering flight [6]. The classic automatic approach [7] [8] [9] [12] cannot be used as it requires too much memory and computing power.

The next part describes the platform and the theory of the toy's operation. The third part presents the methods used to compute the tilt angle, introducing a new low-cost tilt angle piezosensor. The kinematic and dynamic models used for improving the automatic command control are shown in the fourth part. Finally, real and simulated experiments are described in the last part.

[1] Vectron® BlackHawk is a registered trademark and property of Eduscience

Fig. 1 : BlackHawk Vectron® Set

II. BLACKHAWK VECTRON SET DESCRIPTION

The Vectron® set is composed of (Fig. 1) a docking base, a controller, a power supply and the Vectron Black Hawk (Flying saucer). It has been chosen because it is tethered with a power line to the docking base, allowing us to perform as many experiments as needed without spending hours recharging batteries. It is thus lighter and cheaper than if it carried batteries.

Furthermore, the Vectron® uses a new concept in hovering flight since the entire body ring rotates. This provides a gyroscopic effect. The Thrust Vector Computer (TVC) inside the controller continuously changes the power to all three motors as they rotate, in such a way as to provide vectored control of the vehicle. For the TVC to coordinate the control pad and the Vectron®, an Infrared Reference Beam (IRB) is used (Fig. 2). The beam is projected out radially as the Vectron® rotates and it is detected by the IR Dome in front of the controller. The TVC then knows the exact position of all three motors at the moment the IRB strikes the IR Dome.

The first step in developing the autopilot mode of the aircraft is to detect the tilt angle variation and its direction.

III. DETECTING THE TILT ANGLE

To improve the hovering flight abilities, a new low-cost tilt angle sensor has been added to the aircraft (Fig. 3). Once the tilt variation and direction is known, the correct command will be computed automatically and sent to the thrust vector command to stabilize the aircraft and maintain hovering flight.

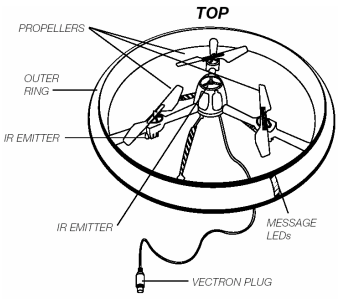

Fig. 2 : The Vectron®

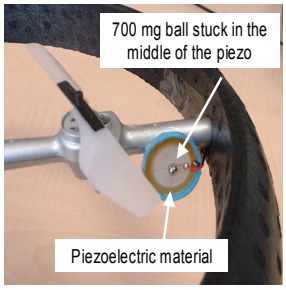

Fig. 3 : Piezosensor for tilt angle detection

A. New Low-cost Tilt Angle Sensor

The main idea for detecting the tilt angle is to combine the aircraft gyroscopic effect with a piezosensor. A low-cost piezosensor has been placed in the periphery of the flying saucer (Fig. 3). After numerous studies, the best exploitable results are given when the piezosensor is in the tangential direction of the rotation.

This sensor detects the tilt angle variation within a range of 0 and 45 degrees to the horizontal line. Moreover, a plastic shell covers the sensor to prevent air turbulence (it has been removed for the picture in Fig. 3).

The active element of the sensor is a piezoelectric material with the help of a compression disk (a capacitor with the piezoceramic material sandwiched between two electrodes). A force applied perpendicular to the disk causes a charge production and a voltage at the electrodes. The sensing element of a piezoelectric accelerometer consists of piezoceramic material and a seismic mass. One side of the piezoelectric material is connected to a rigid post at the sensor base. The so-called seismic mass is attached to the other side. When the accelerometer is subjected to vibration, a force is generated which acts on the piezoelectric element. According to Newton's Law this force is equal to the product of the acceleration and the seismic mass. Because of the piezoelectric effect, a charge output proportional to the applied force is generated. Since the seismic mass is constant the charge output signal is proportional to the acceleration of the mass. With k_p the piezoelectric constant, we have :

$$\vec{V} = k_p \vec{F} = k_p m \vec{a} \qquad (1)$$

$\vec{V}$ is the voltage delivered by the piezosensor.

Over a wide frequency range both the sensor base and the seismic mass have the same acceleration magnitude. Therefore, the sensor measures the acceleration of the test object. Within the useful operating frequency range the sensitivity is independent of frequency. The lower the seismic mass, the lower the sensitivity. After many tests, in our case, a 700 mg ball gave the best results and has been chosen for the seismic mass.

B. Detecting the Tilt Angle Direction

Once the tilt angle variation has been detected, to be able to compute the correct thrust vector command, the aircraft tilt direction must be known. This can be done by analyzing the phase difference between the IR Emitter (IRB) of the Vectron® and the remote control.

The Vectron® is divided into 24 virtual sectors (Fig. 4). The 0 Sector is linked to the IR beam of the flying aircraft.

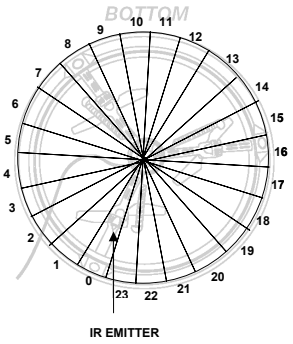

Fig. 4 : The 24 sectors of the Vectron®

Each sector is $\pi/12$ rad (15 deg) wide. The spin rotation due to the gyroscopic effect allows us to compute the sector where the Vectron® tilts, taking into account the remote control synchronization (Fig. 6). The IRB beam is projected out radially as the Vectron® rotates and is detected by the IR Dome in front of the controller. The exact position of the aircraft is known at the moment the IRB strikes the IR Dome.

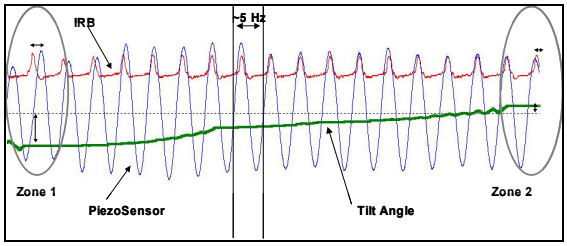

Fig. 5 : Sensor Data

Fig. 5 shows the evolution of the piezosensor signal, the IRB signal and the tilt angle as a function of time. The piezosensor signal is sinusoidal and its frequency is about 5 Hz, which corresponds to the rotation speed of the Vectron®. The amplitude of the piezosensor signal is a function of the tilt angle. If the Vectron is in hovering flight the amplitude is near zero. The more the Vectron® tilts, the more the mass in the piezosensor is accelerated, so, the more the amplitude. With a closer look at zone 1 (Fig. 5), we can notice that the aircraft tilts. The piezosensor signal is sinusoidal and there is a phase difference between the IRB signal and piezosensor signal (–90 deg) which shows the aircraft's tilt direction: Sector 18 in relation to the IR emitter (Sector 0). The focus on Zone 2 shows that, the aircraft is still tilting but that both signals (IRB and piezosensor) are now in phase: the tilt direction is now Sector 0. Obviously, all these signals are strongly filtered. Fig. 5 has been obtained with the Monitor-Simulator Software described in section V.A. With these sensor data, the tilt angle variation and direction are known; the next

part introduces the models used to compute the correct command to stabilize the aircraft.

IV. HOVERING FLIGHT COMMAND

A. Modeling the Vectron® for Stabilization Computation

The Euler angles could be defined in many different ways depending on the domain of use. The aeronautical engineering notation is used in this document.

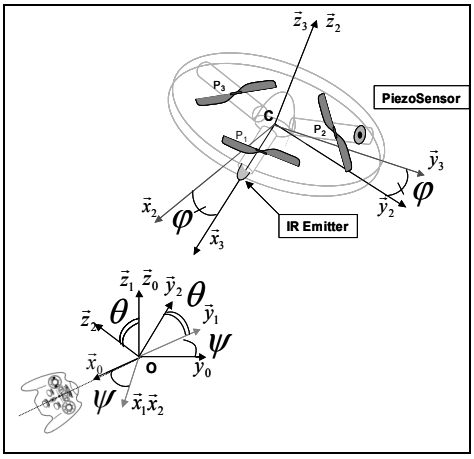

Fig. 6 : Frames of the Vectron®

The Vectron® aircraft is described by the generalized coordinates $v = (x, y, z, \psi, \theta, \varphi)$ where the position of the center of mass is **C** and (ψ, θ, φ) are the three Euler angles (yaw, pitch and roll angles) and represent the orientation of the Vectron® (Fig. 6).

The aim of this part is to find the expressions θ and φ as a function of the outputs of the tilt sensor.

1) Newton's Second Law of Motion on the Piezosensor

In the frame $\Re_3'$, the piezosensor frame, ($\overrightarrow{x'_3}$ holds the piezosensor); see Fig. 3 and Fig. 6. We have : $(\overrightarrow{x_3}C\overrightarrow{x'_3}) = 2\pi/3$, $(\overrightarrow{y_3}C\overrightarrow{y'_3}) = 2\pi/3$, $\overrightarrow{z'_3} = \overrightarrow{z_3}$, let M be the centre of inertia of the piezosensor ball. The acceleration at this point is :

$$d^2\overrightarrow{OM}/dt^2 = d^2(R.\overrightarrow{x'_3})/dt^2 \quad (2)$$

From Newton's Second Law of Motion :

$$m\vec{g} + \vec{F} = m\vec{a} \quad (3)$$

so,

$$\vec{F} = mR\begin{bmatrix} -\dot{\beta}^2 - \dot{\theta}^2\sin^2\beta - (g\sin\theta\sin\beta/R)\overrightarrow{x'_3} \\ \ddot{\beta} - \dot{\theta}^2\sin\beta\cos\beta - (g\sin\theta\cos\beta/R)\overrightarrow{y'_3} \\ \ddot{\theta} + 2\dot{\theta}\dot{\beta}\cos\beta - (g\cos\theta/R)\overrightarrow{z_3} \end{bmatrix} \quad (4)$$

From (1), the piezo voltage along the axis $\overrightarrow{y_3}$ is obtained:

$$V = kmR(\ddot{\beta} - \dot{\theta}^2\sin\beta\cos\beta - g\sin\theta\cos\beta/R) \quad (5)$$

with :

assumption 1: $\ddot{\beta} = 0$, acceleration around $\overrightarrow{z_3}$ is nil, the speed rotation Ω is constant.

assumption 2: $\dot{\theta} = 0$, the tilt angle is constant

(5) becomes :

$$V = -kmg\sin\theta\cos\beta \quad (6)$$

To synchronize the command with the voltage given by the piezosensor, we need to compute the tilt direction of the vectron®. The IR emitter (Fig. 2) on the bottom side of the engine allows us to know which propeller has to be powered to maintain hovering flight (see III.B).

With $\beta = -\varphi - 2\pi/3$, the piezosensor is on a different framework than the IRB emitter (Fig. 6). There is a phase difference equal to $2\pi/3$. We can write,

$$\begin{cases} V(t) = -kmg\sin\theta\cos(\omega t - \varphi - 2\pi/3) \\ \dot{V}(t) = -\omega kmg\sin\theta\sin(\omega t - \varphi - 2\pi/3) \end{cases} \quad (7)$$

To integrate the new thrust vector command into an embedded microcontroller, we have to work in discrete time space.

2) Discrete time space

With the assumption 2, $\dot{\theta} = 0$, θ is assumed to be constant so $\sin\theta$, we have from (7):

$$V_n = A.\cos(\omega n + \varphi) \quad (8)$$

where :

$A = -kmg\sin\theta = cst$ and $\omega = 2\pi/N$ (ω is the rotation speed and N is the number of samples per rotation).

From (8), we deduce :

$$\dot{V}_n = (V_n - V_{n-k})/k \quad (9)$$

with k being the size of the piezoelectric sensor samples stack.

And, thanks to Simpson's Formula[2], we have:

$$\dot{V}_n = -2A\sin(\omega k/2)\sin(w(n - k/2) + \varphi)/k \quad (10)$$

From (10), we deduce:

$$\theta_{n-k/2} = \arcsin\left(-\frac{\sqrt{4\sin^2(\pi k/N)V_{n-k/2}^2 + V_n^2 - 2V_nV_{n-k} + V_{n-k}^2}}{2kmg\sin(\pi k/N)}\right) \quad (11)$$

And also from (10), we find:

$$\varphi = \arctan(V_n - V_{n-k})/(-2\sin(\omega k/2)V_{n-k/2})) - 2\pi((n-k/2)/N + 1/3) \quad (12)$$

B. Newton's Second Law of Motion on the Propellers

The aim of this part is to compute the thrust vector command for each of the propellers as a function of the tilt

[2] $\cos\alpha - \cos\beta = -2\sin\left(\frac{\alpha-\beta}{2}\right)\sin\left(\frac{\alpha+\beta}{2}\right)$

angle and the direction of the aircraft. The details of propeller propulsion are complex because a propeller is like two rotating wings. The propeller used in the Vectron® has two blades. They are long and thin and a cut through the blade perpendicular to the long dimension gives an airfoil shape.

1) Aeronautical results

A wing is a good approximation for a rotor blade, so we can use the wing lift force model.

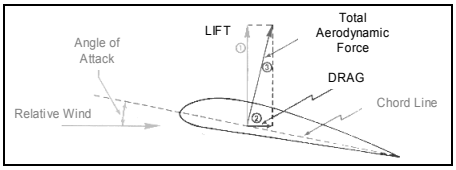

Fig. 7 : Forces acting on an airfoil shape (wing)

a) Lift Force

$$L = \rho V^2 S_{ref} C_L / 2 \quad (13)$$

with : ρ as the air density (kg.m^{-3}), V as the wing velocity (m.s^{-1}), S_{ref} as the reference area: for a helicopter, the reference area is the rotor disk area, or the area of the circle through which the rotor blades turn, C_L is the coefficient of lift: this is a dimensionless variable that changes with the speed as well as the angle of attack (typically between 0.02 and 0.05).

b) Drag Force

$$D = \rho V^2 S_{ref} C_d / 2 \quad (14)$$

The terms are the same as above except for C_d, the drag coefficient.

2) Thrust Force

In fact, we have two blades per propeller with :

$$\begin{cases} F_{blade1} + F_{blade2} = F \\ D_{blade1} + D_{blade2} = 0 \end{cases}$$

The drag forces are balancing each other. The thrust force is the sum of the two forces described above, according to Fig. 7 (total aerodynamic force). We have :

$$F = L + D = K\omega_n{}^2 \quad (15)$$

With K constant.

$$\begin{cases} F_x = 0 \\ F_y = F\sin(\alpha) = -K\omega_n^2 \sin(\alpha) \\ F_z = F\cos(\alpha) = K\omega_n^2 \cos(\alpha) \end{cases} \quad (16)$$

α is the constant tilt angle of the propeller about frame 3 of the Vectron® (Fig. 8).

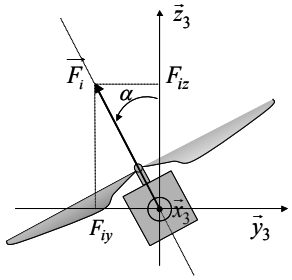

Fig. 8 : Propeller inclination

Let $M_{0\to3} = M_E(\psi,\theta,\varphi)$ be the transition matrix between frame 0 and frame 3. So, $M_{3\to0} = M_E(-\varphi,-\theta,-\psi)$. M_E is the Euler Matrix.

If we define the position of propeller 1 in frame 3, (denoted : ${}_{F3}|P_1$), we have :

${}_{F3}|P_1 = {}_{F3}[R \quad 0 \quad 0]^T$ we could write

$${}_{F0}|P_1 = M_{3\to0} \times \begin{vmatrix} R \\ 0 \\ 0 \end{vmatrix}_{F3} = \begin{vmatrix} c\varphi_3 c\psi_3 - c\theta_3 s\varphi_3 c\psi_3 \\ -s\varphi_3 c\psi_3 - c\theta_3 s\psi_3 c\varphi_3 \\ s\theta_3 s\psi_3 \end{vmatrix}_{F0} \quad (17)$$

If we write the moment of the propeller in frame 3, ${}_{F3}|\mathrm{M(O,F\to P_1)} = \overrightarrow{OP_1} \wedge \vec{F} = \begin{vmatrix} R \\ 0 \\ 0 \end{vmatrix}_{F3} \wedge \begin{vmatrix} 0 \\ -F_y \\ F_z \end{vmatrix}_{F3} = \begin{vmatrix} 0 \\ -RF_z \\ -RF_y \end{vmatrix}_{F3}$ so in the reference frame 0, we have :

$${}_{F0}|\mathrm{M(O,F\to P_1)} = M_{3\to0} \times {}_{F3}|\mathrm{M(O,F\to P_1)}$$

So, for the whole rotorcraft, with the 3 propellers we obtain:

$${}_{F0}|\mathrm{M(O,(F\to C)} = \sum_{i=1}^{3} {}_{F0}|\mathrm{M(O,(F\to P_i)} \quad (18)$$

with C as the center of mass of the Vectron® (Fig. 7):

We find out that the angular momentum is:

$$\vec{\sigma}_{O_S,solid/R} = [I]_{O_S,solid} \times \left|\overrightarrow{\Omega_{S/R}}\right| + m_s \overrightarrow{O_sG_s} \wedge \vec{V}_{O_S/R} \quad (19)$$

The dynamic momentum theorem gives :

$\forall\Sigma$, $\forall t$, $\forall P$:

$$\vec{\delta}_{P,\Sigma/R_g} = \mathrm{M(P,ext\to\Sigma)} \quad (20)$$

with $[I]_{O_S,solid}$ as the inertia operator. The angular momentum theorem which links the angular momentum and the dynamic momentum gives:

$$\vec{\delta}_{P,\Sigma/R} = [\frac{d}{dt}\vec{\sigma}_{P,\Sigma/R}]_R + m_\Sigma \vec{V}_{P/R} \wedge \vec{V}_{G_\Sigma/R} \quad (21)$$

In our case, the Vectron® is symmetrical (x and y) and $\dot{\varphi} >> \dot{\theta}$ and $\dot{\varphi} >> \dot{\psi}$, so, we finally obtain:

$$\vec{\delta}_{O,S/R_0} = C\ddot{\varphi}.\vec{z}_3 + C\dot{\varphi}\left[d\vec{z}_3/dt\right]_{R0} = \sum_{i=1}^{3} \mathrm{M}_{\mathrm{P_i}}(0,\mathrm{F}\to S) \text{ and}$$

$$\sum_{i=1}^{3} \mathrm{M}_{\mathrm{P_i}}(0,\mathrm{F}\to S) = C\Big|_{R_0}\begin{vmatrix} \ddot{\varphi}s\psi s\theta + \dot{\varphi}\dot{\psi}c\psi s\theta + \dot{\varphi}\dot{\theta}s\psi c\theta \\ \dot{\varphi}\dot{\psi}s\psi s\theta - \ddot{\varphi}c\psi s\theta - \dot{\varphi}\dot{\theta}c\psi c\theta \\ \ddot{\varphi}c\theta - \dot{\varphi}\dot{\theta}s\theta \end{vmatrix} \quad (22)$$

3) Thrust Vector Command Generation

From (18) and (22) we have:

$$\begin{cases} \sum_{i=1}^{3} F_{iz}\, s(\varphi + 2\pi(i-1)/3) = C\dot{\varphi}\dot{\psi}s\theta / R \\ \sum_{i=1}^{3} F_{iz}\, c(\varphi + 2\pi(i-1)/3) = C\dot{\varphi}\dot{\theta} / R \\ \sum_{i=1}^{3} F_{iy} = C\ddot{\varphi} / R \end{cases} \quad (23)$$

With (16) and (23), we can write :

$$K\cos(\alpha)\times[A]\left[\omega_n^2\right] = C\begin{bmatrix} \ddot{\varphi}/R \\ \dot{\varphi}\dot{\theta}/R \\ \dot{\varphi}\dot{\psi}\sin(\theta)/R \end{bmatrix} \quad (24)$$

(α tilt angle of the propellers of the Vectron®) with $A = \begin{bmatrix} -\tan(\alpha) & -\tan(\alpha) & -\tan(\alpha) \\ \cos(\varphi) & \cos(\varphi+2\pi/3) & \cos(\varphi-2\pi/3) \\ \sin(\varphi) & \sin(\varphi+2\pi/3) & \sin(\varphi-2\pi/3) \end{bmatrix}$ and $\left[\omega_n\right] = \left[\omega_1 \quad \omega_2 \quad \omega_3\right]^T$ propeller speed rotation vector, so we could write for n=1,2,3:

$$\left[\omega_n^2\right] = (C/RK\cos(\alpha))\left[A^{-1}\right]\begin{bmatrix} \ddot{\varphi} \\ \dot{\varphi}\dot{\theta} \\ \dot{\varphi}\dot{\psi}\sin(\theta) \end{bmatrix} \quad (25)$$

Equation (25) is the Thrust Vector Command for each propeller. It is the rotation speed of each propeller as a function of the tilt and direction angle of the Vectron®.

To validate this command, we have implemented it into a Monitor-Simulator Software linked with the ODE library for more realistic results (3D and physical engines).

V. EXPERIMENTS

We have developed a tool to enable us to trace and monitor the different sensors' output. The software consists of 3 parts: the monitor, the simulator and the ODE[3] library.

3 ODE is an open source, high performance library for simulating rigid body dynamics http://ode.org/

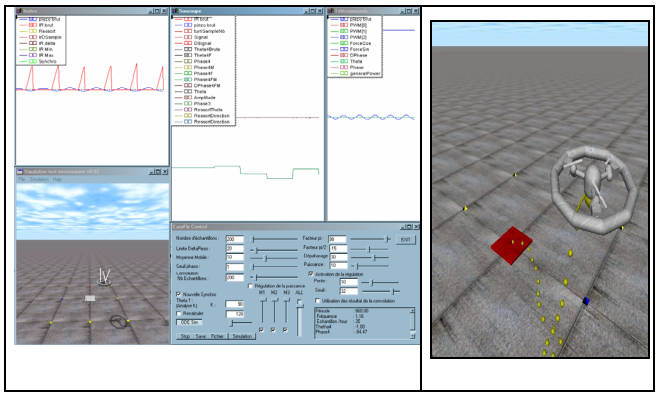

Fig. 9 : The Monitor-Simulator Software

A. The Monitor-Simulator Software

One of the goals of this software is to plot the data sensor outputs to confirm the theoretical results (Fig. 9). The piezosensor signal is sinusoidal and its amplitude varies according to the tilt angle. We also notice the phase difference between the IRB and the piezosensor signal. This is used to compute the tilt angle as explained in III.B, Fig. 5.

1) Plots of the theoretical curves

With the theoretical models calculated, we can also plot the theoretical results and check that the practical results match the theoretical ones. The theoretical results are used to compute the thrust vector command to maintain the hovering flight of the Vectron® according to (25) in the ODE simulation.

2) ODE Simulations

ODE is an open source, high performance library for simulating rigid body dynamics. It is fully featured, stable, mature and platform-independent, with an easy to use C/C++ API. It has advanced joint types and integrated collision detection with friction. ODE is useful for simulating vehicles, objects in virtual reality environments and virtual creatures. It is currently used in many computer games, 3D authoring tools and simulation tools.

A 3D model Vectron® has been designed to be simulated in ODE. ODE is able to simulate the dynamic and kinetic behavior of the Vectron®. A model including the power cable, the 3 propellers and the piezosensor has been created. We have also used the output signals of ODE to compute the correct command to maintain the hovering flight according to (25).

It is also possible to trace all the different signals in the Monitor-Simulator Software, the unfiltered piezosensor signal, the phase difference and the tilt angle of the Vectron®.

B. The experiments

The good results obtained with the simulation encouraged us to test the new thrust vector command with a real prototype. The commands were loaded into the microcontroller and Fig. 10 shows the Vectron® hovering autonomously and the embedded board with the microcontroller.

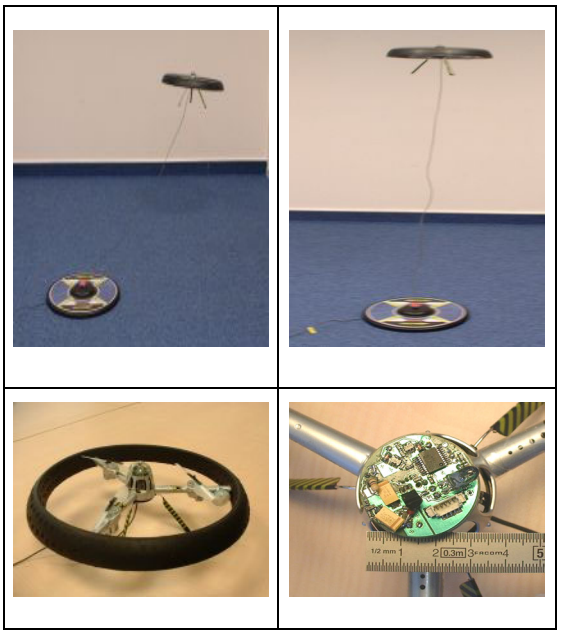

Fig. 10 : Hovering flight command of the Vectron® and PCB

The takeoff phase is automatic and raises the aircraft above the ground base with 75% thrust power, allowing the autopilot mode, when engaged, to control the thrust power between 50% and 100% to maintain hovering flight.

After takeoff, the autopilot mode is engaged and the aircraft can remain in a 50 cm radius circle to the height of approximately 1.7 m, but from time to time the power cable stretches itself, producing a non-elastic shock the control cannot handle. The sensor used is able to detect if the aircraft is stable but not if it has drifted from above the ground base. One idea, without adding any other sensors, lies in the integration of the variation of the tilt angle as a function of time and in order to deduce if it is drifting or not. In this case, a correction command to bring the aircraft back above the ground base is computed; this solution is currently in progress.

However, the aim of this research has been reached. The Vectron® now has an autopilot mode allowing it to avoid crashes and maintain hovering flight using a low-cost solution.

VI. CONCLUSION

We have presented an original stabilization control command for holding a 3-rotor gyro-helicopter toy in hovering flight. A low-cost tilt angle sensor has been presented. This simple sensor is sufficient and no other sensors are needed. Simplified kinetic and dynamic models of the Vectron® with the tilt angle sensor data are used to compute the new thrust vector command. The command is executed in a low computing power embedded microcontroller, not in a remote Pentium PC. The hovering flight command has been validated twice, firstly by software simulation and then with real experiments. The toy is now easier to handle for beginner pilots. The autopilot mode stabilizes the aircraft, applying the command described each time the pilot releases the remote control pad and thus avoiding fatal crashes. Future works will concern adding a new mode to the Vectron®: automatically avoiding obstacles but still using cheap sensors.

ACKNOWLEDGMENT

This work could not have been done without the WANY Robotics R&D engineering team : Laurent Tremel, Loïc Anguera, Christophe Marin, Charles-Marie DeGraeve, Christophe Sevilla, Raphaël Bini and Gemma Griffiths. The authors wish to thank them for their valuable work.

REFERENCES

[1] Etkin B., "Dynamics of Flight", John Wiley and Sons, Inc., New York, 1959.

[2] Duhaut D., Lavarec E., Mateo E., Montillet D., "Robotics for entertainment" in french "Robotique de divertissement" in P. Dauchez "Applications non Manufacturières de la Robotique", Hermès Science Publications, Collection IC2/Systèmes automatisés, chap. 7, pp. 239-278, 2000.

[3] Barnes W. McCormick, "Aerodynamics Aeronautics and Flight Mechanics", John Wiley and Sons, Inc., New York, 1995.

[4] BlackHawk Vectron® HomePage http://www.vectronblackhawk.com/mainpage.htm

[5] Hamel T., Mahony R., Lozano R., Ostrowski J., "Dynamic Modelling and Configuration and Stabilization for an X4-Flyer", In International Federation of Automatic.Control Symposium, IFAC 2002, Barcelona, Spain.

[6] Hamel T., Mahony R., "Visual Servoing of an Under-Actuated Dynamic Rigid-Body System: An Image-Based Approach", IEEE Trans. on Robotics and Automation, vol. 18, No 2, pp. 187-198, April 2002.

[7] Castillo P., Dzul A, Lozano R.," Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft ", IEEE Transactions on Control Systems Technology, vol. 12, Issue 4, pp. 510-516, July 2004.

[8] Lavarec E. "3D Motion Estimation Using a Camera and Proprioceptive Sensors", PhD Thesis, LIRMM, Laboratory of Computer Science, Robotics, and Microelectronics, December 2001.

[9] Zhang Z., Brandt R., "Robust Hovering Control of a PVTOL Aircraft", IEE Transactionc on control systels technology, vol. 7, No. 3, May 1999.

[10] Pierrot F., Marquet F., Company O., Gil T., " H4 parallel robot : Modeling, design and preliminary experiments", Proc ICRA'01, pp. 3256-3261, 2001.

[11] Rongier P., Lucidarme P., "A sizing method for a multi-robot system", Proc. IEEE/IROS'2001 International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, pp. 387-392, , 2001.

[12] Fantoni I., Zavala A., Lozano R., "Global stabilization of a PVTOL aircraft with bounded thrust", Conf. On Decision and Control, pp. 4462-4467, Las Vegas, December 2002.

[13] Mahony R., Hamel T., "Adaptive Compensation of Aerodynamic Effects during Takeoff and Landing Manoeuvres for a Scale Model Autonomous Helicopter", European Journal of Control, pp.1-15, 2001.