Kinematic and Dynamic Modeling and Control of a 3-Rotor Aircraft
Philippe Rongier, Erwann Lavarec, François Pierrot

To cite this version:

HAL Id: lirmm-00106454
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00106454
Submitted on 16 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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® [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 manoeuvrable 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
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:

$$\vec{V} = k_p \vec{F} = k_p m \ddot{a}$$

(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.

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. 4: The 24 sectors of the Vectron®](image)

![Fig. 5: Sensor Data](image)

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.

The Vectron® aircraft is described by the generalized coordinates \( v = (x, y, z, \psi, \theta, \phi) \) where the position of the center of mass is \( C \) and \((\psi, \theta, \phi)\) 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 \( \theta \) and \( \phi \) as a function of the outputs of the tilt sensor.

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

In the frame \( R_x \), the piezosensor frame, \((x', z')\) holds the piezosensor; see Fig. 3 and Fig. 6. We have :

\[
(x' C) = 2\pi/3, \quad (y' z') = 2\pi/3, \quad z'_3 = z_3,
\]

let \( M \) be the centre of inertia of the piezosensor ball. The acceleration at this point is :

\[
d^2\overrightarrow{OM} / dt^2 = d^2(R\overrightarrow{x'_3}) / dt^2
\]

From Newton’s Second Law of Motion :

\[
mg + \overrightarrow{F} = ma
\]

so,

\[
\overrightarrow{F} = mR\left[ -\ddot{\theta}^2 + \ddot{\phi}^2 \sin^2 \beta - (g \sin \theta \sin \beta / R) x'_3 \right] + mR \dot{\phi} \dot{\phi} \cos \beta - (g \sin \theta \cos \beta / R) y'_3
\]

From (1), the piezo voltage along the axis \( y'_3 \) is obtained:

\[
V = kmR(\dot{\theta} - \dot{\phi})^2 \sin \beta \cos \beta - g \sin \theta \cos \beta / R
\]

with :

assumption 1: \( \dot{\beta} = 0 \), acceleration around \( z_3 \) is nil, the speed rotation \( \Omega \) is constant.

assumption 2: \( \theta = 0 \), the tilt angle is constant

(5) becomes :

\[
V = -kmg \sin \theta \cos \beta
\]

To synchronize the command with the voltage given by the piezosensor, we need to compute the tilt direction of the piezosensor. The IR emitter (Fig. 6) 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,

\[
\begin{align*}
V(t) &= -kmg \sin \theta \cos(\alpha \omega - 2\pi / 3) \\
\dot{V}(t) &= -km \cos \theta \sin(\alpha \omega - 2\pi / 3)
\end{align*}
\]

(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, \( \dot{\theta} = 0 \), \( \theta \) is assumed to be constant so \( \sin \theta \), we have from (7):

\[
V_n = A \cos(\omega m + \varphi)
\]

(8)

where :

\[
A = -km \sin \theta = \text{cst} \quad \text{and} \quad \omega = 2\pi / N \quad (\omega \text{ is the rotation speed and } N \text{ is the number of samples per rotation}).
\]

From (8), we deduce :

\[
\dot{V}_n = (V_n - V_{n-k}) / k
\]

(9)

with \( k \) being the size of the piezoelectric sensor samples stack.

And, thanks to Simpson’s Formula\(^2\), we have:

\[
\dot{V}_n = -2A \sin(\omega k / 2) \sin(n(n - k / 2) + \varphi) / k
\]

(10)

From (10), we deduce:

\[
\theta_{n+1/2} = \arcsin\left(\frac{\sqrt{4\sin^2(\omega k / 2) + 2V_n^2 - 2V_{n-k} + V_0^2}}{2km \sin(\omega k / N)}\right)
\]

(11)

And also from (10), we find:

\[
\varphi = \arctan(\alpha / V_{n-k} / V_{n-k+1}) - 2\pi(n-k / 2) / (N+1/3)
\]

(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

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

**a) Lift Force**

\[ L = \rho V^2 S_{ref} C_L / 2 \]  

with \( \rho \) 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**

\[ D = \rho V^2 S_{ref} C_D / 2 \]  

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:

\[
\begin{align*}
F_{blade1} + F_{blade2} &= F \\
D_{blade1} + D_{blade2} &= 0
\end{align*}
\]

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:

\[ F = L + D = K \omega_n^2 \]  

With \( K \) constant.

\[
\begin{align*}
F_x &= 0 \\
F_y &= F \sin(\alpha) = -K \omega_n^2 \sin(\alpha) \\
F_z &= F \cos(\alpha) = K \omega_n^2 \cos(\alpha)
\end{align*}
\]

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

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

If we define the position of propeller 1 in frame 3, (denoted : \( F_3^1 \)), we have:

\[ F_3^1 = [F_3] R 0 0 0 1 \]

we could write

\[ F_0 F_3^1 = M_{3\rightarrow 0} \times [F_3] M(O,F \rightarrow P_1) \]

If we write the moment of the propeller in frame 3, \( \vec{M}(O,F \rightarrow P_1) = \vec{d} \times \vec{F} = 0 \) in the reference frame 0, we have:

\[ F_0 \vec{M}(O,F 

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

\[ F_0 \vec{M}(O,F \rightarrow C) = \sum_{i=1}^{3} F_0 \vec{M}(O,F \rightarrow P_i) \]  

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

We find out that the angular momentum is:

\[ \vec{\sigma}_{\Omega_{solid},R} = \int [\Omega_{solid}, R] \times \vec{\Omega}_{3/R} + m \vec{\Omega}_3 \times \vec{\Omega}_{3/R} \]  

The dynamic momentum theorem gives:

\[ \forall \Sigma, \forall t, \forall P : \]

\[ \overset{\rightarrow}{\sigma}_{P, \Sigma : R} = M(P, ext \rightarrow \Sigma) \]  

with \([\Omega]_{\Omega_{solid}, R} \) as the inertia operator. The angular momentum theorem which links the angular momentum and the dynamic momentum gives:

\[ \overset{\rightarrow}{\sigma}_{P, \Sigma : R} = \int \frac{d}{dt} \sigma_{P, \Sigma : R} + m \vec{\sigma}_{P, /R} \times \vec{\Omega}_{G_z / R} \]  

In our case, the Vectron® is symmetrical (x and y) and \( \dot{\phi} \gg \dot{\theta} \) and \( \dot{\phi} \gg \dot{\psi} \), so, we finally obtain:
\[ \vec{\delta}_{\text{O,S}/\kappa} = C \vec{\phi} \hat{z}_3 + C \vec{\phi}[\vec{e}_3 \cdot d\vec{E}]_{\kappa_0} = \sum_{j=1}^{3} M_{\vec{F}}(0, F \rightarrow S) \] and

\[ \sum_{j=1}^{3} M_{\vec{F}}(0, F \rightarrow S) = C \begin{bmatrix}
\dot{\phi} \\
\dot{\phi} \\
\dot{\phi}
\end{bmatrix} - \begin{bmatrix}
\phi \\
\phi \\
\phi
\end{bmatrix} R.
\]

3) **Thrust Vector Command Generation**

From (18) and (22) we have:

\[ \sum_{j=1}^{3} F_{ix} = C \dot{\phi} / R \]

\[ \sum_{j=1}^{3} F_{iy} = C \dot{\phi} / R \]

\[ \sum_{j=1}^{3} F_{iz} = C \dot{\phi} / R \]

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

\[ K \cos(\alpha) \times [A][\omega_n^2] = C \begin{bmatrix}
\dot{\phi} / R \\
\dot{\phi} / R \\
\dot{\phi} / R
\end{bmatrix} \]

\[ A = \begin{bmatrix}
\cos(\phi) & \cos(\phi + 2\pi / 3) & \cos(\phi - 2\pi / 3) \\
\sin(\phi) & \sin(\phi + 2\pi / 3) & \sin(\phi - 2\pi / 3)
\end{bmatrix} \]

\[ [\omega_n] = [\omega_1 \omega_2 \omega_3]^T \]

The propeller speed rotation vector, so we could write for n=1,2,3:

\[ [\omega_n^2] = (C / RK \cos(\alpha))[A]^{-1} \begin{bmatrix}
\dot{\phi} \\
\dot{\phi} \\
\dot{\phi} \sin(\theta)
\end{bmatrix} \]

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 library.

---

3 ODE is an open source, high performance library for simulating rigid body dynamics.

http://ode.org/
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, Raphael Bini and Gemma Griffiths. The authors wish to thank them for their valuable work.

REFERENCES