

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Intelligent Flight Control of an Autonomous Quadrotor

Syed Ali Raza and Wail Gueaieb
*University of Ottawa,
Canada*

1. Introduction

This chapter describes the different steps of designing, building, simulating, and testing an intelligent flight control module for an increasingly popular unmanned aerial vehicle (UAV), known as a quadrotor. It presents an in-depth view of the modeling of the kinematics, dynamics, and control of such an interesting UAV. A quadrotor offers a challenging control problem due to its highly unstable nature. An effective control methodology is therefore needed for such a unique airborne vehicle.

The chapter starts with a brief overview on the quadrotor's background and its applications, in light of its advantages. Comparisons with other UAVs are made to emphasize the versatile capabilities of this special design. For a better understanding of the vehicle's behavior, the quadrotor's kinematics and dynamics are then detailed. This yields the equations of motion, which are used later as a guideline for developing the proposed intelligent flight control scheme.

In this chapter, fuzzy logic is adopted for building the flight controller of the quadrotor. It has been witnessed that fuzzy logic control offers several advantages over certain types of conventional control methods, specifically in dealing with highly nonlinear systems and modeling uncertainties. Two types of fuzzy inference engines are employed in the design of the flight controller, each of which is explained and evaluated.

For testing the designed intelligent flight controller, a simulation environment was first developed. The simulations were made as realistic as possible by incorporating environmental disturbances such as wind gust and the ever-present sensor noise. The proposed controller was then tested on a real test-bed built specifically for this project. Both the simulator and the real quadrotor were later used for conducting different attitude stabilization experiments to evaluate the performance of the proposed control strategy. The controller's performance was also benchmarked against conventional control techniques such as input-output linearization, backstepping and sliding mode control strategies. Conclusions were then drawn based on the conducted experiments and their results.

1.1 Quadrotor background

Louis Bréguet and Jacques Bréguet, two brothers working under the guidance of Professor Charles Richet, were the first to construct a quadrotor, which they named Bréguet Richet Gyroplane No. 1 Breguet-Richet-1907. The first flight demonstration of Gyroplane No. 1

Source: Motion Control, Book edited by: Federico Casolo,
ISBN 978-953-7619-55-8, pp. 580, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

with no control surfaces was achieved on 29 September 1907. Figure 1 shows the huge quadrotor with double layered propellers being prepared for its first manned flight.

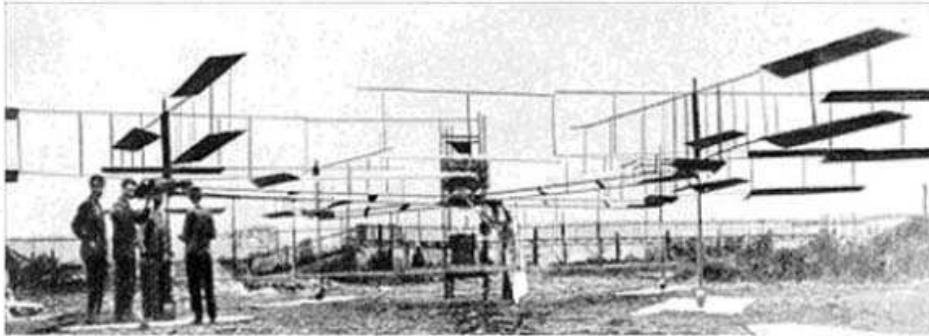


Fig. 1. Bréguet Richet Gyroplane No. 1 Rumerman (2002).

Later, two additional designs were developed and experimental flights were conducted. The first, by Georges de Bothezat and Ivan Jerome in 1922, had six-bladed rotors placed at each end of an X-shaped truss structure, as shown in Figure 2.



Fig. 2. Quadrotor designed by George De Bothezat, February 21, 1923 Rumerman (2002).

The second, shown in Figure 3, was built by Étienne Cehmichen in 1924, and set distance records, including achieving the first kilometer long helicopter flight.

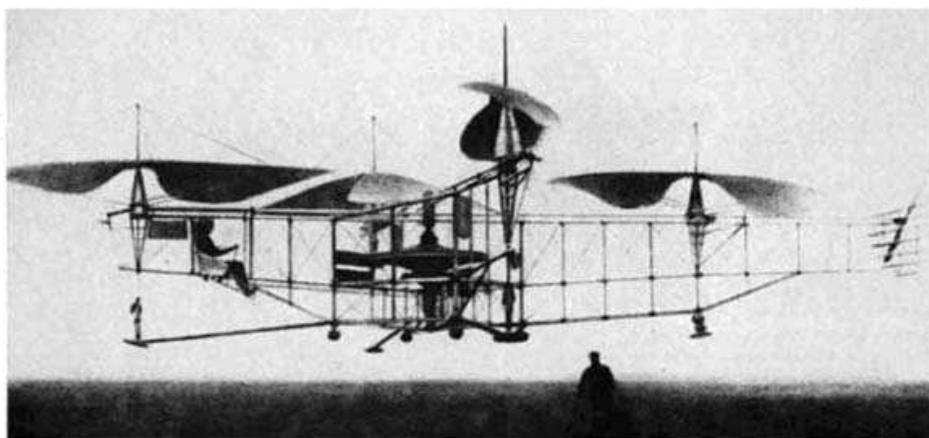


Fig. 3. Cehmichen quadrotor designed in 1924 Rumerman (2002).

At present, apart from military endeavours, UAVs are also being employed in various commercial and industrial applications. In particular, these include the use of unmanned helicopters for crop dusting or precision farming Sugiura et al. (2003), and microwave

autonomous copter systems for geological remote sensing Archer et al. (2004). STARMAC Waslander et al. (2005) is a multi-agent autonomous rotorcraft, which has potential in security-related tasks, such as remote inspections and surveillance. The commercially available quadrotor kit called DraganFlyer Inc. (2008) has become a popular choice for aerial mapping and cinematography.

UAVs are subdivided into two general categories, fixed wing UAVs and rotary wing UAVs. Rotary winged crafts are superior to their fixed wing counterparts in terms of achieving higher degree of freedom, low speed flying, stationary flights, and for indoor usage. A quadrotor, as depicted in Figure 4, is a rotary wing UAV, consisting of four rotors located at the ends of a cross structure. By varying the speeds of each rotor, the flight of the quadrotor is controlled. Quadrotor vehicles possess certain essential characteristics, which highlight their potential for use in search and rescue applications. Characteristics that provide a clear advantage over other flying UAVs include their Vertical Take Off and Landing (VTOL) and hovering capability, as well as their ability to make slow precise movements. There are also definite advantages to having a four rotor based propulsion system, such as a higher payload capacity, and impressive maneuverability, particularly in traversing through an environment with many obstacles, or landing in small areas.

As illustrated by the conceptual diagram in Figure 4, the quadrotor attitude is controlled by varying the rotation speed of each motor. The front rotor (M_f) and back rotor (M_b) pair rotates in a clockwise direction, while the right rotor (M_r) and left rotor (M_l) pair rotates in a counter-clockwise direction. This configuration is devised in order to balance the drag created by each of the spinning rotor pairs. Figure 5 shows the basic four maneuvers that can be accomplished by changing the speeds of the four rotors. By changing the relative speed of the right and left rotors, the roll angle of the quadrotor is controlled. Similarly, the pitch angle is controlled by varying the relative speeds of the front and back rotors, and the yaw angle by varying the speeds of clockwise rotating pair and counter-clockwise rotating pair. Increasing or decreasing the speeds of all four rotors simultaneously controls the collective thrust generated by the robot. A roll motion can be achieved while hovering by increasing the speed of the right rotor, while decreasing the speed of the left rotor by the same amount. Hence, the overall thrust is kept constant.

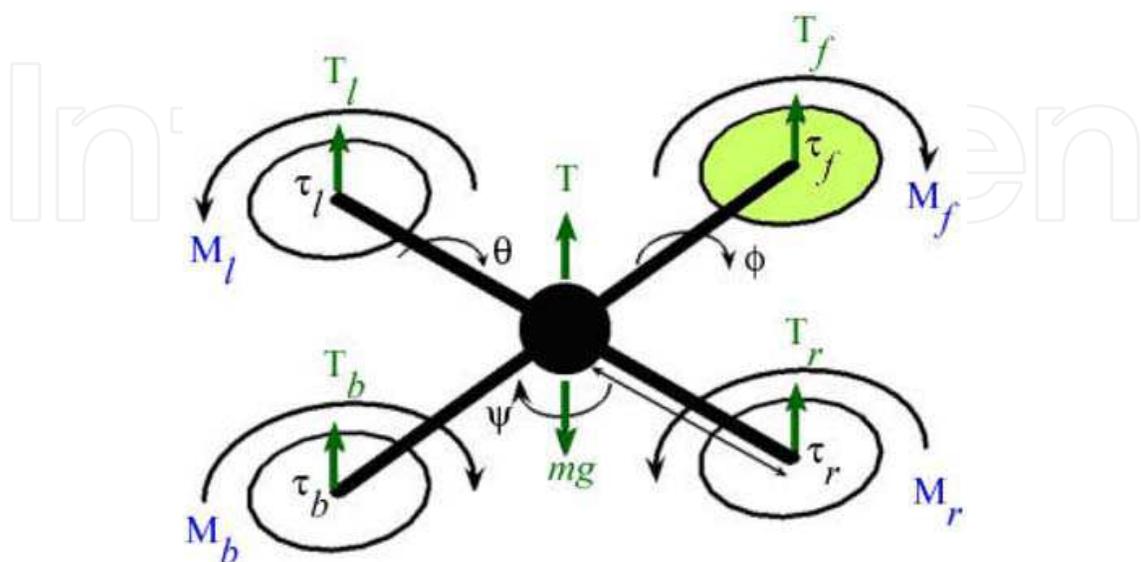


Fig. 4. Conceptual diagram of a quadrotor.

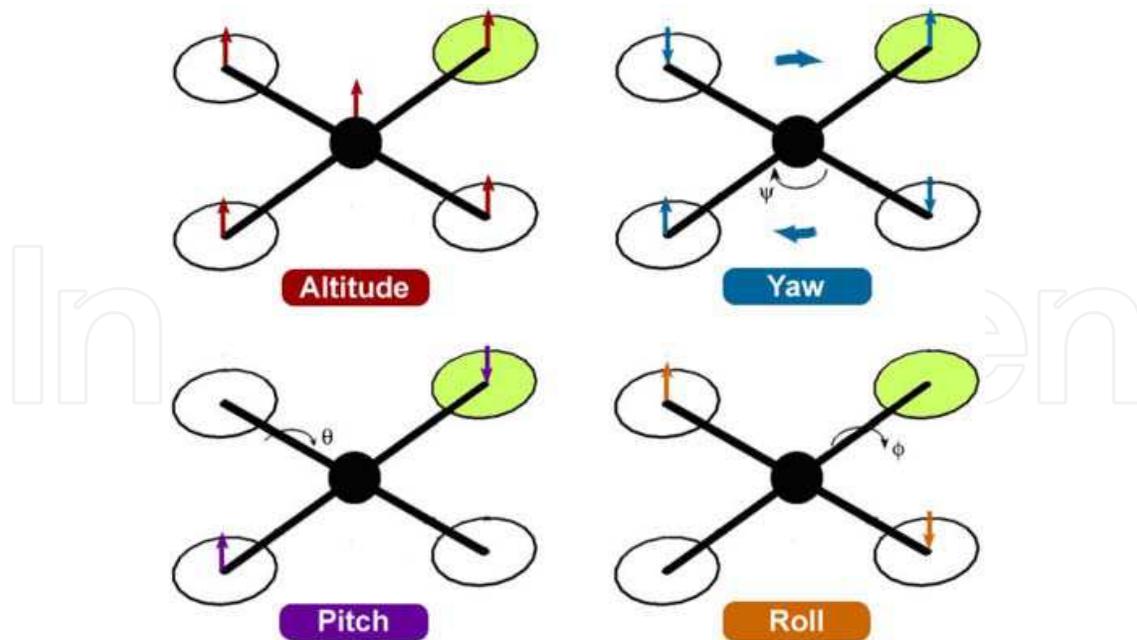


Fig. 5. Quadrotor dynamics.

In the past few years, much research has already been conducted on the modeling and control of quadrotors. Many control techniques, as summarized in Table 1, are proposed in the literature, however, excluding STARMAC, their primary focus is mostly for indoor flight control and therefore do not account for uncertainties and external disturbances. Lyapunov stability theory is used for stabilization and control of the quadrotor in Bouabdallah et al. (2004a) and Dzul et al. (2004). Conventional PD² feedback and PID structures are used for simpler implementation of control laws, and comparison with LQR based optimal control theory is presented in Tayebi and McGillvray (2006) and Bouabdallah et al. (2004b). Backstepping control is also proposed with the drawback of higher computational loads in Guenard et al. (2005). Visual feedback is applied in many cases, using onboard or offboard cameras for pose estimation by Altug et al. (2002) and Guenard et al. (2008). Fuzzy logic control techniques have also been proposed Coza and Macnab (2006), along with neural networks Tarbouchi et al. (2004) and reinforcement learning Waslander et al. (2005).

Many quadrotor test-beds have been constructed in different research projects, where simulators are also developed for testing the control laws beforehand. In Kivrak (2006), LQR is used for attitude stabilization of a commercially available Draganflyer Vti quadrotor model in MATLAB Simulink. In another project, the modeling, design, and control of a Miniature Flying Robot (MFR), named OS4 was accomplished Bouabdallah (2007), where a mathematical model was developed for the simulation and control of a mini quadrotor using linear and nonlinear control methods.

2. Quadrotor's kinematics and dynamics

Mathematical modelling provides a description of the behaviour of a system. The flight behaviour of a quadrotor is determined by the speeds of each of the four motors, as they vary in concert, or in opposition with each other. Hence, based on its inputs, a mathematical representation of the system can be used to predict the position and orientation of the quadrotor. The same can further be used to develop a control strategy, whereby manipulating the speeds of individual motors results in achieving the desired motion.

Project	Control Technique	Picture
STARMAC, Stanford University, 2005 Waslander et al. (2005)	Reinforcement Lrn.	
OS4, EPFL, December 2006 Bouabdallah (2007)	Backstepping	
Pennsylvania State University, Hanford, 2005 Hanford (2005)	PI	
Helio-copter, Brigham Young University, Fowers, 2008 Fowers (2008)	Visual feedback	
HMX-4, Pennsylvania State University, 2002 Altug et al. (2002)	Feedback Lin.	
Quad-Rotor UAV, University of British Columbia Chen and Huzmezan (2003)	MBPC and H_{∞}	
Quad-Rotor Flying Robot, Universiti Teknologi Malaysia Weng and Shukri (2006)	PID	

Table 1. Quadrotor flight control techniques used in various projects.

To derive the full mathematical model of the quadrotor, we need to define its kinematics and dynamics first. The kinematic equations provide a relation between the vehicle's position and velocity, whereas the dynamic model defines the relation governing the applied forces and the resulting accelerations.

2.1 Reference frames

Before getting into the equations of kinematics and dynamics of the quadrotor, it is necessary to specify the adopted coordinate systems and frames of reference, as well as how transformations between the different coordinate systems are carried out.

The use of different coordinate frames is essential for identifying the location and attitude of the quadrotor in six degrees of freedom (6 DOF). For example, in order to evaluate the equations of motion, a coordinate frame attached to the quadrotor is required. However, the forces and moments acting on the quadrotor, along with the inertial measurement unit (IMU) sensor values, are evaluated with reference to the body frame. Finally, the position and speed of the quadrotor are evaluated using GPS measurements with respect to an inertial frame located at the base station.

Thus, three main frames of reference are adopted, as shown in Figure 6:

1. The inertial frame, $\mathcal{F}_i = (\vec{x}_i, \vec{y}_i, \vec{z}_i)$, is an earth-fixed coordinate system with the origin located on the ground, for example, at the base station. By convention, the x-axis points towards the north, the y-axis points towards the east, and the z-axis points towards the center of the earth.
2. The body frame $\mathcal{F}_b = (\vec{x}_b, \vec{y}_b, \vec{z}_b)$, with its origin located at the center of gravity (COG) of the quadrotor, and its axes aligned with the quadrotor structure such that the x-axis \vec{x}_b is along the arm with front motor, the y-axis \vec{y}_b is along the arm with right motor, and the z-axis $\vec{z}_b = \vec{x}_b \times \vec{y}_b$, where 'x' denotes the cross product.
3. The vehicle frame, $\mathcal{F}_v = (\vec{x}_v, \vec{y}_v, \vec{z}_v)$, is the inertial frame with the origin located at the COG of the quadrotor. The vehicle frame has two variations, \mathcal{F}_ϕ and \mathcal{F}_θ . \mathcal{F}_ϕ is the vehicle frame, \mathcal{F}_v , rotated about its z-axis \vec{z}_v by an angle ψ so that \vec{x}_v and \vec{y}_v are aligned with \vec{x}_b and \vec{y}_b , respectively. \mathcal{F}_θ is frame \mathcal{F}_ϕ rotated about its y-axis, \vec{y}_ϕ , by a pitching angle, θ , such that \vec{x}_ϕ and \vec{z}_ϕ are aligned with \vec{x}_b and \vec{z}_b , respectively.

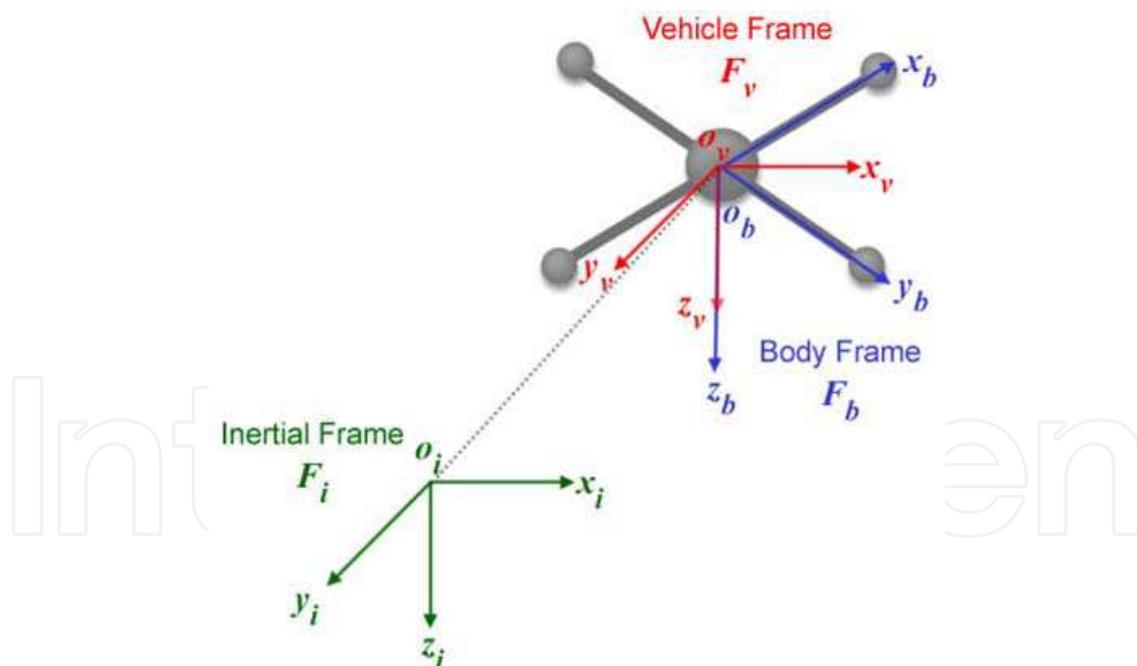


Fig. 6. The inertial, body and vehicle frames of reference.

Translation and rotation matrices are used to transform one coordinate reference frame into another desired frame of reference. For example, the transformation from \mathcal{F}_i to \mathcal{F}_v provides the displacement vector from the origin of the inertial frame to the center of gravity (COG) of the quadrotor. Also, the transformation from \mathcal{F}_v to \mathcal{F}_b is rotational in nature, therefore yielding the roll, pitch and yaw angles.

2.2 Quadrotor's kinematics

Let $P_{\mathcal{F}}^T = [p_x, p_y, -p_z]$ and $\Omega_{\mathcal{F}}^T = [\phi, \theta, \psi]$ denote the quadrotor's position and orientation within a given frame \mathcal{F} . The relation between the quadrotor's speed in the three predefined frames is expressed as

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ -\dot{p}_z \end{bmatrix}_{\mathcal{F}_i} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix}_{\mathcal{F}_v} = [R_{\mathcal{F}_v}^{\mathcal{F}_b}]^T \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix}_{\mathcal{F}_b} \quad (1)$$

Where $[R_{\mathcal{F}_v}^{\mathcal{F}_b}]^T \in \mathbb{R}^{3 \times 3}$ is the rotation matrix that maps frame \mathcal{F}_b to frame \mathcal{F}_v and is defined by

$$[R_{\mathcal{F}_v}^{\mathcal{F}_b}]^T = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

with $s\theta = \sin\theta$ and $c\theta = \cos\theta$. The same notation applies for $s\phi$, $c\phi$, $s\psi$, and $c\psi$.

The rotational motion relationship can therefore be derived using the appropriate state variables, such as the vehicle frame angles (ϕ , θ , and ψ) and the body frame angular rate ($\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$). However, in order to do so, these variables need to be brought into one common frame of reference. Using rotation matrices to transform vehicle frames \mathcal{F}_ϕ , \mathcal{F}_θ , and \mathcal{F}_ψ into the body frame of reference \mathcal{F}_b , we get

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{\mathcal{F}_b} = R_{\mathcal{F}_\phi}^{\mathcal{F}_b}(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{\mathcal{F}_\theta}^{\mathcal{F}_b}(\theta) R_{\mathcal{F}_\phi}^{\mathcal{F}_\theta}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{\mathcal{F}_\psi}^{\mathcal{F}_b}(\psi) R_{\mathcal{F}_\theta}^{\mathcal{F}_\psi}(\theta) R_{\mathcal{F}_\phi}^{\mathcal{F}_\psi}(\phi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

where $R_{\mathcal{F}_\theta}^{\mathcal{F}_\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix}$, $R_{\mathcal{F}_v}^{\mathcal{F}_\theta} = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}$ and $R_{\mathcal{F}_\phi}^{\mathcal{F}_b}(\phi) = R_{\mathcal{F}_\theta}^{\mathcal{F}_b}(\theta) = R_{\mathcal{F}_\psi}^{\mathcal{F}_b}(\psi) = I$.

Therefore,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{\mathcal{F}_b} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{\mathcal{F}_v}$$

It follows that,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{\mathcal{F}_v} = \begin{bmatrix} 1 & s\phi \tan \theta & c\phi \tan \theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi \sec \theta & c\phi \sec \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{\mathcal{F}_b} \quad (2)$$

Equations (1) and (2) represent the quadrotor's equations of motion.

2.3 Quadrotor's dynamics

To build the dynamic model of the quadrotor we will use Newton-Euler formalism, while adopting the following assumptions:

1. The quadrotor structure is a rigid body.
2. The quadrotor frame is symmetrical.

3. The COG of the quadrotor coincides with the center of the rigid frame.

The moment of inertia is calculated by assuming the quadrotor as a central sphere of radius r and mass M_0 surrounded by four point masses representing the motors. Each motor is supposed to have a mass m and attached to the central sphere through an arm of length l , as shown in Figure 7.

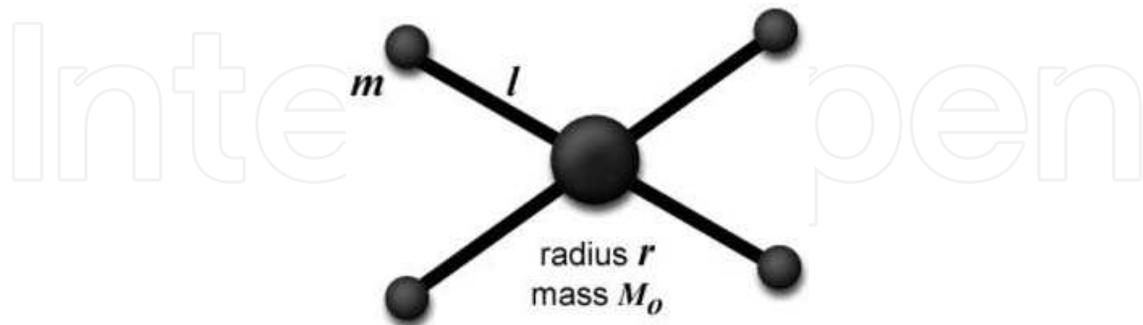


Fig. 7. Moment of inertia.

Due to the symmetry of the quadrotor about all three axes, its inertial matrix becomes symmetrical and is defined by

$$J = \begin{bmatrix} j_x & 0 & 0 \\ 0 & j_y & 0 \\ 0 & 0 & j_z \end{bmatrix}$$

where $j_x = j_y = j_z = \frac{2M_0r^2}{5} + 2l^2m$.

The dynamics of the quadrotor under external forces applied to its COG and expressed in the body frame is derived by applying Newton-Euler formulation Beard (2008)

$$\begin{bmatrix} MI_{3 \times 3} & 0 \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} \ddot{P}_{\mathcal{F}_b} \\ \ddot{\Omega}_{\mathcal{F}_b} \end{bmatrix} + \begin{bmatrix} \dot{\Omega}_{\mathcal{F}_b} \times M\dot{P}_{\mathcal{F}_b} \\ \dot{\Omega}_{\mathcal{F}_b} \times J\dot{\Omega}_{\mathcal{F}_b} \end{bmatrix} = \begin{bmatrix} F_{\mathcal{F}_b} \\ \tau_{\mathcal{F}_b} \end{bmatrix}$$

where M is the quadrotor's total mass, and $F^T = [f_x \ f_y \ f_z]$ and $\tau^T = [\tau_\phi \ \tau_\theta \ \tau_\psi]$ are the external force and torque vectors applied on the quadrotor's COG. The terms τ_ϕ , τ_θ , and τ_ψ are the roll, pitch and yaw torques respectively.

Thus, the translational dynamic model can be written as

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{bmatrix}_{\mathcal{F}_b} = \begin{bmatrix} \dot{\psi}\dot{p}_y - \dot{\theta}\dot{p}_z \\ \dot{\phi}\dot{p}_z - \dot{\psi}\dot{p}_x \\ \dot{\theta}\dot{p}_x - \dot{\phi}\dot{p}_y \end{bmatrix}_{\mathcal{F}_b} + \frac{1}{M} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_{\mathcal{F}_b}$$

while the rotational model is

$$\begin{aligned} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= J^{-1} \left\{ \begin{bmatrix} 0 & \dot{\psi} & -\dot{\theta} \\ -\dot{\psi} & 0 & \dot{\phi} \\ \dot{\theta} & -\dot{\phi} & 0 \end{bmatrix} J \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \right\} \\ &= \begin{bmatrix} \frac{j_y - j_z}{j_x} \dot{\theta}\dot{\psi} \\ \frac{j_z - j_x}{j_y} \dot{\phi}\dot{\psi} \\ \frac{j_x - j_y}{j_z} \dot{\phi}\dot{\theta} \end{bmatrix}_{\mathcal{F}_b} + \begin{bmatrix} \frac{1}{j_x} \tau_\phi \\ \frac{1}{j_y} \tau_\theta \\ \frac{1}{j_z} \tau_\psi \end{bmatrix}_{\mathcal{F}_b} \end{aligned}$$

2.4 Aerodynamic forces and torques

With the derived kinematic and dynamic model, we will now define the forces and torques acting on the quadrotor. The forces include the aerodynamic lift generated by each rotor, and the gravitational pull acting in counter to the total lift generated. The moments are the torques generated in order to achieve the roll, pitch and yaw movements. The following forces and torques are produced:

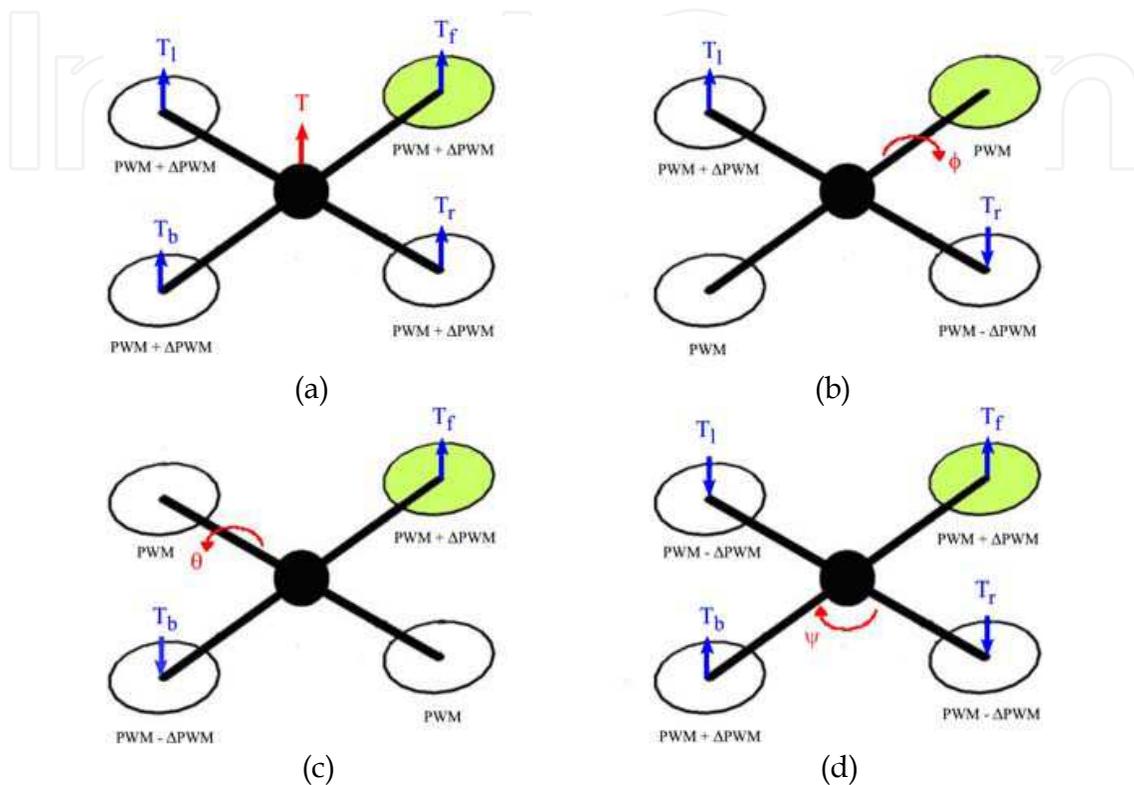


Fig. 8. Forces and moments acting on the quadrotor: (a) Quadrotor thrust; (b) Rolling torque; (c) Pitching torque; and (d) Yawing torque.

Upward Force (Thrust): The total quadrotor thrust is the sum of the thrust produced by each propeller, as depicted in Figure 8(a):

$$T = T_f + T_r + T_b + T_l$$

Rolling Torque: This is the torque produced by increasing the left rotor's thrust while decreasing that of the right rotor, or vice versa, as shown in Figure 8(b):

$$\tau_\phi = l(T_l - T_r)$$

Pitching Torque: The pitching torque in Figure 8(c) is produced by increasing the front rotor's thrust while decreasing that of the back rotor, or vice versa:

$$\tau_\theta = l(T_f - T_b)$$

Yawing Torque: The yawing torque is the result of all four individual torques generated due to the spinning rotors. The front and back rotors spin in the clockwise direction, while the left and right rotors spin in the counterclockwise direction. As shown in Figure 8(d), an imbalance between these two pairs results in a yawing torque causing the quadrotor to rotate about its z-axis:

$$\tau_\psi = \tau_f + \tau_b - \tau_r - \tau_l$$

Gravitational Force (weight): Along with the other forces, the gravitational force acts on the COG of the quadrotor. In the vehicle frame this force is expressed as

$$W_{\mathcal{F}_v} = \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix}$$

with g being the gravitational constant. Therefore, in the body frame, the weight can be written as

$$W_{\mathcal{F}_b} = R_{\mathcal{F}_v}^{\mathcal{F}_b} \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix} = \begin{bmatrix} -Mgs\theta \\ Mgc\theta s\phi \\ Mgc\theta c\phi \end{bmatrix}$$

Including the forces and torques acting on the system, the equations of motion become as defined below.

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{bmatrix}_{\mathcal{F}_b} = \begin{bmatrix} \dot{\psi}\dot{p}_y - \dot{\theta}\dot{p}_z \\ \dot{\phi}\dot{p}_z - \dot{\psi}\dot{p}_x \\ \dot{\theta}\dot{p}_x - \dot{\phi}\dot{p}_y \end{bmatrix} + \begin{bmatrix} -gs\theta \\ gc\theta s\phi \\ gc\theta c\phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{f_z}{M} \end{bmatrix}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}_{\mathcal{F}_b} = \begin{bmatrix} \frac{j_y - j_z}{j_x} \dot{\theta}\dot{\psi} \\ \frac{j_z - j_x}{j_y} \dot{\phi}\dot{\psi} \\ \frac{j_x - j_y}{j_z} \dot{\phi}\dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{j_x} \tau_\phi \\ \frac{1}{j_y} \tau_\theta \\ \frac{1}{j_z} \tau_\psi \end{bmatrix}$$

3. Flight controller design

This section details the development of a fuzzy logic flight controller for the quadrotor. A generalized overview of fuzzy logic control and the advantages it offers for nonlinear control applications are presented. Based on the dynamics and kinematics derived in the previous section, the autonomous flight control strategy is thereby introduced. The proposed fuzzy logic controller is implemented with two types of inference engines for comparison.

3.1 Fuzzy logic control

Since its inception in Zadeh (1965), fuzzy logic has been applied to various fields of engineering, manufacturing, business, and medicine, among others. Within the area of engineering, control systems offer significant applications for fuzzy logic, designated as fuzzy logic control. Before getting into details with regards to fuzzy logic control, we would first like to provide some basic facts about fuzzy systems.

Fuzzy logic control offers a great advantage over some conventional control methods which heavily depend on the exact mathematical model of the control system, specifically in dealing with nonlinear systems subjected to various types of uncertainties. Being independent of the plant's parameters sets fuzzy controllers apart from their conventional counterparts. Fuzzy controllers in general can be designed intuitively in light of the

knowledge acquired on the behavior of the system in hand. This knowledge is often gained through experience and common sense, regardless of the mathematical model of the dynamics governing this behavior. For example, in learning how to ride a bike, humans try to build a set of common sense rules and learn from their failures without paying any attention to the dynamic model of the bike. Fuzzy logic control tries to mimic this type of human-like reasoning and embrace it within a pre-defined mathematical model to automate the control of complex systems characterized by ill-defined mathematical models, for example.

3.2 Flight control algorithm

The quadrotor is an under-actuated system with four actuators controlling its six degrees-of-freedom position/orientation. The flight controller is responsible for achieving two challenging goals simultaneously: (i) controlling the quadrotor's position, while (ii) stabilizing its attitude, i.e., orientation (roll, pitch and yaw angles). More specifically, given a desired position (p_x, p_y, p_z) and yaw angle ψ , the goal is to design a controller to force these control states to converge to their respective desired values, while maintaining the pitch and roll angles as close to zero as possible.

Let PWM_{mot} denote the PWM value of motor $mot \in \{f, r, b, l\}$ for the front, right, back, and left motors, respectively. Then, the thrust and torque applied on the quadrotor by motor mot can be expressed as

$$\begin{aligned} T_{mot} &= K_T \times PWM_{mot} \\ \tau_{mot} &= K_\tau \times PWM_{mot} \end{aligned}$$

where K_T and K_τ are motor-dependent parameters. This yields

$$\begin{bmatrix} PWM_f \\ PWM_r \\ PWM_b \\ PWM_l \end{bmatrix} = G \times \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

with

$$G = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -l \times K_T & 0 & l \times K_T \\ l \times K_T & 0 & -l \times K_T & 0 \\ -K_\tau & K_\tau & -K_\tau & K_\tau \end{bmatrix}^{-1}$$

The above equations provide a basic understanding of how the angular speed of each motor contribute to the overall thrust and torques exerted on the quadrotor. This knowledge will serve as a guideline in developing a rule base of the direct fuzzy logic controller, as depicted in Figure 9.

Three fuzzy controllers are designed to control the quadrotor's roll (ϕ), pitch (θ) and yaw (ψ) angles, denoted by FLC_ϕ , FLC_θ , and FLC_ψ , respectively, with the former two serving as attitude stabilizers. Three fuzzy controllers, FLC_x , FLC_y and FLC_z , are further designed to control the quadrotor's position. All six fuzzy controllers have identical inputs, (i) the error $e = (\tilde{\cdot}) = (\cdot)_d - (\cdot)$, which is the difference between the desired signal $(\cdot)_d$ and its actual

value (\cdot), and (ii) the error rate \dot{e} . The first input (error) is normalized to the interval $[-1,+1]$, while the second (error rate) is normalized to the interval $[-3,+3]$.

$$\begin{aligned} U_x &= \text{FLC}_x(\tilde{p}_x, \dot{\tilde{p}}_x) & U_\theta &= \text{FLC}_\theta(\tilde{\theta}, \dot{\tilde{\theta}}) \\ U_y &= \text{FLC}_y(\tilde{p}_y, \dot{\tilde{p}}_y) & U_\phi &= \text{FLC}_\phi(\tilde{\phi}, \dot{\tilde{\phi}}) \\ U_z &= \text{FLC}_z(\tilde{p}_z, \dot{\tilde{p}}_z) & U_\psi &= \text{FLC}_\psi(\tilde{\psi}, \dot{\tilde{\psi}}) \end{aligned}$$

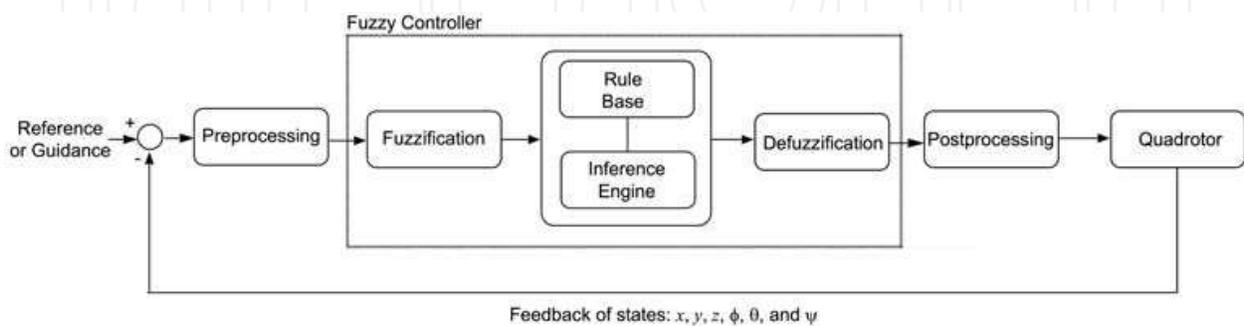


Fig. 9. Control scheme.

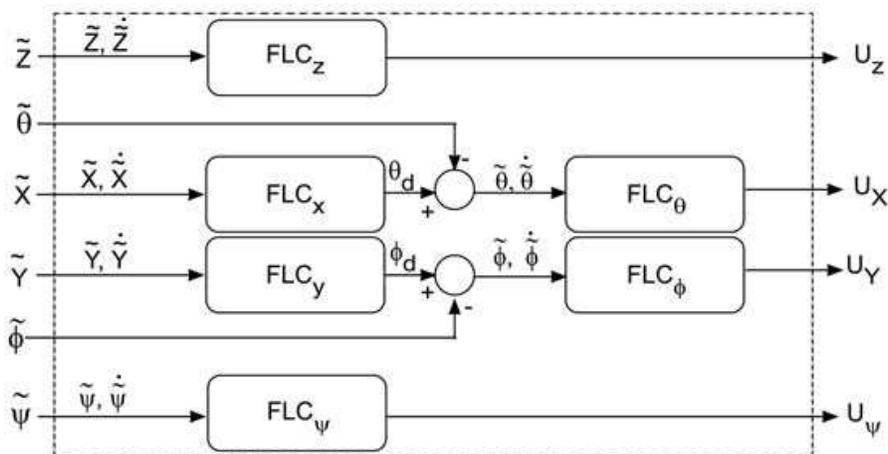
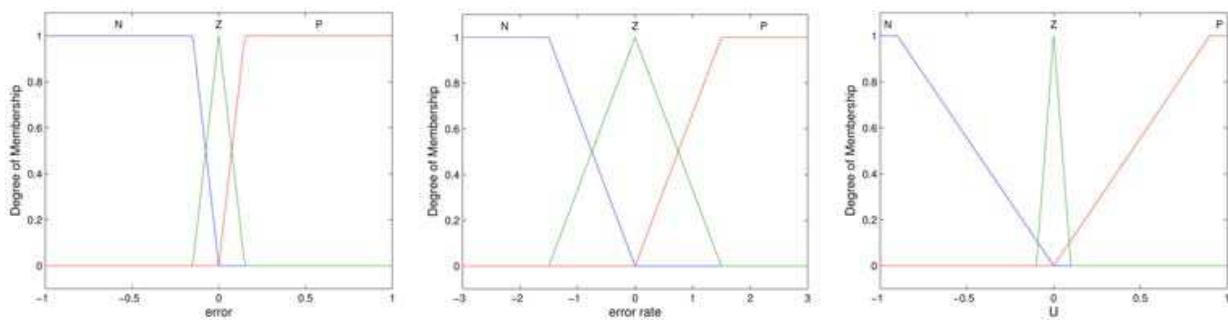


Fig. 10. Flight controller block diagram.

In this control strategy, the desired pitch and roll angles, θ_d and ϕ_d , are not explicitly provided to the controller. Instead, they are continuously anticipated by controllers FLC_x and FLC_y in such a way that they stabilize the quadrotor's attitude. The input and output membership functions (Figure 11) of each FLC are tuned empirically and are finalized as follows:

$$\begin{aligned} \mu_N(e) &= \text{trapezoid}(-1, -0.15, 0) & \mu_Z(e) &= \text{triangle}(-0.15, 0, 0.15) \\ \mu_P(e) &= \text{trapezoid}(0, 0.15, 1) & \mu_N(\dot{e}) &= \text{trapezoid}(-3, -1.5, 0) \\ \mu_Z(\dot{e}) &= \text{triangle}(-1.5, 0, 1.5) & \mu_P(\dot{e}) &= \text{trapezoid}(0, 1.5, 3) \\ \mu_N(U) &= \text{trapezoid}(-1, -0.85, 0) & \mu_Z(U) &= \text{triangle}(-0.1, 0, 0.1) \\ \mu_P(U) &= \text{trapezoid}(0, 0.85, 1) \end{aligned}$$

A unified rule base comprising nine IF-THEN rules is developed and is presented in Table 2.



(a) Input variable error e . (b) Input variable error rate \dot{e} . (c) Output variable U .

Fig. 11. Input and output membership functions.

		e		
		N	Z	P
\dot{e}	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

Table 2. The rule base of the fuzzy controller.

For it to be modular and independent of the quadrotor’s parameters, the fuzzy logic controllers are bounded by pre-processing and post-processing blocks (Figure 9). The pre-processing module calculates the error e and error rate \dot{e} and normalizes them to the intervals $[-1,+1]$ and $[-3,+3]$, respectively. The post-processing block uses the controllers output signals to calculate the PWM value of each motor as follows:

$$\begin{aligned}
 PWM_f &= Sat(U_Z + U_X - U_\psi + Offset) \\
 PWM_r &= Sat(U_Z + U_Y + U_\psi + Offset) \\
 PWM_b &= Sat(U_Z - U_X - U_\psi + Offset) \\
 PWM_l &= Sat(U_Z - U_Y + U_\psi + Offset)
 \end{aligned}$$

where ‘Offset’ is a priori-defined bias to counter balance the weight of the quadrotor. The resultant PWM values are saturated to a maximum threshold that depends on the maximum possible speed of the motors used.

It is important to note that this control scheme does not depend on the kinematic and dynamic equations derived in section 2. Those equations are only used to build the quadrotor’s model in the simulator, which would be unnecessary with a real quadrotor. Being independent of the plant’s parameters sets the fuzzy controllers apart from conventional control systems, which depend in one way or the other on the plant’s mathematical model. The fuzzy controllers are designed in light of the knowledge acquired on the quadrotor’s behavior and from its dynamic model. Therefore, changing the quadrotor or some of its physical parameters like the mass and inertia does not require redesigning the fuzzy logic controller. Instead, the postprocessing module may need to be fine-tuned to optimize the controller’s performance, such as to calibrate the offset, for instance.

Two different fuzzy inference engines are implemented: (i) a Mamdani, and (ii) a Takagi-Sugeno-Kang (TSK) fuzzy model. The Mamdani fuzzy inference method uses a min-max operator for the aggregation and the centroid of area method for defuzzification. One known problem with this type of controller is the high computational burden associated to it, especially when implemented on an embedded system. To alleviate this problem, a zero

order TSK fuzzy inference engine is implemented for comparison. In this model, the output membership functions of the Mamdani fuzzy controller are replaced with three fuzzy singletons $N = -1$, $Z = 0$ and $P = +1$.

4. Numerical and experimental results

To test the proposed fuzzy logic flight controller with both inference engines and study their performances, a simulation environment is first developed. After satisfactory performance results are attained, the controller is implemented on a quadrotor testbed. The details of the quadrotor simulator and the real-world test-bed are presented in the following subsections.

4.1 Simulation results

The quadrotor simulator is implemented in MATLAB Simulink, as shown in Figure 12. The equations of motion derived earlier are used to model the quadrotor. The inputs to the quadrotor are taken as the four PWM speed values of the motors. To make the simulations more realistic, sensory noise and environmental disturbances such as wind are also taken into account. Medium wind gust speeds are generated based on real data from Canada Weather Statistics, Statistics (2009). The wind disturbance is incorporated as two further inputs representing the north and east wind condition. The quadrotor model outputs are the linear and angular accelerations that are integrated twice to obtain the position and orientation vectors. The angular accelerations $\ddot{\phi}$, $\ddot{\theta}$, and $\ddot{\psi}$ are degraded with a white noise and then used as a feedback to the fuzzy controller.

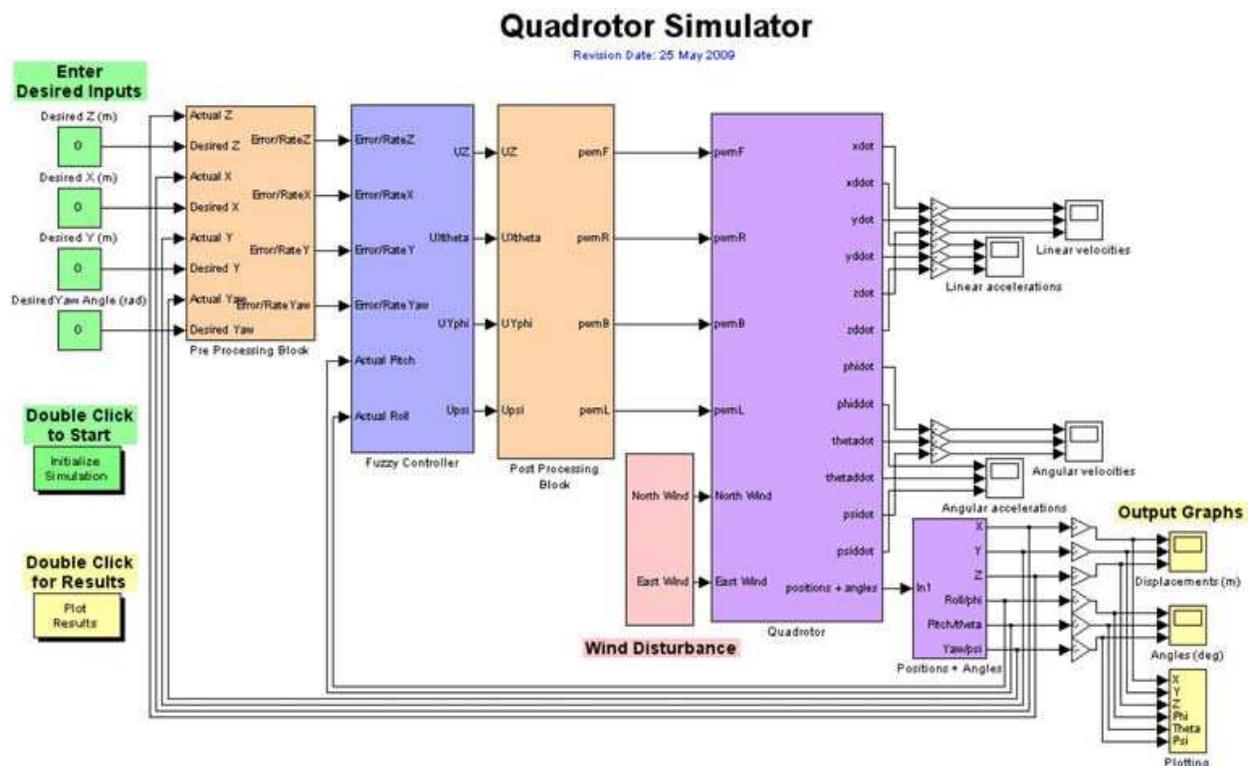


Fig. 12. MATLAB Simulink block diagram of the quadrotor simulator.

The values used for the quadrotor's dynamic parameters are: $M = 0.765$ Kg, $l = 0.61$ m, $J_x = J_y = 0.08615$ Kg·m², $J_z = 0.1712$ Kg·m², $K_T = 5.45$, $K_{tau} = 0.0549$, $\text{Offset} = Mg/(4K_T) = 0.344$.

The six identical fuzzy controllers are developed using the MATLAB Fuzzy Logic Toolbox. The input and output variables with membership functions described earlier are set accordingly. The fuzzy controllers are used in a configuration as shown in Figure 10. The inputs and outputs of the flight controller are pre- and post-processed, respectively.

The environmental disturbances are introduced such that a white noise is added to the angular accelerations $\ddot{\phi}$, $\ddot{\theta}$, and $\ddot{\psi}$ for emulating the inertial measurement unit (IMU) sensor. The IMU signals are further processed through rate transitions to incorporate the ADC sampling rate.

The user-defined inputs are the desired translatory coordinates $P_{\mathcal{F}_i}$ with respect to the inertial frame, and the desired yaw angle ψ . The desired pitch and roll angles are implicitly set to zero to achieve attitude stabilization.

Three simulations are conducted to test the performance of the proposed fuzzy logic flight controllers with both inference engines. The system's initial states are set to zero, while the desired quadrotor's position and orientation are set to $P_{\mathcal{F}_i}^T = [10, 10, 25]$ m and $\Omega_{\mathcal{F}_i}^T = [0, 0, 30]$ degrees in all three simulations. The purpose of the simulation is to assess the performance of the fuzzy logic controller and compare the accuracy of the two fuzzy inference engines under different disturbance conditions. The first simulation is run without any disturbances. In the second simulation, the controller is subjected to sensor noise. In the third simulation, it is subjected to sensor noise and medium north-east wind gust of 10 m/s.

The simulation results presented in Figures 13 and 14, demonstrate the satisfactory performance of the proposed controller despite the presence of sensor noise and wind disturbances. The Mamdani fuzzy controller converges to the desired states relatively faster than its TSK counterpart. The yaw angle drift under wind disturbance is clearly visible with the TSK controller.

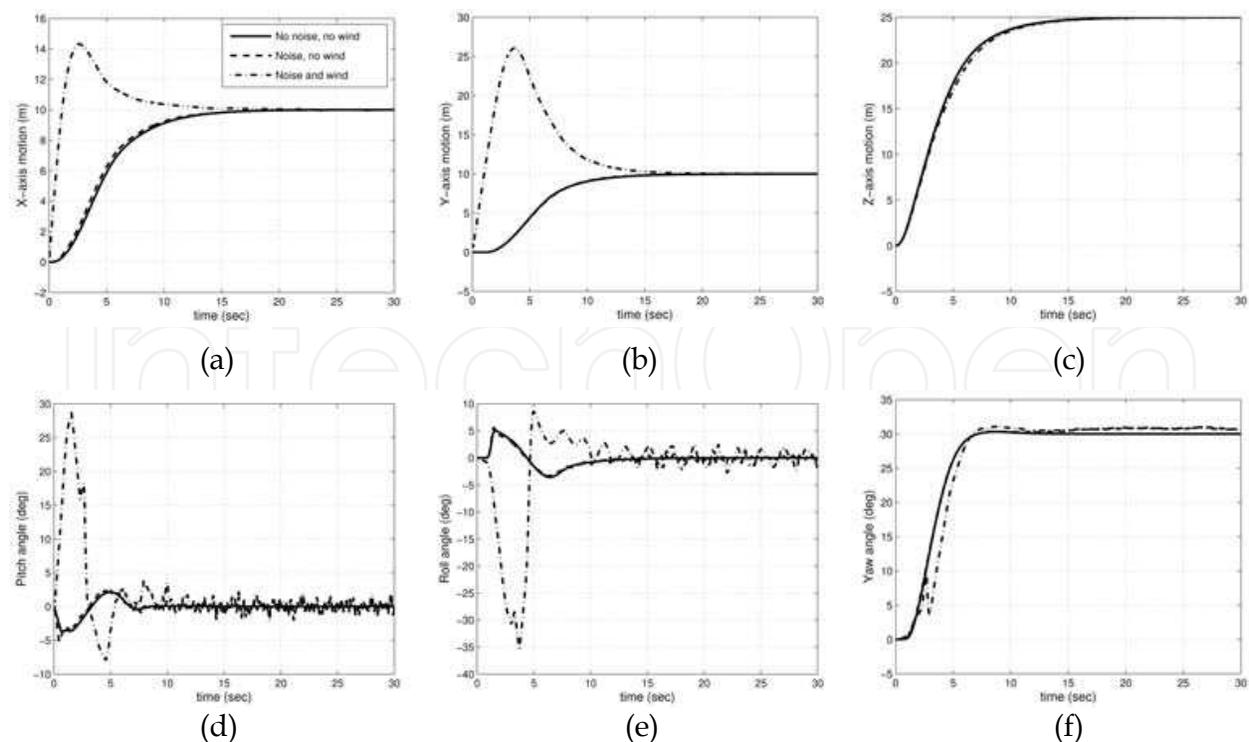


Fig. 13. Simulation results of the Mamdani controller. Quadrotor states: (a) x-axis; (b) y-axis; (c) z-axis (altitude); (d) pitch (θ); (e) roll (ϕ); and (f) yaw (ψ).

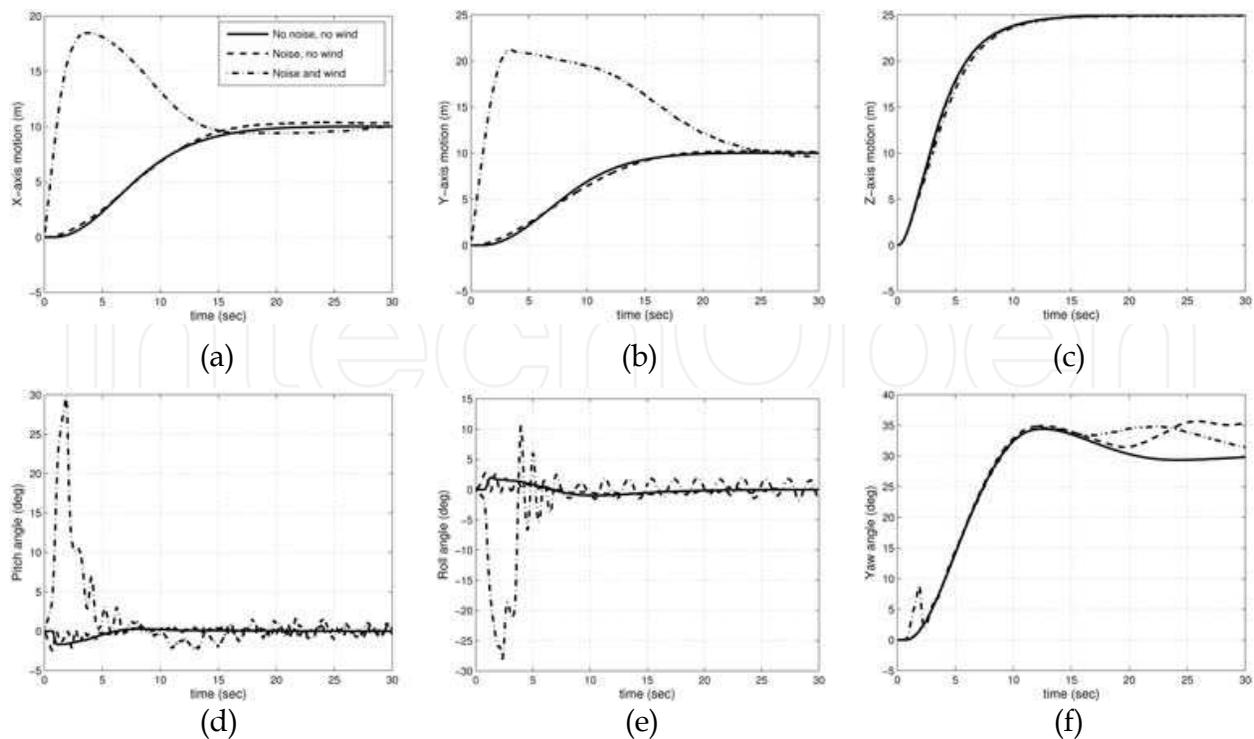


Fig. 14. Simulation results of the TSK controller. Quadrotor states: (a) x-axis; (b) y-axis; (c) z-axis (altitude); (d) pitch (θ); (e) roll (ϕ); and (f) yaw (ψ).

4.2 Quadrotor test-bed and experimental results

The quadrotor test-bed, as shown in Figure 15, comprises light weight carbon fiber rods of length 0.61 m connected to a center piece forming the desired cross structure for the frame. The four propulsion units are made of brushless DC motors connected to electronic speed controllers to provide actuation. Different propellers are also tested to find the optimal thrust to power consumption ratio. The system is powered by a 2100 mAh, 11.1-Volt lithium polymer battery. The quadrotor's specifications are summarized in Table 3.



Fig. 15. The quadrotor test-bed.

The IMU is designed using a triple axis 3g accelerometer ADXL330, a dual axis 500 degree-per-sec rate gyroscope IDG300 for pitch and roll angle rates. The yaw angle acceleration is measured using a single axis 300 degree-per-sec ADXRS300 gyroscope. The bandwidths of all the sensors are trimmed to 10 Hz. The MEMS inertial sensors are combined with complementary filtering techniques for the online estimation of the attitude angles.

Component	Specifications	Weight (g)
4 Carbon fiber rods	Frame arms of 0.61m length	52
Center piece	Frame center joint	58
4 Motor mounts	Aircraft grade aluminum sheets	40
4 Brushless DC motors	RIMFIRE 1000kV, 135Watts	164
4 Electronic Speed Controllers	ElectriFly 12A ESC 150Watts	56
4 Composite propellers	Counter rotating sets L=10", P=4.5"	24
4 Propeller adapters	Collet type aluminum 3-5mm	20
Lithium polymer battery	11.1V 2100mAh	150
Power supply board	Hybrid PSU	70
IMU sensor board	ADXL330, IDG300, ADXRS300	44
Axon board	ATmega640	34
Standoffs and screws	Aluminum frame assembly	53
	Total Weight	765

Table 3. The quadrotor test-bed component summary.

The processing is accomplished with a 34-gram Axon board based on ATmega640 microcontroller running at 16 MHz clock. The IMU data is sampled using an onboard 10-bit ADC. The proposed flight control is implemented using the TSK fuzzy controllers converted to a look-up table for a higher computational efficiency. The flight controller, elaborated in Algorithm 1, is set to operate at a bandwidth of 100 Hz.

Algorithm 1 Quadrotor Flight Control Algorithm

Initialize:

UART 0,1,2,3

Analog and Digital Ports

Timer 0,1,2,3,4

ADC clock = 125 KHz

PWM Timers 3,4

Body:

IF (button pressed) THEN Arm the motors.

Desired Coordinates

GPS Coordinates

ADC Gyro and Accelerometer Signals

IMU Measurement Calibrations

Complementary Filtering for Attitude Estimates

Transmit the Attitude Estimates

Pre-processing: Error and Error Rate Calculation

CALL the 6 Fuzzy Controllers

Calculate $U_x, U_y, U_z, U_\phi, U_\theta, U_\psi$

Post-processing: Calculate $PWM_f, PWM_b, PWM_r, PWM_l$

Set PWM timers

Repeat

END

The first experiment is designed to test the quadrotor's hovering and attitude stabilization capabilities. So, the desired position $P_{\mathcal{F}_i}$ is pre-defined as the quadrotor's current position. Both, experimental and simulation results are reported in Figure 16. In simulation, the controller managed to keep the pitch and roll angles within the interval of $[-3,+4]$ degrees. However, in reality, the errors of these angles were fluctuating between -8 and $+7$ degrees for the pitch, and -6 and $+12$ degrees for the roll. The main difference between the simulation and experimental results stems from the vibration of the frame from which the test-bed was made. In addition, the 0.61-m long carbon fiber arms were bending and twisting when excited by the motors. This put the fuzzy logic controller to a higher challenge than what was originally anticipated. Yet, it was successful in forcing the pitch and roll angles to within an acceptable range.

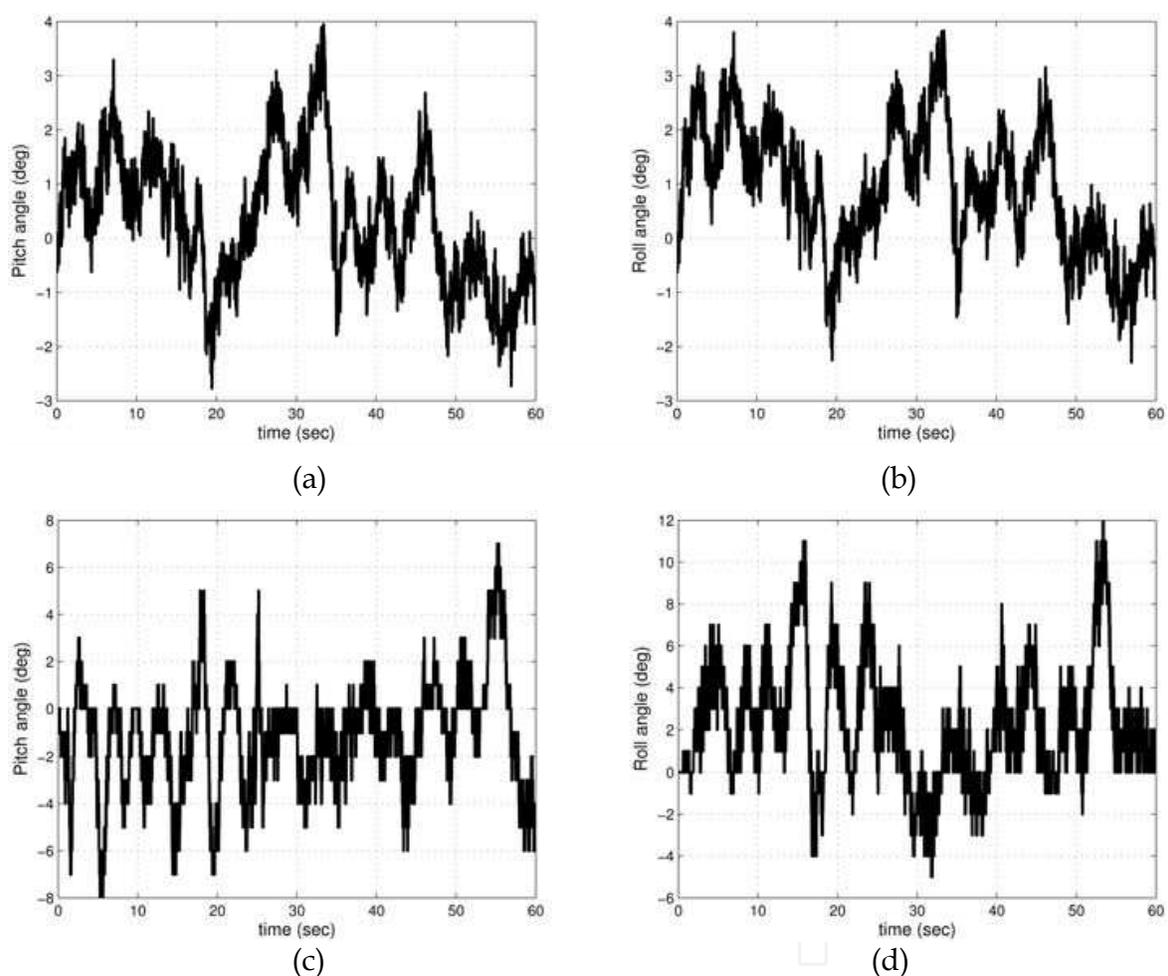


Fig. 16. Simulation and experimental attitude stabilization results: (a) simulator pitch (θ); (b) simulator roll (ϕ); (c) test-bed pitch (θ); and (d) test-bed roll (ϕ).

In the second experiment, the controller was tested under harsher conditions so as to evaluate its behavior if the quadrotor collides into an obstacle, or if it is faced with other types of disturbances. Hence, the previous experiment was repeated, but this time one of the quadrotor's arms is abruptly tapped in the middle of the flight. The results are shown in Figure 17. As can be seen, the controller was able to quickly bring the pitch and roll angle errors back to within a safe range.

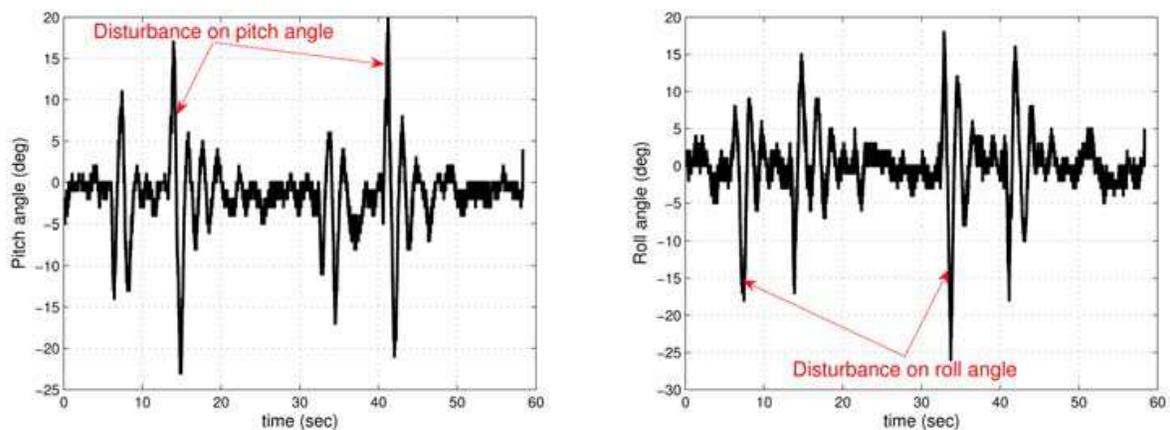


Fig. 17. Experimental results of attitude stabilization under external disturbances: (a) test-bed pitch θ ; and (b) test-bed roll ϕ .

5. Conclusion and future directions

This chapter addressed the problem of autonomous flight control of a quadrotor UAV. It was conducted as a research project at the School of Information Technology and Engineering (S.I.T.E.), University of Ottawa. Detailed mathematical modeling of the quadrotor's kinematics and dynamics was provided. A modular fuzzy logic approach was proposed for the autonomous control of quadrotors in general, without the need for a precise mathematical model of their complex and ill-defined dynamics. The fuzzy technique was implemented through Mamdani and TSK inference engines for comparison purposes. The controller comprises six individual fuzzy logic modules designated for the control of the quadrotor's position and orientation. The investigation on the two types of control methodologies was conducted in a simulator environment, where disturbances such as wind conditions and sensor noise were incorporated for a more realistic simulation. The fuzzy flight controller was eventually implemented on a quadrotor test-bed specifically designed and built for the project.

The experiments were first conducted on the simulator before being validated on the test-bed. The results demonstrated a successful control performance especially with the Mamdani inference engine. When compared to other conventional techniques applied for a similar purpose Altug et al. (2002), Bouabdallah et al. (2004b), the proposed methodology showed a higher robustness despite the induced disturbances.

The future work is directed towards achieving fully autonomous flight in outdoor environments. Furthermore, adaptive fuzzy control techniques will be investigated to automatically tune some of the controller's parameters online, to further optimize its performance.

6. References

- E. Altug, J.P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 72–77, 2002.
- F. Archer, A. Shutko, T. Coleman, A. Haldin, E. Novichikhin, and I. Sidorov. Introduction, overview, and status of the microwave autonomous copter system MACS. *Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium*, 5:3574–3576, Sep 2004.

- Randal W. Beard. Quadrotor dynamics and control. Brigham Young University, February 19 2008. URL <http://www.et.byu.edu/groups/ece490quad/control/quadrotor.pdf>. lecture notes.
- S. Bouabdallah. Design and control of quadrotors with application to autonomous flying. Master's thesis, Swiss Federal Institute of Technology, 2007.
- S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Proceedings of the International Conference on Robotics and Automation*, 2004a.
- S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *International Conference on Intelligent Robots and Systems*, 2004b.
- M. Chen and M. Huzmezan. A combined MBPC/ 2 DOF H_∞ controller for a quad rotor UAV. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003.
- C. Coza and C.J.B. Macnab. A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization. In *Annual meeting of the North American Fuzzy Information Processing Society*, pages 454–458, 2006.
- A. Dzul, P. Castillo, and R. Lozano. Real-time stabilization and tracking of a four rotor mini rotorcraft. *IEEE Transaction on Control System Technology*, 12(4): 510–516, 2004.
- S. G. Fowers. Stabilization and control of a quad-rotor micro-UAV using vision sensors. Master's thesis, Brigham Young University, August 2008.
- N. Guenard, T. Hamel, and V. Moreau. Dynamic modeling and intuitive control strategy for an X4-flyer. In *International Conference on Control and Automation*, pages 141–146, 2005.
- N. Guenard, T. Hamel, and R. Mahony. A practical visual servo control for an unmanned aerial vehicle. *IEEE Transactions on Robotics*, 24(2):331–340, 2008.
- S. D. Hanford. A small semi-autonomous rotary-wing unmanned air vehicle. Master's thesis, Pennsylvania State University, December 2005.
- Draganfly Innovations Inc. Industrial aerial video systems and UAVs, October 2008. URL <http://www.draganfly.com/>.
- A.Ó. Kivrak. Design of control systems for a quadrotor flight vehicle equipped with inertial sensors. Master's thesis, The Graduate School of Natural and Applied Sciences of Atilim University, December 2006.
- J. Rumerman. Helicopter development in the early twentieth century. [Online] Available, December 2002. URL www.centennialofflight.gov.
- Canada Weather Statistics. Courtesy of Environment Canada. [Online] Available, February 2009. <http://www.weatherstats.ca/>.
- R. Sugiura, T. Fukagawa, N. Noguchi, K. Ishii, Y. Shibata, and K. Toriyama. Field information system using an agricultural helicopter towards precision farming. *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2:1073–1078, July 2003.
- M. Tarbouchi, J. Dunfied, and G. Labonte. Neural network based control of a four rotor helicopter. In *International Conference on Industrial Technology*, pages 1543–1548, 2004.
- A. Tayebi and S. McGilvray. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Transaction on Control System Technology*, 14(3):562–571, May 2006.
- S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin. Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning. In *International Conference on Intelligent Robots and Systems*, pages 468–473, 2005.
- K. W. Weng and M. Shukri. Design and control of a quad-rotor flying robot for aerial surveillance. *4th Student Conference on Research and Development (SCOReD 2006)*, pages 173–177, 2006.
- L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.



Motion Control

Edited by Federico Casolo

ISBN 978-953-7619-55-8

Hard cover, 590 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

The book reveals many different aspects of motion control and a wide multiplicity of approaches to the problem as well. Despite the number of examples, however, this volume is not meant to be exhaustive: it intends to offer some original insights for all researchers who will hopefully make their experience available for a forthcoming publication on the subject.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Syed Ali Raza and Wail Gueaieb (2010). Intelligent Flight Control of an Autonomous Quadrotor, Motion Control, Federico Casolo (Ed.), ISBN: 978-953-7619-55-8, InTech, Available from:

<http://www.intechopen.com/books/motion-control/intelligent-flight-control-of-an-autonomous-quadrotor>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen