Controlling an unmanned quad-rotor aerial vehicle with model parameter uncertainty and actuator failure

Abdel Ilah Nour Alshbatat*

Communications and Computer Engineering Department,
Tafila Technical University,
Tafila, 66110, Jordan
Email: a.alshabatat@ttu.edu.jo
Email: abdnoor80@yahoo.com
*Corresponding author

Liang Dong

Electrical and Computer Engineering Department,
Baylor University,
Waco, TX 76798-7356, USA
Email: liang_dong@baylor.edu

Peter James Vial

School of Electrical, Computer and Telecommunication Engineering,
University of Wollongong,
Wollongong, NSW 2522, Australia
Email: peter_vial@uow.edu.au

Abstract: It is challenging to stabilise an unmanned quad-rotor aerial vehicle when a dynamic change in its model parameters or failure of its actuator occurs. In this paper, a quad-rotor unmanned aerial vehicle (UAV) is controlled based on model reference adaptive control (MRAC) and a linear quadratic regulator (LQR). The kinematics and dynamics of the quad-rotor are calculated, and Lyapunov's direct stability method is used to design the MRAC. In order to evaluate the performance of the adaptive control algorithms in the presence of thrust loss that may occur due to component failure or physical damage, a real quad-rotor is built from scratch using commercial components. Both controllers are designed, implemented and tested using AVR microcontrollers. Comparison is made between the controllers under normal and faulty situations and the effectiveness of the proposed control strategy is verified. Simulation and experimental results show that both controllers have satisfactory performance under normal conditions and even in the presence of the partial loss of thrust that may occur due to the loss of control effectiveness in one of the rotors or the damage of one propeller, superior system performance is observed using the proposed MRAC controller.

Keywords: MRAC; model reference adaptive control; LQR; linear quadratic regulator; quad-rotor unmanned aerial vehicle; Lyapunov stability method.

Reference to this paper should be made as follows: Alshbatat, A.I.N., Dong, L. and Vial, P.J. (2016) 'Controlling an unmanned quad-rotor aerial vehicle with model parameter uncertainty and actuator failure', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 15, No. 4, pp.295–322.

Biographical notes: Abdel Ilah Nour Alshbatat received his BS in Electrical Engineering from Mutah University, Jordan, in 1991, his MS in Computer Engineering from Yarmouk University, Jordan, in 2004, and his PhD in Electrical and Computer Engineering from Western Michigan University, USA, in 2010. In 1991, he joined the Royal Jordanian Air Force and served in the Directorate of Communication and Electronics. In 2010, he joined the Faculty of Tafila Technical University and is currently an Associate Professor in the Department of Computer Engineering. His current research interests include computer networks, wireless networks, embedded systems, unmanned aerial vehicles, brain-computer interface and communication systems.

Liang Dong received the BS in Applied Physics with minor in Computer Engineering from Shanghai Jiao Tong University, China, in 1996, and the MS and PhD in Electrical and Computer Engineering from the University of Texas at Austin in 1998 and 2002, respectively. He joined the Faculty of Baylor University in August 2011 as an Associate Professor of Electrical and Computer Engineering.

Peter James Vial (M'89, SM'11) received the BE (Electrical) in 1987, the ME (Honours) on Telecommunications in 1996, Graduate Diploma in Education (Mathematics) in 2000 and PhD on Telecommunications in 2009 all from the University of Wollongong, Wollongong, New South Wales, Australia. He is currently an Assistant Professor at the University of Wollongong.

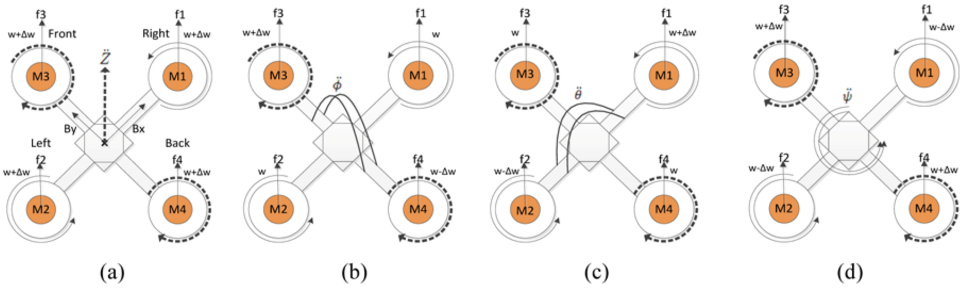
1 Introduction

The quad-rotor unmanned aerial vehicle (UAV) is an unpiloted aircraft capable of flight without a human operator. It is a small flying robot with four propellers driven by four DC motors: front, back, left and right (Das et al., 2009; Beji et al., 2005; Castillo et al., 2005). As shown in Figure 1, the front (M3) and back (M4) motors rotate clockwise while the left (M2) and right (M1) motors rotate counter-clockwise. The basic motion of this type of UAVs is generated by varying the rotor speeds of the four motors. Each rotor produces both thrust and torque with respect to the aircraft's centre of mass. Because the rotors spin in opposite directions, the torque on one side of the aircraft cancels out the torque on the other side. Therefore, if all rotors spin with the same angular velocity, the net torque equals zero and the thrust accelerates the aircraft up in the air (Figure 1 (a)). To change the direction of the quad-rotor, a mismatch needs be induced in the aerodynamic torque balance by increasing or decreasing the speed of the rotors rotating in the same direction.

The quad-rotor has emerged lately as a popular UAV platform. It can be controlled by the rotational speed of the rotors with the potential to take-off, fly, and land in a small area (Hoffmann et al., 2007; Hanford et al., 2005; Pounds et al., 2002; Tayebi and McGilvray, 2006). In hovering flight, to gain the rolling angle movement (Figure 1(b)), the front rotor's (M3) speed is increased and the back rotor's (M4) speed is decreased by the same amount while maintaining the same thrust on the other two rotors.

Likewise, to achieve the pitching angle movement (Figure 1(c)), the right rotor's (M1) speed is increased and the left rotor's (M2) speed is decreased while maintaining the same thrust on the other two rotors. Similarly, yaw moment (Figure 1(d)) is obtained by increasing the speed of rotors (M3) and (M4) and decreasing the speed of rotors (M1) and (M2).

Figure 1 Vertical and attitude movements of the quad-rotor: (a) vertical movement; (b) roll movement; (c) pitch movement and (d) yaw movement (see online version for colours)



Quad-rotor dynamics are normally controlled by four physical inputs: thrust, pitch, role, and yaw. Several control algorithms have been investigated to stabilise the quad-rotor (Bouabdallah and Noth, 2004; Madani and Benallegue, 2006; Tarbouchi et al., 2004; Mokhtari and Benallegue, 2004; Benallegue et al., 2001; Hamel et al., 2007). For example, the classical proportional-integral-derivative (PID) control algorithm can be used to control it. Since the quad-rotor is by nature an under-actuated system, which means that it is able to control all six degrees of freedom with only four inputs, the classical PID control algorithm does not work well on this UAV. On the other hand, adaptive control algorithms have been used to accommodate systems with parameters that are unknown or changing (Morel and Leonessa, 2006; Hamel et al., 2005; Middleton and Goodwin, 1988). The adaptive technique is different from other controllers in that it does not need a priori information about the uncertain parameters. The output of a plant is compared against the output of a model that is driven by a reference signal. The error between the model output and the plant output is then used to drive the plant to the desired reference model.

The MRAC algorithm is one of the best control techniques that maintain the stability of a system while the system parameters vary slowly or are uncertain (Hsu, 1990; Qu et al., 1994). This algorithm creates a closed loop controller with parameters that can be updated to change the response of the system so that it tracks the output of a reference model. Several approaches such as the MIT rule, Lyapunov stability theory, passivity theory, and theory of augmented error were used to design the MRAC. All methods except Lyapunov theory were considered very sensitive to the amplitude of the reference signal and therefore they do not guarantee convergence or stability (Swarnkar et al., 2011). MRAC provides a systematic approach for adjusting controller parameters and, therefore, has the potential to improve the performance of the quad-rotor in the presence of parameter uncertainty. It also has the capacity to react rapidly to any failure without a priori knowledge of such variations in the dynamic characteristics of the quad-rotor.

The main contributions of the paper are:

- Using Lyapunov's direct stability method in the designing of MRAC in order to stabilise quad-rotor in the presence of malfunction in actuator's bearing.
- Augmentation of baseline controller with Lyapunov based MRAC for the purpose of fast adaptation.

Facing the problem of malfunction in actuator's bearing, the motivation of this project is to develop a control mechanism as a hand on for those quad-rotor having an original controller. This controller can be implemented for any military or commercial controller without replacing or changing anything in the used one. Furthermore, the project aims to assist technical universities in enhancing the performance of educational controllers that they have, and use it as a practical demonstration for the impact of malfunction in actuator's bearing in the control process of a quad-rotor.

Quad-rotors possess tremendous advantages over other types of UAVs in terms of manoeuvring capability, safety, weight, and cost. It can be used for applications such as search and rescue, surveillance, and remote inspection (Jaimes et al., 2008). One of the challenges when constructing this type of vehicles is the stabilisation problem with dynamic changes in the model parameters. In this work, we model and control a quad-rotor UAV with parameter uncertainty using the MRAC. The controller is designed as the LQR whose parameters are updated by the MRAC using the Lyapunov stability method. The remainder of this paper is organised as follows: Section 2 reveals some recent research on control algorithms developed for quad-rotor vehicles. In Section 3, the kinematic and dynamic model of the quad-rotor is derived. In Section 4, the quad-rotor structure is presented. In Section 5, we describe the LQR and the MRAC and explain how Lyapunov's stability theory is used in the design of the MRAC. In Section 6, we present the simulation and experimental results as the performance of the MRAC is compared with the performance of the LQR when the system is subjected to parameter uncertainty and actuator failure. Finally, we summarise the main results of the work in Section 7.

2 Related works

In recent years, a large number of studies have been conducted on the modelling and controlling of quad-rotors (Castillo and Dzul, 2004). Various control algorithms have been proposed for indoor (Bouabdallah et al., 2005) and outdoor flights. One of these control algorithms is based on the linear quadratic regulator (LQR). The LQR was implemented to minimise tracking errors by minimising a cost function (Castillo et al., 2005; Tayebi and McGilvray, 2006). This provides the best possible performance with respect to given measurements. To account for uncertainties and external disturbances during the outdoor flight, model reference adaptive control (MRAC) was investigated with different types of quad-rotors (Whitehead and Bieniawskiy, 2010). Morel and Leonessa (2006) presented an innovative adaptive control algorithm using back stepping to solve the problem of trajectory tracking. They obtained the control law and then tested it through numerical simulations. Coza and Macnab (2006) proposed a new adaptive fuzzy control to stabilise the quad-rotor helicopter in the presence of sinusoidal wind disturbance. They developed a set of membership functions as a guide to the adaptation process. The adaptive fuzzy control method does not require an accurate system model and has been shown to be robust against disturbances.

MRAC techniques based on Lyapunov stability theory have been developed for adaptive control (Sastry and Bodson, 1989; Costa et al., 2003). Kaufman et al. (1998) presented the adaptive technique based on the concept of a generator tracker. This technique allows the system to follow sinusoidal reference commands. Sadeghzadeh et al. (2011) tested two popular controllers, i.e., the PID controller and the adaptive controller. Both controllers work well for controlling the height of the quad-rotor. They claimed that MRAC ensures stability of the quad-rotor immediately after a failure occurs. Palunko and Fierro (2011) addressed the problem of quad-rotor stabilisation and trajectory tracking with dynamic changes in the aircraft's centre of gravity. They designed three controllers: a linear PD controller, a feedback linearisation controller, and an adaptive controller. They proved that the adaptive controller is able to stabilise the quad-rotor and compensate for any changes in the centre of gravity, while the PD controller and feedback linearisation controller are not able to cope with dynamic changes in the centre of gravity. The same technique was implemented as Antonelli et al. (2013) considered some external disturbance in quad-rotor design such as the possibility of wrong estimation of the centre of mass.

There has also been research on the integration of multiple control techniques to develop a robust controller for quad-rotors under external disturbance and parameter uncertainty. Bouabdallah and Noth (2004) compared the PID controller with the LQ controller for stabilising an indoor quad-rotor. They concluded that the PID controller is able to stabilise the quad-rotor and achieve more robust results than the LQ controller. Adigbli et al. (2007) compared the back-stepping and sliding-mode control techniques against the PID controller. They showed that the PID controller cannot be used as an effective technique for tracking problems. Varga and Bogdan (2009) showed that a fuzzy Lyapunov-based controller can be used as an effective technique for tracking tasks with a predefined trajectory. Wai (2007) presented an adaptive fuzzy sliding-mode controller. Adjustment of the fuzzy parameters was based on Lyapunov stability theory. The controller was robust to uncertainties that exist in practical applications. Bouadi et al. (2011) handled modelling inaccuracies and presented an adaptive sliding mode controller for the quad-rotor attitude stabilisation and altitude trajectory tracking. Their adaptation law was based on Lyapunov stability theory.

3 Kinematics and dynamics of quad-rotors

To develop a control strategy, a mathematical model of the quad-rotor is derived based on its kinematics and dynamics. The kinematic equations provide a means to understand the motion of the vehicle, whereas the dynamic model provides an in-depth view of the relation that governs the forces and the resulting accelerations.

3.1 Kinematics of quad-rotors

In order to describe a quad-rotor's motion, two reference frames are needed and these are shown in Figure 2: an earth frame (E-frame) and a body frame (B-frame). A quad-rotor has 12 governing state variables which can be divided into four groups: three linear position variables, three linear velocity variables, three attitude variables, and three angular velocity variables. The linear position of the quad-rotor centre of mass is determined by the coordinates of the vector from the origin of the E-frame to the origin of the B-frame. The vector with respect to the E-frame is $P_E = [x \quad y \quad z]^T$. The vehicle's attitude is defined by the orientation of the B-frame with respect to the E-frame given by $A_E = [\Phi \quad \theta \quad \psi]^T$, where ϕ, θ and ψ denote the vehicle's roll, pitch, and yaw along the three orthogonal body axes *x*, *y*, *z*, respectively. The linear velocity of the quad-rotor is denoted by vector $V_B = [u \quad v \quad w]^T$, while the angular velocity is expressed with respect to the reference frame as $W_B = [p \quad q \quad r]^T$. Therefore, the linear position and the angular position are defined in the E-frame, while the linear velocity, the angular velocity, the forces and torques are defined in the B-frame. To calculate the linear velocity in the E-frame (V_E), V_B must be transformed using the rotation matrix *R* that goes from the B-frame to the inertial frame.

$$V_E = \dot{P}_E = R \cdot V_B \tag{1}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R. \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{2}$$

where *R* is the airframe orientation in space which can be written as $R = R(\Phi) \times R(\theta) \times R(\psi)$. Here, $R(\phi)$, $R(\theta)$, and $R(\psi)$ denote the rotations along *x*-axis (roll), *y*-axis (pitch), and *z*-axis (yaw), respectively.

$$R(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \text{-}\sin(\Phi) \\ 0 & \sin(\Phi) & \cos(\Phi) \end{bmatrix}, \quad R(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix},$$

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$R = \begin{bmatrix} \cos(\theta).\cos(\psi) & \sin(\Phi).\sin(\theta).\cos(\psi) - \cos(\Phi).\sin(\psi) & \cos(\Phi).\sin(\theta).\cos(\psi) + \sin(\Phi).\sin(\psi) \\ \cos(\theta).\sin(\psi) & \cos(\Phi).\cos(\psi) + \sin(\theta).\sin(\Phi).\sin(\psi) & \sin(\theta).\cos(\Phi).\sin(\psi) - \sin(\Phi).\cos(\psi) \\ -\sin(\theta) & \sin(\Phi).\cos(\theta) & \cos(\Phi).\cos(\theta) \end{bmatrix}. \tag{4}$$

Using equation (2), $\dot{x}, \dot{y}, \dot{z}$ can be written as:

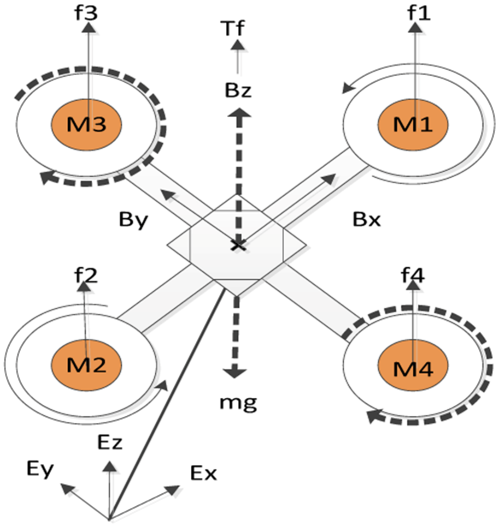
$$\dot{x} = (C\theta.C\psi\,)u + (-C\Phi.S\psi + S\Phi.C\psi.S\theta\,)V + (S\Phi.S\psi + C\Phi.S\theta.C\psi\,)w \tag{5}$$

$$\dot{y} = (C\theta.S\psi)u + (C\Phi.C\psi + S\Phi.S\psi.S\theta\,)V + (-S\Phi.C\psi + C\Phi.S\theta.S\psi\,)w \tag{6}$$

$$\dot{z} = (-S\theta)u + (S\Phi.C\theta)V + (C\Phi.C\theta)w, \tag{7}$$

where, *C* = cos(), *S* = sin(), *T* = tan().

Figure 2 The body and earth frames of a quad-rotor (see online version for colours)



As we did for the linear velocity, the angular velocity of the quad-rotor in the B-frame can be transferred to obtain one in the E-frame as:

$$W_E = \dot{A}_E = L_{ib}.W_B \tag{8}$$

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = L_{ib}.\begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{9}$$

where, L_{ib} is a rotation matrix on components from the inertial frame to the B-frame. It is given by:

$$L_{ib} = \begin{bmatrix} 1 & \tan(\theta).\sin(\Phi) & \tan(\theta).\cos(\Phi) \\ 0 & \cos(\Phi) & -\sin(\Phi) \\ 0 & \sec(\theta).\sin(\Phi) & \sec(\theta).\cos(\Phi) \end{bmatrix}. \tag{10}$$

Therefore, we have

$$\dot{\Phi} = p + q.\sin(\Phi).\tan(\theta) + r.\cos(\Phi).\tan(\theta) \tag{11}$$

$$\dot{\theta} = (\cos(\Phi)).q - (\sin(\Phi)).r \tag{12}$$

$$\dot{\psi} = q.\sin(\Phi).\sec(\theta) + r.\cos(\Phi)\ .\sec(\theta). \tag{13}$$

3.2 Dynamics of quad-rotors

The quad-rotor movement is controlled by the angular speed of the four rotors (W_i). Each rotor produces a thrust and a torque, and these two parameters generate the main thrust (T_f), airframe roll torque (τ_ϕ), pitch torque (τ_θ), and yaw torque (τ_ψ). The upward lift force and drag torque generated by each motor propeller are given by $f_i = D_m w_i^2$ and $\tau_i = D_t w_i^2$, respectively, where D_m and D_t are constant and represent the thrust and drag torque factors. As shown in Figure 2, the only forces acting on the quad-rotor are the total propeller thrust (T_f) and the gravitational force f_g. Therefore, the net force acting on the quad-rotor with respect to the E-frame is:

$$f = \mathrm{R}(F - f_d) + f_g, \tag{14}$$

where F is the thrust vector in the B-frame, f_g is the gravitational force, and f_d is the drag force. The total thrust applied to the quad-rotor is given by:

$$T_f = \sum_{i=1}^{4} fi = D_m \sum_{i=1}^{4} w_i^2, \tag{15}$$

Hence

$$F = \begin{bmatrix} 0 \\ 0 \\ T_f \end{bmatrix}. \tag{16}$$

The gravitational force applied to the quad-rotor with respect to the E-frame is given by:

$$f_g = -mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{17}$$

where m is the total mass of the quad-rotor and g is the gravity constant. A drag force acts on the quad-rotor body opposite to the direction it moves can be written as:

$$f_d = K_i \dot{P} = \begin{bmatrix} K_x \dot{x} \\ K_y \dot{y} \\ K_z \dot{z} \end{bmatrix} \tag{18}$$

where K_i is the drag coefficients and f_d is a function of several parameters such as aircraft speed, wing area, and air density. Let ρ, A, V and r denote the air density, frontal area perpendicular to the axis of motion, velocity relative to the air, and radius of rotation, respectively. By assuming that the density of the air is constant, we can express the above equation as:

$$f_{di} = \frac{1}{2} K_i \rho A V^2. \tag{19}$$

f_d is negligible at low speed and approximately zero when considering the dynamic behaviour of the quad-rotor. Therefore, the final equations that described the dynamic behaviour of the quad-rotor are given by:

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = R \begin{pmatrix} 0 \\ 0 \\ T_f \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \tag{20}$$

$$\ddot{x} = \frac{1}{m} T_f (\cos(\Phi).\sin(\theta).\cos(\psi) + \sin(\Phi).\sin(\psi)) \tag{21}$$

$$\ddot{y} = \frac{1}{m} T_f (\sin(\theta).\cos(\Phi).\sin(\psi) - \sin(\Phi).\cos(\psi)) \tag{22}$$

$$\ddot{z} = \frac{1}{m} T_f (\cos(\Phi).\cos(\theta)) - g. \tag{23}$$

The above equations can be simplified by defining a control input that represents the total thrust generated by the four propellers along z-axis. Let us define u_1 as:

$$u_1 = \sum_{i=1}^{4} w_i^2. \tag{24}$$

Then, the equation of motion can be reformulated as:

$$\ddot{x} = \frac{D_m}{m} u_1 (\cos(\Phi).\sin(\theta).\cos(\psi) + \sin(\Phi).\sin(\psi)) \tag{25}$$

$$\ddot{y} = \frac{D_m}{m} u_1 (\sin(\theta).\cos(\Phi).\sin(\psi) - \sin(\Phi).\cos(\psi)) \tag{26}$$

$$\ddot{z} = \frac{D_m}{m} u_1 (\cos(\Phi).\cos(\theta)) - g. \tag{27}$$

Airframe torques in the B-frame are denoted by roll, pitch, and yaw as:

$$j \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} - C_o \begin{pmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}, \tag{28}$$

Where, $C_o \begin{bmatrix} \dot{\Phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ represents the Coriolis term and will be ignored in the following equations:

$$\tau_\phi = j_x \ddot{\Phi} = \Delta Fx\, l \tag{29}$$

$$\tau_\theta = j_y \ddot{\theta} = \Delta Fy\, l \tag{30}$$

$$\tau_\psi = j_z \ddot{\psi} = \Delta Fz\, l. \tag{31}$$

In terms of the torque exerted by each motor, the above equations become:

$$\tau_A = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} (f_3 - f_4) l \\ (f_1 - f_2) l \\ \sum_1^4 \tau_{M_i} \end{bmatrix}. \tag{32}$$

Therefore,

$$\tau_\phi = (f_3 - f_4)l = lD_m(w_3^2 - w_4^2) = j_x\ddot{\Phi} \tag{33}$$

$$\tau_\theta = (f_1 - f_2)l = lD_m\left(w_1^{\,2} - w_2^{\,2}\right) = j_y\ddot{\theta} \tag{34}$$

$$\tau_\psi = \sum_1^4 \tau_{M_i} = \tau_{M_3} + \tau_{M_4} - \tau_{M_1} - \tau_{M_2} = j_z\ddot{\psi}, \tag{35}$$

where, l is the distance between the motors and the centre of gravity, and τ_{Mi} is the torque produced by each motor. The total torque produced by each rotor is equal to:

$$\tau_{mi} = j_r\dot{w}_i + D_t w_i^{\,2}, \tag{36}$$

where j_r is the rotational inertia of rotor i about its z-axis, and $\dot{w}_i$ is the angular acceleration of the rotor. In steady state, w is constant and $\dot{w}_i = 0$. Therefore, the torque produced by the rotor's propeller is equal to the reactive torque.

$$\tau_{mi} = D_t w_i^2 = \frac{D_t}{D_m} fi. \tag{37}$$

The total torque about the quad-rotor z-axis is given by the sum of all the torques as:

$$\tau_\psi = \sum_1^4 \tau_{M_i} = D_t\left(w_3^2 + w_4^2 - w_1^2 - w_2^2\right) = j_z\ddot{\psi} \tag{38}$$

$$\tau_\psi = \sum_1^4 \tau_{M_i} = \frac{D_t}{D_m}(f_3 + f_4 - f_1 - f_2) = j_z\ddot{\psi}. \tag{39}$$

Typically, equation (36) represents the motor dynamics as a simple first order differential equation. The above equations can be simplified by defining control inputs u_2, u_3 and u_4 that correspond to the collective roll, pitch, and yaw forces generated by the four propellers, respectively.

$$u_2 = (w_3^2 - w_4^2). \tag{40}$$

$$u_3 = (w_1^2 - w_2^2) \tag{41}$$

$$u_4 = f_3 + f_4 - f_1 - f_2. \tag{42}$$

Then the equation of motion can be reformulated as:

$$\ddot{\Phi} = u_2\, lD_m / jx \tag{43}$$

$$\ddot{\theta} = u_3\, lD_m / jy \tag{44}$$

$$\ddot{\psi} = u_4 C / jz, \tag{45}$$

where C is the force-to-moment scaling factor and j is the rotational inertia around x, y, and z. Finally, the equations of motions of the quad-rotor on the x, y and z become:

$$\ddot{x} = \frac{D_m}{m} u_1 (\cos(\Phi).\sin(\theta).\cos(\psi) + \sin(\Phi).\sin(\psi)) \tag{46}$$

$$\ddot{x} = \frac{D_m}{m} u_1 (\cos(\Phi).\sin(\theta).\cos(\psi) + \sin(\Phi).\sin(\psi)) \tag{47}$$

$$\ddot{z} = \frac{D_m}{m} u_1 (\cos(\Phi).\cos(\theta)) - g \tag{48}$$

$$\ddot{\Phi} = u_2 \, lD_m \, / \, jx \tag{49}$$

$$\ddot{\theta} = u_3 \, lD_m \, / \, jy \tag{50}$$

$$\ddot{\psi} = u_4 C \, / \, jz. \tag{51}$$

4 Quad-rotor structure

Figure 3 shows the system architecture of the quad-rotor. The hardware structure of the quad-rotor has two stages. The first stage consists of the mechanical parts, and the second stage consists of the electronic parts. As shown in Figure 4(a)–(c), the mechanical parts include a central hub, four aluminium booms, four motor mounts, four brushless DC motors, and four propellers (two of these rotate clockwise, and the other two rotate counter-clockwise). The frame of the quad-rotor is composed of four aluminium booms attached to the hub which carries the avionics and power supply. The 30-cm aluminium booms with 2 × 2 cm^2 cross-area are used because of their light weight. Motors and propellers are attached at the ends of the aluminium booms. Brushless three-phase out-runner DC motors rated for 1350 KV/310 W are used in this project. Four propellers 10 × 45 are mounted on the motors, two pieces for the standard rotation and two pieces for the right-hand rotation. The propeller is attached to the DC motor through a standard accessory pack.

Figure 3 Quad-rotor system architecture (see online version for colours)

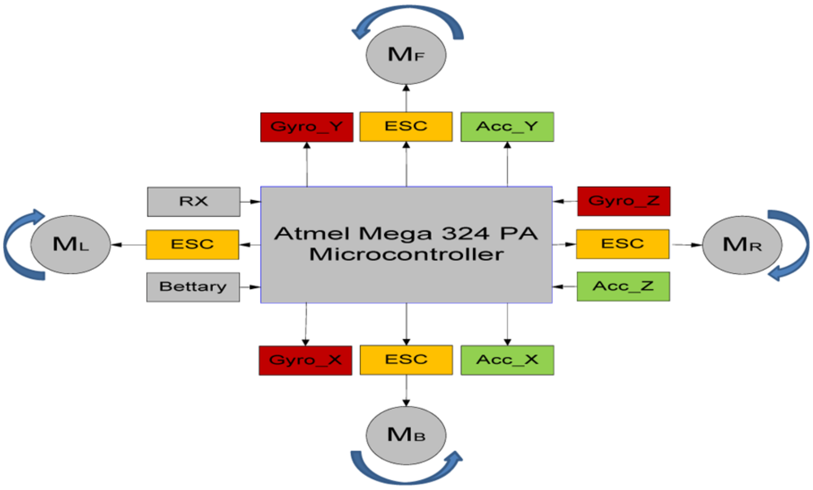
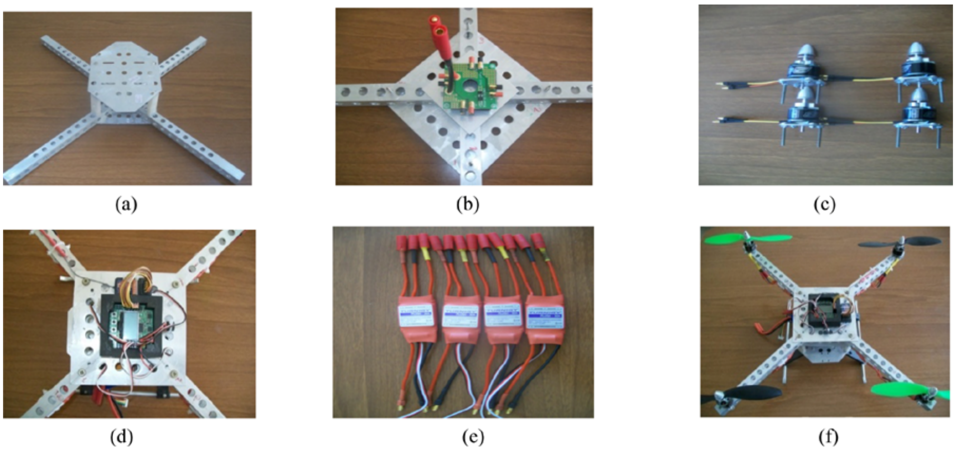


Figure 4 Structure of the quad-rotor: (a) aluminium based structure; (b) power distribution board; (c) brushless DC motors; (d) flight control board; (e) speed controllers and (f) quad-rotor frame (see online version for colours)



(a) (b) (c)

(d) (e) (f)

As shown in Figure 4(d)–(f), a Hobby King KK2.0 multi-rotor flight control board is attached to the central hub. This board consists of the AtmelMega324PA microcontroller, dual chip three-axis gyros, a single chip three-axis accelerometer, and an LCD for on-board configuration. To control the speed of the DC motors in order to control the quad-rotor, the flight control board is attached to the 30-A brushless speed controller (ESC). The ESC converts PWM signals into a three-phased signal, which continuously rotates the motor. The Atmel Mega324PA microcontroller takes sensor inputs along with the signal coming from the receiver and performs a direct MRAC on the four motors by varying the PWM signals. The system is powered by a 4000 mAh, 11.1 volt lithium polymer battery.

5 Flight controller design

This section details the application of the MRAC algorithm to the quad-rotor platform. The Lyapunov stability argument is used to design the adaptive controller and the reference model used for the controller is generated using the LQR and the quad-rotor dynamics and kinematics equations derived in the previous section. The control problem is formulated in the presence of the thrust loss that may occur due to component failure or physical damage.

5.1 The model reference adaptive control (MRAC) algorithm

Two major categories of the MRAC exist in the literature: direct and indirect. The direct method is selected in this work and its control structure is shown in Figure 5. It consists of a quad-rotor model to be controlled, a reference model, a controller, and an adjustment mechanism. The desired behaviour of the quad-rotor is represented by a reference model. The output of the quad-rotor model in this technique is compared against the output of the reference model that is driven by a reference signal. The error between the output of the reference model and the output of the quad-rotor model is used to drive the adjustment mechanism. The parameters of the controller are adjusted so as to eliminate the error between the quad-rotor model and the reference model. The LQR controller generates an input signal for the quad-rotor in order to follow a command signal, and Lyapunov stability theory is used for the adjustment mechanism to update the controller parameters so that the quad-rotor output will better match that of the desired reference model.

5.2 *Flight control algorithm*

In this section, we present the design of the flight control system that can stabilise the quad-rotor based on the two algorithms: LQR and MRAC.

Figure 5 Structure of a model reference adaptive control system (see online version for colours)

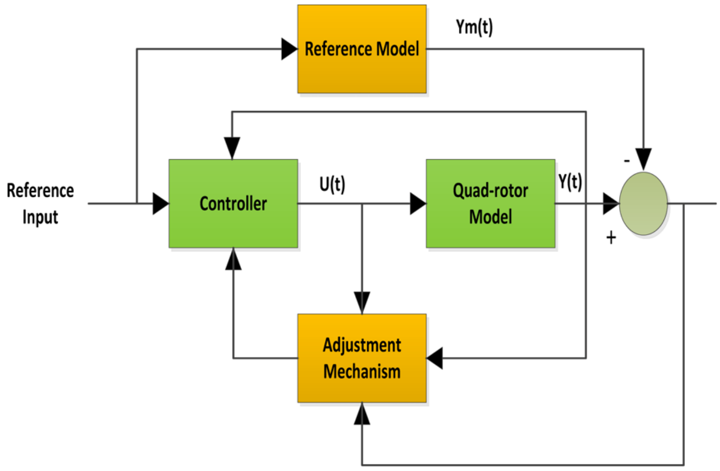
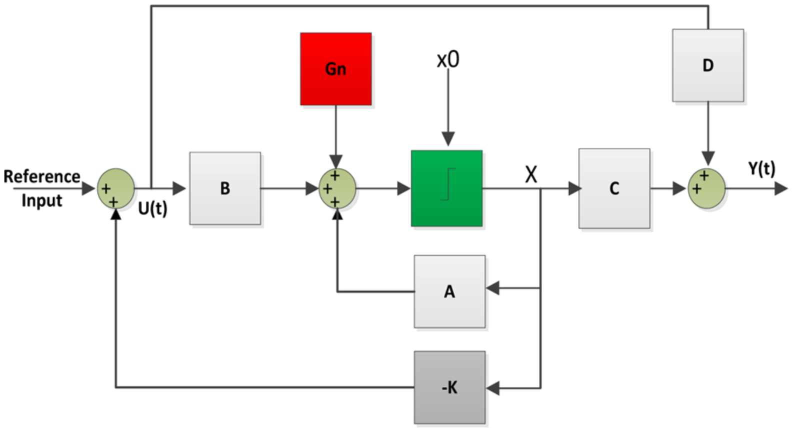


Figure 6 Full state feedback LQR controller (see online version for colours)



5.2.1 *Linear quadratic regulator (LQR)*

The full state feedback LQR controller is one of the most important state-space-based optimal controllers that provides practical feedback gains (Figure 6). In LQR problems, the system dynamics are described by a set of linear differential equations and the system cost is described by a quadratic function. The continuous-time linear system defined on the interval $t \in [t_0, t_1]$ can be written in the state-space form as:

$$\dot{x} = Ax + Bu + G_n \tag{52}$$

$$y = Cx + Du, \tag{53}$$

where $x \in R^n$, $\in R^m$, and $y \in R^p$. x, u, y, and G are the state vector of the system, the control input, the measured output, and the disturbance signal, respectively. $A \in R^{n \times n}$, $B \in R^n$, and C are the system matrix, control input matrix, and the output matrix, respectively. If the disturbance signal G_n is ignored, the LQR controller is given by:

$$u = -Kx + x_r = -\left[K_1\ K_2, \ldots, K_n\right] \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} x_{r1} \\ x_{r2} \\ \cdot \\ x_{rn} \end{bmatrix}. \tag{54}$$

The close-loop system using this control becomes:

$$\dot{x} = \left(A - BK\right)x + Bx_r, \tag{55}$$

where x_r is the reference value, which is assumed to be zero since it does not affect the stability of the system but affects the steady state error. The gain matrix K of the close-loop system which solve the LQR problem is:

$$K = R^{-1}B^T P, \tag{56}$$

where P is a unique, positive semi-definite solution to the Riccati equation given in equation (57), in order to minimise the cost function given in equation (58).

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{57}$$

$$J = \int_0^x \left(x^T Qx + u^T Ru\right) \mathrm{d}t. \tag{58}$$

Using the LQR-based controller, the first step is to select and tune the weighting matrices Q and R by simulations. Q is an $n \times n$ semi-positive definite symmetric matrix that weighs the states, and R is an $m \times m$ positive definite symmetric matrix that weighs the inputs. By solving equation (57) using MATLAB routine $\left[K, P\right] = \mathrm{lqr}(A, B, Q, R)$, the feedback K can be computed and $u = -R^{-1}B^T Px$ is optimal for any initial state $x(0)$. In order to use the designed LQR controller, the nonlinear system derived in equations (46)–(51) must be presented in a state-space form and linearised around a certain operating point. The state and input vectors are given in equations (59) and (60).

$$x = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \Phi\ \dot{\Phi}\ \theta\ \dot{\theta}\ \psi\ \ \dot{\psi}\ \ ]^T \tag{59}$$

$$u = [u_1\ u_2\ \ u_3\ \ u_4]^T \tag{60}$$

$$\begin{gathered} x_1 = x,\ x_2 = \dot{x}\ , x_3 = y, x_4 = \dot{y},\ x_5 = z,\ x_6 = \dot{z},\ x_7 = \Phi, \\ x_8 = \dot{\Phi},\ x_9 = \theta,\ x_{10} = \dot{\theta},\ x_{11} = \psi,\ x_{12} = \dot{\psi} \end{gathered} \tag{61}$$

$$\begin{gathered}\dot{x}_1 = x_2,\ \dot{x}_2 = \ddot{x},\ \dot{x}_3 = x_4,\ \dot{x}_4 = \ddot{y},\ \dot{x}_5 = x_6,\ \dot{x}_6 = \ddot{z},\ \dot{x}_7 = x_8,\ \dot{x}_8 = \ddot{\Phi}\ , \\ \dot{x}_9 = x_{10},\ \dot{x}_{10} = \ddot{\theta},\ \dot{x}_{11} = x_{12},\ \dot{x}_{12} = \ddot{\psi}. \end{gathered} \tag{62}$$

The non-linearised equations of motion can be written as:

$$\dot{x} = f\left(x,u\right) = \mathrm{d}/\mathrm{d}t \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{D_m}{m} u_1 (\cos x_7 . \sin x_9 . \cos x_{11} + \sin x_7 . \sin x_{11}) \\ x_4 \\ \frac{D_m}{m} u_1 (\sin x_9 . \cos x_7 . \sin x_{11} - \sin x_7 . \cos x_{11}) \\ x_6 \\ \frac{D_m}{m} u_1 (\cos x_7 . \cos x_9) - g \\ x_8 \\ u_2 l D_m / jx \\ x_{10} \\ u_3 l D_m / jy \\ x_{12} \\ u_4 C / jz \end{bmatrix}. \tag{63}$$

The nonlinear system $\dot{x} = f\left(x,u\right),\ \ y = h(x)$ should be linearised around an operating point (x_0, u_0). Consider the state as $x = x_0 + \Delta x,$ where x is the actual state, x_0 is the operating point, and Δx is the deviation from the operating point. $\Delta\dot{x} = \dot{x} - \dot{x}_0$, x_0 is constant and $\Delta\dot{x} = \dot{x} = f\left(x,u\right) = f\left(x_0 + \Delta x, u_0 + \Delta u\right)$. Simplify this equation using Taylor expansion by assuming that $\Delta x, \Delta_u$ are small, thus $\Delta\dot{x} = \dot{x} = f\left(x_0,u_0\right) + \partial f / \partial x\left(x_0,u_0\right)\Delta x$ $+\partial f / \partial u\left(x_0,u_0\right)\Delta u.$ By assuming that $\left(x_0,u_0\right) = 0,$ the previous equation becomes $\Delta\dot{x} = \dot{x} = A\Delta x + B\Delta u.$ Doing the same thing for the output $y,$ $y = h\left(x_0 + \Delta x\right)$ $= h\left(x_0\right) + \partial f / \partial x\left(x_0\right)\Delta x = C\Delta x$, where $h\left(x_0\right) = 0,$ and the matrices are:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \quad C = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix}. \tag{64}$$

As a starting point, assuming that all the states are measurable, let $u_2 = (u_{20} + \Delta u_2)/D_m$, $u_3 = (u_{30} + \Delta u_3)/D_m$, $u_4 = (u_{40} + \Delta u_4)/c,$ and let the total thrust be equal to $u_1 = u_{10} + \Delta u_1$. As we seek the hovering state, the following conditions should be hold: $x_0 = [0\ 0\ 0\ 0\ Z\ 0\ 0\ 0\ 0\ 0\ 0\ 0],$ $u_{10} = mg,$ $u_{20} = u_{30} = u_{40} = 0$. Applying equation (64) at the operating point, we have matrices A and B as:

$$A=\begin{bmatrix}0&1&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&g&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&-g&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\quad B=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\1/m&0&0&0\\0&0&0&0\\0&l/jx&0&0\\0&0&0&0\\0&0&l/jy&0\\0&0&0&0\\0&0&0&1/jz\end{bmatrix}. \tag{65}$$

5.2.2 Tracking a reference input

Our goal in this subsection is to drive the output $y(t)$ to a given bounded time-varying reference input $r(t)$. As shown in Figure 7, we add an integrator to the error signal and create a state within the controller that computes the integral of the error signal. The augmented open-loop dynamics of the system is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}x\\w\end{bmatrix}=\begin{bmatrix}Ax+Bu\\y-r\end{bmatrix}=\begin{bmatrix}Ax+Bu\\Cx-r\end{bmatrix}, \tag{66}$$

where w is a new state which integrates the controller error. Equation (66) can be written in state space as:

$$\begin{bmatrix}\dot{x}\\\dot{w}\end{bmatrix}=\begin{bmatrix}A&0\\C&0\end{bmatrix}\begin{bmatrix}x\\w\end{bmatrix}+\begin{bmatrix}B\\0\end{bmatrix}u+\begin{bmatrix}0\\-I\end{bmatrix}r\ . \tag{67}$$

This leads to the open-loop system

$$\dot{x}_a=A_a x_a+B_a u+B_r r. \tag{68}$$

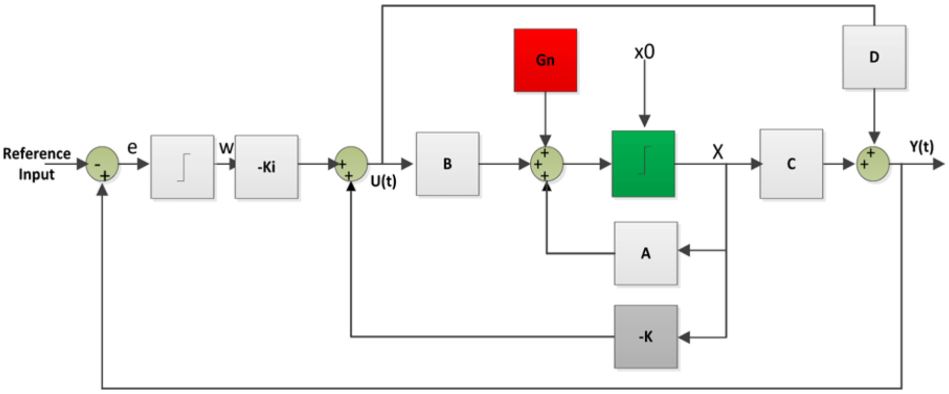
With the output $y=\begin{bmatrix}C & 0\end{bmatrix}x_a=C_a x_a$.

The output tracking error is given by $e_y(t)=y(t)-r(t)=\dot{w}$. The control input is $u=-Kx-K_i w$. The gain K_i does not affect the stability of the system, but it affects the steady state error. Control input stated above can be simplified to $=-K_a x_a$, where $K_a=\begin{bmatrix}K & K_i\end{bmatrix}$ and $x_a=\begin{bmatrix}x & w\end{bmatrix}^T$. The feedback gain K_a (nominal feedback gain K and feed forward gain K_i) is selected using the LQR technique. Therefore, under the assumption that all model parameters are known, the augmented close-loop system becomes:

$$\dot{x}_a=\left(A_a-B_a K_a\right)x_a+B_r r=\breve{A}_a x_a+B_r r, \tag{69}$$

where $x_a(t)$ is the augmented state and $r(t)$ is a continuous reference input.

Figure 7 Full state feedback LQR controller with an integrator (see online version for colours)



5.2.3 Design of adaptive controller

The MRAC approach based on the Lyapunov's stability theory is presented in this subsection. In this approach, the output response of the quad-rotor model is forced by the adaptive controller to track the response of a reference model irrespective of the quad-rotor parameter variations. The approach consists of two controllers: a baseline controller and an adaptive controller. The baseline controller is an augmented LQR controller that yields nominal performance in the absence of failure, while the adaptive controller is used to adjust controller parameters to eliminate the error between the quad-rotor model and the reference model outputs. Two parameters $\tilde{\theta}_1(t)$ and $\tilde{\theta}_2(t)$ are added to the baseline controller to define the adaptive controller law. These parameters mainly depend on the gain and are adjusted by the adaptation mechanism which is driven by the Lyapunov stability arguments.

Let the plant in the MRAC be modelled as $\dot{x}_p = A_p x_p + B_p \sigma u$ with two unknown parameters A_p, and σ. Here, $\sigma \in R^{m \times m}$ is an unknown diagonal matrix with strictly positive diagonal elements $0 < \sigma_i \leq 1$ that model actuator failures, and A_p models the parametric uncertainty. A_p, B_p, and σ_i can be written in the canonical form as:

$$A_a = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_1 & -a_2 & \ldots & -a_n \end{bmatrix} \quad B_p = \begin{bmatrix} B \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \quad \sigma_i = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_i \end{bmatrix}. \tag{70}$$

The reference model is given by the close-loop system $\dot{x}_m = A_m x_m + B_m \sigma_m r$, with the baseline controller $u = -K_m x_m$. A_m and B_m have the same shape as A_p and B_r given above except that a_i becomes $a_{mi} = 1$. By assuming that $\sigma_m = I^{m \times m}$. and A_m and B_m are known to the controller, the reference model becomes similar to the one given in equation (69).

To eliminate the effects of the parametric uncertainty such as mass, system inertia, thrust, and drag factors and the loss of actuator effectiveness, we choose the adaptive controller to be $u_{ad} = \tilde{\theta}(t) Z(t)$ such that $x_p \rightarrow x_m$ for any reference input signal $r(t)$. $\tilde{\theta}(t) = \left[\tilde{\theta}_1(t) \quad \tilde{\theta}_2(t) \right]$ represents the adaptive control gains that need to be adjusted through the adaptive law, and $Z = \left[x_p(t) \quad r(t) \right]^T$ represents the measured values. Adding the baseline to the adaptive controller, the proposed control law becomes $u(t) = u_{ad}(t) + u_{bl}(t) = \tilde{\theta}_1(t)x_p(t) + \tilde{\theta}_2(t)r(t) - k_p x_p(t)$. Substitute this control law in the model that is dedicated to the MRAC, the close-loop system becomes:

$$\dot{x}_p = \left(A_p + B_p\sigma\tilde{\theta}_1 - \sigma B_p K_p\right)x_p + \left(B_p\sigma\tilde{\theta}_2\right)r. \tag{71}$$

If we define the adaptive error terms as:

$$\theta_1(t) = \tilde{\theta}_1(t) - \theta_1^* \tag{72}$$

$$\theta_2(t) = \tilde{\theta}_2(t) - \theta_2^*. \tag{73}$$

Equation (71) becomes:

$$\dot{x}_p = \left(A_p + B_p\sigma\theta_1 + B_p\sigma\theta_1^* - B_p\sigma K_p\right)x_p + \left(B_p\sigma\theta_2 + B_p\sigma\theta_2^*\right)r. \tag{74}$$

It is clear from the above equation that there exist constant parameters like θ_1^* and θ_2^* such that $(A_p + B_p\sigma\theta_1^* - B_p\sigma K_p) = \tilde{A}_m$, and $\left(B_p\sigma\theta_2^*\right) = \sigma_m B_m = \tilde{B}_m$. In other words, if the adaptive term $\tilde{\theta}$ is adjusted as it approaches θ^*, the adaptive error term θ will converge to zero, and the dynamics of the close-loop system will match the dynamics of the new reference model:

$$\dot{x}_m = \tilde{A}_m x_m + \tilde{B}_m r \tag{75}$$

Define the close-loop tracking error dynamics of the MRAC as $e(t) = x_p(t) - x_m(t)$. Therefore, differentiating $e(t)$ yields

$$\dot{e}(t) = \tilde{A}_m e(t) + (B_p\sigma\theta_1(t))x_p(t) + (B_p\sigma\theta_2(t))r(t). \tag{76}$$

If we define the parameter estimate error to be $\theta(t) = \left[\theta_1(t) \quad \theta_2(t)\right]^T$ and $(t) = \left[x_p(t) \quad r(t)\right]^T$, we have:

$$\dot{e}(t) = \tilde{A}_m e(t) + (B_p\sigma\theta(t))x(t). \tag{77}$$

To design a suitable adaptation law, let us define the Lyapunov function as:

$$v\left(e(t), \theta(t)\right) = e^T(t)Pe(t) + |\sigma|\theta^T(t)\Gamma^{-1}\theta(t), \tag{78}$$

where, $\Gamma > 0$ is a diagonal positive definite matrix of the adaptive gains, and P is a unique symmetric positive definite solution to the algebraic Lyapunov equation given by:

$$\tilde{A}_m^T P + P\tilde{A}_m = -Q, \tag{79}$$

with Q being any symmetric positive definite matrix. By differentiating $V(e, \theta)$ with respect to time, we have:

$$\begin{aligned}\dot{V} &= \dot{e}^T Pe + e^T P\dot{e} + |\sigma|\dot{\theta}^T\Gamma^{-1}\theta + |\sigma|\theta^T\Gamma^{-1}\dot{\theta} \\ &= 2e^T P\dot{e} + 2|\sigma|\theta^T\Gamma^{-1}\dot{\theta} \\ &= 2e^T P\left[\tilde{A}_m e + B_p\sigma\theta^T x\right] + 2|\sigma|\theta^T\Gamma^{-1}\dot{\theta} \\ &= -e^T Qe + 2\sigma\theta^T(e^T PB_p x + \Gamma^{(-1)}\dot{\theta}).\end{aligned} \tag{80}$$

Clearly, by choosing the parameter update rule to be $\dot{\theta} = -\Gamma x e^T P B_p$, the Lyapunov theorem is satisfied and error is minimised. Finally, the adaptive terms are given as:

$$\begin{bmatrix} \dot{\hat{\theta}}_1 \\ \dot{\hat{\theta}}_2 \end{bmatrix} = -\operatorname{sgn}(\sigma)\Gamma \begin{bmatrix} x_p \\ r \end{bmatrix} e^T P B_p. \tag{81}$$

6 Simulation and experimental results

A simulation and a flight test in an outdoor environment were conducted to validate the control approaches.

6.1 Quad-rotor parameters

As shown in Figure 8, the arm length (L) of the quad-rotor is 30 cm and the total mass (m_t) of the quad-rotor is 1619 g. Each single rotor weighs 57 g and thus the net weight of the rotors $m_r = m_1 + m_2 + m_3 + m_4 = 228$ g. To simplify the process, we assume that the total mass of the quad-rotor excluding the net weights of the rotors is homogeneous and distributed inside the sphere of radius R = 7 cm. The moment of inertia of the sphere is $j_R = \frac{2}{5}(m_t - m_r)R^2$, and the moment of inertia for a point-mass (motor) is $j_r = m_r L^2$. The moment of inertia along axes x, y and z can be calculated as:

$$j_x = j_R + 2j_r = \frac{2}{5}(1.391)(0.07)^2 + 2(0.057)(0.3)^2 = 0.001298636 k_g.m^2$$

$$j_y = j_R + 2j_r = \frac{2}{5}(1.391)(0.07)^2 + 2(0.057)(0.3)^2 = 0.01298636 k_g.m^2$$

$$j_z = j_R + 4j_r = \frac{2}{5}(1.391)(0.07)^2 + 4(0.057)(0.3)^2 = 0.02324636 k_g.m^2.$$

Therefore, the inertial matrix is given by:

$$J = \begin{bmatrix} j_x & 0 & 0 \\ 0 & j_y & 0 \\ 0 & 0 & j_z \end{bmatrix} = \begin{bmatrix} 0.013 & 0 & 0 \\ 0 & 0.013 & 0 \\ 0 & 0 & 0.023 \end{bmatrix}.$$

The thrust and drag torque created by each rotor due to the rotation of the propeller can be calculated as:

$$f_i = D_m w_i^2, D_m = c_m \pi\rho(r)^4$$

$$\tau_i = D_t w_i^2, D_t = c_t \pi\rho(r)^5,$$

where C_m is the propeller thrust coefficient, C_t is the propeller drag torque coefficient, ρ is the density of the air, and r is the radius of the rotor blade (12.7 cm). Consider $C_m = 0.1154$, $C_t = 0.0743$ for a propeller 1045, air density is 1.225 kg/m^3. Therefore, D_m = 0.000092 kg.m, D_t = 0.0000071 kg.m^2, and the force-to-moment scaling factor can be calculated as:

$$\tau_i = (D_t / D_m).f_i$$

$$C = \frac{D_t}{D_m} = 0.114 \text{ m.}$$

Other parameters are shown in Table 1 and Matrices A and B are shown below.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.617 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 23.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 23.1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 43.5 \end{bmatrix}.$$

Consequently, reference inputs are chosen as $R = [x_r \ y_r \ z_r \ \psi_r]^T$, and the four desired outputs are chosen as *x*, *y*, *z*, and yaw angle *ψ*. Therefore the output vector, *y*, is defined as $y = [x \ y \ z \ \psi]^T$, the output matrices *D* = 0, and *C* is given as:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

On the other hand, each rotor has an angular speed according to $w = \frac{2\pi}{60} K_v (v_m - v_0)$, where v_m is the motor's power supply voltage, v_0 is the motor's dead zone voltage, Kv is the RPM per volt, and the term $\frac{2\pi}{60} K_v$ represents the propeller's angular speed per volt. In this work, a brushless three phase out-runner DC motors rated for 1350 KV/310 W is used, powered by a 4000 mAh, 11.1 Volt lithium polymer battery, hence w = 1414 rad/s. On the other hand, to achieve the hover state, the net force acting on the quad-rotor is set to zero, and the angular speed is given by $w_i = \sqrt{mg / 4D_m}$, hence w_i = 208 rad/s. When this value is applied to the formula $f_i = D_m w_i^2$, it results that each propeller should provide 4 N (0.408 kg) in order to achieve hovering. This is acceptable since the total thrust required by the four propellers is equal to (9.81 × 1.619) N.

Table 1 Parameters of a quad-rotor

Parameters	*Value*	*Unit*	*Description*
L	0.30	m	Arm length
G	9.81	m/s^2	Gravitational constant
m_r	0.228	kg	Rotors mass
m_t	1.619	kg	Total mass of the quad-rotor
j_x	0.013	kg.m^2	Moment of inertia around *X* axes
j_y	0.013	kg.m^2	Moment of inertia around *Y* axes
j_z	0.023	kg.m^2	Moment of inertia around *Z* axes
W	990	rad/s	Rotor angular speed
D_m	0.000092	kg.m	Thrust coefficient
D_t	0.0000071	kg.m^2	Torque coefficient

6.2 Simulation results

To design a position controller for the quad-rotor under study, the weighting matrices of the LQR controller are chosen to be R = diag(0.01, 0.01, 0.015, 0.015) and Q = diag(600 100 600 100 150 30 1 0 1 0 1400 60). The gain matrix K is computed using a linearised model of the quad-rotor, and then the performance of the LQR controller is evaluated on the nonlinear model. As a starting point, let the initial position of the quad-rotor be P_i = (2, –3, 4) and initial attitude α_i = (0, 0, pi/4). The task is to stabilise and take the aircraft from this situation to a new one with the reference inputs P_r = (0,0,0), and α_i = (0,0,0). Figure 9 shows the simulation result under ideal condition without any external disturbance.

Figure 8 Quad-rotor schematic showing the total mass of the quad-rotor (see online version for colours)

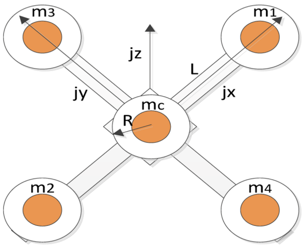
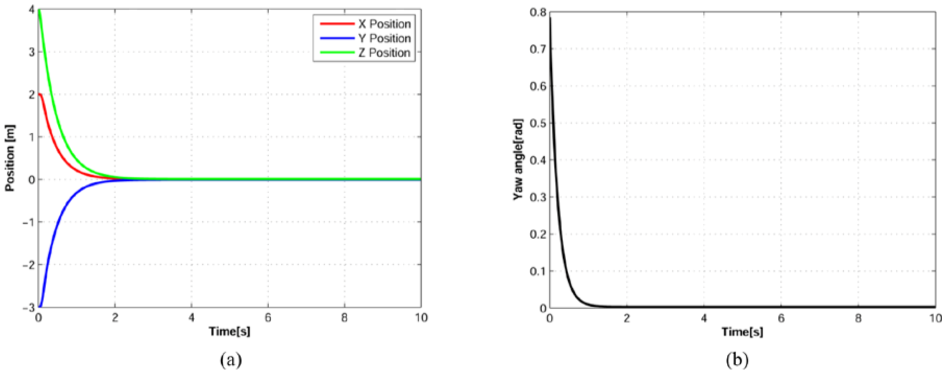


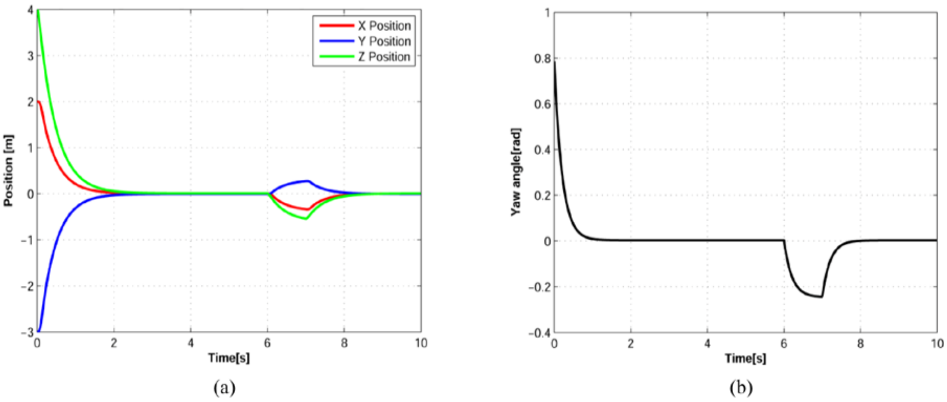
Figure 9 Time response of the LQR controller under no uncertainty condition: (a) evolution of *x*, *y*, and *z* positions and (b) evolution of yaw angle (see online version for colours)



As can be seen in Figure 9, the output of the quad-rotor is successfully driven towards the reference input. This implies that with a reasonable choice of weighting matrices, the performance of the LQR controller is satisfactory under the absence of parametric uncertainties and it is able to make the quad-rotor follows the desired trajectory. This verifies the theoretical claim in equation (55) since disturbance signal G_n is ignored.

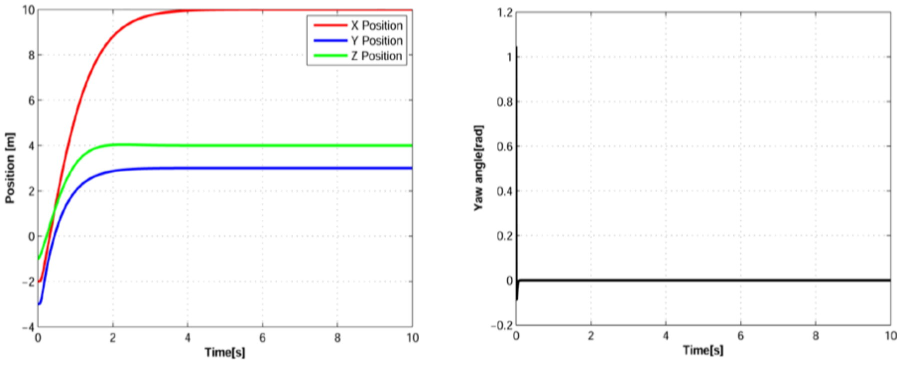
In the second scenario, the quad-rotor UAV is commanded to hover at a fixed position. A simulated loss of actuator thrust is injected into the system at time $t = 6$ s. The resulting performance is shown in Figure 10.

Figure 10 Time response of the LQR controller under a partial loss of actuator thrust: (a) evolution of *x*, *y*, and *z* positions and (b) evolution of yaw angle (see online version for colours)



As can be seen in Figure 10, the LQR controller exhibits some degradation in the performance after 6 s of simulation and does not respond quickly to the loss in actuator thrust. This is expected because LQR is a fixed gain controller. In the third scenario, our goal is to drive the output $y(t)$ to a given bounded time-varying reference input $r(t)$. We add an integrator to the error signal and then the LQR controller is evaluated on the nonlinear model. The initial position of the quad-rotor is Pi = (–2, –3, –1) and initial attitude is αi = (0, 0, pi/3). The task is to stabilise and take the aircraft from this situation to a new one with the reference inputs Pr = (10, 3, 4) and $\alpha i = (0, 0, 0)$. Figure 11 shows the simulation result of the augmented close-loop dynamics of the system under ideal condition without any external disturbance. As can be seen, the output of the quad-rotor is successfully driven towards the desired trajectory. This verifies the theoretical claim in equation (69).

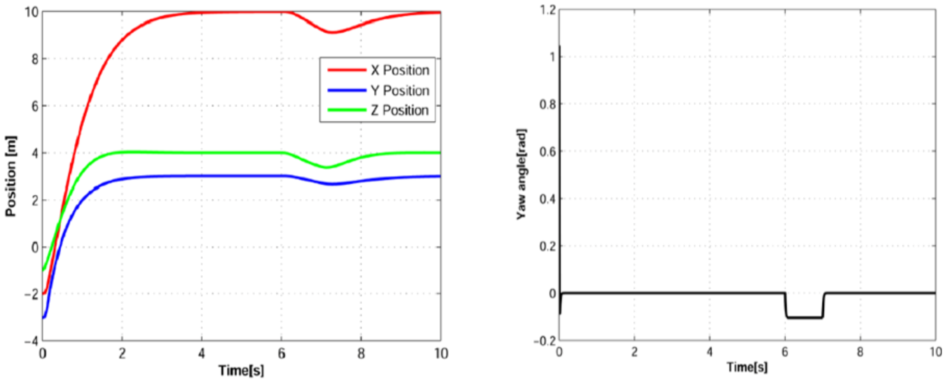
Figure 11 Time response of the augmented LQR controller under no uncertainty condition: (a) evolution of *x*, *y*, and *z* positions and (b) evolution of yaw angle (see online version for colours)



(a) (b)

In the fourth scenario, 25% loss of control effectiveness is initiated in the fourth actuator at time $t = 6$ s. As shown in Figure 12, the LQR experiences a deviation of 45 cm in the position and over five degrees in yaw for the quad-rotor under test. This deviation causes the quad-rotor to fly at an altitude of 4.55 m during the loss of 25% of thrust and then return back to the normal situation. This is because the controller has been augmented with an integrator to the error signal.

Figure 12 Time response of the augmented LQR controller under a partial loss of actuator thrust: (a) evolution of *x*, *y*, and *z* positions and (b) evolution of yaw angle (see online version for colours)

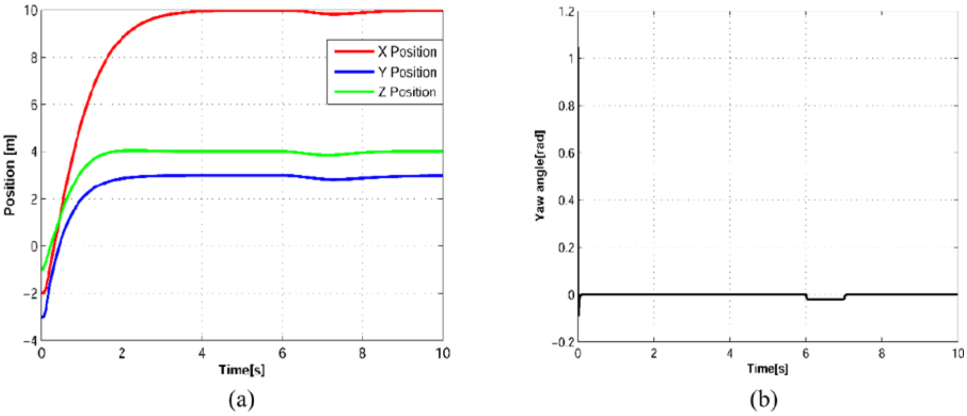


(a) (b)

In the fifth scenario, we examine the performance of the MRAC controller by commanding the quad-rotor to hover at a fixed position. The setting for this scenario is similar to the fourth scenario and the resulting performance is shown in Figure 13. As can be seen, the MRAC controller responds quickly to the partial loss of actuator thrust. This is because the adjustable controller parameters changed accordingly to compensate for the loss of control effectiveness immediately after the loss of 25% of thrust.

Finally, performances of the baseline controller and the adaptive controller are compared in Figure 14. The dashed lines and the solid lines represent results from the LQR and MRAC controllers respectively. The results indicate that the LQR experiences a deviation and is unable to keep the quad-rotor hovering at a fixed position, while the MRAC controller responds quickly and is able to maintain stability. This verifies the theoretical claim since LQR is a fixed gain controller, and it cannot provide robustness towards parametric uncertainties as the varying gain adaptive controller.

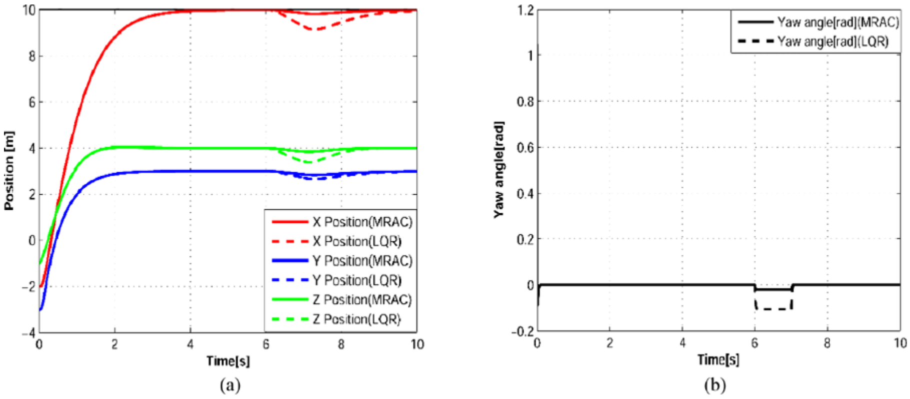
Figure 13 Time response of the MRAC controller under a partial loss of actuator thrust: (a) evolution of *x*, *y*, and *z* positions and (b) evolution of yaw angle (see online version for colours)



(a)

(b)

Figure 14 Comparison between the LQR controller and the MRAC controller: (a) evolution of *x*, *y*, and *z* positions and (b) evolution of yaw angle (see online version for colours)



(a)

(b)

6.3 Experimental results

As shown in Figure 4(f), many holes were drilled to decrease the total weight of the quad-rotor. The Atmel Mega324PA microcontroller, the dual chip three-axis gyros and the single chip three-axis accelerometer are located in a box so as to avoid vibration from the propellers. To the right of the flight controller, an RF receiver is attached to the central hub and connected to the microcontroller. ECEs are fixed on the aluminium booms using plastic cable ties. Several experiments have been conducted to balance the propellers and the actuators. As shown in Figure 15, two experiments were conducted to test the quad-rotor's hovering and attitude stabilisation capabilities with a faulty bearing installed in motor number four.

Brushless three-phase out-runner DC motors are used in this project; namely model NTM (26-28 1350Kv/310 W). This model required two kinds of bearings (3 × 7 × 3 mm and 3 × 8 × 4 mm) and have a 3mm shaft. Experiments were conducted on four motors: three of them are undamaged while the fourth one is damaged with a faulty bearing. The fault in bearing was made by drilling a hole of 1mm diameter in its outer race.

In the first experiment, the LQR was implemented, and the quad-rotor was commanded to maintain a fixed hover position. As demonstrated in Figure 15(a) and (b), the controller was unable to quickly bring the pitch angle error back to a safe range, which resulted in a crash. In the second experiment, MRAC was implemented, and the quad-rotor was commanded as we did in the first experiment to hover at a fixed position. As demonstrated in Figure 15(c) and (d), the controller was able to keep the pitch angle within an acceptable range, which indeed maintained the quad-rotor hovering at 4 metre above ground.

Figure 15 Snapshots of the quad-rotor in flight demonstrating the difference in behaviour between the LQR and the MRAC controller: (a) the LQR responses at low altitude using a single faulty bearing; (b) the LQR responses at high altitude using a single faulty bearing; (c) the MRAC responses at low altitude using a single faulty bearing and (d) the MRAC responses at high altitude using a single faulty bearing (see online version for colours)

(a)

(b)

(c)

(d)

In both experiments, bearing fault is practically implemented and their effects on quad-rotor stability are studied with the help of LQR and MRAC controllers. There were no significant differences between simulation results and flight test results. The only difference is the crash that occurred during the first test which may be undesirable especially if, for example, the quad-rotor is operating at low altitude.

7 Conclusion

This paper presents a quad-rotor UAV which was developed and tested. First, in order to eliminate the effects of parametric uncertainty and the loss of actuator effectiveness during outdoor flight, the MRAC algorithm based on Lyapunov stability was designed, tested, and compared with the LQR controller with full state feedback and integral action. Second, both controllers were experimentally tested using an AtmelMega 324PA microcontroller. They provided good results for height and attitude stabilisation under normal conditions. In the presence of partial loss of thrust, the MARC controller was shown to be more effective at stabilising the quad-rotor, whereas the LQR controller was not able to compensate for the dynamic changes in the quad-rotor.

References

Adigbli, P., Grand, C., Mouret, J. and Doncieux, S. (2007) 'Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques', Paper presented at the *7th European Micro Air Vehicle Conference*, 17–21 September, Toulouse, France, pp.1–9.

Antonelli, G., Arrichiello, F., Chiaverini, S. and Robuffo, P. (2013) 'Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances', *ASME 2013: Proceedings IEEE International Conference on Advanced Intelligent Mechatronics*, Wollongong, AU, pp.1337–1342.

Beji, L., Zemalache, K. and Marref, H. (2005) 'Control of an under-actuated system: application to a four rotors rotorcraft', *ROBIO* 2005*: IEEE International Conference on Robotics and Biomimetic*, Shatin, China, pp.404–409.

Benallegue, A., Mister, V. and Sirdi, N. (2001) 'Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback', *Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication*, Bordeaux, Paris, pp.586–593.

Bouabdallah, S. and Noth, A. (2004) 'PID vs LQ control techniques applied to an indoor micro quadrotor', *IROS 2004: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp.2451–2456.

Bouabdallah, S., Murrieri, P. and Siegwart, R. (2005) 'Towards autonomous indoor micro VTOL', *Autonomous Robots*, Vol. 18, No. 2, pp.171–183.

Bouadi, H., Cunha, S., Drouin, A. and Camino, F. (2011) 'Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking', *Proceeding of 12th IEEE International Symposium on Computational Intelligence and Informatics*, Budapest, Hungary, pp.449–455.

Castillo, P. and Dzul, A. (2004) 'Real-time stabilization and tracking of a four rotor mini rotorcraft', *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 4, pp.510–516.

Castillo, P., Lozano, R. and Dzul, A. (2005) 'Stabilization of a mini rotorcraft with four rotors', *IEEE Control Systems*, Vol. 25, No. 6, pp.45–55.

Costa, R., Hsu, L., Imai, A. and Kokotovic, P. (2003) 'Lyapunov-based adaptive control of MIMO systems', *Automatic*, Vol. 39, No. 7, pp.1251–1257.

Coza, C. and Macnab, C. (2006) 'A new robust Adaptive-fuzzy control method applied to quadrotor helicopter stabilization', *NAFIPS 2006*: *fuzzy Information Processing Society*, pp.454–458.

Das, A., Subbarao, K. and Lewis, F. (2009) 'Dynamic inversion with zero-dynamics stabilization for quadrotor control', *IET Control Theory and Applications*, Vol. 3, No. 3, pp.303–314.

Hamel, T., Guenard, N. and Mahony, R. (2007) 'A practical visual servo control for unmanned aerial vehicle', *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, pp.1342–1348.

Hamel, T., Metni, N. and Derkx, F. (2005) 'Visual tracking control of aerial robotic systems with adaptive depth estimation', *CDC-ECC 2005: Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, 12–15 December, Spain, pp.6078–6084.

Hanford, S.D., Long, L.N. and Horn, J.F. (2005) 'A small semi-autonomous rotary-wing unmanned air vehicle (UAV)', Paper presented at the *Infotech Aerospace Conference 2005*, Washington DC, USA.

Hoffmann, G.M., Huang, H., Wasl, S.L. and Tomlin, C.J. (2007) 'Quadrotor helicopter flight dynamics and control: theory and experiment', *AIAA 2007: Proceedings of the AIAA Guidance, Navigation,* and *Control Conference*, Hilton Head, South Carolina, pp.1–20.

Hsu, L. (1990) 'Variable structure model reference adaptive controller (VSMRAC) 'using only input and output measurement: the general case', *IEEE Transactions on Automatic Control*, Vol. 35, No. 11, pp.1238–1243.

Jaimes, A., Kota, S. and Gomez, J. (2008) 'An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles UAV(s)', *SoSE 2008: Proceedings of the IEEE International Conference on System of Systems Engineering*, Singapore, pp.1–6.

Kaufman, H., Barkana, I. and Sobel, K. (1998) *Direct Adaptive Control Algorithms: Theory and Applications*, 2nd ed., New York.

Madani, T. and Benallegue, A. (2006) 'Control of a quadrotor mini-helicopter via full state back stepping technique', *Decision and Control 2004: Proceedings of 45th IEEE Conference on Decision and Control*, San Diego, USA, pp.1515–1520.

Middleton, R. and Goodwin, G. (1988) 'Adaptive control of time-varying linear system', *IEEE Transactions on Automatic Control*, Vol. 33, No. 2, pp.150–155.

Mokhtari, A. and Benallegue, A. (2004) 'Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle', *ICRA 2004: Proceedings of the IEEE International Conference on Robotics and Automation*, 18–22 April, Barcelona, Spain, pp.2359–2366.

Morel, Y. and Leonessa, A. (2006) 'Direct adaptive tracking control of quadrotor aerial vehicles', *FCRAR 2006: Florida Conference on Recent Advances in Robotics*, Miami, Florida, pp.155–161.

Palunko, I. and Fierro, R. (2011) 'Adaptive control of a quadrotor with dynamic changes in the center of gravity', Paper Presented at the *18th IFAC World Congress*, 28 August 2011, Milano, Italy.

Pounds, P., Mahony, R. Hynesand, P. and Roberts, J. (2002) 'Design of a four-rotor aerial robot', *ACRA 2002: Proceedings of the Australasian Conference on Robotics and Automation*, Auckland NZ.

Qu, Z., Dorsey, J.F. and Dawson, D.M. (1994) 'Model reference robust control of a class of SISO systems,' *IEEE Transactions on Automatic Control*, Vol. 35, No. 11, pp.2219–2234.

Sadeghzadeh, I., Mehta, A., Zhang, Y. and Rabbath, C. (2011) 'Fault-tolerant trajectory tracking control of a quadrotor helicopter using gain-scheduled PID and model reference adaptive control', Paper presented at the *Annual Conference of the Prognostics and Health Management Society*, 23 August 2011, Montreal, QC, Canada.

Sastry, S. and Bodson, M. (1989) *Adaptive Control-Stability, Convergence and Robustness*, Prentice-Hall Advanced Reference Series, Republished by Dover Publications in 2011.

Swarnkar, P., Jain, S. and Nema, R. (2011) 'Comparative analysis of MIT rule and Lyapunov in model reference adaptive control scheme', *Innovative Systems Design and Engineering*, Vol. 2, No. 4, pp.154–162.

Tarbouchi, M., Dunfied, J. and Labonte, G. (2004) 'Neural network based control of a four rotor helicopter', *ICIT 2004: Proceedings of the IEEE International Conference on Industrial Technology*, pp.1543–1548.

Tayebi, A. and McGilvray, S. (2006) 'Attitude stabilization of a VTOL quadrotor aircraft', *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 3, pp.562–571.

Varga, M. and Bogdan, S. (2009) 'Fuzzy-Lyapunov based quadrotor controller design', *Proceedings of the European Control Conference*, Budapest, Hungary, pp.928–933.

Wai, R. (2007) 'Fuzzy sliding mode control using adaptive tuning technique', *IEEE Transaction on Industrial Electronics*, Vol. 54, No. 1, pp.586–594.

Whitehead, B. and Bieniawskiy, S. (2010) 'Model reference adaptive control of a quadrotor UAV', Paper presented at the *AIAA Guidance, Navigation, and Control Conference*, 2–5 August, Toronto, Ontario, Canada, pp.4944–4957.