文章编号：1671-7848(2016)12-1874-09 **DOI:** 10.14107/j.cnki.kzgc.SI16001

Design and Implementation of Fault-Tolerant Control Algorithms for an Unmanned Quadrotor System

ZHANG You-min[1,2], *YU Xiang*[2,3], *WANG Ban*[2], *LIU Ding*[1]

(1. a. Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing; b. Department of Information and Control Engineering, Xi'an University of Technology, Xi'an 710048, China; 2. Department of Mechanical and Industrial Engineering, Concordia University, Montreal H3G 1M8, Canada; 3. a. State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body; b. College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410072, China)

Abstract: Unmanned aerial vehicles (UAVs) have become more promising and developed very rapidly over the past decade. As a key and enabling technique for ensuring the safe flight as well as wide and various practical uses of UAVs, fault-tolerant control (FTC) techniques have been drawn more attention and developed in the presence of actuator faults in recent years worldwide. This paper first presents a brief overview of several existing UAVs platforms and brief review to the existing FTC techniques applied to UAVs. Then, descriptions on a developed unmanned quadrotor helicopter system (named as Qabll-X4) are stated with emphasis on the hardware, software, and modeling aspects of a UAV system from control engineering system viewpoint. Finally, two newly investigated FTC algorithms with experimental tests to the Qball-X4 UAV are presented.

Key words: Unmanned aerial vehicle; fault-tolerant control; unmanned quadrotor system

中图分类号：TP18 文献标识码：A

四旋翼无人系统的容错控制算法设计与实现

张友民 [1,2]，余翔 [2,3]，王斑 [2]，刘丁 [1]

(1. 西安理工大学 a. 陕西省复杂系统控制和智能信息处理重点实验室；b. 自动化与信息工程学院信息与控制工程系，西安 710048；2. 康考迪亚大学 机械与工业工程系，蒙特利尔 H3G 1M8，加拿大；3. 湖南大学 a. 汽车车身先进设计与制造国家重点实验室；b. 机械与运载工程学院，长沙 410072)

摘　要：无人机在过去十年中发展非常迅速且应用越来越广泛。作为确保无人机安全飞行和实际使用的关键技术，容错控制技术近年来也已经在世界范围内引起了极大的关注、并针对无人机执行器故障开展了相当多的研究。首先概述了无人机的分类及容错控制技术的研究概况。然后对一款基于工业界与学术界有效合作之成果的四旋翼无人直升机系统 (Qball-X4) 进行了描述，并重点介绍了此无人机系统的硬件和软件组成。最后介绍了基于 Qabll-X4 的 2 个最近研究的容错控制算法的设计和实现，并进行了实验测试与验证。

关键词：无人机；容错控制；四旋翼无人系统

1 Introduction

As compared to a *manned* aircraft, a powered aerial vehicle without a pilot onboard, which is capable of 1) flying in an autonomous or remotely controlled manner and 2) carrying a payload to complete specific missions, is named as an *unmanned* aerial vehicle (UAV) [1]. UAVs have gained increasing interest in

收稿日期：2016-01-17；**修回日期：**2016-10-14

基金项目：国家自然科学基金(61573282, 61603130)；陕西省自然科学基金(2015JZ020).

作者简介：张友民(1963-)，男，陕西大荔人，博士，教授，博士生导师，主要从事无人系统（容错）导航、制导与控制及其应用以及智能电网系统故障诊断与容错控制等方面的教学与科研工作；余翔(1981-)，男，湖南株洲人，博士，副教授，主要从事核心安全系统容错控制和智能无人系统导航、制导与控制等方面的教学与科研工作；王斑(1989-)，男，山东烟台人，博士生，主要研究方向为飞行器容错控制；刘丁(1957-)，男，陕西西安人，博士，教授，博士生导师，主要从事自动制理论与应用、复杂系统与工业过程控制等方面的教学与科研工作。

both civilian and military domains due to the promising potential in intelligence, surveillance, and reconnaissance (ISR) applications. As shown in Fig. 1, existing UAVs can be grouped into fixed-wing, rotary-wing, and tilt-wing types, respectively. According to the characteristics of UAV weight, flying range, endurance, and altitude, UAVs can also be classified into several categories as indicated in Tab. 1 and Tab. 2.

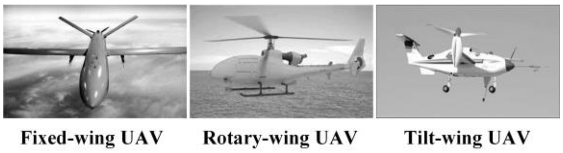

Fig. 1 UAV classification based on wing configuration

Fig. 1 基于机翼布局的无人机分类

Tab. 1 Classification of UAVs by weight

Tab. 1 基于重量的无人机分类

Category	Gross Weight
Micro	≤5 kg
Light	5~50 kg
Medium	50~200 kg
Heavy	200~2 000 kg
Super Heavy	≥2 000 kg

Tab. 2 Classifications of UAVs by endurance, range, and altitude

Tab. 2 基于续航、航程和高度的无人机分类

Category	Flying Range	Endurance	Altitude
Low	≤100 km	≤5 hours	≤1 km
Medium	100~400 km	5~24 hours	1~10 km
Long/High	≥400 km	≥24 hours	≥10 km

New generations of UAVs are designed to achieve missions not only with increased efficiency, but also with more safety. UAV component malfunctions or structure failures may occur during the course of a mission, which can significantly affect the flight safety and the mission accomplishment. In this sense, UAVs are regarded as mission-critical systems, demanding a specific level of fault-tolerant capability. Moreover, how to meet the reliability requirements of the UAV when flying in complex environments is still very challenging, as pointed out in [2]. For more realistic applications, a variety of fault-tolerant control (FTC) methodologies have been proposed to guarantee the UAV safety and survivability in the presence of faults [3-12]. This paper is an extension of the published papers with emphasis on a system viewpoint towards a practical and reliable UAV system. Moreover, FTC, as one of key challenging and necessary techniques making UAV more applicable and reliable for future practical uses in various civilian applications, has been emphasized.

Existing literature concerning UAV FTC system design has been dominated by two sorts: *passive* and *active* strategies [13-14]. With application to UAVs, normal flight conditions as well as catastrophic and high probability failures are prescribed in the design process. Thus, a passive FTC is capable of ensuring flight safety and an acceptable level of performance when the pre-considered faults occur. On the other hand, a fault detection and diagnosis (FDD) unit, a reconfigurable controller, and a control reconfiguration mechanism construct a typical active FTC [13-14]. Based on the information from an FDD unit, the flight controller can be reconfigured/adapted to manage system redundancy against the identified faults.

UAVs possessing four rotors (also named as unmanned quadrotor) have been widely used as a new platform for development and validation of FTC techniques in view of their simple structure and existing hardware redundancy although limited. The main focus of this paper is on 1) reviewing an unmanned quadrotor system; 2) developing several FTC techniques to accommodate the quadrotor faults; and 3) implementing the designed FTC algorithms into the unmanned quadrotor system for experimental validation.

The remainder of this paper is organized as follows. Overview of an unmanned quadrotor system is presented in Section 2, where the primary subsystems and quadrotor system modeling are detailed. Two typical FTC methods are developed with application to the quadrotor UAV in Section 3. Section 4 analyzes the flight test results, followed by some concluding remarks in Section 5.

2 Overview of an Unmanned Quadrotor System

As it is well-known and shown in Fig. 2, for building a control system, it is necessary for having suitable actuator(s), sensor(s), and controller(s) to provide satisfactory performance of the system. Appropriate controllers play an important role in forming the closed-loop through feedback from sensors information for linking to actuators to govern the behavior (stability, performance, robustness, and fault-tolerance etc.) of the system.

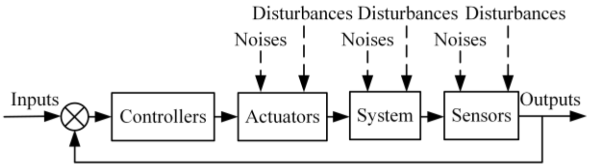

Fig. 2 General architecture and components of a control system

Fig. 2 控制系统的基本结构和组成部分

2.1 Architecture and Components of a Quadrotor UAV (Qball-X4)

As a typical control system, the architecture of the developed quadrotor UAV (Qball-X4) and its associated system components for flight control applications are illustrated in Fig. 3.

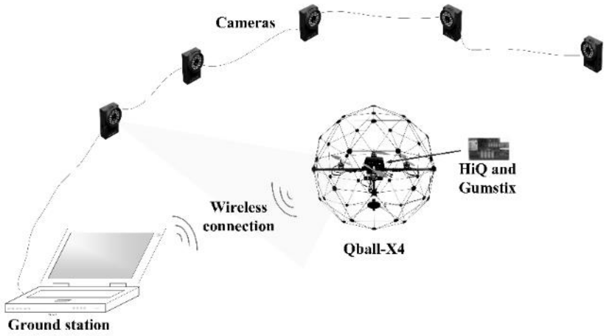

Fig. 3 The architecture of unmanned quadrotor system

Fig. 3 无人四旋翼系统的架构

The quadrotor UAV developed by Quanser Inc. at Canada is enclosed within a protective carbon round cage (thus called as Qball-X4) to ensure the safe operation. Such a quadrotor UAV was initially developed through a Strategic Project Grant (SPG) during 2007-2010 and a Discovery Project Grant (DPG) during 2007-2011 leading by the first author at Concordia University and financially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) with Quanser Inc. as one of three industrial supporting organizations. As the *actuators* of the Qball-X4 system, four propellers of 10-inch length and standard RC motors together with speed controllers are mounted in Qball-X4. The Quanser embedded control module (QECM) consists of a Quanser HiQ aero data acquisition card (which can provide high-resolution accelerometer, gyroscope, and inertial measurement unit (IMU) sensors and servo outputs to control four motors) and a QuaRC-powered Gumstix embedded processor, as the hardware and software platform of the Qball-X4 system *controller*. QuaRC is a Quanser designed software environment which converts MATLAB/Simulink codes systematically and automatically into C-code to be complied by Real-Time Workshop of the MATLAB/Simulink environment for real-time control applications. The Gumstix single-chip processor has been selected as the microcontroller which allows for fast developing and deploying controllers in MATLAB/Simulink environment of the ground station to control the flight of Qball-X4 through the QuaRC interface. The interaction between the ground station and Qball-X4 is completed via wireless communications. Moreover, cameras, as *sensors* of the UAV system, are applied for in-door experiments, such that the position information of the Qball-X4 can be obtained in real-time for feedback control purpose.

As can be observed from Fig. 4, the Qball-X4 quadrotor UAV consists of three major parts: 1) the actuators; 2) the geometry; and 3) Qball-X4 dynamics. The *actuators* are composed of the electronic speed controllers (ESCs), motors, and propellers in a set of four. The input vector of the propeller devices is $u=[u_1,u_2,u_3,u_4]^{\mathrm{T}}$, which is in the form of pulse width modulation (PWM) signals. The output vector of the propeller devices is the thrust vector $T=[T_1,T_2,T_3,T_4]^{\mathrm{T}}$. The geometry linking the generated thrusts to the applied lifts and torques corresponds to the position and orientation of the propellers with respect to the center of the Qball-X4 mass. The dynamics of the Qball-X4 UAV represents the relationships between the output of geometry and the position, velocity, and acceleration.

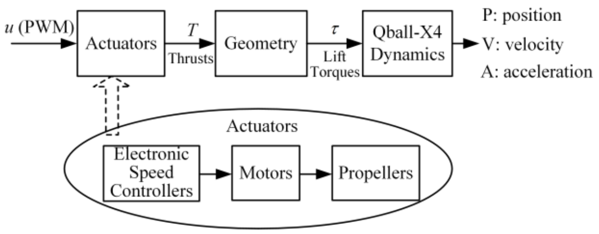

Fig. 4 The block diagram of Qball-X4 UAV

Fig. 4 Qball-X4 无人机框图

2.2 Modeling of the Qball-X4 UAV

2.2.1 Modeling of Actuators

The motors of the Qball-X4 UAV are out-runner brushless motors. A first-order linear transfer function is to describe the relationship between the produced thrust T_i of the *i*th motor and the *i*th PWM input u_i:

$$T_i = K\frac{\omega}{s+\omega}u_i, i = 1, \cdots, 4 \tag{1}$$

where K and ω denote a positive gain and motor bandwidth, respectively.

2.2.2 Modeling of the Control Action/Concept of the Qball-X4 Based on Its Special Geometry

As shown in Fig. 5 for the schematic representation of the Qball-X4, the motors and propellers are configured in such a way so that the back and front (#1 and #2) motors spin counter-clockwise and the left and right (#3 and #4) spin clockwise.

The distance from the location of each motor and the center mass of the UAV is denoted by *L*. A torque τ_i is generated by the spinning of the *i*th motor. The origin of the body-fixed frame is the UAV's center of mass with the *x*-axis pointing from back to front, while the *y*-axis pointing from right to left. The thrust T_i produced by the *i*th propeller always points upward in the *z*-direction in parallel to the motor's rotation axis.

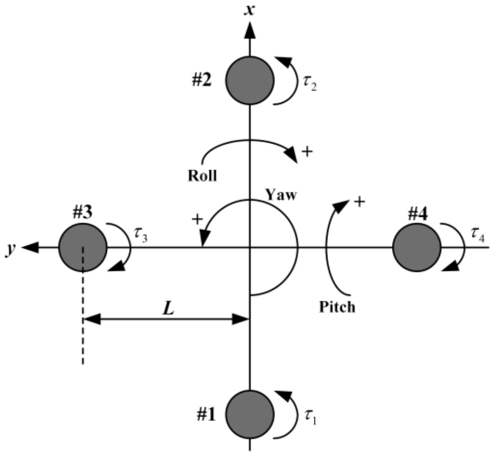

Fig. 5 The schematic representation of Qball-X4

Fig. 5 Qball-X4 示意图

The relationship between the lift/torques and the thrusts of four propellers can be described as:

$$\begin{cases} u_z = T_1 + T_2 + T_3 + T_4 \\ u_\theta = L(T_1 - T_2) \\ u_\phi = L(T_3 - T_4) \\ u_\psi = \tau_1 + \tau_2 - \tau_3 - \tau_4 \end{cases} \tag{2}$$

Moreover, the torque τ_i (i =1, …, 4) can be represented by $\tau_i = K_\psi T_i$, where K_ψ is a constant. By setting $T_i = Ku_i$, Eq. (2) can be rewritten as:

$$\begin{cases} u_z = K(u_1 + u_2 + u_3 + u_4) \\ u_\theta = KL(u_1 - u_2) \\ u_\phi = KL(u_3 - u_4) \\ u_\psi = KK_\psi(u_1 + u_2 - u_3 - u_4) \end{cases} \tag{3}$$

where u_z stands for the total lift by four propellers to be applied to the Qball-X4 UAV in the *z*-direction. u_θ, u_ϕ, and u_ψ are the applied torques in pitch (θ), roll (ϕ), and yaw (ψ) rotations, respectively.

2.2.3 Modeling of Qball-X4 Dynamics

The Qball-X4 dynamics in a hybrid coordinate system is described by position dynamics expressed in the inertial reference frame, and the angular dynamics represented in the body-fixed reference frame, for the convenience of modeling and controller design:

$$\begin{cases} m\ddot{x} = u_z(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - k_x\dot{x} \\ m\ddot{y} = u_z(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) - k_y\dot{y} \\ m\ddot{z} = u_z(\cos\phi\cos\theta) - mg - k_z\dot{z} \\ J_x\dot{p} = u_p + (J_y - J_z)qr - J_T q\Omega - k_p p \\ J_y\dot{q} = u_q + (J_z - J_x)pr - J_T p\Omega - k_q q \\ J_z\dot{r} = u_r + (J_x - J_y)pq - k_r r \end{cases} \tag{4}$$

where x, y, and z are the coordinates of the center of the Qball-X4 mass in the inertial frame. m denotes the Qball-X4 mass, J_x, J_y, and J_z are the moments of inertia along x, y, and z directions, respectively. θ, ϕ, and ψ denote the pitch, roll, and yaw Euler angles, while p, q, and r represent the angular velocities in the body-fixed frame. k_x, k_y, k_z, k_p, k_q, and k_r are drag coefficients. J_T is the moment of inertia for each motor and Ω stands for the overall speed of propellers:

$$\Omega = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4 \tag{5}$$

where Ω_i is the *i*th propeller speed.

The angular rates in the inertial frame (Euler rates) can be related to those in the body-fixed frame as follows:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{6}$$

Under hovering conditions, the matrix in Eq. (6) is close to an identity matrix and the angular rates in the body-fixed reference frame can thereby be treated as the angular velocities in the inertial reference frame. In consequence, Eq. (4) can be rewritten as:

$$\begin{cases} m\ddot{x} = u_z\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) - k_x\dot{x} \\ m\ddot{y} = u_z\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) - k_y\dot{y} \\ m\ddot{z} = u_z\left(\cos\phi\cos\theta\right) - mg - k_z\dot{z} \\ J_x\ddot{\theta} = u_\theta + \left(J_y - J_z\right)\dot{\phi}\dot{\psi} - J_T\dot{\phi}\Omega - k_\theta\dot{\theta} \\ J_y\ddot{\phi} = u_\phi + \left(J_z - J_x\right)\dot{\theta}\dot{\psi} - J_T\dot{\theta}\Omega - k_\phi\dot{\phi} \\ J_z\ddot{\psi} = u_\psi + \left(J_x - J_y\right)\dot{\theta}\dot{\phi} - k_\psi\dot{\psi} \end{cases} \tag{7}$$

where u_p, u_q, u_r, k_p, k_q, and k_r are replaced by u_θ, u_ϕ, u_ϕ, k_θ, k_ϕ, and k_ψ for notation convenience. When the Qball-X4 is flying at low speeds, a simplified nonlinear model of Eq. (7) can be obtained by neglecting the drag terms and gyroscopic and Coriolis-centripetal effects:

$$\begin{cases} m\ddot{x} = u_z\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \\ m\ddot{y} = u_z\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) \\ m\ddot{z} = u_z\left(\cos\phi\cos\theta\right) - mg \\ J_x\ddot{\theta} = u_\theta \\ J_y\ddot{\phi} = u_\phi \\ J_z\ddot{\psi} = u_\psi \end{cases} \tag{8}$$

By assuming also hovering conditions ($u_z \approx mg$) under the condition with small pitch and roll angles and without yaw motion ($\psi = 0$), the above model can be further simplified as:

$$\begin{cases} \ddot{x} = \theta g \\ \ddot{y} = -\phi g \\ \ddot{z} = u_z / m - g \\ J_x\ddot{\theta} = u_\theta \\ J_y\ddot{\phi} = u_\phi \\ J_z\ddot{\psi} = u_\psi \end{cases} \tag{9}$$

3 Recent New Development on Fault-tolerant Control Algorithms Applied to the Qball-X4 UAV

Recently, FDD and FTC design approaches have been developed and applied to rotary-wing UAVs. Rotary-wing UAVs are capable of completing vertical take-off and landing, allowing smaller area for flight tests as compared to fixed-wing UAVs. Hence, rotary-wing UAVs have been recognized as excellent platforms for validating UAV techniques, attracting increasing research interest [3]. On the other hand, rotary-wing types of aircraft present great challenges in FDD and FTC, mainly due to the highly nonlinear characteristics and less hardware redundancy available.

In this section, FTC methods are proposed to maintain the flight safety of Qball-X4 UAV in the event of actuator faults. Two typical FTC approaches, based on the concepts of sliding mode control (SMC) and linear quadratic regulator (LQR), are presented as an example of nonlinear model-based and linear model-based control strategy, respectively. Some of these methods require the information with regard to the time of faults occurrence, the location and the amplitude of faults whereas others do not. In the former case, an FDD module is required to obtain the information of the faults. For the sake of brevity, the FDD design is omitted herein.

3.1 SMC-based FTC Design

SMC, possessing a strong capability of handling uncertainties and disturbances [15], is an excellent option to be adopted in the passive FTC design process. In this section, an active FTC strategy is designed further based on the SMC technique in the presence of faults, and only actuator type of faults are considered in the design procedure.

When the actuator faults occur, the system dynamics and mathematical model will be changed. Considering the actuator effectiveness loss is constant after the fault occurrence, therefore, the remaining actuator power can be calculated by subtracting the lost power from the original one. The state equation can be expressed as [7]:

$$\dot{x}(t) = Ax(t) + B\left(I - \Gamma(t)\right)u(t) \tag{10}$$

and

$$u(t) = \left[u_1(t) \quad \cdots \quad u_m(t)\right]^{\mathrm{T}}$$
$$\Gamma(t) = \begin{bmatrix} \gamma_1(t) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma_m(t) \end{bmatrix} \tag{11}$$

where $A \in \Re^n$, $B \in \Re^m$, $u_i(t) \in u(t)$, $\gamma_i(t) \in \Gamma(t)$ represents the loss of control effectiveness in the *i*th control actuator, with $0 \le \gamma_i(t) \le 1$.

If $\gamma_i(t) = 0$, the *i*th actuator is functioning perfectly; If $\gamma_i(t) = 1$, the *i*th actuator has failed completely (total loss/failure); If $0 < \gamma_i(t) < 1$, the *i*th actuator has failed partially (partial loss).

The following equation can be defined:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 1/m & 1/m & 1/m & 1/m \\ 1/J_x & -1/J_x & 0 & 0 \\ 0 & 0 & 1/J_y & -1/J_y \\ 1/J_z & 1/J_z & -1/J_z & -1/J_z \end{bmatrix} \begin{bmatrix} u_z \\ u_\theta \\ u_\phi \\ u_\psi \end{bmatrix} \tag{12}$$

To generalize the above equation, it is defined as:

$$U = \Lambda F \tag{13}$$

where Λ is the mapping matrix, U is the control inputs, and F is the actuator inputs, respectively.

Since the actuator failure applies directly on F, the following relation is satisfied:

$$F_f(t) = \Gamma(t) F(t) \tag{14}$$

Therefore, from Eq. (10), the new states equation can be obtained as following:

$$\dot{x}(t) = Ax(t) + Bu_i(t) - Bu_{fi}(t) \tag{15}$$

where $u_{fi}(t)$ is the control input with fault.

Furthermore, the following equations can be derived:

$$\dot{x}(t) = Ax(t) + B\left(u_i(t) - u_{fi}(t)\right) \tag{16}$$

In general,

$$\dot{X}(t) = AX(t) + B\left(U(t) - \Lambda F_f(t)\right) \tag{17}$$

$$\dot{X}(t) = AX(t) + B\left(U(t) - \Lambda\left(\Gamma(t)\left(\Lambda^{-1} U_f(t)\right)\right)\right) \tag{18}$$

where $U(t)$ indicates the new control inputs of the quadrotor UAV.

The states after the actuator faults occurrence are redefined as:

$$\begin{aligned} \dot{X}_f &= \left[z_f \ \dot{z}_f \ \theta_f \ \dot{\theta}_f \ \phi_f \ \dot{\phi}_f \ \psi_f \ \dot{\psi}_f \ \right]^{\mathrm{T}} = \\ &\left[x_{f1} \ x_{f2} \ x_{f3} \ x_{f4} \ x_{f5} \ x_{f6} \ x_{f7} \ x_{f8} \ \right]^{\mathrm{T}} \end{aligned} \tag{19}$$

Then one can obtain the following form:

$$\dot{X}_f = \left[\dot{x}_{f1} \quad \dot{x}_{f2} \quad \dot{x}_{f3} \quad \dot{x}_{f4} \quad \dot{x}_{f5} \quad \dot{x}_{f6} \quad \dot{x}_{f7} \quad \dot{x}_{f8} \right]^{\mathrm{T}}$$

$$\begin{bmatrix} x_{f2} \\ -g - d_z x_{f2} + (\cos\phi\cos\theta)\left(u_1 - u_{f1}\right) \\ x_{f4} \\ \dfrac{J_y - J_z}{J_x} x_{f6} x_{f8} - \dfrac{J_r x_{f6} \Omega}{J_x} - d_\theta x_{f4} + l\left(u_2 - u_{f2}\right) \\ x_{f6} \\ \dfrac{J_z - J_x}{J_y} x_{f4} x_{f8} + \dfrac{J_r x_{f4} \Omega}{J_y} - d_\phi x_{f6} + l\left(u_3 - u_{f3}\right) \\ x_{f8} \\ \dfrac{J_x - J_y}{J_z} x_{f4} x_{f6} - d_\psi x_{f8} + c\left(u_4 - u_{f4}\right) \end{bmatrix} \tag{20}$$

with

$$\Gamma(t) = \begin{bmatrix} \gamma_1(t) & 0 & 0 & 0 \\ 0 & \gamma_2(t) & 0 & 0 \\ 0 & 0 & \gamma_3(t) & 0 \\ 0 & 0 & 0 & \gamma_4(t) \end{bmatrix} \tag{21}$$

where $0 \le \gamma_i(t) \le 1$.

The sliding surface is established as following:

$$s_j(t) = \dot{e}_j + \lambda_j e_j + k_{pj} \int e_j \tag{22}$$

The derivative of the sliding surface in this case is:

$$\dot{s}_{fj} = \ddot{e}_j + \lambda_{aj} \dot{e}_j + k_{apj} e_j \tag{23}$$

If the 4th actuator failed, the approximation of the control input for attitude roll $\hat{u}_{f\phi}$ can be derived as following:

$$\hat{u}_{f\phi} = \frac{2J_y}{l\left(k_4 J_y - J_y\right)} \begin{pmatrix} -\ddot{\phi}_d - \dfrac{l\left(m - mk_4\right)}{4J_y} u_{fz} - \\ \dfrac{l\left(k_4 J_z - J_z\right)}{4J_y} u_{f\psi} + \dfrac{J_z - J_x}{J_y} x_{f4} x_{f8} \\ + \dfrac{J_r x_{f4} \Omega}{J_y} - d_\phi x_{f6} \\ -\lambda_{a5}\left(\dot{\phi}_{fd} - \dot{\phi}_f\right) - 2k_{ap5} e_\phi \end{pmatrix} \tag{24}$$

where k_{ap5} and λ_{a5} are all positive gains.

The sliding condition can be easily proven to meet the following condition:

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t} s_{fj}^{\ 2} = s_{fj} \dot{s}_{fj} \le 0 \tag{25}$$

Therefore, the closed-loop system is stable.

In general, the overall control inputs are expressed as:

$$U_{fj} = \hat{U}_{fj} + k_{aj} \operatorname{sign}\left(s_{fj}\right) \tag{26}$$

where λ_{aj}, k_{aj}, and k_{apj} are all positive gains of the changed sliding mode based fault-tolerant control.

A new saturation function is introduced for an active FTC to eliminate and reduce chattering, and to achieve a relatively faster convergence of the system.

$$\begin{cases} u_{fj} = \hat{u}_{fj} + k_{aj}\left(-\delta_a\right), & \text{if} \quad s_{aj} \le -\delta_a \\ u_{fj} = \hat{u}_{fj} + k_{aj} s_{aj}, & \text{if} \quad -\delta_a < s_{aj} < \delta_a \\ u_{fj} = \hat{u}_{fj} + k_{aj}\left(\delta_a\right), & \text{if} \quad s_{aj} \ge \delta_a \end{cases} \tag{27}$$

where δ_a is a mall scalar representing the boundary of the active FTC saturation.

3.2 LQR-based FTC Design under Polytopic Representation

LQR is a widely used optimal multi-variable feedback control approach that minimizes the excursion in state trajectories of a system while requiring minimum controller effort. The behavior of a LQR controller is determined by two parameters: state and control weighting matrices. These two matrices are main design parameters to be selected by designer, greatly influencing the success of the LQR controller synthesis. With the goal of representing nonlinear Qball-X4 UAV behavior by linear model suitable for real-time control based on LQR controller design under both normal and prescribed fault cases, both the normal and prescribed fault cases are considered and formulated by the concept of polytope during the control design stage. To be more specific, one convex of the polytope can represent the normal case, while each of the rest can denote the individual actuator fault. Faults occurring in actuators can be treated as an uncertainty of the control input which can be modeled by the polytopic uncertainties. The uncertainty is defined as a polytope in which the model parameters must lie. Each case in the considered fault sets can be regarded as the corner of the polytope. It is advantageous to form one model including the fault and fault-free cases at the design stage of FTC. Therefore, the faulty system falls into the convex hull of these corners. As a consequence, combining LQR with polytopic representation can achieve optimal control performance under both normal and faulty conditions based on the models represented by the polytopes.

Focusing on the altitude motion of the Qball-X4 UAV, the dynamics in the fault-free case is represented as:

$$\dot{x}_z = f\left(x_z\right) + Bu \tag{28}$$

The mathematical model of the Qball-X4 with actuator faults can therefore be expressed as:

$$\dot{x}_z = f\left(x_z\right) + Bu_f = f\left(x_z\right) + B(I - \Gamma_A)u \tag{29}$$

where

$$\Gamma_{\mathrm{A}} = \mathrm{diag}\left\{\gamma_1^a, \gamma_2^a, \cdots, \gamma_m^a\right\}, \gamma_i^a \in \Re \tag{30}$$

$$f\left(x_z\right) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}_t \\ z_t \\ s \end{bmatrix} + \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix} = A \begin{bmatrix} \dot{z}_t \\ z_t \\ s \end{bmatrix} + G \tag{31}$$

$$B = \begin{bmatrix} K/M & K/M & K/M & K/M \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{32}$$

In such a way, $\gamma_i^a = 1$ implies a complete failure of the *i*th actuator of the Qball-X4. On the other hand, $\gamma_i^a = 0$ indicates that the *i*th actuator operates normally.

In this study, both the normal and faulty cases are considered at the FTC design stage. Each case in the considered fault sets is known *a priori*, which can be seen as the corner of the polytope. In addition, the faulty system falls into the convex hull of these corners. The system matrix

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{33}$$

does not change as only the actuator faults are concerned. The control input matrix with the considered actuator fault sets can be modeled by the polytopic uncertainties as:

$$B\left(\vartheta\right) = \vartheta_0 B + \vartheta_1 B\Gamma_{\mathrm{A}}, \vartheta_0 + \vartheta_1 = 1, \vartheta_0, \vartheta_1 > 0 \tag{34}$$

Applying a LQR controller requires the system to be linearized to the state space representation as $\dot{x} = Ax + B\left(\vartheta\right)u + G$. By designing a state feedback control law $u = -K_u x$, the closed-loop system becomes $\dot{x} = \left(A - B\left(\vartheta\right)K_u\right)x$. It is noted that the design of K_u is a trade-off between the transient response and the control effort.

The optimal control approach is to design the control law $u = -K_u x$ such that it minimizes the following performance index,

$$J = \frac{1}{2}\int_0^{\infty}\left(x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru\right)\mathrm{d}t \tag{35}$$

where $x \in \Re^n$ and $u \in \Re^m$ denote the state and control input vectors, respectively. The weighting matrices Q and R are symmetric positive semi-definite and symmetric positive definite, respectively.

Substituting the state-variable feedback control $u = -K_u x$ into Eq. (35) yields:

$$J=\frac{1}{2}\int_0^\infty\left[x^{\mathrm{T}}\left(Q+K_u{}^{\mathrm{T}}RK_u\right)x\right]\mathrm{d}t \tag{36}$$

In Eq. (36), K_u is the control gain matrix given by Eq. (37),

$$K_u=R^{-1}B^{\mathrm{T}}\left(\vartheta\right)P \tag{37}$$

where P is the unique symmetric positive semi-definite solution to the algebraic Riccati equation as shown in Eq. (38),

$$PA+A^{\mathrm{T}}P+Q-PB\left(\vartheta\right)R^{-1}B^{\mathrm{T}}\left(\vartheta\right)P=0 \tag{38}$$

The objective in this optimal control design is to select the K_u minimizing the above performance index J.

4 Implementation and Flight Test Results

The practical view of the Qball-X4 UAV platform used in this research is presented in Fig. 6.

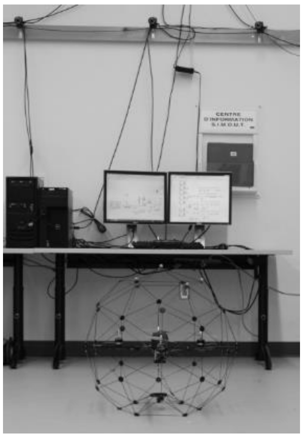

Fig. 6 The Qball-X4 platform

Fig. 6 Qball-X4 平台

The FTC algorithms for the Qball-X4 UAV are implemented in MATLAB/Simulink environment of the ground station and subsequently downloaded on the Gumstix embedded processor to be run on-board in real-time with a frequency of 200 Hz. The flight tests are conducted indoor without GPS signals while a group of the OptiTrack cameras are deployed to offer the UAV position in the three-dimensional space.

4.1 Flight Test Results of the SMC-based FTC Method

In this scenario, a 15 % propeller damage occurring at 20 s is considered to validate the SMC-based FTC. Figs. 7-9 exhibit the Qball-X4 UAV positions along *x*, *y*, and *z* directions.

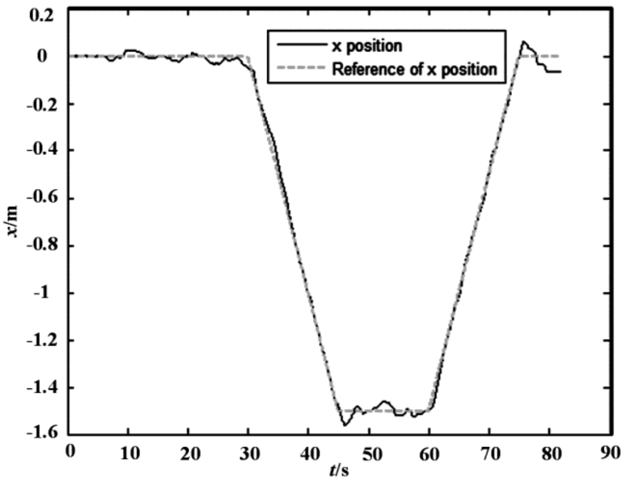

Fig. 7 Tracking performance in *x* direction with SMC-based FTC

Fig. 7 基于 SMC 的 FTC 在 *x* 方向的跟踪性能

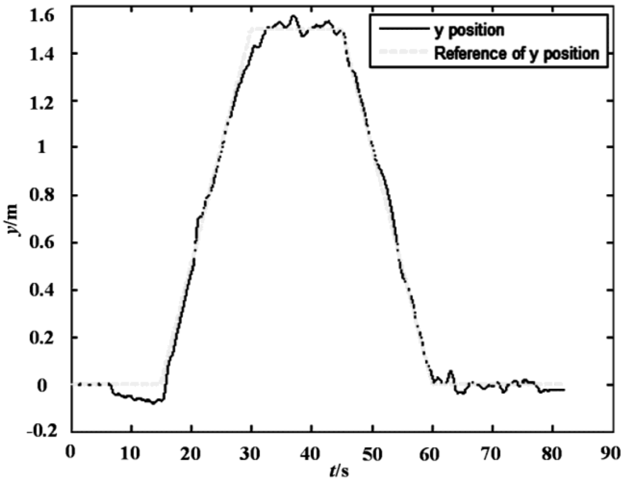

Fig. 8 Tracking performance in *y* direction with SMC-based FTC

Fig. 8 基于 SMC 的 FTC 在 *y* 方向的跟踪性能

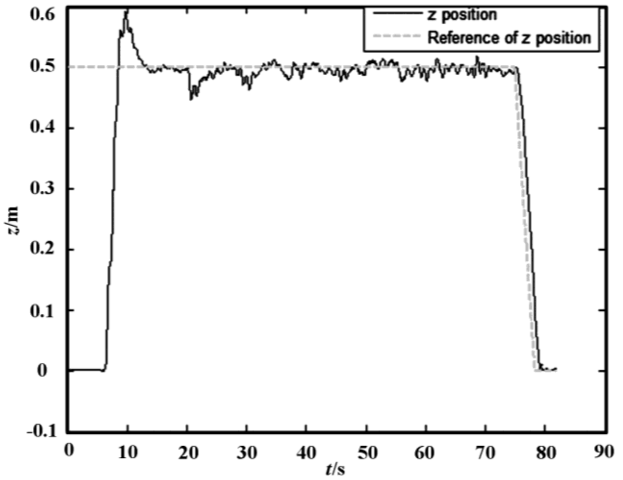

Fig. 9 Tracking performance in *z* direction with SMC-based FTC

Fig. 9 基于 SMC 的 FTC 在 *z* 方向的跟踪性能

In the event of propeller fault, the Qball-X4 UAV can still follow the reference of *z* direction after prompt fault accommodation. The reference signals with respect to *x* and *y* directions can be tracked as well after the propeller fault takes place. Therefore, it is evident from the resulting responses that the SMC-based FTC is capable of guaranteeing the flight safety in the absence and presence of the considered fault.

4.2 Flight Test Results of the LQR-based FTC Method

In this flight test, 40 % loss of effectiveness occurrs in all of the four motors at 45 s. The height response of the Qball-X4 UAV is depicted in Fig. 10.

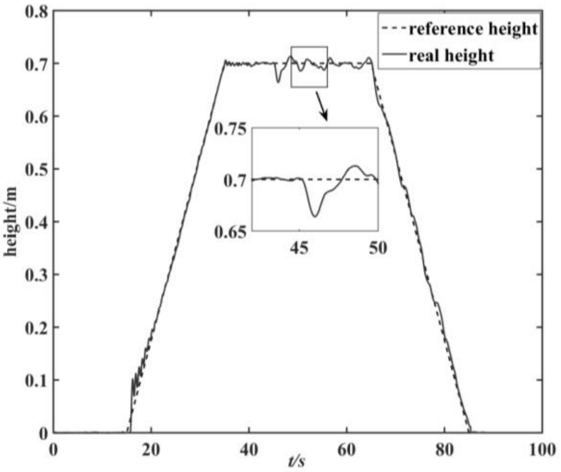

Fig. 10 The flight test result for LQR-based FTC

Fig. 10 基于 LQR 的 FTC 飞行试验结果

The Qball-X4 UAV can track the height reference signal when all the motors are operating normally. After the partial fault takes place, the LQR-based FTC can ensure the flight safety and an acceptable level of tracking performance. Using the LQR-based FTC, the UAV can accomplish the landing maneuver as the reference input required.

5 Conclusion

This paper mainly focuses on the research and development works on fault-tolerant control (FTC) for the unmanned rotorcraft systems. The developed Qball-X4 unmanned aerial vehicle (UAV) system is used as a test platform of FTC approaches. In order to maintain the flight safety of the UAV system, two newly investigated FTC algorithms are proposed and subsequently implemented into the Qball-X4 UAV system. The flight tests are carried out using the Qball-X4 platform, considering a variety of actuator fault scenarios. The results have demonstrated the effectiveness of the developed FTC techniques.

References(参考文献)

[1] Valavanis K P, Vachtsevanos G J. Handbook of unmanned aerial vehicles[M]. Dodrecht, Netherlands: Springer, 2015.

[2] NASA. Unmanned aerial vehicles roadmap 2002-2027[R]. Office of the Secretary of Defense, Washington DC, 2002.

[3] Y. M. Zhang, Chamseddine A, Rabbath CA, et al. Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed[J]. Journal of Franklin Institute, 2013, 350 (9): 2396-2422.

[4] Drozeski G R, Saha B, Vachtsevanos G J. A fault detection and reconfigurable control architecture for unmanned aerial vehicles[C]. Proceedings of IEEE Aerospace Conference, Big Sky, 2005: 1-9.

[5] Waydo S, Hauser J, Bailey R, et al. UAV as a reliable wingman: a flight demonstration[J]. IEEE Transactions on Control Systems Technology, 2007, 15(4): 680-688.

[6] Izadi H A, Gordon B W, Zhang Y M. Decentralized receding horizon control for cooperative multiple vehicles subject to communication delay[J]. AIAA Journal of Guidance, Control, and Dynamics, 2009, 32(6): 1959-1965.

[7] Zhang Y M, Jiang J. An active fault-tolerant control system against partial actuator failures[J]. IEE Proceedings – Control Theory and Applications, 2002, 149(1): 95-104.

[8] Chamseddine A, Zhang Y M, Rabbath C A, et al. Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle[J]. IEEE Transactions on Aerospace and Electronic Systems, 2012, 48(4): 2832-2848.

[9] Chamseddine A, Zhang Y M, Rabbath C A, et al. Trajectory planning and replanning strategies applied to a quadrotor unmanned aerial vehicle[J]. AIAA Journal of Guidance, Control, and Dynamics, 2012, 35(5): 1667-1670.

[10] Tousi M M, Khorasani K. Optimal hybrid fault recovery in a team of unmanned aerial vehicles, Automatica, 2012, 48(2): 410-418.

[11] Qian M S, Jiang B, Xu D Z. Fault tolerant control scheme design for the formation control system of unmanned aerial vehicles[J]. Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems & Control Engineering, 2013, 227(8): 626-634.

[12] Yu X, Liu Z X, Zhang Y M. Fault-tolerant formation control of multiple UAVs in the presence of actuator faults[J]. International Journal of Robust and Nonlinear Control, 2016, 26(12): 2668-2685.

[13] Zhang Y M, Jiang J. Bibliographical review on reconfigurable fault-tolerant control systems[J]. Annual Reviews in Control, 2008, 32(2): 229-252.

[14] Yu X, Jiang J. A survey of fault-tolerant controllers based on safety-related issues[J]. Annual Reviews in Control, 2015, 39(1): 46-57.

[15] Edwards C, Lombaerts T, Smaili H. Fault tolerant flight control: a benchmark challenge[M]. Beilin, Germany: Springer, 2010.