AIAA Guidance, Navigation, and Control Conference
08 - 11 August 2011, Portland, Oregon

AIAA 2011-6716

Fault/Damage Tolerant Control of a Quadrotor Helicopter UAV using Model Reference Adaptive Control and Gain-Scheduled PID

Iman Sadeghzadeh[1], Ankit Mehta[2], and Youmin Zhang[3]

Concordia University, Montreal, Quebec, H3G 1M8, Canada

In this paper, two useful approaches to Fault Tolerant Control (FTC) for a quadrotor helicopter Unmanned Aerial Vehicle (UAV) in the presence of fault(s) in one or more actuators during flight have been investigated and experimentally tested based on a Model Reference Adaptive Control (MRAC) and a Gain-Scheduled Proportional-Integral-Derivative (GS-PID) control. A Linear Quadratic Regulator (LQR) controller is used in cooperation with the MRAC and the GS-PID to control the pitch and roll attitudes of the helicopter. Unlike the MRAC, the GS-PID is used only to control the helicopter in height control mode. MRAC is used to control the helicopter in both height control as well as trajectory control. For damage tolerant control the MRAC is evaluated based on partial damage of one of propellers during flight. Finally, the experimental flight testing results of both controllers are presented for the fault tolerant control performance comparison in the presence of actuator faults in the quadrotor UAV.

I. Introduction

SSafety, reliability and acceptable level of performance of dynamic control systems are key performance measures in control systems not only in normal operation conditions but also in the presence of partial faults or failures in the components of the controlled system. Hence, the role of Fault Tolerant Control Systems (FTCS) is revealed evidently. In fact, when a fault occurs in a system, it suddenly starts to behave in an unanticipated manner. So, the fault tolerant controller should be able to handle the fault and to guarantee system stability and acceptable performance in the presence of faults[1].

There are different techniques to handle such faults. Adaptive control techniques are one of the mostly used techniques for such situations. In fact, adaptive control is originally a control technique which bases on a concept that controllers must adapt to a controlled system with parameters which vary, or are initially uncertain. For example, as an aircraft flies, its mass will slowly decrease as a result of fuel consumption; a control law is needed to adapt itself to such changing conditions.

As one of adaptive control techniques, Model Reference Adaptive Control (MRAC) is concerned with forcing the dynamic response of the controlled system to asymptotically approach that of reference system, despite parametric uncertainties in the plant.

On the other hand, Proportional-Integral-Derivative (PID) controllers are the most widely used controllers in industry. PID controllers are reliable and easy to be used and tuned. In this paper PID controllers are designed and tuned to control the quadrotor helicopter UAV under the normal and faulty flight conditions. For such a purpose the gain scheduling strategy is applied to PID controllers for achieving fault-tolerant control of the quadrotor helicopter. Gain-scheduled (GS) PID is a useful technique which we use for different sections of the flight envelope by properly tuning the controller gains in the sense that it is assumed that the time and the magnitude of the fault are

[1]Ph.D. Student, Department of Mechanical and Industrial Engineering, i_sade@encs.concordia.ca
[2]M.Eng. Student, Department of Mechanical and Industrial Engineering, ankitmehta_ecer@yahoo.co.in
[3]Associate Professor, Department of Mechanical and Industrial Engineering, ymzhang@encs.concordia.ca, Senior Member of AIAA, Corresponding Author.

predetermined or on-line diagnosed. Therefore, the GS-PID controller takes the fault-related gains to handle the faults during the flight of the helicopter.

During recent years, Unmanned Aerial Vehicles (UAVs) have proved to hold a significant role in the world of aviation. Among the rotorcrafts, quadrotor helicopters can usually afford a larger payload than conventional helicopters due to their four rotors configuration. Moreover, small quadrotor helicopters possess a great manoeuvrability and are potentially simpler to manufacture. For these advantages, quadrotor helicopters have received much and increased interest in UAV research. The quadrotor helicopter we consider is an under-actuated system with six outputs and four inputs and the states are highly coupled. There are four fixed-pitch-angle blades whereas single-rotor helicopters have variable-pitch-angle (collective) blades.

Control of a quadrotor helicopter is performed by varying the speed of each rotor. The configuration and structure of a quadrotor, especially the Quanser quadrotor unmanned helicopter known as Qball-X4, which was developed in collaboration between Concordia University and Quanser Inc. through an NSERC (Natural Sciences and Engineering Research Council of Canada) Strategic Project Grant (SPG), are presented in the next parts of this paper, with also related hardware/software of the quadrotor helicopter[1].

II. System Configuration and Mathematical Model of a Quadrotor Helicopter UAV: Qball-X4

A. General and Qball-X4 Quadrotor Helicopter Structure

A concept of the quadrotor helicopter is shown in Fig.1. Each rotor produces a lift force and moment. The two pairs of rotors, i.e., rotors (1, 3) and rotors (2, 4) rotate in opposite directions so as to cancel the moment produced by the other pair. To make a roll angle (φ) along the *x*-axis of the body frame, one can increase the angular velocity of rotor (2) and decrease the angular velocity of rotor (4) while keeping the whole thrust constant. Likewise, the angular velocity of rotor (3) is increased and the angular velocity of rotor (1) is decreased to produce a pitch angle (θ) along the *y*-axis of the body frame. In order to perform yawing motion (ψ) along the *z*-axis of the body frame, the speed of rotors (1, 3) is increased and the speed of rotors (2, 4) is decreased.

The quadrotor helicopter is assumed to be symmetric with respect to the *x* and *y* axes so that the center of gravity is located at the center of the quadrotor and each rotor is located at the end of bars.

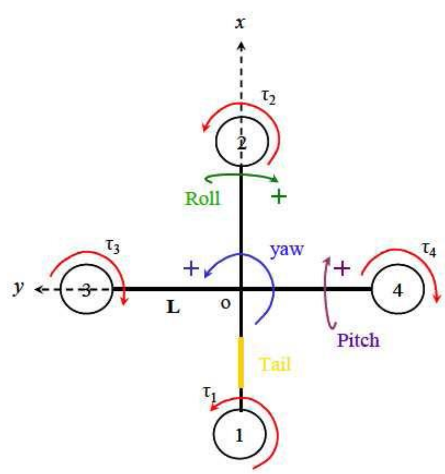

Figure 1. Quadrotor helicopter configuration with Roll-Pitch-Yaw Euler angles [φ,θ, ψ][2]

The present test-bed quadrotor (Qball-X4) at MIE department of Concordia University which has been designed and made by Quanser Inc. is shown in Fig. 2. Qball-X4 is an innovative rotary-wing vehicle platform suitable for a wide variety of UAV research applications. The Qball-X4 is a quadrotor helicopter design propelled by four motors fitted with 10-inch propellers. The entire quadrotor is enclosed within a protective carbon fibre cage. The Qball-X4's proprietary design ensures safe operation as well as opens the possibilities for a variety of novel applications.

The protective cage is a crucial feature since this unmanned aerial vehicle was designed for use in an indoor laboratory. To obtain the measurement from on-board sensors and to drive the motors connected to the four propellers, the Qball-X4 utilizes Quanser's onboard avionics Data Acquisition Card (DAQ), the HiQ, and the embedded Gumstix computer. The HiQ DAQ is a high-resolution Inertial Measurement Unit (IMU) and avionics Input/Output (I/O) card designed to accommodate a wide variety of research applications. QuaRC, which is the Quanser's real-time control software, allows researchers and developers to rapidly develop and test controllers on actual hardware through a MATLAB/Simulink interface. QuaRC's open-architecture hardware and extensive Simulink blockset provides users with powerful control development tools. QuaRC can target the Gumstix embedded computer, automatically generating code and executing controllers on-board the vehicle. During flights, while the controller is executing on the Gumstix, users can tune parameters in real-time and observe sensor measurements from a host ground station computer (PC or laptop)[2].

The interface to the Qball-X4 is MATLAB/Simulink with QuaRC. The controllers are developed in Simulink with QuaRC on the host computer, and these models are downloaded and compiled into executable codes on the target (Gumstix) seamlessly[2]. A diagram of this configuration is shown in Figure 3.

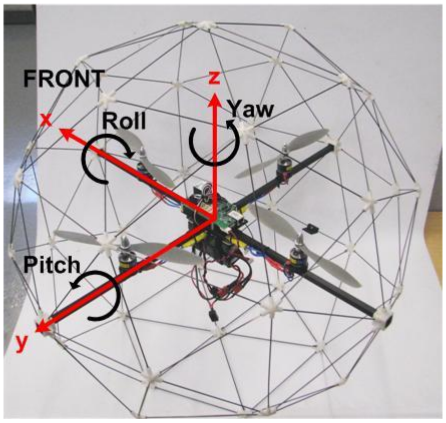

Figure 2. The Quanser Qball-X4 quadrotor UAV[2]

For Qball, the following hardware and software are embedded:

- **Qball-X4:** Qball-X4 as shown in Figure 2 above
- **HiQ:** QuaRC aerial vehicle data acquisition card (DAQ).
- **Gumstix:** The QuaRC target computer. An embedded, Linux-based system with QuaRC runtime software installed
- **Batteries:** Two 3-cell, 2500 mAh Lithium-Polymer batteries
- **Real-Time Control Software:** The QuaRC-Simulink configuration, as detailed in Ref. 2

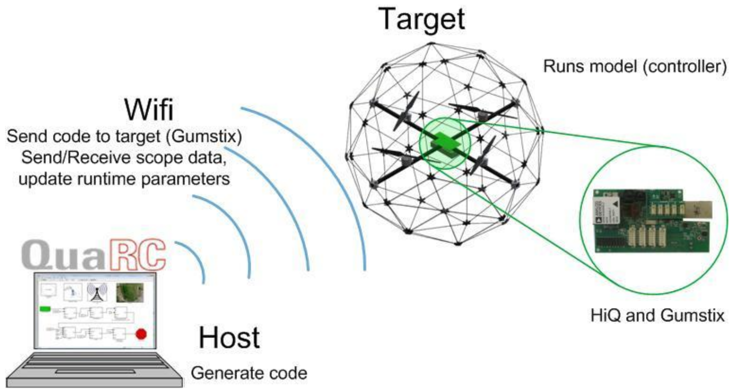

Figure 3. Communication hierarchy and communication diagram[2]

B. Qball Model using Conventional Model of Quadrotor Helicopter

There are some papers published for quadrotor helicopter using different types of model depending on the structure of each quadrotor. In Qball-X4, there are four (E-flite Park 400) brushless motors, using 10×4.7" propeller. As explained before, in order to cancel the moment of each pair of motors, the rotors 1 and 2 have clockwise rotation and the motors 3 and 4 have counterclockwise rotation.

For every attitude change the angular velocity of motors is changed, but the total thrust of all the four motors is constant in order to maintain the height. For instant, to make a pitch angel (θ) along the *Y*-axis of the body frame one can increase the angular velocity of rotor (2) and increase the angular velocity of rotor (1), while keeping the trust constant. Likewise the angular velocity of rotor (3) is increased and the angular velocity of rotor (4) is decreased in order to make a roll angel (φ) along the *X*-axis of the body frame.

It can be understood easily that yaw motion along the *Z* axis of body frame will be implemented by increasing total angular velocity of rotors (1, 2) and decreasing the angular velocity of opposite rotation rotors (3, 4). Motors of Qball-X4 are not exactly located at the end of the aluminum rods, but 6 inches from the end point not to touch the fiber carbon rod cage by propellers and the *L* is the length of road between every motor rotational axis and the center of gravity of the Qball-X4[2].

While flying there are four downwash thrust vectors generated by four rotors, if we neglect the drag of four rotors we can present the equations of motion as follows:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m}\left(\sum_{i=1}^{4} F_i\right) \mathrm{R} e_3 + (g_r(z) - g) e_3 \tag{1}$$

$$\begin{aligned} \ddot{\phi} &= l(F_3 - F_4)/J_1 \\ \ddot{\theta} &= l(F_1 - F_2)/J_2 \\ \ddot{\psi} &= \rho(F_1 + F_2 - F_3 - F_4)/J_3 \end{aligned} \tag{2}$$

where *J* is the moment of inertia with respect to each axis and ρ is the force-to-moment scaling factor.

In this equation of motion, [*x*, *y*, *z*] are the position of the quadrotor in earth position and [φ, θ, Ψ] are respectively roll, pitch and yaw angle. As mentioned before, we need to transform the matrix from body frame to the Earth frame, thus *R* is the coordinate transformation matrix from body frame to earth frame and $e_3 = [0,0,1]^T$.

$$R = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\phi\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3}$$

We can assume that below a certain height of the quadrotor, certain ground effects will affect Qball-X4 and we define $g_r(z)$ for such an effect as shown below:

$$g_r(z) = \begin{cases} \dfrac{A}{(z + z_{cg})^2} - \dfrac{A}{(z_0 + z_{cg})^2} & 0 < z \le z_0 \\ 0 & else \end{cases} \tag{4}$$

In this equation we consider *A* as ground effect and z_{cg} is the *Z* component of CoG. Because it is very difficult to derive the exact equations for the ground effect, the term $g_r(z)$ is considered an unknown perturbation in designing a controller, which requires compensation or adaptation. We can simplify (1) and (2), by defined input terms as (5). u_1 is the normalized total lift force, and u_2, u_3 and u_4 correspond to the control inputs of roll, pitch and yaw moments, respectively.

$$\begin{aligned} u_1 &= (F_1 + F_2 + F_3 + F_4)/m \\ u_2 &= (F_3 - F_4)/J_1 \\ u_3 &= (F_1 - F_2)/J_2 \\ u_4 &= \rho(-F_1 - F_2 + F_3 + F_4) \end{aligned} \tag{5}$$

We can rewrite the equation of motion as below:

$$\ddot{x} = u_1(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \tag{6}$$

$$\ddot{y} = u_1(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \tag{7}$$

$$\ddot{z} = u_1(\cos\phi \cos\theta) - g + g_r(z) \tag{8}$$

$$\ddot{\phi} = u_2 l \tag{9}$$

$$\ddot{\theta} = u_3 l \tag{10}$$

$$\ddot{\theta} = u_3 l \tag{11}$$

We can present equivalently, $\boldsymbol{x} = [x, y, z, \varphi, \theta, \psi]$ and $\boldsymbol{u} = [u_1, u_2, u_3, u_4]$ in the vector form as:

$$\ddot{x} = f(x) + g(x)u + f_r(x), \tag{12}$$

$$f(x) = \begin{bmatrix} 0 \\ 0 \\ -g \\ 0 \\ 0 \\ 0 \end{bmatrix}, f_r(x) = \begin{bmatrix} 0 \\ 0 \\ g_r(z) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{13}$$

and to define $g(\boldsymbol{x})$ as follows:

$$g(x) = \begin{bmatrix} \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi & 0 & 0 & 0 \\ \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi & 0 & 0 & 0 \\ \cos\phi \cos\theta & 0 & 0 & 0 \\ 0 & l & 0 & 0 \\ 0 & 0 & l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

C. Dynamic State-Space Model of the Qball-X4

This section describes the dynamic model of the Qball-X4. The nonlinear models are described as well as linearized models for use in controller development. For the following discussion, the axes of the Qball-X4 vehicle are denoted as (*x*, *y*, *z*). Roll, pitch, and yaw are defined as the angles of rotation about the *x*, *y*, and *z* axis, respectively. The global workspace axes are denoted as (*X*, *Y*, *Z*) and are defined with the same orientation as the Qball-X4 sitting upright on the ground.

Actuator Dynamics

The thrust generated by each propeller is modeled using the following first-order system

$$F = k \frac{\omega}{s + \omega} u \tag{15}$$

where *u*, is the PWM input to the actuator, ω is the actuator bandwidth and *K* is a positive gain. These parameters were calculated and verified through experimental studies [2]. A state variable, *v*, will be used to represent the actuator dynamics, which is defined as follows:

$$v = \frac{\omega}{s + \omega} u \tag{16}$$

Roll and Pitch Model

Assuming that rotations about the *x* and *y* axes are decoupled, the motion in roll/pitch axis can be modeled as shown in Figure 4. As illustrated in this figure, two propellers contribute to the motion in each axis. The thrust generated by each motor can be calculated from Eq. (15) and using its corresponding input. The rotation around the center of gravity is produced by the difference in the generated thrusts. The roll/pitch angle can be formulated using the following dynamics [2]:

$$J\ddot{\theta} = \Delta FL \tag{17}$$

where

$$J = J_{roll} = J_{Pitch} \tag{18}$$

are the rotational inertia of the device in roll and pitch axes. *L* is the distance between the propeller and the center of gravity, and

$$\Delta F = F1 - F2 \tag{19}$$

represents the difference between the forces generated by the motors.

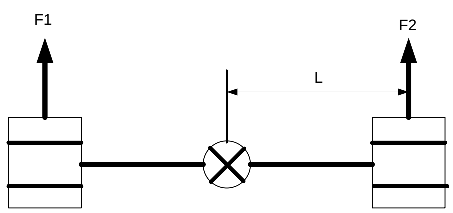

Figure 4. Roll/pitch axis model

By combining the dynamics of motion for the roll/pitch axis and the actuator dynamics for each propeller the following state-space equations can be derived[2]

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\nu} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{KL}{J} \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \nu \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \Delta F \tag{20}$$

To facilitate the use of an integrator in the feedback structure a fourth state can be added to the state vector, which is defined as $\dot{S} = \theta$.

Height Model

The motion of the Qball-X4 in the vertical direction (along the *Z* axis) is affected by all the four propellers. The dynamic model of the Qball-X4 height can be written as[2]:

$$M\ddot{Z} = 4F\cos(r)\cos(p) - Mg \tag{21}$$

where *F* is the thrust generated by each propeller, *M* is the total mass of the device, *Z* is the height and *r* and *p* represent the roll and pitch angles, respectively. As expressed in this equation, if the roll and pitch angles are nonzero the overall thrust vector will not be perpendicular to the ground. Assuming that these angles are close to zero, the dynamics equations can be linearized to the following state space form[2]:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M} & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -g \\ 0 \\ 0 \end{bmatrix} \tag{22}$$

X-Y Position Model

The motion of the Qball-X4 along the *X* and *Y* axes is caused by the total thrust and by changes of the roll/pitch angles. Assuming that the yaw angle is zero, the dynamics of motion in *X* and *Y* axes can be written as:

$$\begin{aligned} M\ddot{X} &= 4F\sin(p) \\ M\ddot{Y} &= -4F\sin(r) \end{aligned} \tag{23}$$

Assuming that the roll and pitch angles are close to zero, the following linear state-space equations can be derived for *X* and *Y* positions[2]:

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{V} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M}P & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u \tag{24}$$

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{V} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M}P & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u \tag{25}$$

Yaw Model

The torque generated by each motor, τ , is assumed to have the following relationship with respect to the PWM input,

$$\tau = K_y u \tag{26}$$

where K_y is a positive gain. The motion in the yaw axis is caused by the difference between the torques exerted by the two clockwise and the two counterclockwise rotating propellers. The motion in the yaw axis can be modeled using the following equation:

$$J_y \ddot{\theta}_y = \Delta\tau \tag{27}$$

In this equation, θ_y is the yaw angle and J_y is the rotational inertia about the z axis. The resultant torque of the motors, $\Delta\tau$, can be calculated from[2]:

$$\Delta\tau_y = -\tau_1 - \tau_2 + \tau_3 + \tau_4 \tag{28}$$

The yaw axis dynamics can be rewritten in the state-space form as:

$$\begin{bmatrix} \dot{\theta}_y \\ \ddot{\theta}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_y \\ \dot{\theta}_y \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{K_y}{J_y} \end{bmatrix} \Delta\tau_y \tag{29}$$

III. Model Reference and Gain-Scheduled Proportional-Derivative-Integral (GS-PID) Adaptive Fault/Damage Tolerant Controllers

Model Reference Adaptive Control (MRAC) is concerned with forcing the dynamic response of the controlled system to asymptotically approach that of a reference system, despite parametric uncertainties in the plant. Two major subcategories of MRAC are those of indirect methods, in which the uncertain plant parameters are estimated and the controller redesigned online based on the estimated parameters, and direct methods, in which the tracking error is forced to zero without regard to parameter estimation accuracy (though under certain conditions related to the level of excitation in the command signal, the adaptive laws often can converge to the proper values). MRAC for linear systems has received, and continues to receive, considerable attention in the literature. Robustness of the adaptive algorithms has been a persistent issue in the literature. It is well-known that the presence of unmodeled dynamics in the system under control can lead to divergence of the parameter estimates and, ultimately, instability of the closed-loop system; this phenomenon can occur even in the case of state feedback. A model reference adaptive control structure is presented in Fig. 5.

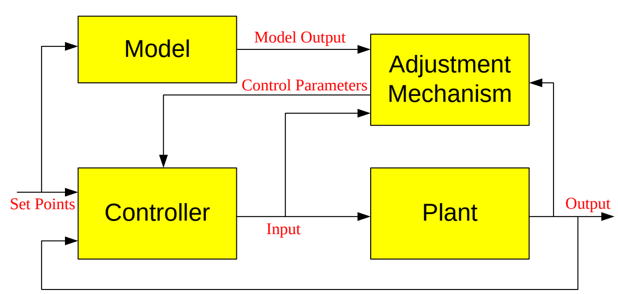

Figure 5. Model reference adaptive control structure

Also, there are different approaches to MRAC such as:

- The MIT rule
- Lyapunov stability theory
- Design of MRAS based on Lyapunov stability theory
- Hyperstability and passivity theory
- The error model
- Augmented error
- A model-following MRAC

In this paper MIT rule is used to control the height of Qball-X4. However, the schemes based on the MIT rule and other approximations may go unstable.

We illustrate the use of the MIT rule for the design of an MRAC scheme for the plant

$$\ddot{y} = -a_1 \dot{y} - a_2 y + u \tag{30}$$

where a_1 and a_2 are the unknown plant parameters, and $\dot{y}$ and y are available for measurement.

The reference model to be matched by the closed-loop plant is given by

$$\ddot{y}_m = -2\ddot{y}_m - y_m + r \tag{31}$$

The control law

$$u = \theta^*_1 \dot{y} + \theta^*_2 y + r \tag{32}$$

where

$$\theta^*_1 = a_1 - 2, \theta^*_2 = a_2 - 1 \tag{33}$$

will achieve perfect model following. The equation (33) is referred to as the matching equation. Because a_1 and a_2 are unknown, the desired values of the controller parameters θ_1^* and θ_2^* cannot be calculated from (34).

Therefore, instead of (32) we use the control law

$$u = \theta_1 \dot{y} + \theta_2 y + r \tag{34}$$

where θ_1 and θ_2 are adjusted using the MIT rule as

$$\dot{\theta}_1 = -\gamma e_1 \frac{\partial y}{\partial \theta_1}, \dot{\theta}_2 = -\gamma e_2 \frac{\partial y}{\partial \theta_1} \tag{35}$$

where $e_1 = y - y_m$. To implement (35), we need to generate the sensitivity functions $\frac{\partial y}{\partial \theta_1}, \frac{\partial y}{\partial \theta_2}$ online.

A. MRAC simulation and experimental test results

In this part, the simulation and implementation of the controller with fault-free and post-fault are illustrated. In fault situation the MRAC acts as a Fault Tolerant Controller (FTC) and compensates the loss of the power caused by actuator gain in brushless motors by retuning the gains of actuators in real-time control.

The simulation of the MRAC in cooperation with LQR showed the best result. As it is shown in Fig. 6, the output of the controller is tracking the reference signal in a good manner.

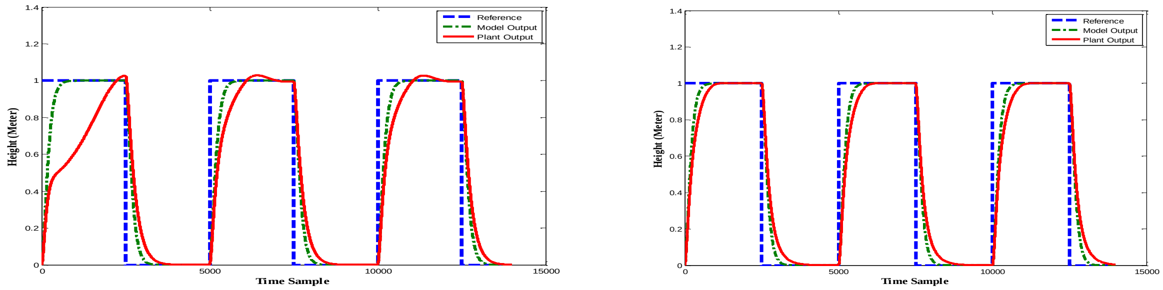

Figure 6. MRAC response (left) and combination of MRAC and LQR in simulation (right)

In fault-free implementation to quadrotor, the controller showed to be robust enough against modeling uncertainties and external disturbances. Different trajectories are applied to the Qball such as square, triangle and circle, and the results were satisfactory .

In FTC part, the same fault is modeled and injected to all four motors at the same time. The magnitude of the fault was considered as 14.2% loss of the actuator power. This fault is injected in different parts of the flight, like in hover condition and trajectory tracking scenarios. In hover mode the Qball lost the height in the first few seconds after fault occurrence in actuator, but the controller was able to maintain the desired altitude in few latter seconds.

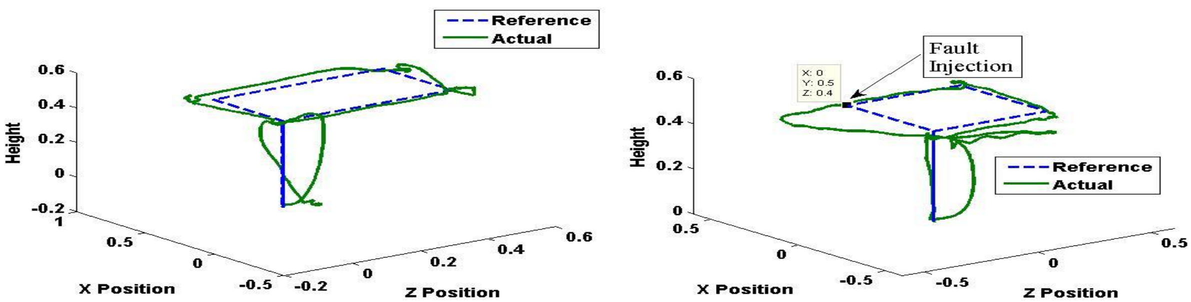

Figure 7. Square trajectory tracking in fault-free and faulty conditions (left and back motors) using the combination MRAC and LQR controller

In Figure 8, change of all four PWM signals is shown in both pre-fault and post-fault periods.

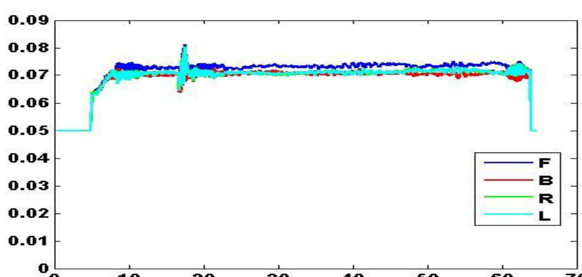

Figure 8. PWM input for 14.2% of power loss in all 4 actuators

Some other fault injection scenarios are applied to quadrotor helicopter like power loss of one, two, and three an all four actuators in hovering mode and trajectory tracking mode. In trajectory tracking mode the Qball was successful to regain the desired height and to continue the trajectory tracking and to land safely.

In the damage-tolerant last part of the work with MRAC, 15% of the propeller was broken during flight in order to test the reaction of the controller and Qball based on the LQR and the MRAC. Qball could land safely. The result is shown in Fig. 9 and also a video that can be obtained at: http://users.encs.concordia.ca/~ymzhang/UAVs.htm.

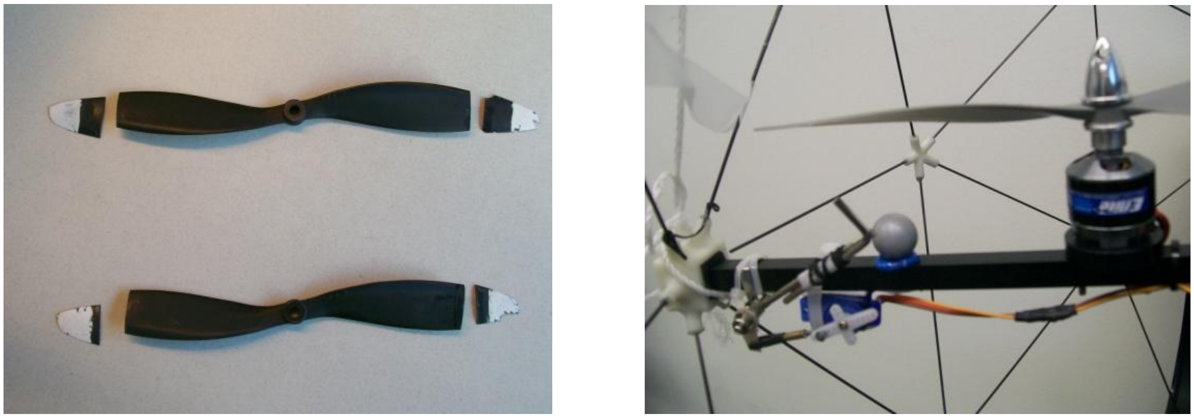

Figure 9. Damaged propellers (15% and 23% of power loss) with the fault injection mechanism

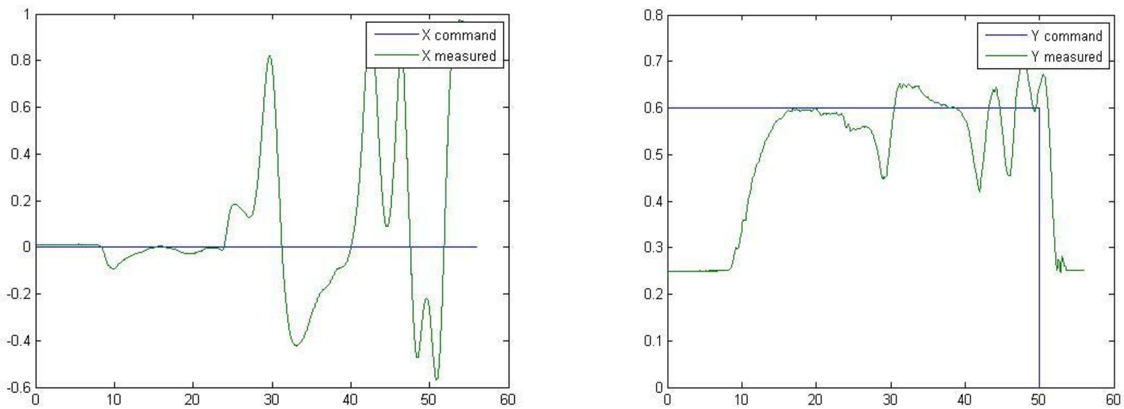

Figure 10. The Qball-X4 response to a 15% of propeller damage in X and Y directions

B. Gain-Scheduled Proportional-Derivative-Integral (GS-PID) Controller

In view of the advantages of widely used Proportional-Integral-Derivative (PID) controller and gain scheduling control strategy in aerospace and industrial applications, a control strategy using gain scheduling based PID controller is proposed for fault tolerant control of our test-bed Qball-X4.

For GS-PID controller, many sets of pre-tuned gains are applied to the controllers in different flight conditions under both fault-free and faulty cases. In the next step, attempts to obtain the best stability and performance of Qball-X4 in height control under both cases and to switch the controller gains from every pre-tuned set to another set of the gains in the presence of different levels of actuator faults have been carried out. The results are compared to the single PID controller and the GS-PID showed to be fairly reliable with a high reliability, stability and performance of the Qball-X4. One of the main parameters to consider in GS-PID is the switching time between the time of fault occurrence and the time of switching to new set of gains. In other words, if this transient (switching) time is held long (more than one second) it can cause the Qball-X4 to hit the ground and cause a crash, since the operating height was considered as 70 cm to 1 meter. The structure of a PID controller implemented in the Qball-X4 software environment is shown in Figure 11.

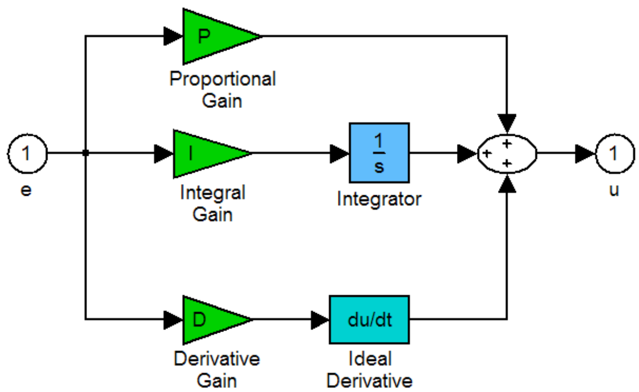

Figure 11. PID controller structure

For PID controllers, it can be seen from the plots that the single PID controller, which is tuned well for normal take-off and hovering, is not able to handle the faulty situation, but the GS-PID handles the fault better than the single PID. The fault occurrence and detection time is vital for the stability and the acceptable performance of the Qball-X4. The comparison is shown in Figure 12. Better performance with a shorter time delay of 0.5s between fault injection and the switching time has been achieved which verified the importance of fast and correct fault detection and control switching (reconfiguration) after fault occurrence.

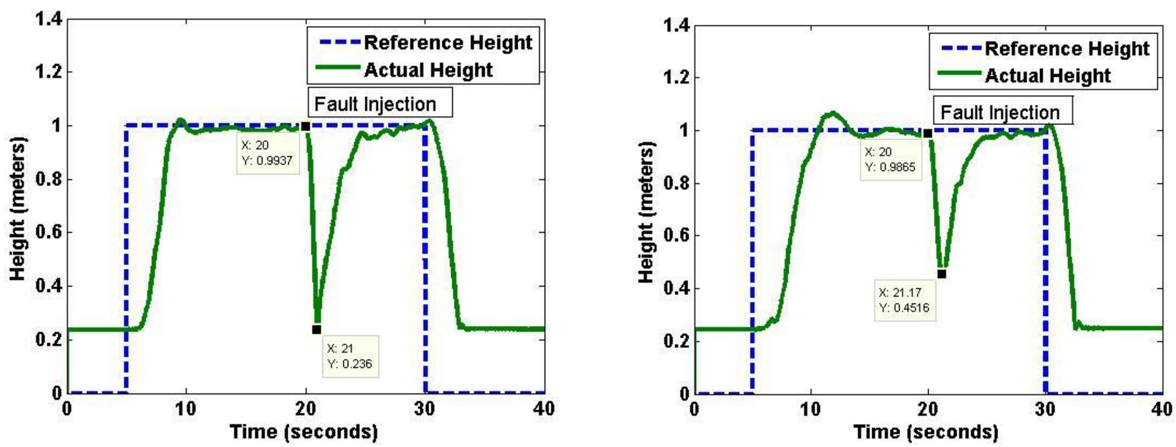

Figure 12. PID controller response (left) and GS-PID controller response (right) to the same fault

V. Conclusion and Future Work

In this work, we used two types of controllers to handle the faults in a quadrotor helicopter. Model Reference Adaptive Control (MRAC) combined with a Linear Quadratic Regulator (LQR) controller is proposed for fault tolerant control of the height and trajectory tracking of the Qball-X4 quadrotor helicopter UAV. MRAC is concerned with forcing the dynamic response of the controlled system to asymptotically approach that of a reference system, despite parametric uncertainties and faults in the UAV. Gain-Scheduled Proportional-Integral-Derivative (GS-PID) uses different sets of control gains with proportional fault magnitudes. GS-PID as well as MRAC showed to be able to compensate the fault/damage injected during hovering and flight conditions. Also a damage tolerant scenario is tested by MRAC. Regarding MRAC, the best result was obtained by combining the MIT rule with the LQR technique and the response of Qball-X4 was much better in 14.2% of fault in actuator comparing to 15% of fault (physical damage) in the propeller caused by fault injection mechanism in the propeller end tips. The test results also showed that 23% of power loss caused by the propeller damage will be over the control and safety limits of the actuators and the Qball-X4 was not able to compensate the fault and to land safely. Further, the GS-PID showed to be robust after a good tuning to the same fault. Overall, the result of both controllers were very close, but the MRAC showed to be more accurate and responsive rather than GS-PID.

As one of future work we are working on the GS-PID controller for trajectory tracking control. Attempt will be made also to extend the MRAC for roll, pitch and yaw controls. Videos showing the flight testing results presented in this paper and any new development can be found in: http://users.encs.concordia.ca/~ymzhang/UAVs.htm.

Acknowledgments

This work is mainly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Strategic Project Grant (SPG). Development and support of the Qball-X4 quadrotor helicopter UAV by Quanser Inc. is also highly appreciated.

References

[1] Sadeghzadeh, I., Mehta, A., and Zhang, Y. M., "Fault Tolerant Control of a Quadrotor Umanned Helicopter using Model Reference Adaptive Control," *The Third Symposium on Small Unmanned Aerial Vehicle Technologies and Applications (SUAVTA'11) in ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications (MESA2011)*, Washington, DC, USA, August 28-31, 2011.

[2] Quanser Inc., *Qball Manual*, available at http://www.quanser.com/english/html/UVS_Lab/fs_Qball_X4.htm

[3] Whitehead, B. T., and Bieniawskiy, S. R., "Model Reference Adaptive Control of a Quadrotor UAV," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[4] Gadient, R., Levin, J., and Lavretsky, E.,"Comparison of Model Reference Adaptive Controller Designs Applied to the NASA Generic Transport Model," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[5] Levin, J., "Alternative Model Reference Adaptive Control," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[6] Guo, J. and Tao, G., "A Multivariable MRAC Scheme Applied to the NASA GTM with Damage," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[7] Lemon, K. A., Steck, J. E., and Hinson, B. T., "Model Reference Adaptive Fight Control Adapted for General Aviation: Controller Gain Simulation and Preliminary Flight Testing on a Bonanza Fly-By-Wire Testbed," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[8] Bierling, T., Hocht, L., and Holzapfel, F., "Comparative Analysis of MRAC Architectures in a Unified Framework," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[9] Crespo, L. G., Matsutani, M., and Annaswamy, A. M., "Design of a Model Reference Adaptive Controller for an Unmanned Air Vehicle," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[10] Stepanyan, V., Campbell, S., and Krishnakumar, K., "Adaptive Control of a Damaged Transport Aircraft Using M-MRAC," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[11] Dydek, Z. T. and Annaswamy, A. M., "Combined/Composite Adaptive Control of a Quadrotor UAV in the Presence of Actuator Uncertainty," *AIAA Guidance, Navigation, and Control Conference*, 2-5 August 2010, Toronto, ON, Canada.

[12] Milhim, A. B., Zhang, Y. M., and Rabbath, C.-A., "Gain Scheduling Based PID Controller for Fault Tolerant Control of a Quad-Rotor UAV," *AIAA Infotech@Aerospace 2010*, 20-22 April 2010, Atlanta, Georgia, USA.