2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Active Fault Tolerant Control of Quadrotor UAV Using Sliding Mode Control

Abdel-Razzak Merheb[1], Hassan Noura[2] and François Bateman[3]

Abstract— **In this paper, an active fault tolerant controller using Sliding Mode Observers and Sliding Mode Control for fault tolerant control of a quadrotor UAV is proposed. The sliding mode observer estimates online the amount of fault injected in one of the quadrotor motors. The estimated fault information is then used to update the reconfigurable sliding mode controller responsible for the UAV control. Controller reconfiguration is done using two different methods, adding a control residue and scaling the controls by the amount of fault. The controller is tested in Simulink under partial loss of one motor speed actuator fault. Results demonstrated the effectiveness of the controller despite the lack of hardware redundancy of the quadrotor system.**

I. INTRODUCTION

Quadrotors are under-actuated systems with highly coupled states subjected to complicated aerodynamic situations, high disturbances, actuator failures, and system faults [1] [2]. Sliding Mode Controllers (SMCs) are powerful control strategies that have been widely used to construct passive fault tolerant schemes for quadrotors ([3], [4], [5]). Passive Fault Tolerant Controllers (PFTC) are designed for a previously considered failure types and magnitudes, and the behavior of the system under different faults can not be predicted. On the other hand, Active Fault Tolerant Controllers (AFTC) react properly to any failure by reconfiguring the control actions according to the magnitude and type of the fault occurred.

Authors in [6] develop a novel active fault tolerant control based on sliding mode control. Matrix full-rank factorization technique is used to design the sliding mode surface. Faults are tolerated by using flexible design parameters to update online the gain of the discontinuous part of the SMC, and no fault detection and isolation (FDI) scheme is used. Faults considered are system faults resulting from model uncertainties and disturbances, input faults resulting from actuator outage, loss of effectiveness, actuator stuck, and total actuator failure under a special actuator redundancy assumption. In [7], a cascaded sliding mode observer method to reconstruct actuator faults for a class of descriptor linear systems is developed. A new method to discuss the existence conditions of the sliding mode observer based on a new canonical form is also presented. Simulation results has shown promising results that encourage the application of the new method for sensor fault reconstruction problems. Chamseddine et al. proposed an SMC based active fault tolerant controller for a system formed of a nonlinear vehicle active suspension with its hydraulic actuation [8]. The controller is designed to tolerate terrain disturbances and heave sensor breakdown which affects the attitude of the system. FDI unit is based on sliding mode observer scheme which estimate any loss in sensor data. Whenever a fault occurs, estimated values are injected in the controller and system response is corrected. Authors pointed that with some critical sensor faults the system looses its observability rendering the fault uncorrected. They suggest also the design of a bank of observers to handle multiple faults. Simulation results proved the effectiveness of the proposed technique.

Quadrotor active and passive fault-tolerant controllers based on sliding mode control are developed in [9]. In active controller there exists a pool of predesigned controllers suitable for each fault case, and the controller switches to the convenient controller whenever a fault is detected. Simulation and experimental results show the robustness of the active FTC representing better tracking performance in presence of faults. In [10], Sliding Mode Control and Control Allocation scheme are combined to form an online Fault Tolerant controller for UAV applications. Whenever an actuator fault occurs, the new controller uses the effectiveness level of the actuators to redistribute the control signals to the remaining non-faulty actuators. The new scheme implemented on ADMIRE flying UAV model has shown that with good FTC, reconfiguration of the controller may not be necessary, and faults and total actuator failures might be handled directly. In [11], authors develop two Active Fault Tolerant Controllers based on SMC. A standard linear Observer is used to calculate the residual vector and detect system and actuator faults of linear MIMO systems. The residual vector can then be used to update the gain of the discontinuous term of the SMC in an adaptive manner. The second approach is to update the equivalent control term of the SMC with respect to fault distribution information. Simulation results show that both approaches were able to maintain acceptable performance of a linear MIMO system and avoid its dynamics to wind-up, however, the second approach exhibits better performance. Li et al. use the augmented sliding mode techniques to design passive and active fault tolerant control sliding mode controllers [12]. Both controllers are tested in simulation and experimentally on a Qball-X4 quadrotor under real propeller damage. The active controller shows better

[1]A. Merheb is a PhD student Aix-Marseille University, France, and Lab instructor at the Lebanese International University, Lebanon abdelrazzac10@yahoo.fr

[2]Prof. H. Noura is the Chairman of EENG Department at United Arab Emirates University, Al Ain, UAE hnoura@uaeu.ac.ae

[1]François Bateman, PhD, is at the French Air Force Academy, Salon-de-Provence, France francois.bateman@club-internet.fr

tracking performance compared to the passive controller, but both controllers drive the quadrotor successfully under fault. Authors , in addition, present a comparison between the two approaches, and address the advantages, disadvantages, and limitations of each approach. Authors develop an active fault tolerant sliding mode controller for quadrotor UAV in [13]. Using the robustness property inherent in SMC and a reconfiguration technique, the new controller is able to handle both external disturbances and actuator faults. Fault Detection and Isolation (FDI) is based on the residual vector generated by a linear observer, and is able to distinguish between external disturbances and actuator faults. On fault detection, the reconfiguration scheme of the FTC changes the rotors trust forces rather than the control signals. SIMULINK results verify the effectiveness of the new controller where a partial loss in a rotor trust is immediately detected and tolerated enabling the quadrotor to follow the desired trajectory for a safe landing. Zhang and Chamseddine present different fault tolerant control techniques applied to the Qball-X4 quadrotor testbed under propeller damage [14]. Sliding Mode Control (SMC) and Backstepping Control (BSC) are used as passive FTCs, Gain-Scheduled PID, Model Reference Adaptive Control, and Model Predictive Control are used as active fault tolerant controllers to ensure the reference tracking of the quadrotor in presence of a partial loss in the control effectiveness of one or more actuators. The fault detection and diagnosis (FDD) scheme is not shown in the paper, and authors refer to recent articles of the Networked Autonomous Vehicles Laboratory of Concordia University. Results emphasize the power of the presented controllers. Authors in [15] design a robust sliding mode controller with fault tolerant properties for the highly nonlinear Machan UAV. The controller has two cascaded loops where the inner loop uses feedback linearization technique to linearize and decouple online the nonlinear system into three linear SISO second order sub-systems. Sliding mode controllers developed using a simple and effective phase modulation method form the outer loop controlling the new linear subsystems. Simulation results emphasize the importance of the new controller and prove the efficiency and effectiveness of the proposed strategy.

In this paper, the inherent robustness of Sliding Mode Control is used with the quadrotor UAV not only to compensate for disturbances and system uncertainties, but also to tackle for actuator faults. The controller proposed here differs from the one presented in [5] in that it is active i.e. the SMC outputs change according to fault magnitude.

II. QUADROTOR DYNAMICS

Quadrotors are under-actuated rotorcrafts that use four propellers to perform motion in the space (Figure 1). They are called under-actuated because their actuators number is less than their Degree-Of-Freedom: only four actuators (propellers) are used to control six variables, the coordinates x, y, and z, and the altitude variables ϕ, θ, and ψ. The

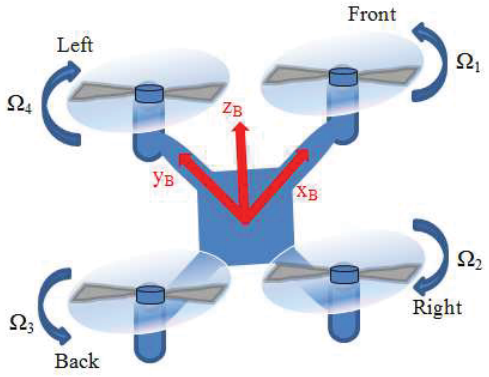


Fig. 1: Quadrotor schematic.

dynamic equations of a quadrotor are [16]

$$\ddot{x} = \frac{U_1}{m}(sin\psi sin\phi + cos\psi sin\theta cos\phi) - \frac{K_{ftx}}{m}\dot{x} \quad (1)$$

$$\ddot{y} = \frac{U_1}{m}(-cos\psi sin\phi + sin\psi sin\theta cos\phi) - \frac{K_{fty}}{m}\dot{y} \quad (2)$$

$$\ddot{z} = \frac{U_1}{m}cos\theta cos\phi - \frac{K_{ftz}}{m}\dot{z} - g \quad (3)$$

$$\ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{I_{rotor}}{I_x}\dot{\theta}\gamma - \frac{K_{fax}}{I_x}\dot{\phi}^2 + \frac{lU_2}{I_x} \quad (4)$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} - \frac{I_{rotor}}{I_y}\dot{\phi}\gamma - \frac{K_{fay}}{I_y}\dot{\theta}^2 + \frac{lU_3}{I_y} \quad (5)$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} - \frac{K_{faz}}{I_z}\dot{\psi}^2 + \frac{U_4}{I_z} \quad (6)$$

Note that the effect of aerodynamic forces is presented by K_{fax}, K_{fay}, and K_{faz} which are the aerodynamic friction coefficients. K_{ftx}, K_{fty}, and K_{ftz} are the coefficients of the translation drag forces affecting the coordinates of the quadrotor. U_1, U_2, U_3, and U_4 are the controls, and γ is defined as

$$\gamma = U_1 - U_2 + U_3 - U_4 \quad (7)$$

Because no sensor is implemented on the motors, it is impossible from a practical point of view to measure the speed of each motor (however prediction is possible using observers). This means that the value of γ is not available for the controller of the quadrotor, and the term $\frac{I_{rotor}}{I_i}.\dot{j}.\gamma$ with $i = x, y$, and $j = \phi, \theta$ cannot be used in the controller design. This is why we assume the effect of γ as disturbance, well known for us but impossible to measure practically. The controls can be presented by the speeds of the four propellers as

$$U_1 = b(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (8)$$

$$U_2 = M_x = bl(-\Omega_2{}^2 + \Omega_4{}^2) \quad (9)$$

$$U_3 = M_y = bl(-\Omega_3{}^2 + \Omega_1{}^2) \quad (10)$$

$$U_4 = M_z = d(-\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 + \Omega_4{}^2) \quad (11)$$

Where Ω_i with $i = 1, 2, 3, 4$ is the speed of motor i. x, y, and z are the coordinates of the quadrotor with respect to the base frame, ϕ, θ, and ψ are respectively the roll, pitch,

TABLE I: Variables Used In Quadrotor Modeling

I_x	$8.1e^{-3}N.m.s^2$	b	$54.2e^{-6}N.s^2$
I_y	$8.1e^{-3}N.m.s^2$	l	$0.24m$
I_z	$14.2e^{-3}N.m.s^2$	d	$1.1e^{-6}m.s^2$
I_{rotor}	$104e^{-6}N.m.s^2$	g	$9.8m.s^{-2}$
m	$1Kg$	K_{fax}, K_{fay}	$5.56e^{-4}N/rad/s$
K_{faz}	$6.3540e^{-4}N/rad/s$	K_{ftx}, K_{fty}	$5.56e^{-4}N/m/s$
K_{ftz}	$6.3540e^{-4}N/m/s$		

and yaw angles of the quadrotor. $I_{x,y,z}$ are the quadrotor moments of inertia with respect to the corresponding axis, m is the mass of the quadrotor, and g is the gravitational constant. Table I shows the parameters of the Asctec Pelican quadrotor used in this paper.

By taking $X = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^t$ as state vector, the dynamics of the quadrotor can easily be expressed in the state space form

$$\dot{X} = f(X) + g(X)u(t) \tag{12}$$

Where $u(t) = [U_1\ U_2\ U_3\ U_4]^t$ is the control input.

III. ACTIVE FAULT TOLERANT CONTROL BASED ON SLIDING MODE CONTROL

In this section, the design of the SMC based active fault tolerant controller is provided. The controller here is based on the regular SMC with a Fault Diagnosis and Identification (FDI) unit, and an adaptive control scheme. The FDI unit is a sliding mode observer used to detect and estimate the fault magnitude. The adaptive control scheme uses the fault information given by the observer to reconstruct the control laws applied to the quadrotor. Control reconstruction is made by using the fault magnitude to calculate the amount of control loss, and adding this values again to the control laws. Another approach is to use the amount of control loss to scale the control laws. Both approaches are interpreted in this section.

A. Design of Sliding Mode Controller

The Sliding Mode Control law has two parts, the discontinuous control used to bring the system states towards the pre-designed sliding surface $s = 0$, and the equivalent control which maintains the states on this surface.

We choose the sliding surface $s = e + c\int e(\tau)d\tau$, with $e = x_d - x$ for $x = \phi, \theta, \psi$, and z. Let $V = (1/2)s^2$ be a positive definite Lyapunov function, if $\dot{V}$ is negative semi definite then the sliding condition is satisfied. The derivative of the Lyapunov function is $\dot{V} = s\dot{s} = s(\dot{e} + ce) = s(\dot{x}_d - \dot{x} + ce)$. For $\dot{V}$ to be negative semi definite $\dot{x}_d - \dot{x} + ce$ should be less than zero. The dynamic equations of the quadrotor are then used to derive the equivalent control

$$u_{eq} = g^{-1}(X)\left[\dot{X}_d + ce - f(X)\right] \tag{13}$$

With $c = diag([c_z;\ c_\phi;\ c_\theta;\ c_\psi])$ a positive gain matrix. The discontinuous control is designed using the saturation function, a continuous switching function that takes the system towards the designed sliding surface regardless the sign of the states

$$u_{dis} = -ksat(s) \tag{14}$$

With $k = [k_z\ k_\phi\ k_\theta\ k_\psi]$ a positive gain vector affecting the conversion speed of the discontinuous control. The Sliding Mode Control of the quadrotor is then

$$u = g^{-1}(X)\left[\dot{X}_d + ce - f(X)\right] - ksat(s) \tag{15}$$

Vectors c and k should be chosen carefully to ensure the reachability condition or the negative definite of the derivative of the Lyapunov function. A good approach is to use a stochastic search algorithm to pick the best values of c and k; we choose a biologically-inspired algorithm called Ecological System Algorithm (ESA) to perform the search. While tuning the controllers, ESA first provides the controller with suggested values of c and k vectors, and by using an SMC controlled quadrotor model with step inputs it measures the feasibility of the recommended vectors by measuring the amount of error made. ESA then updates the vector values using ecological rules, and the controller with the new values is tested again. These steps repeat until the optimal vectors producing the maximum value of a fitness function are found. The optimal values for the SMC controller used in this paper are shown in table II. More information about the Ecological Systems Algorithm and its application to find the quadrotor SMC vectors can be found in [5] and [17] (see also Appendix C).

TABLE II: Best Vector Values Found Using ESA

	z	ϕ	θ	ψ
c	16.56	6.84	8.14	4.37
k	155.5	0.916	7.86	2.68

The stability of the controller is tested by choosing a Lyapunov function $V = 1/2s^2$, finding its derivative $\dot{V} = s(\dot{X}_d - \dot{X} + ce) = s(\dot{X}_d - f(X) - g(X)u + ce)$ and testing whether it is negative or not. With the controller law inserted, the Lyapunov function becomes $\dot{V} = s(f(X) - \tilde{f}(X) - kg(X)sign(s))$, where $\tilde{f}(X)$ is identical to $f(X)$ with no γ multiplier term, and it is used in the controller equations instead of $f(X)$ because there are no sensor implemented practically on the quadrotor to measure the speeds of the motors. Let $F(X)$ be $f(X) - \tilde{f}(X)$, $\dot{V} = sF(X) - kg(X)|s|$ is negative if we choose $k \geq \dfrac{F(X) + \eta}{g(X)}$, with η is a small positive number. Using the quadrotor parameters, the manufacturer constraints, and the fact that $\gamma_{max} = 1046$ (equation (7)), it is easy to find that the gains k should be $k_z \geq 0$, $k_\phi \geq 0.4532\dot{\phi}$, $k_\theta \geq 0.4532\dot{\theta}$, and $k_\psi \geq 0$ in order for the controller to be Lyapunov stable. Note that $g(X)$ is a positive function because $\dfrac{l}{I_i} \geq 0$ with $i = x, y, z$, and $g_z(X) = \dfrac{cos(\phi)cos(\theta)}{m} \geq 0$ because the quadrotor angles are within $[-0.31\ 0.31]rad$ interval as stated by the manufacturer. To ensure the safety of the quadrotor, we allow

its angles to change from minimum to maximum value in no less than $0.5sec$. This means that $\dot{\phi} \leq 1.24rad/sec$, which makes k_ϕ and k_θ greater than 0.562 in order for the controller to be stable. SMC controller gains shown in table II respect the constraint found previously which means that the controller is Lyapunov stable.

B. Design of Sliding Mode Observer

The dynamics of the quadrotor system under faults is expressed in the following form

$$\dot{X} = f(X) + g(X)u(t) + f(t) \tag{16}$$

Where $f(t)$ is a time-varying fault function. All faults are represented here as an additive function, rather than being represented by changes in the state-space matrices of the system. A convenient observer for the above faulty system is the augmented Luenberger form

$$\dot{\tilde{X}} = f(\tilde{X}) + g(\tilde{X})u(t) + v(t) + L(Y - \tilde{Y}) \tag{17}$$

$$\tilde{Y} = C\tilde{X} \tag{18}$$

Where $\tilde{Y}$ is the output of the system, $C^{1\times 12}$ is a constant vector with all its elements alternating between the unity and zero, L is the observer gain vector, and $v(t)$ is the time-varying estimator of the fault. As a first step, the Luenberger observer for fault-free situation will be designed. This observer has identical equation as shown in equation (17) but with no fault estimator term $v(t)$. The error dynamics is $\dot{e} = \dot{X} - \dot{\tilde{X}}$ or

$$\dot{e} = f(X) - f(\tilde{X}) + \left[g(X) - g(\tilde{X})\right] u - L(Y - \tilde{Y}) \tag{19}$$

Let $e_y = Y - \tilde{Y}$ and $V = \frac{1}{2}{e^2}_y$ be a Lyapunov function. The observer gain will be chosen so that the derivative of the Lyapunov function be negative definite, so that the output error e_y goes to zero in finite time

$\dot{V} \leq 0$ or $e_y\dot{e_y} = -\eta{e^2}_y \implies f(X) - f(\tilde{X}) + \left[g(X) - g(\tilde{X})\right] u - Le_y = -\eta e_y$ and thus

$$L = \frac{f(X) - f(\tilde{X}) + \left[g(X) - g(\tilde{X})\right] u}{e_y} + \eta \tag{20}$$

Where η is a positive design parameter affecting the convergence speed of the output error to zero.

Once the fault-free observer is designed, it is augmented by a sliding mode term responsible for online estimation of any system fault (equation (17)). The sliding mode fault estimation term v has two parts, the discontinuous part responsible for driving the estimation to a pre-designed sliding surface, and the equivalent part responsible for maintaining the estimation on this surface, $v = v_{eq} + v_{dis}$. We define by the sliding surface the estimation error $S = e = X - \tilde{X}$; the aim is to keep this error zero despite the presence of fault. The equivalent part of the fault estimation term is used to keep the estimation on the sliding surface when $\dot{e} = 0$

$\dot{e} = f(X) - f(\tilde{X}) + \left[g(X) - g(\tilde{X})\right] u - Le_y - v_{eq} + f(t) = 0$, and the equivalent part of the fault estimation term is then found as

$$v_{eq} = f(X) - f(\tilde{X}) + \left[g(X) - g(\tilde{X})\right] u - Le_y + f(t) \tag{21}$$

The next step is to design the discontinuous part of the fault estimation term responsible for bringing the estimation to the pre-designed sliding surface. This includes switching the estimation to the negative side of the sliding surface if it is initially at its positive side and vice versa. A good choice is to use the signum function $v_{dis} = -Ksign(S)$. To reduce chattering, the saturation function is used instead of the signum function

$$v_{dis} = -Ksat(S) \tag{22}$$

Where K is the discontinuous part gain affecting the conversion speed of v_{dis}. By trial and error, K_i is found as $K_\phi = K_\theta = K_\psi = 0.05$, and $K_x = K_y = K_z = 0.1$. The state space representation of the observer can be found in Appendix B.

By using the sliding mode observer of equation (17), one can estimate the actual states of the faulty system (16). Once the trajectory of the observer reaches the sliding mode, all the observed states converge to their actual value $\tilde{X} \longrightarrow X$. This means that the observer error converges to zero. Under this situation, the equivalent part of the estimation term can be used to reconstruct the fault $f(t) \longrightarrow v_{eq}$. To estimate the value of the fault, we injected many known faults to the sliding mode controlled system and measured the v_{eq} vector. We draw the graph of fault-v_{eq} values using MATLAB and extracted the equation of the graph (the fault magnitude with respect to v_{eq}) using $polyfit$ function. With unknown fault injected, calculating v_{eq} and using its value in the equation is enough to find the magnitude of the injected fault.

C. Design of Adaptive SMC

Once the fault is detected and its magnitude is estimated, it is now possible to use this fault to adapt the controller and keep the quadrotor following the desired path. In this part, the SMC designed previously will be upgraded with the fault magnitude to end up with an active fault tolerant control SMC. The design is somehow similar to the SMC based active fault tolerant controller designed in [12]. The faulty system has the following dynamics

$$\dot{X} = f(X) + g(X)u_R(t) + f(t) \tag{23}$$

Where $u_R(t)$ is the remaining control after a fault has occurred. Note that the faults interpreted in this paper affects the motors' speeds and result in a sudden drop in the effectiveness of the motors. $u_R = u - u_f$, where u_f is the control percentage loosed when the fault occurs, and u is the optimal control. The aim of fault tolerant control is to find the optimal control u and use it to control the faulty system. Following the same design steps of regular SMC, the equivalent part of the remaining control is found

as $u_{eq\,R} = g^{-1}(X)\left[\dot{X}_d + ce - f(X) + f(t)\right]$, and thus the optimal control is found as

$$u_{eq} = g^{-1}(X)\left[\dot{X}_d + ce - f(X) + f(t)\right] + u_f \tag{24}$$

To bring the system dynamics to the sliding surface, the same discontinuous control of the regular SMC is used (equation 14). The main objective of the design now is to find a parametric interpretation of u_f.

The relation between the quadrotor controls and its motors' speeds is expressed through its physical constraints

$$U = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ bl & 0 & -bl & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} {\Omega^2}_1 \\ {\Omega^2}_2 \\ {\Omega^2}_3 \\ {\Omega^2}_4 \end{bmatrix} = \Gamma\Omega \tag{25}$$

For faulty situation, the control-speed relation will be updated by the fault magnitude or

$$U_R = \Gamma_R\Omega = [\Gamma - \Gamma_f]\,\Omega \tag{26}$$

Where U_R is the remaining control after fault occurrence, Γ_R is the remaining Γ matrix, and Γ_f is the faulty matrix defined as

$$\Gamma_f = \begin{bmatrix} f_1 b & f_2 b & f_3 b & f_4 b \\ 0 & -f_2 bl & 0 & f_4 bl \\ f_1 bl & 0 & -f_3 bl & 0 \\ -f_1 d & f_2 d & -f_3 d & f_4 d \end{bmatrix} \tag{27}$$

Where f_i is the percentage of error motor i is subjected to. By defining the elements of vector $F^{1\times 4}$ as

$$F_i(X) = \begin{cases} 0, & \text{if there is no fault} \\ \frac{E_i}{100}, & \text{if a fault of percentage } E_i \quad \exists \end{cases} \tag{28}$$

We can express Γ_f as $\Gamma_f = \Gamma\mathbb{I}F$, and the faulty control will be then

$$U_f = \Gamma\mathbb{I}F\Omega \tag{29}$$

To correct the controls, the adaptive SMC controller adds the faulty controls to the remaining controls $U = U_R + U_f$ or

$$\begin{cases} U_1 = U_{R1} + b(F_1.{\Omega^2}_1 + F_2{\Omega^2}_2 + F_3{\Omega^2}_3 + F_4{\Omega^2}_4) \\ U_2 = U_{R2} + bl(-F_2{\Omega^2}_2 + F_4{\Omega^2}_4) \\ U_3 = U_{R3} + bl(F_1{\Omega^2}_1 - F_3{\Omega^2}_3) \\ U_4 = U_{R4} + d(-F_1{\Omega^2}_1 + F_2{\Omega^2}_2 - F_3{\Omega^2}_3 + F_4{\Omega^2}_4) \end{cases} \tag{30}$$

Where U is the optimal controller capable of driving the quadrotor under fault, and U_R is the remaining controller generated under fault with no fault tolerant properties.

Another approach based on multiplication can also be used to reconstruct the control laws. Here, the control laws are scaled directly without the need for the remaining controls

$$U = \begin{bmatrix} \frac{b}{1-F_1} & \frac{b}{1-F_2} & \frac{b}{1-F_3} & \frac{b}{1-F_4} \\ 0 & \frac{-bl}{1-F_2} & 0 & \frac{b.l}{1-F_4} \\ \frac{b.l}{1-F_1} & 0 & \frac{-b.l}{1-F_3} & 0 \\ \frac{-d}{1-F_1} & \frac{d}{1-F_2} & \frac{-d}{1-F_3} & \frac{d}{1-F_4} \end{bmatrix} \begin{bmatrix} {\Omega^2}_1 \\ {\Omega^2}_2 \\ {\Omega^2}_3 \\ {\Omega^2}_4 \end{bmatrix} \tag{31}$$

or

$$\begin{cases} U_1 = b(\frac{{\Omega^2}_1}{1-F_1} + \frac{{\Omega^2}_2}{1-F_2} + \frac{{\Omega^2}_3}{1-F_3} + \frac{{\Omega^2}_4}{1-F_4}) \\ U_2 = bl(-\frac{{\Omega^2}_2}{1-F_2} + \frac{{\Omega^2}_4}{1-F_4}) \\ U_3 = bl(\frac{{\Omega^2}_1}{1-F_1} - \frac{{\Omega^2}_3}{1-F_3}) \\ U_4 = d(-\frac{{\Omega^2}_1}{1-F_1} + \frac{{\Omega^2}_2}{1-F_2} - \frac{{\Omega^2}_3}{1-F_3} + \frac{{\Omega^2}_4}{1-F_4}) \end{cases} \tag{32}$$

Simulations show that both approaches have similar results with more realistic and safer response for the addition approach. Rotor speeds in the scaling approach oscillate rapidly as shown in figure (11) which is not realistic for the UAV control. Note that Ω_i which is the speed of motor i is found from the remaining controls using the following equations

$${\Omega^2}_1 = \frac{1}{4b}U_{R1} - \frac{1}{2bl}U_{R3} - \frac{1}{4d}U_{R4} \tag{33}$$

$${\Omega^2}_2 = \frac{1}{4b}U_{R1} - \frac{1}{2bl}U_{R2} + \frac{1}{4d}U_{R4} \tag{34}$$

$${\Omega^2}_3 = \frac{1}{4b}U_{R1} + \frac{1}{2bl}U_{R3} - \frac{1}{4d}U_{R4} \tag{35}$$

$${\Omega^2}_4 = \frac{1}{4b}U_{R1} + \frac{1}{2bl}U_{R2} + \frac{1}{4d}U_{R4} \tag{36}$$

IV. SIMULATION RESULTS

The sliding mode controller designed in the previous section is used to control the attitude of the Asctec Pelican quadrotor along with its altitude variables (z, ϕ, θ, and ψ) in SIMULINK environment. The quadrotor is used to follow two different paths in the space, a continuous path of helical shape, and a closed square of $10m$ long sides at $5m$ height (figure 2). To ensure the position control of the quadrotor, a simple PD controller is used. This controller (equation (37)) is used to check the deviation of the quadrotor and generate the suitable controls

$$\ddot{i} = k_{ip}(i_d - i) - k_{id}\dot{i} \tag{37}$$

Where i is x, y, or z, and $k_{ip} = 2$ and $k_{id} = 3$ are respectively the proportional and derivative gains found by trial and error. The controls generated by this controller are used to generate the desired values of altitude and height variables using the following equations

$$\phi_d = \arctan(\frac{-\ddot{y}}{\sqrt{(\ddot{x}^2 + (\ddot{z} + g)^2)}}) \tag{38}$$

$$\theta_d = \arctan(\frac{\ddot{x}}{\ddot{z} + g}) \tag{39}$$

$$\psi_d = 0 \tag{40}$$

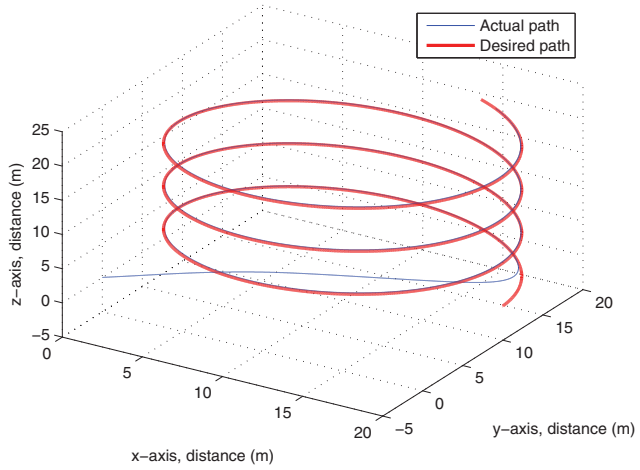
The equation of the continuous helical path is

$$x_d = 10cos(0.1t) + 10 \tag{41}$$

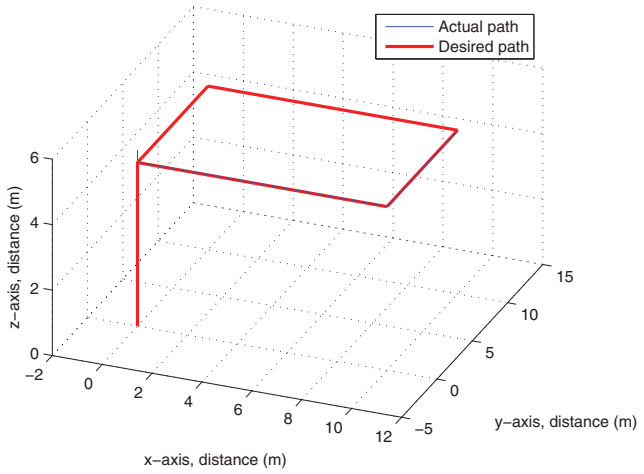
$$y_d = 10sin(0.1t) + 10 \tag{42}$$

$$z_d = 0.1t \tag{43}$$

The effectiveness of the active fault tolerant controller is tested using partial loss in one of the motors speed resulted



(a) Helical Path



(b) Square Path

Fig. 2: Paths of the SMC controlled Quadrotor (55% fault in motor 1)

from mechanical or voltage problems. This fault results in a partial loss of effectiveness for the infected motor which affects the lift force generated by the corresponding rotor. Results of the active fault tolerant sliding mode controller show better tracking performance and higher robustness compared to the passive fault tolerant control of the same sliding mode controller in [5].

The active controller is able to handle up to 55% of effectiveness loss of one of the quadrotor motors. This is very close to results of [5] but with important difference: no great position error increasing with the magnitude of the faults is seen at the altitude response. The remaining error at the attitude is very small (in terms of milli-radiant), and has very little effect on the motion of the quadrotor. To compensate this small error it is enough to augment the controller with an integrator, and the Sliding Mode Controller of the quadrotor becomes

$$u(t) = g^{-1}(X)\left[\dot{X}_d + ce - f(X)\right] - ksat(s) + k_i \int e(\tau)d\tau \quad (44)$$

Where $k_i = [k_{iz}\ k_{i\phi}\ k_{i\theta}\ k_{i\psi}]$ is the integrator gain vector, chosen by trial and error to have all its elements equal to unity.

Figures (3) to (10) illustrate the active FTSMC test using the addition control reconstruction approach (equation (30)) with 55% fault in motor 1 injected at $t = 40s$. The scaling control reconstruction approach gives similar attitude and height response, but with oscillating rotor speeds values (figure (11)) which makes it less applicable and less safe.

V. CONCLUSION

A Sliding Mode based active Fault Tolerant controller for Asctec Pelican quadrotor is developed. Fault Diagnosis and Identification unit uses a sliding mode observer which is a linear Luenberger observer for state estimation augmented with a sliding term for fault identification and estimation. Two approaches were presented for the reconfiguration scheme responsible for updating the control laws based on information provided by the FDI unit: reconfiguration using addition and reconfiguration using control scaling. SIMULINK results show good fault tolerant performance of the controller despite the under-actuated system used. Active controller shows better tracking error compared with the passive fault tolerant regular SMC controlling the same quadrotor. Real implementation of the proposed controller on the Asctec Pelican quadrotor will be conducted in the near future.

ACKNOWLEDGMENT

This research is supported by the United Arab Emirates University, Al-Ain, UAE.

REFERENCES

[1] I. Sadeghzadeh, A. Mehta, Y. Zhang, and C.-A. Rabbath, "Fault-tolerant control of quadrotor helicopter using gain-scheduled pid and model reference adaptive control," in *2011 Annual Conference of the Prognostics and Health Management Society*, Montreal, Quebec, Canada, September 2011.

[2] H. Huang, G. Hoffmann, S. Waslander, and C. TomlinTomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *IEEE International Conference on Robotics and Automation*. Japan: IEEE, May 12-17 2009.

[3] R. Youssef and H. Peng, "Piecewise sliding mode decoupling fault tolerant control system," *American Journal of Applied Sciences*, vol. 7, no. 1, pp. 102–109, 2010.

[4] J. Wu, Z. Weng, Z. Tian, and S. Shi, "Fault tolerant control for uncertain time-delay systems based on sliding mode control," *KYBERNETIKA*, vol. 44, no. 5, p. 617 – 632, 2008.

[5] A. Merheb, H. Noura, and F. Bateman, "Passive fault tolerant control of quadrotor uav using regular and cascaded sliding mode control," in *2nd International Conference on Control and Fault-Tolerant Systems, SysTol'13*, Nice, France, October 2013.

[6] L.-Y. Hao and G. Yang, "Robust fault tolerant control based on sliding mode method for uncertain linear systems with quantization," *ISA Transactions*, April 2013.

[7] J. Yu, G. Sun, and H. Karimi, "Fault-reconstruction-based cascaded sliding mode observers for descriptor linear systems," *Mathematical Problems in Engineering*, vol. 2012, no. 623426, July 2012.

[8] A. Chamseddine, H. Noura, and M. Ouladsine, "Sensor fault-tolerant control for active suspension using sliding mode techniques," in *Workshop on networked control systems and fault-tolerant control*, Ajaccio, Corsica, France, October 2005.
[9] T. Li, Y. Zhang, and B. W. Gordon, "Passive and active nonlinear fault-tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering published online*, October 2012.
[10] H. Alwi and C. Edwards, "Fault tolerant control using sliding modes with on-line control allocation," *Automatica*, vol. 44, pp. 1859–1866, April 2008.
[11] U. Demirci and F. Keresteciogl u, "Fault tolerant control with re-configuring sliding-mode schemes," *Turk J Elec Engin*, vol. 13, no. 1, 2005.
[12] T. Li, Y. Zhang, and B. Gordon, "Passive and active nonlinear fault-tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, pp. 1–12, October 2012.
[13] F. Sharifi, M. Mirzaei, B. Gordon, and Y. Zhang, "Fault tolerant control of a quadrotor uav using sliding mode control," in *Conference on Control and Fault Tolerant Systems (2010)*, Nice, France, October 2010, pp. 239–244.
[14] Y. Zhang and A. Chamseddine, "Fault tolerant flight control techniques with application to a quadrotor uav testbed," in *Automatic Flight Control Systems - Latest Developments*, D. T. Lombaerts, Ed., no. 5. InTech, 2012, pp. 119–150.
[15] Y. Tang and R. Patton, "Phase modulation of robust variable structure control for nonlinear aircraft," in *UKACC International Conference on Control (CONTROL 2012)*, Cardiff, UK, September 2012.
[16] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an uav type-quadrotor," in *International Journal of Applied Mathematics and Computer Sciences, Vol.4, No.1*, 2007, pp. 12–17.
[17] A. Merheb and H. Noura, "Novel bio-inspired stochastic tuning of a quadrotor PD controller," in *The second annual Australian Control Conference (AUCC 2012)*, Sydney, Australia, November 2012.

APPENDIX

A. State Space Representation of the quadrotor

The state space representation of the quadrotor dynamics with $X = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}\ ]^t = \left[x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ \right]^t$ as state vector, and $Y = C.X$ as output vector is

$$\dot{X} = f(X) + g(X).u(t)$$

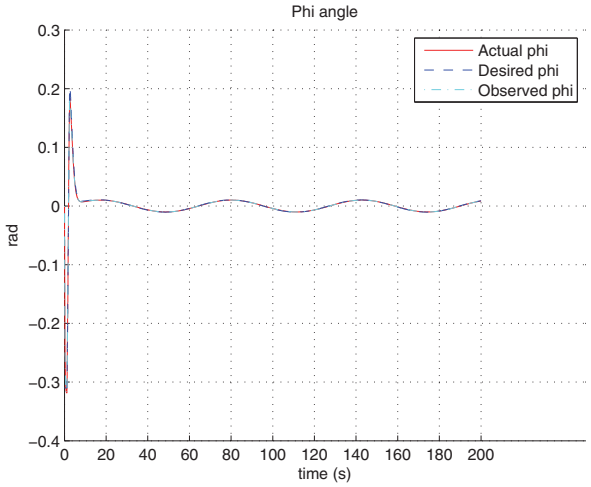
Note that $Y = [x\ 0\ y\ 0\ z\ 0\ \phi\ 0\ \theta\ 0\ \psi\ 0\ ]^t$. Matrix $f(X)$ is 12×1 matrix, matrix $g(X)$ is 12×4, and the control matrix is 4×1.

$f_1(X) = x_2$, $f_2(X) = -\frac{K_{ftx}}{m}.\dot{x}$, $f_3(X) = x_4$, $f_4(X) = -\frac{K_{fty}}{m}.\dot{y}$, $f_5(X) = x_6$, $f_6(X) = -\frac{K_{ftz}}{m}.\dot{z} - g$, $f_7(X) = x_8$, $f_8(X) = \frac{I_y - I_z}{I_x}.\dot{\theta}.\dot{\psi} + \frac{I_{rotor}}{I_x}.\dot{\theta}.\gamma - \frac{K_{fax}}{I_x}.\dot{\phi}^2$, $f_9(X) = x_{10}$, $f_{10}(X) = \frac{I_z - I_x}{I_y}.\dot{\phi}.\dot{\psi} - \frac{I_{rotor}}{I_y}.\dot{\phi}.\gamma - \frac{K_{fay}}{I_y}.\dot{\theta}^2$, $f_{11}(X) = x_{12}$, $f_{12}(X) = \frac{I_x - I_y}{I_z}.\dot{\phi}.\dot{\theta} - \frac{K_{faz}}{I_z}.\dot{\psi}^2$.

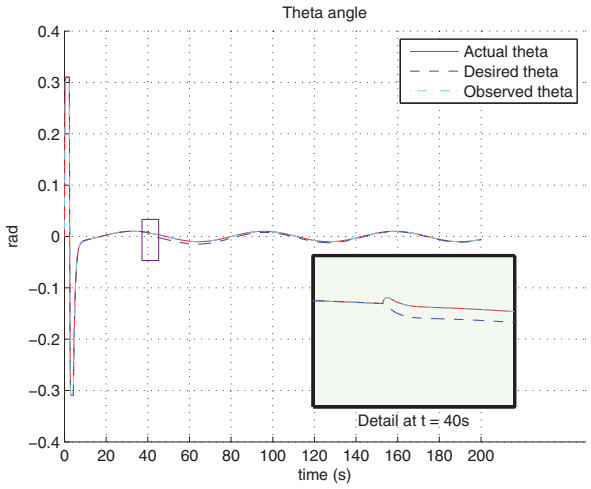
All the elements of matrix $g(X)$ are zero except $g_{2,1}(X) = \frac{(sin\psi.sin\phi + cos\psi.sin\theta.cos\phi)}{m}$, $g_{4,1}(X) = \frac{(-cos\psi.sin\phi + sin\psi.sin\theta.cos\phi)}{m}$, $g_{6,1}(X) = \frac{cos\theta.cos\phi}{m}$, $g_{8,2}(X) = \frac{l}{I_x}$, $g_{10,3}(X) = \frac{l}{I_y}$, and $g_{12,4}(X) = \frac{1}{I_z}$.

B. State Space Representation of the Sliding Mode Observer

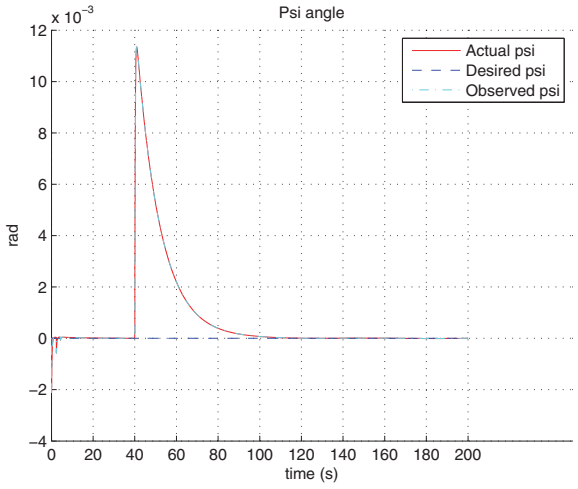
Before showing the state space form of the observer, we have to state that the sliding mode estimate vector of the fault v is 12×1 matrix with all its odd elements equal to



(a) Roll response



(b) Pitch response



(c) Yaw response

Fig. 3: Altitude response with 55% fault for the quadrotor following the continuous path.

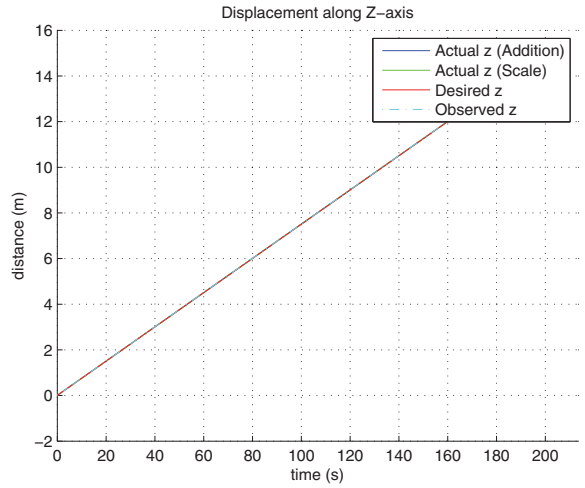


Fig. 4: Height response with 55% fault for the quadrotor following the continuous path.

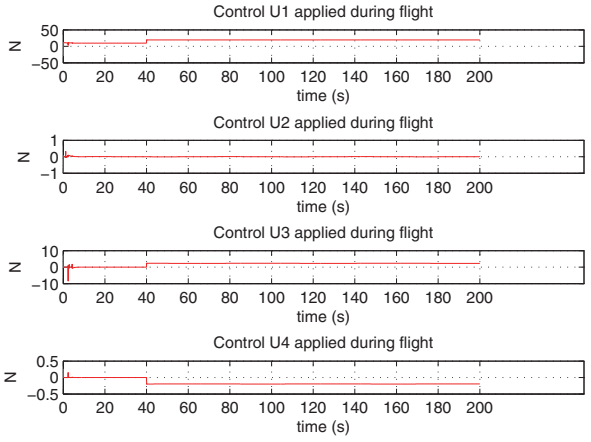


Fig. 5: Controls of the quadrotor following the continuous path.

zero. The state space form of the observer is shown in the

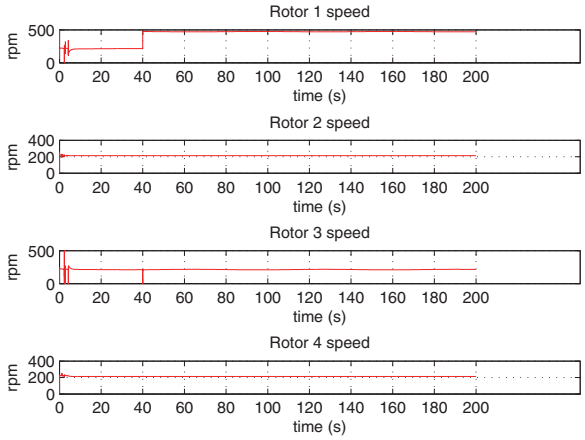


Fig. 6: Rotor speeds of the quadrotor following the continuous path.

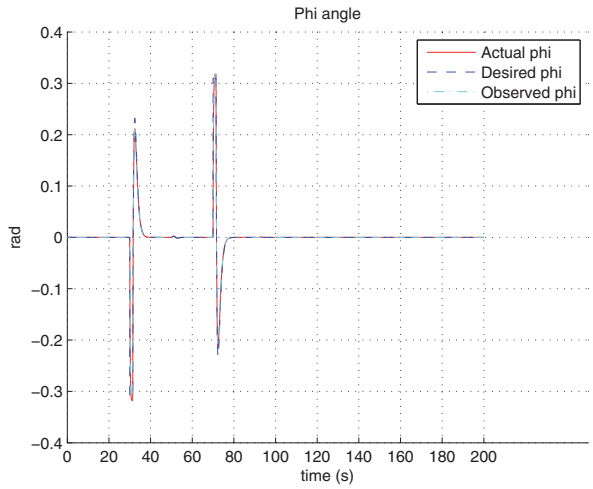
equations below

$$\dot{\tilde{x}}_1 = \tilde{x}_2 + L_1.(y_1 - \tilde{y}_1) = \tilde{x}_2 + L_1.e_{y_1}$$
$$\dot{\tilde{x}}_2 = \frac{U_1}{m}.(sin\psi.sin\phi + cos\psi.sin\theta.cos\phi) - \frac{K_{ftx}}{m}.\dot{x} + \\ + L_2.(x_2 - \tilde{x}_2) + v_{x_2}$$
$$\dot{\tilde{x}}_3 = \tilde{x}_4 + L_3.(y_3 - \tilde{y}_3) = \tilde{x}_4 + L_3.e_{y_3}$$
$$\dot{\tilde{x}}_4 = \frac{U_1}{m}.(-cos\psi.sin\phi + sin\psi.sin\theta.cos\phi) - \frac{K_{fty}}{m}.\dot{y} + \\ + L_4.(x_4 - \tilde{x}_4) + v_{x_4}$$
$$\dot{\tilde{x}}_5 = \tilde{x}_6 + L_5.(y_5 - \tilde{y}_5) = \tilde{x}_6 + L_5.e_{y_5}$$
$$\dot{\tilde{x}}_6 = \frac{U_1}{m}.cos\theta.cos\phi - \frac{K_{ftz}}{m}.\dot{z} - g + \\ + L_6.(x_6 - \tilde{x}_6) + v_{x_6}$$

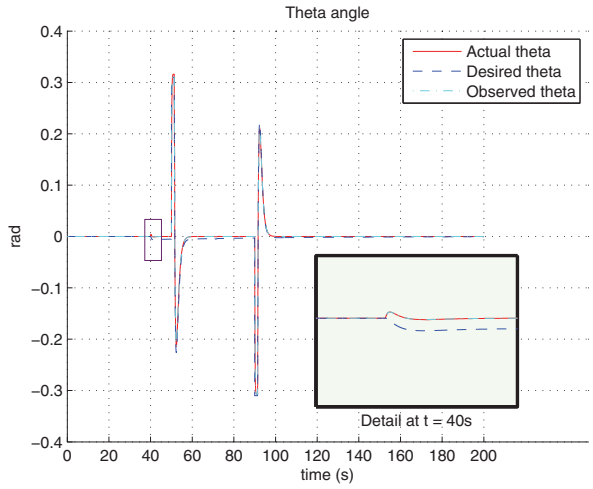
$$\dot{\tilde{x}}_7 = \tilde{x}_8 + L_7.(y_7 - \tilde{y}_7) = \tilde{x}_8 + L_7.e_{y_7}$$
$$\dot{\tilde{x}}_8 = \frac{I_y - I_z}{I_x}.\dot{\theta}.\dot{\psi} + \frac{I_{rotor}}{I_x}.\dot{\theta}.\gamma - \frac{K_{fax}}{I_x}.\dot{\phi}^2 + \frac{l.U_2}{I_x} + \\ + L_8.(x_8 - \tilde{x}_8) + v_{x_8}$$
$$\dot{\tilde{x}}_9 = \tilde{x}_{10} + L_9.(y_9 - \tilde{y}_9) = \tilde{x}_{10} + L_9.e_{y_9}$$
$$\dot{\tilde{x}}_{10} = \frac{I_z - I_x}{I_y}.\dot{\phi}.\dot{\psi} - \frac{I_{rotor}}{I_y}.\dot{\phi}.\gamma - \frac{K_{fay}}{I_y}.\dot{\theta}^2 + \frac{l.U_3}{I_y} + \\ + L_{10}.(x_{10} - \tilde{x}_{10}) + v_{x_{10}}$$
$$\dot{\tilde{x}}_{11} = \tilde{x}_{12} + L_{11}.(y_{11} - \tilde{y}_{11}) = \tilde{x}_{12} + L_{11}.e_{y_{11}}$$
$$\dot{\tilde{x}}_{12} = \frac{I_x - I_y}{I_z}.\dot{\phi}.\dot{\theta} - \frac{K_{faz}}{I_z}.\dot{\psi}^2 + \frac{U_4}{I_z} + \\ + L_{12}.(x_{12} - \tilde{x}_{12}) + v_{x_{12}}$$

C. Ecological Systems Algorithm

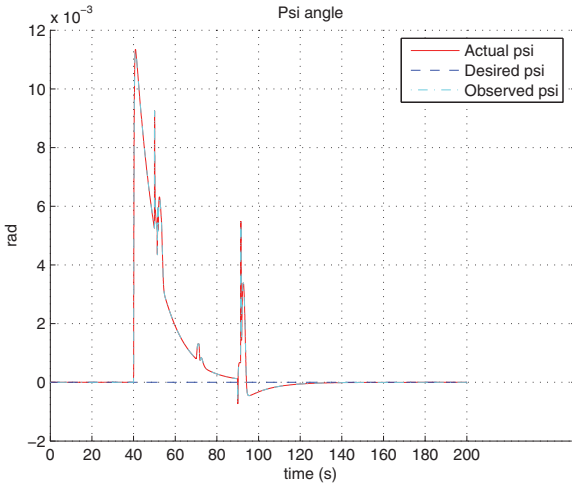
In this paper, Ecological Systems Algorithm (ESA) [17] is used to tune the quadrotor sliding mode controller (Fig.



(a) Roll response



(b) Pitch response



(c) Yaw response

Fig. 7: Altitude response with 55% fault for the quadrotor following the discontinuous path.

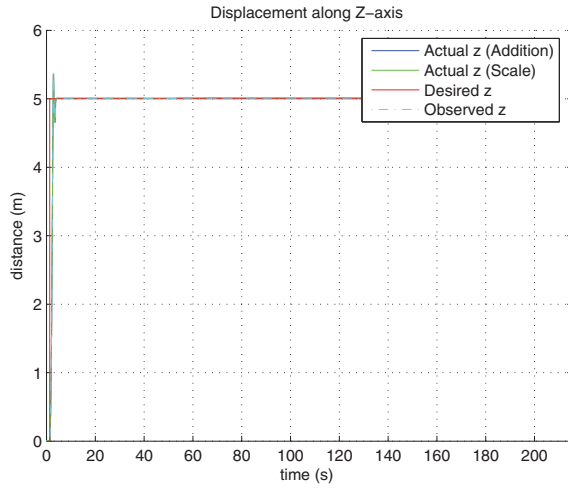


Fig. 8: Yaw and height response with 55% fault for the quadrotor following the discontinuous path.

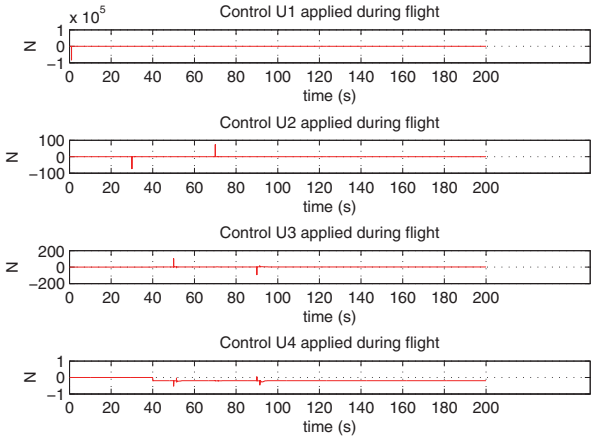


Fig. 9: Controls of the quadrotor following the discontinuous path.

12). There are two positive parameter vectors that need to be tuned: the sliding surface slope vector c, and the discontinuous control convergence vector k. These vectors contain elements for each control variable, mainly we have $c = [c_z \ c_\phi \ c_\theta \ c_\psi]$ and $k = [k_z \ k_\phi \ k_\theta \ k_\psi]$. Vectors c and k should be tuned carefully in order to ensure the reachability condition. Tuning the SMC controller is an optimization problem where the user searches for the optimal values giving the best response of controllers. Moreover, this is a black-box problem: although the quadrotor dynamic equations are well known, it is impossible to predict the effectiveness of the controller with new values until it is implemented and tested. Biologically inspired search algorithms make use of many random decisions, they also search iteratively and experimentally and the values are found as approximations to the optimal ones. This means that with each iteration, the algorithm gets closer to the

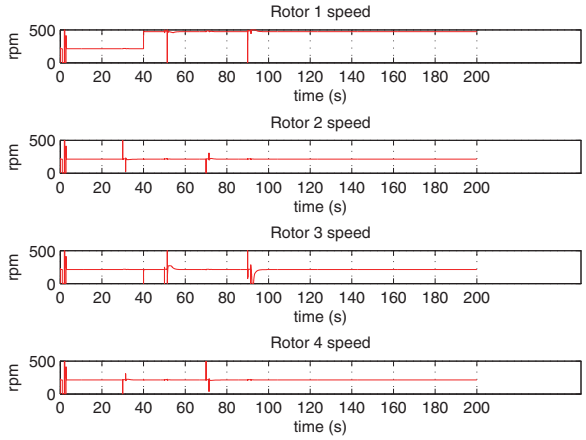


Fig. 10: Rotor speeds of the quadrotor following the discontinuous path.

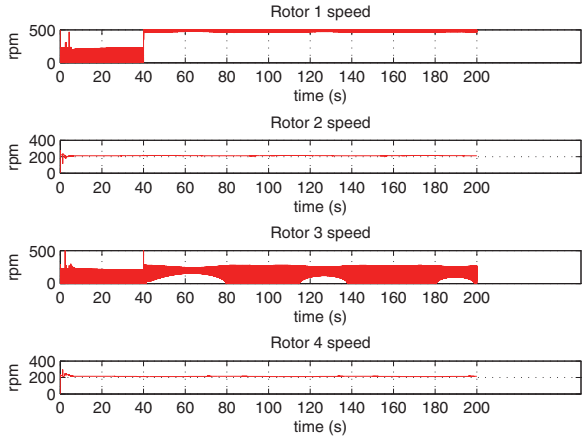


Fig. 11: Rotor speeds of the quadrotor using the scaling adaptive controller (oscillation in rotors 1 and 3).

optimal values. In natural ecological systems, a group of animal species and natural phenomena interact so that all the species benefit. A herbivores herd searching randomly for the best foraging location is conducted to this location by means of randomness, high speed, and perhaps diffusion of the herd when a carnivores group attacks it. To tune the SMC using ESA, the candidate solutions generated randomly are imitated by a herd of herbivores (zebras for example) that moves randomly searching for the best foraging place. Each element in the herd has two coordinate variables (showing its position), a health and an age variables. By randomly changing the coordinates of an individual, it moves in the environment searching for food. The concentration of food at each location is checked by using the x and y coordinates of this location as c and k values of the sliding mode controller. This controller is then used to control the quadrotor, and a fitness function using state errors is formed. If the fitness

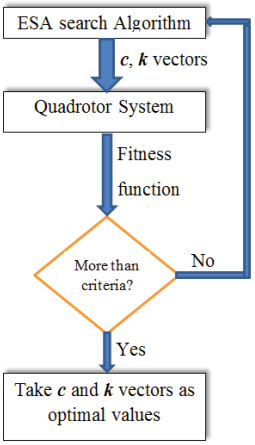
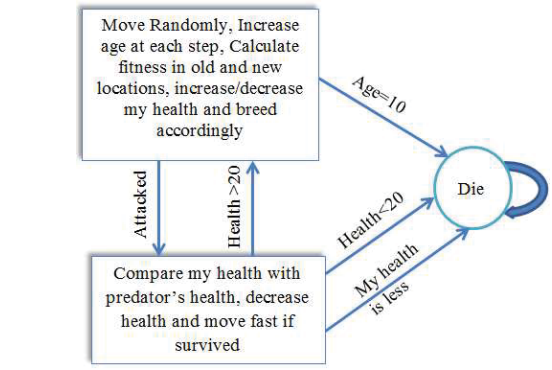
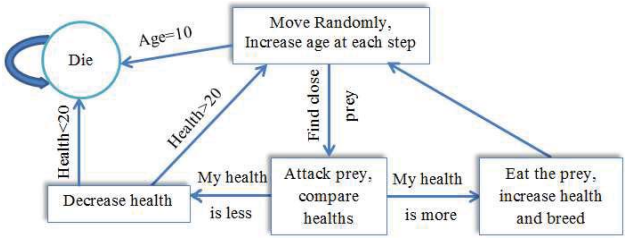


Fig. 12: Tuning the controllers using ESA.

function gives high value (i.e. c and k are close to optimal ones), it means that the individual has moved to a location with better fitness function value, so its health increases and it breeds resulting in a new "baby" individual at that location. On the other hand, if the new location has worse fitness value, the individual health decreases and no baby is delivered. If the fitness value of the individual location is higher than a given threshold, its health increases even if it comes from a better location. Whenever an individual health value drops below 10, this individual dies (Fig. 13). Moreover, each individual has a life of 10 moves; after ten moves the individual dies regardless its health value. After a while, all the population will be gathered in the highest fitness value location thus the optimal c and k vectors are found. Under some conditions, the herbivores population is trapped in local minimum where the fitness value is high but might not be the highest in the environment. Here, there is a need of an external factor that moves some herbivores individuals out of the local minimum to keep on searching for better locations. This factor is a carnivores population (a lion herd for example) added to the environment. Carnivores population individuals have the same variables a herbivores individual has; they also move randomly in the environment searching for herbivores preys (Fig. 13). Whenever a carnivores individual becomes close to a herbivores prey (within predefined distance), it attacks it and a health comparison is applied. If the herbivores individual is healthier it manages to escape the fight to a random direction but with injuries that decrease its health. The carnivores health is also decreased as a result of famine, but it follows its prey trying to attack it again. On the other hand, if the carnivores health is more, it manages to eat the herbivores individual resulting in the death of the prey, an increase in the predator health, and the birth of a new predator baby. More information on ESA and its application can be found in [17]. The fitness function used to check the effectiveness of a value group is a Mean Squared Error based fitness function. Candidate Vectors c and k values are used to control the quadrotor, the mean squared error of each variable (z, ϕ, θ and ψ) is calculated,



(a) Prey FSM



(b) Predator FSM

Fig. 13: Finite State Machines of ESA search individuals.

and the total fitness is found as follows,

$$Fitness_i = \frac{1}{\frac{\sum_{t=0}^{tf} e_{it}^2}{size(E_i)} + 0.1} \tag{45}$$

Where i is the control variable (z, ϕ, θ or ψ), and e_{it} is the deviation of the variable i from its desired value at time t. The environment borders of ϕ, θ and ψ for SMC ESA tuning are $[0.2 \;\; 10]$ for both x and y intervals, and $[1 \;\; 20]$ for x interval, and $[30 \;\; 200]$ for y interval of z environment. The step sizes of search individuals are 0.5 and 5 for respectively x and y coordinates of z environment, and 0.2 for both coordinates of the remaining variables. The initial herbivores population is 80 individuals, while the predators initial population is only 5 individuals all with initial health of 100. The minimum attack distance is 0.3 for z environment and 0.08 for ϕ, θ and ψ environments. When an agent eats, its health is increased by a factor of 3 (AugA); when it doesn't eat in an iteration, its health decreases by a factor of 0.98 (DecA), and by a factor of 0.8 (Dec) when it survives a predator attack. On the other hand, the predator health increases by a factor of 1.2 (Aug) if it eats, but decreases by 0.8 (DecP) when it doesn't manage to eat in an iteration. Each individual has a life of 10 iterations and a health threshold value 20. A fitness value of 1 is the minimum accepted value to increase the health of a herbivores individual (EpsF), and a population of 4 is the minimum accepted population for the algorithm to continue its execution.

Table III show the best c and k vector values of SMC found using ESA. ESA parameters are set by trial and error; for example by decreasing the parameter AugA below the optimal value agents loose rapidly their health and the algorithm fails to find the optimal region. On the other hand, by increasing AugA over the optimal value, agents found in bad regions survive many iterations and it takes ESA a long time to find the optimal region. The fitness function is found by studying the step response of the quadrotor using optimal c and k vectors with inputs of $z = 1m$ and $\phi = \theta = \psi = 0.3rad$ all starting at $t = 1sec$.

TABLE III: Best Vector Values Found Using ESA

	z	ϕ	θ	ψ
c	16.56	6.84	8.14	4.37
k	155.5	0.916	7.86	2.68