

## DESIGN OF PASSIVE FAULT-TOLERANT CONTROLLERS OF A QUADROTOR BASED ON SLIDING MODE THEORY

ABDEL-RAZZAK MERHEB <sup>a,b</sup>, HASSAN NOURA <sup>c,\*</sup>, FRANÇOIS BATEMAN <sup>d</sup>

<sup>a</sup>LSIS Lab

Aix-Marseille University, Domaine Universitaire de Saint-Jérôme, Batiment Polytech  
Avenue Escadrille Normandie-Niemen, 13397 Marseille, France

<sup>b</sup>Electrical & Electronics Engineering (EEE) Department

Lebanese International University, Tripoli campus, Dahr el Ain Road, Al Haykalieh, Lebanon  
e-mail: abdelrazzac10@yahoo.fr

<sup>c</sup>Electrical Engineering Department

United Arab Emirates University, P. O. Box 15551, Al Muwaiji, Al Ain, United Arab Emirates  
e-mail: hnoura@uaeu.ac.ae

<sup>d</sup>French Air Force Academy, Base aérienne 701, Chemin de Saint Jean, 13661, Salon-de-Provence, France

e-mail: francois.bateman@club-internet.fr

In this paper, sliding mode control is used to develop two passive fault tolerant controllers for an AscTec Pelican UAV quadrotor. In the first approach, a regular sliding mode controller (SMC) augmented with an integrator uses the robustness property of variable structure control to tolerate partial actuator faults. The second approach is a cascaded sliding mode controller with an inner and outer SMC loops. In this configuration, faults are tolerated in the fast inner loop controlling the velocity system. Tuning the controllers to find the optimal values of the sliding mode controller gains is made using the ecological systems algorithm (ESA), a biologically inspired stochastic search algorithm based on the natural equilibrium of animal species. The controllers are tested using SIMULINK in the presence of two different types of actuator faults, partial loss of motor power affecting all the motors at once, and partial loss of motor speed. Results of the quadrotor following a continuous path demonstrated the effectiveness of the controllers, which are able to tolerate a significant number of actuator faults despite the lack of hardware redundancy in the quadrotor system. Tuning the controller using a faulty system improves further its ability to afford more severe faults. Simulation results show that passive schemes reserve their important role in fault tolerant control and are complementary to active techniques.

**Keywords:** quadrotor UAV control, fault tolerant control, sliding mode control, cascaded SMC, bio-inspired gain tuning.

### 1. Introduction

Under-actuated systems are those with the number of control inputs less than that of their degrees of freedom. Quadrotors are an example of under-actuated systems that have six degrees of freedom and only four control inputs. Moreover, quadrotors are subjected to complicated aerodynamic situations and high disturbances; they need robust controllers to overcome all these difficulties and ensure safe flight. Recent research focuses on the design of controllers that ensure safe flight for quadrotors while

suffering actuator damage. Robustness to modeling errors, parametric uncertainties, and external disturbances is an inherent property of sliding mode controllers (SMCs).

Das *et al.* (2011) developed a cascaded controller with inner feedback linearizing the control loop and the outer SMC loop for quadrotor control. The outer one controls the position for the quadrotor while the inner control loop is responsible of stabilizing altitude and attitude values. Despite the presence of external disturbances, results based on simulation show the

\*Corresponding author

efficiency of the proposed controller. Adigbli (2007) investigates the performance of three control algorithms designed using a simplified model of a four-rotor flying robot. Simulation and experimental results show that backstepping control and sliding mode control laws both exhibited robustness in the stabilization and the setpoint tracking of a complete UAV model more than the feedback control law.

Mnasri and Gasmi (2011) presented an integral sliding mode controller for a class of uncertain systems in the presence of mismatched uncertainties, norm-bounded nonlinearities, and external disturbances. A sufficient condition for the quadratic stability of the sliding mode is established in terms of linear matrix inequalities. An adaptive fuzzy integral sliding mode control law is used to eliminate the chattering of the sliding mode and to estimate the norm bounds of uncertainties. The efficiency of the proposed controller is investigated using a sixth-order uncertain mechanical system.

Li *et al.* (2012) investigate the fault-tolerance property of sliding mode controllers. Both passive and active FTCs were tested. In passive FTC, the inherent robustness property of the SMC is used for fault tolerance. On the other hand, the active SMC switches to different pre-designed controllers whenever a fault is detected. Simulation and experimental results using the Qball-X4 system model emphasize the effectiveness and robustness of both algorithms, but still active FTC is more robust and shows better tracking performance in the presence of faults. The faults addressed are small actuator faults or propeller damage, and they are kept small in the experimental tests because of the lack of hardware redundancy in the system. A detailed comparison emphasizing the advantages, disadvantages, and limitations of each type is given.

A neural network based adaptive sliding mode controller is designed and presented by Boudjedi *et al.* (2012). This approach offers a solution to chattering and dynamical model uncertainties problems. Moreover, no prior knowledge about the dynamic model and external disturbances is required. Simulation results pointed out the efficiency of the new controller despite the presence of parametric and external disturbances.

Zhang and Chamseddine (2012) present different fault tolerant control algorithms applied to quadrotors with partial actuator failures. FTC algorithms are tested and compared based on simulation and experimental results using the Qball-X4 testbed. Sliding mode control (SMC), backstepping control (BSC) (both are of passive type), gain-scheduled PID, model reference adaptive control, and model predictive control (active type) are all used to control the quadrotor with partial loss in the control effectiveness of one or more actuators. The SMC based fault tolerant controller shows good results by compensating rapidly the damage in Propeller 4.

Hao and Yang (2013) use a matrix full-rank factorization technique to design a new sliding mode surface for a novel fault tolerant control sliding mode controller. By updating online the gain of the discontinuous part of the SMC using flexible design parameters, system faults resulting from model uncertainties and disturbances as well as input faults resulting from actuator outage, the loss of effectiveness and stuck, were all compensated. The main contribution of the new approach is that it does not require a fault detection and isolation (FDI) scheme, and that it is able to compensate some total actuator failure under a special actuator redundancy assumption.

Wu *et al.* (2008) design a fault tolerant controller for a class of uncertain systems with time-varying state-delay based on a sliding mode control scheme. The main idea of the design is to include the delay in the condition of the existence of the sliding surface. The proposed controller applied to a linear model of an aircraft exhibits good performance by tolerating parametric uncertainties, actuator faults, and external disturbances.

Tang and Patton (2012) propose the design of a robust sliding mode controller for the highly nonlinear Machan UAV. The inner loop of the controller is based on the online feedback linearization technique for linearization and decoupling of the nonlinear system into three SISO second order sub-systems. The linearized and decoupled system is then controlled using three sliding mode controllers. The SMCs are developed using a simple and effective phase modulation method that rectifies the controller structure and parameters, and guarantees robust performance and robust stability. Numerical results show that the new controller improves the system response against disturbances and chattering, showing promising fault tolerant control scheme for UAV systems.

Edwards *et al.* (2012) use the sliding mode technique to develop an observer for fault detection and fault tolerant control of aerospace systems. Both first-order and cascaded observer schemes are discussed and checked in simulation using the ADMIRE model. A fault tolerant controller using fault information from the sliding mode observer is developed based on the control allocation method. Results on the SIMONA motion flight simulator show a successful real-time implementation of the proposed fault tolerant control scheme.

Jain *et al.* (2012) use the concept of system interconnection to develop a model-free reconfiguration mechanism for fault tolerant control using a PI controllers bank. A function evaluates the closed-loop performance of the system based on the input-output trajectories information. Whenever some undesired behavior is detected, an appropriate interconnection is achieved so that the control objective is satisfied. Simulations on aircraft during the landing phase show that the real time fault tolerant control reacts successfully and switches to

the fault-case controller when the complete loss of one of the control surfaces is detected.

Xu *et al.* (2012) propose two different methods to estimate actuator faults of a class of nonlinear systems. A robust sliding mode observer is designed based on a Takagi–Sugeno (T–S) fuzzy model, where an inverse system method is used to estimate the actuator fault. The new methods presented in this paper are easy to implement compared to adaptive and other sliding mode methods. The efficiency of the proposed techniques is demonstrated using two numerical examples.

Cascaded SMCs are developed by Mahjoub *et al.* (2011) and Yu *et al.* (2010). The controllers are designed using multi-layer interlaced sliding surfaces and used respectively in the control of a 2DOF under-actuated manipulator and a bicycle robot. The robustness and effectiveness of both algorithms are shown using simulation results. Merheb *et al.* (2014) propose an active fault tolerant sliding mode control for the AscTec Pelican quadrotor. A sliding mode observer is used as a fault detection and identification unit that estimates online fault magnitudes. Two adaptive schemes based on addition and scaling are used to update the SMC of the quadrotor whenever a fault is detected. Simulation based experiments show promising results of the proposed controller.

In this paper, two passive fault tolerant controllers are developed for the AscTec Pelican quadrotor suffering partial loss of effectiveness in its motors. Stability analysis of the controllers is provided, and the controllers are upgraded with an integral term to compensate for the small remaining error. The controllers are then tuned using the faulty quadrotor system to extend their fault tolerant capabilities. Finally, a comparison is made between the passive FTC SMCs developed here and the active FTC SMC of Merheb *et al.* (2014). The comparison shows that despite passive fault tolerant schemes are static and the correction is not made according to the fault magnitude, they can be matched (under specific conditions) to active fault tolerant schemes if the design is unconventional (cascaded SMC) or the tuning is unconventional (tuning under a fault). The work done in this paper differs from that of Das *et al.* (2011) in that both controllers control the altitude and attitude values of the quadrotor, while a simple PD controller is used to control the position of the UAV. Unlike in the work of Das *et al.* (2011), here both the loops of the cascaded SMC tackle faults and better fault tolerance is achieved. The cascaded controller designed in this paper also differs from those by Mahjoub *et al.* (2011) and Yu *et al.* (2010) in that four control inputs are generated for the system, and the same control input is not used for all the states. Moreover, instead of using multiple layer interlaced sliding surfaces, two sliding manifolds are designed separately for the position and velocity control (respectively outer and inner

loops).

The rest of the paper is organized as follows, Section 2 presents the quadrotor dynamic equations and illustrates the design of regular and cascaded sliding mode controllers. Section 3 explains the tuning process of the controllers and provides a detailed explanation of the ecological systems algorithm. Section 4 provides the stability analysis of the controllers for a scalar case where the stability of each state is analyzed individually. Section 4 gives information on the experiments conducted to check the capabilities of the controllers. Equations of the path followed and equations of the position PD controller along with the gains used are given, and fault types injected are defined. Moreover, tuning the controllers using a faulty system to extend their fault tolerant capabilities is discussed in that section. Results of different experiments are also provided there. Finally, Section 5 provides concluding ideas and hints for further future improvements.

## 2. Quadrotor model and sliding mode controllers design

**2.1. Quadrotor model.** Quadrotors are under-actuated rotor-crafts that use four propellers rotating in an opposite pairs direction configuration to move in space (Fig. 1). The dynamic model of a quadrotor is found by applying Newton's second law on the forces and torques the quadrotor is subjected to. A typical quadrotor model is represented by the following equations (Bouadi *et al.*, 2007):

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

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

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

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

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

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

where  $K_{ftx} = K_{f ty} = 5.5670e^{-4}$  N/m/s, and  $K_{ftz} = 6.3540e^{-4}$  N/m/s are the translation drag coefficients, and  $K_{f ax} = K_{f ay} = 5.5670e^{-4}$  N/rad/s, and  $K_{f az} = 6.3540e^{-4}$  N/rad/s are the friction aerodynamic coefficients.  $U_i$ ,  $i = 1, 2, 3, 4$  are respectively the position, roll, pitch, and yaw controls, and  $\gamma$  is a

disturbance due to rotor speed imbalance (Bouabdallah, 2007). Here  $x$ ,  $y$ , and  $z$  are the coordinates of the quadrotor in the base frame; the roll, pitch, and yaw angles of the quadrotor are respectively  $\phi$ ,  $\theta$ , and  $\psi$ .  $I_{x,y,z}$  are the quadrotor moments of inertia with respect to the corresponding axes,  $m$  is the mass of the quadrotor,  $g$  is the gravitational constant,  $b$  and  $d$  are the thrust and drag factors of the propellers, and  $l$  is the distance between the center of the quadrotor and the center of each propeller. The quadrotor parameters are shown in Table 1.

By taking  $X = [z, \phi, \theta, \psi]^T$  as the state vector and  $u(t) = [U_1 \ U_2 \ U_3 \ U_4]^T$  as the control input, the dynamics of the quadrotor can easily be expressed in the state space form,

$$\ddot{X} = f(X) + g(X)u(t) \tag{7}$$

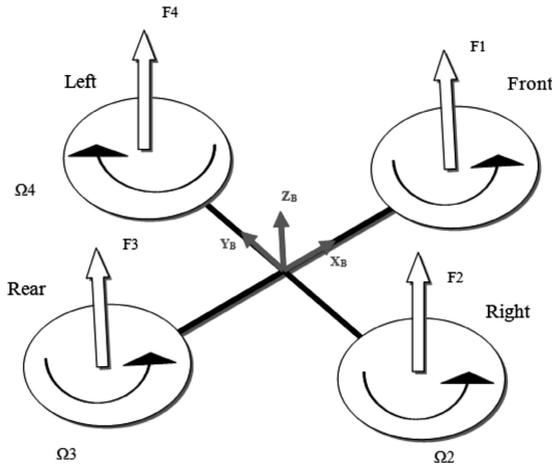


Fig. 1. Quadrotor schematic and the AscTec Pelican quadrotor at UAE University.

Table 1. Variables used in quadrotor modeling.

Variable	Value	Variable	Value
$I_x$	$8.1e^{-3} \text{ N m s}^2$	$b$	$54.2e^{-6} \text{ N s}^2$
$I_y$	$8.1e^{-3} \text{ N m s}^2$	$l$	$0.24 \text{ m}$
$I_z$	$14.2e^{-3} \text{ N m s}^2$	$d$	$1.1e^{-6} \text{ m s}^2$
$I_{\text{rotor}}$	$104e^{-6} \text{ N m s}^2$	$g$	$9.8 \text{ m s}^{-2}$
$m$	$1 \text{ kg}$		

where

$$f(X) = \text{diag}([f_1(X), f_2(X), f_3(X), f_4(X)])$$

with

$$\begin{aligned} f_1(X) &= -g, \\ f_2(X) &= \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{I_{\text{rotor}}}{I_x} \dot{\theta} \dot{\gamma}, \\ f_3(X) &= \frac{I_z - I_x}{I_y} \dot{\phi} \dot{\psi} - \frac{I_{\text{rotor}}}{I_y} \dot{\phi} \dot{\gamma}, \\ f_4(X) &= \frac{I_x - I_y}{I_z} \dot{\phi} \dot{\theta}, \end{aligned}$$

$$g(X) = \text{diag} \left( \left[ \frac{\cos(\phi) \cos(\theta)}{m}, \frac{l}{I_x}, \frac{l}{I_y}, \frac{1}{I_y} \right] \right).$$

The controllers in this paper are designed according to the AscTec Pelican quadrotor testbed available at UAE University, and values taken from the real quadrotor are used to set some bounds and restrictions on the design. The velocities of the motors are measured using a tachometer and found to be between 1200 rpm and 5000 rpm (125–523 rad/sec); these values are used to bound the calculated angular velocities needed to find  $\gamma$ . The maximum thrust of the quadrotor is given by the constructor as 15.7 N (used to bound  $U_1$  control), and  $\phi$ ,  $\theta$ , and  $\psi$  angles are within the  $-18^\circ$  and  $+18^\circ$  interval ( $-0.31$  to  $+0.31$  rad). The maximum motor voltages are 12 V. The relation between the controls and the speeds of the motors is

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ -b & 0 & b & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \tag{8}$$

The faults used in this paper are partial loss of motor power affecting all the motors instantly, and partial loss of effectiveness of one motor. The faults affect rotor speeds resulting in a sudden decrease in the speed of the faulty motor(s). The faulty control vector can then be found as

$$u_f(t) = F_a u(t), \tag{9}$$

where  $F_a$  is a  $4 \times 4$  diagonal matrix containing the fault magnitude between 0 and 1 (1: no fault, 0.5: 50% faults, 0: 100% faults).

**2.2. Regular SMC design.** In sliding mode control, the intention is to drive the state trajectory of the nonlinear system to a pre-specified linear manifold and to maintain the state trajectory on this manifold. The design of a sliding control law then requires two steps: the design of a sliding manifold such that the system presents the desirable behavior when trapped on, and the design of a switching control law that drives the plant state to the manifold and maintains it on the surface for all the future time. The SMC law will have then two components, the equivalent and the discontinuous control law,

$$u = u_{eq} + u_{dis}. \tag{10}$$

From a control point of view, good behavior of the system can be interpreted by a decrease in the state error. In other words, if the state error is changing in its negative direction, we say that the error between the desired and actual states is disappearing. If  $e = X_d - X$  is the state error vector, with  $X$  and  $X_d$  respectively the actual and desired state vectors, then good system behavior is defined by  $\dot{e} = -ce$ . The sliding manifold is then chosen as  $s = \dot{e} + ce$ , and when the system states are on this manifold we have  $s = 0$ . The matrix  $c$  is the manifold slope which affects the conversion speed of the state error to zero, defined for the quadrotor as  $c = \text{diag}([c_z, c_\phi, c_\theta, c_\psi])$ . To find the equivalent controller responsible for maintaining the system on the manifold,  $V = (1/2)s^2$  is chosen as a Lyapunov function, and its derivative  $\dot{V} = s\dot{s} = s(\dot{e} + c\dot{e}) = s(\ddot{X}_d - \ddot{X} + c\dot{e})$  is calculated. By setting  $\dot{V}$  equal to zero, the equivalent control is found to be

$$u_{eq} = g^{-1}(X) [\ddot{X}_d + c\dot{e} - f(X)]. \tag{11}$$

Since the system states may start at any position in the plane, there is a need for a controller that brings the states to the manifold. This controller pulls the states up if they are located below the sliding manifold, and pushes them down if they are above it. A suitable controller could be formed using the signum function, or the saturation function if we intend to avoid discontinuities,

$$u_{dis} = -ksat(s), \tag{12}$$

where  $k = \text{diag}([k_z, k_\phi, k_\theta, k_\psi])$  is a positive gain affecting the conversion speed of the discontinuous control, and the saturation function is defined as

$$sat(s) = \begin{cases} 1, & \text{if } s > 1, \\ s, & \text{if } |s| \leq 1, \\ -1, & \text{if } s < -1. \end{cases} \tag{13}$$

The sliding mode controller is then

$$u(t) = g^{-1}(X) [\ddot{X}_d + c\dot{e} - f(X)] - ksat(s) \tag{14}$$

Tests on the proposed controller show that the responses of the  $\theta$  and  $\psi$  angles exhibit a small remaining error (less than 5% with 55% faults) when one or more motors are affected by faults. This remaining error rises only in faulty cases, and no error is seen when the quadrotor motors work normally. To compensate for the remaining error, the SMC is augmented with an integrator

$$u(t) = g^{-1}(X) [\ddot{X}_d + c\dot{e} - f(X)] - ksat(s) + k_i \int e(\tau) d\tau, \tag{15}$$

where  $k_i = \text{diag}([k_{i_z}, k_{i_\phi}, k_{i_\theta}, k_{i_\psi}])$  is the integrator gain matrix. Note that the term  $g^{-1}(X)$  always exists for the control variables. For  $\phi$ ,  $\theta$ , and  $\psi$ ,  $g^{-1}(X)$  is respectively  $I_x/l$ ,  $I_y/l$ , and  $I_z$ , which always have defined values. For the altitude variable

$$g^{-1}(X) = \frac{m}{\cos \theta \cos \phi},$$

which also exists since both  $\phi$  and  $\theta$  are within the  $-18^\circ$  and  $+18^\circ$  interval ( $-0.31$  to  $+0.31$  rad) as given by the manufacturer. Note that, by assuming that the desired values change slowly, the acceleration of the state vector  $\ddot{X}_d$  is taken as zero.

**2.3. Cascaded SMC design.** A cascaded SMC consists of two SMC based control loops: an outer loop controlling the control variables  $z$ ,  $\phi$ ,  $\theta$ , and  $\psi$ , and an inner loop controlling the velocities. As an example, the cascaded SMC of the quadrotor height is shown in Fig. 2.

The design of a cascaded SMC starts by choosing the sliding manifold of the inner loop which has velocities as the input,

$$s_{(inner)} = e_v + c_{(inner)} \int e_v(\tau) d\tau, \tag{16}$$

$e_v$  being the velocity error vector or  $e_v = \dot{e} = \dot{X}_d - \dot{X} = u_{(outer)} - \dot{X}$ , and  $c_{(inner)}$  representing a positive design diagonal matrix. Similar to the design of a regular SMC, the equivalent control is found by defining a Lyapunov function and verifying the reachability condition,

$$u_{eq(inner)} = g^{-1}(X) [\ddot{X}_d + c_{(inner)}e_v - f(X)]. \tag{17}$$

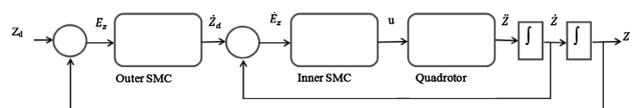


Fig. 2. Cascaded SMC control of the height.

The discontinuous control of the inner loop is designed using the saturation function,

$$u_{\text{dis(inner)}} = -k_{\text{(inner)}}\text{Sat}(s_{\text{(inner)}}). \quad (18)$$

The control input of the quadrotor is then  $u = u_{\text{eq(inner)}} + u_{\text{dis(inner)}}$ . This control input contains the term  $e_v = u_{\text{(outer)}} - \dot{X}$  that needs to be designed in the outer loop using the sliding mode technique. Similarly to the inner loop controller design, the outer loop controller design requires the design of a sliding manifold,

$$s_{\text{(outer)}} = e + c_{\text{(outer)}} \int e(\tau) d\tau, \quad (19)$$

where  $e = X_d - X$ , and  $c_{\text{(outer)}}$  is a diagonal matrix with positive elements. The Lyapunov function is chosen similar to the inner loop case, and the equivalent control is found as

$$u_{\text{eq(outer)}} = \dot{X}_d \text{ eq} = \dot{X} - c_{\text{(outer)}}e \quad (20)$$

while the discontinuous outer control is chosen as

$$u_{\text{dis(outer)}} = \dot{X}_d \text{ dis} = -k_{\text{(outer)}}\text{Sat}(s_{\text{(outer)}}). \quad (21)$$

The control input of the outer loop is then

$$u_{\text{(outer)}} = \dot{X} - c_{\text{(outer)}}e - k_{\text{(outer)}}\text{Sat}(s_{\text{(outer)}}). \quad (22)$$

Finally, the control input of the quadrotor is

$$u(t) = g^{-1}(X) \left[ \ddot{X}_d + c_{\text{(inner)}}e_v - f(X) \right] - k_{\text{(inner)}}\text{Sat}(e_v + c_{\text{(inner)}} \int e_v(\tau) d\tau) \quad (23)$$

with  $e_v = u_{\text{(outer)}} - \dot{X}$ . Here also the acceleration of the state vector  $\ddot{X}_d$  is taken to be zero. Note that these equations contain the velocity matrix  $\dot{X}$  which might be difficult to obtain practically in the presence of noise and disturbance. Either filters or other mathematical techniques could be used to estimate derivatives. The reader is directed to the works of Fliess and Join (2013), Fliess et al. (2008), and Mboup et al. (2009) for more information on the differentiation of noisy signals.

### 3. Tuning the controllers

Sliding mode controller design requires a good choice of design parameters to ensure the reachability condition which is defined by  $\dot{V} = s\dot{s} \leq -\eta|s|$ , where  $\eta$  is a strictly positive real constant determining the convergence speed of the trajectory to the sliding surface. Both the sliding surface slope matrix  $c$  and the discontinuous control convergence matrix  $k$  should be chosen carefully so that the reachability condition holds. Classic design approaches solve the reachability condition equations

ending with constraints on the  $c$  and  $k$  values, and then matrices are chosen randomly while respecting these constraints. Matrices chosen using this method ensure the stability of the controller but might not be the “best” values that have the minimum conversion time. This method requires good mathematical skills and careful solution.

A new trend in engineering is to use artificial intelligence based algorithms to solve mathematical problems. These problems vary from finding optimal PID controller gains of an industrial system to finding the shortest path of a navigating robot. In our case, we can use a search algorithm to find the optimal values of the matrices  $c$  and  $k$  of regular and cascaded SMCs. We choose a novel biologically inspired algorithm called the ecological systems algorithm (ESA) (Merheb and Noura, 2012) that imitates the equilibrium in nature to tune our controllers. The algorithm, as shown in Fig. 3, suggests random values for the matrices  $c$  and  $k$  and checks whether these values are optimal (produce a small response error). The matrices are then updated using ecological rules and checked again. The last step is repeated iteratively until matrices giving the least error values are found. It is important to emphasize that it is easy to solve the reachability condition for the regular SMC case by hand. This means the reachability condition of the cascaded SMC is an equation with four variable matrices ( $c$  and  $k$  for inner and outer loops), with each matrix having four elements corresponding to the four controlled variables ( $z$ ,  $\phi$ ,  $\theta$ , and  $\psi$ ). This means that the number of reachability condition equations is 16 in the cascaded SMC case, which makes it hard to solve using conventional methods.

**3.1. Ecological systems algorithm.** To explain how the ESA finds the optimal values, we will use two insect species: the aphid species (agent) which feed on plant liquid, and ladybug species (predator) which eat aphids. For simplicity, only two species are considered in the ESA; in nature, however, thousands of species interact, forming an ecological system. Aphids move randomly in the environment searching for places with a high nutrient density. If the nutrient in a place is high, all the individuals of an aphid swarm exhibit an increase in their health. In consequence, every individual gives birth to a new “baby” in the swarm. On the other hand, if the nutrient is scarce, all the individuals suffer from malnutrition, all have their health decreased, and some die if their health drops below the vital threshold. Note that every individual in the swarm has a limited life time or a predefined number of iterations to perform. When a search individual dies, other search individuals along with the new born “babies” continue their search until optimal values are found or all the individuals die. If we put an aphid group with such properties in an environment, we realize that after some

time all the aphids are gathered in the high nutrient density place as healthy individuals give birth to new generations in the plenty regions, and individuals in the poor regions die.

Using only aphid species is enough to find an optimal foraging region, so why using the ladybug species? Most of the search algorithms suffer from local minimum problems where good results are found but the optimal values are missed. In our algorithm, the aphids might have found a place with very high nutrient density but did not find the best location. In such situations, there is a need for an external factor that drives the aphids out of the “good” place to a better one. Ladybugs move randomly in the environment searching for aphids to eat and attack any close prey. When a ladybug attacks an aphid individual, if the ladybug is healthier, it manages to eat its prey, its health increases and it gives birth to a ladybug baby. On the other hand, if the aphid is healthier, it manages to escape rapidly in a random direction but with injuries, which results in the decrease of its own health. The ladybug individual that is not able to forage in an iteration suffers from malnutrition and has its health decreased. With the ladybug species introduced, we notice that the algorithm becomes faster, and its chance to find “the” optimal place is higher. Note that the introduction of the ladybug species is not an ultimate solution for the local minimum problem. Actions taken by aphid and ladybug individuals are shown in Figs. 4 and 5.

**3.2. Tuning the controllers.** To tune the regular SMC of the quadrotor using ESA, we use one search space with aphid (agent) and ladybug (predator) groups for

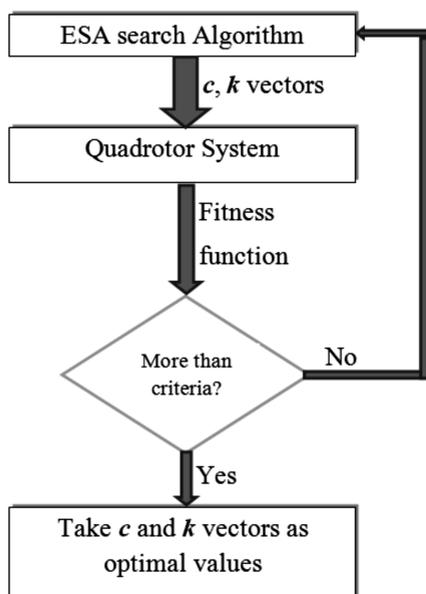


Fig. 3. Tuning the controllers using the ESA.

each control variable ( $z$ ,  $\phi$ ,  $\theta$ , and  $\psi$ ). Figure 6 shows the search space of the height controller with aphid and ladybug individuals. Aphid individuals move in the search space by changing their coordinates randomly according to a predefined step size. These coordinates are taken respectively as the  $c_z$  and  $k_z$  gains. To measure the efficiency of the gain matrices, they are used to control the  $z$ ,  $\phi$ ,  $\theta$ , and  $\psi$  variables of the quadrotor with step function inputs as shown in Fig. 6. Errors between the actual and desired variable values are calculated and used to validate a mean squared error based fitness function (Eqn. (25)).

The more this fitness function gives high value, the

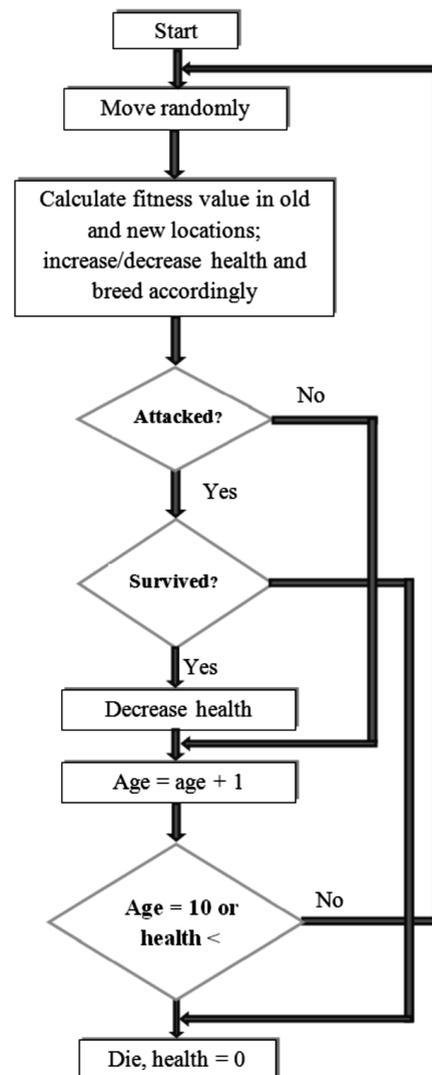


Fig. 4. Aphid individual actions.

more efficient the new  $c$  and  $k$  are:

$$Fitness_i = \frac{1}{MSE_i + 0.1} \quad (24)$$

with

$$MSE_i = \frac{1}{size(E_i)} \sum_{t=0}^{t_f} e_{it}^2, \quad (25)$$

where  $i$  is the control variable ( $z, \phi, \theta,$  or  $\psi$ ), and  $e_{it}$  is the deviation of the variable  $i$  from its desired value at time  $t$ . To tune the regular SMC, four agent groups search four different spaces simultaneously, and the  $c$  and

$k$  matrices each with four elements are loaded and their fitness values are checked. This iteration continues until the optimal matrices  $c$  and  $k$  (with a minimum error) are found. For the cascaded SMC, we have two matrices  $c$  and  $k$ , one for each control loop (inner and outer). This means that the algorithm searches for 16 different values at once. During the tuning process, the controller is used to control the quadrotor performing a motion defined by a step function of 1 m for the height, and step functions of 0.3 rad for the three remaining variables  $\phi, \theta,$  and  $\psi$ . The position error is then calculated and used to find the fitness function.

To fasten the tuning of the cascaded SMC, we first tune it only for attitude variable gains. Errors of only angle variables are used to construct the fitness function. Once the best angle controller gains are found, they are used in the tuning process of the cascaded height controller. This method reduces the tuning time of the cascaded controller remarkably. Table 2 shows the best  $c$  and  $k$  matrix element values for both regular and cascaded SMCs.

The tuning of the regular SMC is accomplished using the unfaulty quadrotor system without the integrator part of the controller. The integrator gain matrix  $k_i$  used to compensate for the remaining error which appears only in faulty cases is found by a trial-and-error procedure while testing the control law in the presence of faults. The integrator gain matrix has all its elements equal to 2. Note that tuning the controllers is made offline, and in some scarce cases the algorithm search individuals extinct and the search process fails to find the appropriate gains. The search process is then repeated to find the optimal

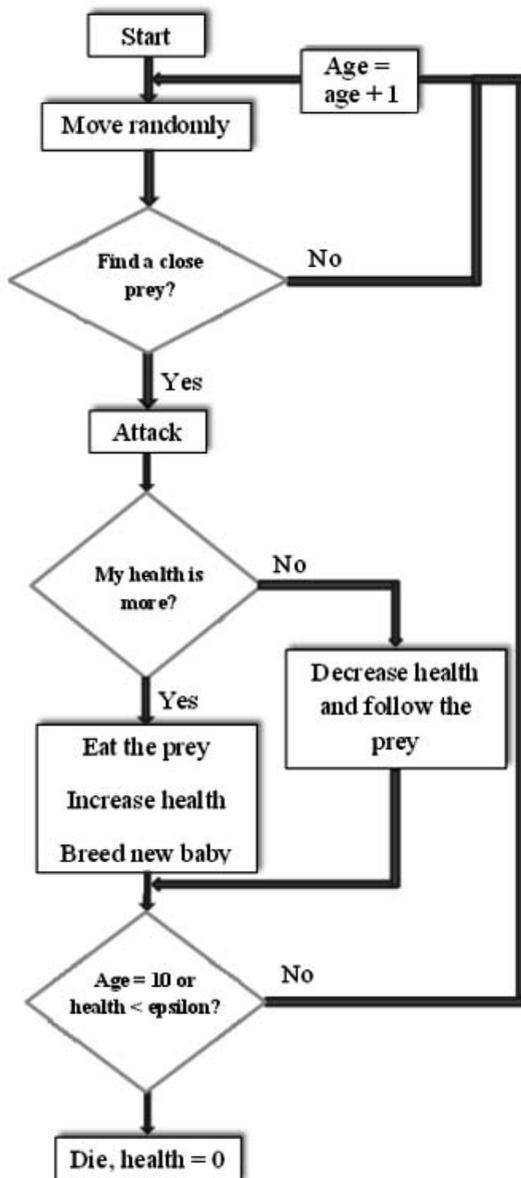


Fig. 5. Ladybug individual actions.

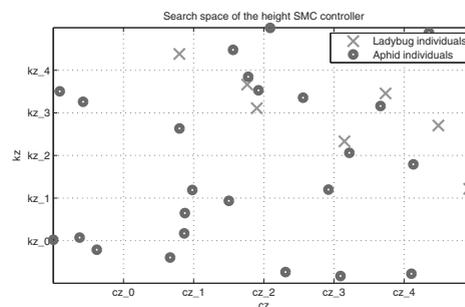


Fig. 6. Search space of the height SMC.

Table 2. Best matrix element values found using the ESA.

	$z$	$\phi$	$\theta$	$\psi$
$c$	16.56	6.84	8.14	4.37
$k$	155.5	0.916	7.86	2.68
$c_{(inner)}$	16	10	11	7
$k_{(inner)}$	15	13	13	10
$c_{(outer)}$	26	7	1	3
$k_{(outer)}$	4	4	4	1

gains. Tuning the controllers using the ESA simplifies and fastens the design since the ESA takes between 12 to 55 seconds to converge using a 2.4 GHz dual core computer with 3 GB RAM. Moreover, conventional design strategies give only an interval to choose the gains and not the optimal gain values. On the other hand, ESA gives the best gain values found within the search interval.

**3.3. ESA parameters.** All the search spaces are bounded, and any individual choosing a step exceeding the boundaries is directed towards the space by negating the last step direction. The initial agent and predator population, their initial health values and step sizes, and the minimum distance that a predator can detect its prey are all set before starting the search. When an agent individual forages, its health increases by a factor of AugA (3 in regular SMC tuning, 25 in cascaded SMC tuning), it decreases by DecA (0.98 in regular, 0.999 in cascaded) when it does not eat, and by a factor Dec (0.8 in regular, 0.99 in cascaded) when it suffers from injuries after a predator attack. Similarly, the predator health increases by factor Aug (1.2 in regular and cascaded) if it forages, and decreases by DecP (0.8) when it does not manage to find a prey. Each individual has a life of 10 iterations and a health threshold value equal to 20. When the health of an individual falls below the health threshold value, it dies immediately. A fitness value of EpsF (0.01) is the minimum accepted value to increase the health of an aphid individual, and a population of 4 is the minimum accepted aphid population for the algorithm to continue its execution.

Tables 3, 4, and 5 show respectively the parameters used in the ESA to tune the regular and the cascaded SMCs of the quadrotor. The parameters of the algorithm are chosen carefully by a trial-and-error procedure. A very high value of the AugA factor, for example, causes slow elimination of the individuals found in bad regions, resulting in low algorithm conversion speed. Contrarily, if this factor is kept low, all the agent individuals will die before the algorithm converges.

Table 3. Variables used in the ESA to tune regular SMC variables.

Variable	Value
$\phi, \theta, \psi$ intervals	$x \in [1 \ 20], y \in [30 \ 200]$
$z$ intervals	$x \in [1 \ 20], y \in [30 \ 200]$
Angles step sizes	$x : 0.5, y : 5$
$z$ step size	$x : 0.2, y : 0.2$
Prey population	80
Angles attack distance	0.05
Predator population	5
Initial health	20

#### 4. Stability test of the proposed controllers

In this section, the stability of the proposed controllers is checked for scalar case (i.e., each state is checked individually).

**Proposition 1.** *The regular sliding mode controller designed in (14) is stable if the discontinuous controller constants are chosen according to the following conditions:*

$$k_\phi \geq 0.562, \tag{26}$$

$$k_\theta \geq 0.562, \tag{27}$$

$$k_\psi > 0 \text{ and } k_z > 0. \tag{28}$$

*Proof.* Proving the proposition above requires examining the sliding condition using the Lyapunov method. Let  $V = (1/2)s^2$  be a positive definite Lyapunov function. To satisfy the sliding condition, elements of diagonal matrix  $k$  have to be chosen so that  $\dot{V}$  is negative semi definite. Since

$$\dot{V} = s\dot{s} = s(\ddot{e} + c\dot{e}) = s(\ddot{X}_d - \ddot{X} + c\dot{e}),$$

by replacing  $\ddot{X}$  with the state space equations, we get

$$\dot{V} = s(\ddot{X}_d - f(X) - g(X)u + c\dot{e}).$$

We replace the control with its equivalent and discontinuous parts so that the derivative of the Lyapunov function becomes  $\dot{V} = s(-kg(X)\text{sign}(s)) = -kg(X)|s|$ . With  $k$  being a positive matrix,

Table 4. Variables used in the ESA to tune cascaded SMC angles variables.

Variable	Value
Prey population	20
Attack distance	0.3
Step size	0.2
Predator population	2
$\phi_{(\text{inner})}$ intervals	$x \in [1 \ 20], y \in [1 \ 20]$
$\phi_{(\text{outer})}$ intervals	$x \in [1 \ 10], y \in [1 \ 10]$
$\theta_{(\text{inner})}$ intervals	$x \in [1 \ 20], y \in [1 \ 20]$
$\theta_{(\text{outer})}$ intervals	$x \in [1 \ 5], y \in [1 \ 5]$
$\psi_{(\text{inner})}$ intervals	$x \in [1 \ 10], y \in [1 \ 15]$
$\psi_{(\text{outer})}$ intervals	$x \in [1 \ 5], y \in [1 \ 5]$

Table 5. Variables used in the ESA to tune cascaded SMC height variables.

Variable	Value
Prey population	100
Attack distance	0.1
Predator population	5
Step size	0.5
$z_{(\text{inner})}$ intervals	$x \in [1 \ 30], y \in [1 \ 70]$
$z_{(\text{outer})}$ intervals	$x \in [1 \ 30], y \in [1 \ 5]$

$\dot{V} = -kg(X)|s|$  is negative only if  $g(X)$  is a positive function. This is true for

$$\begin{aligned} g_\phi(X) &= \frac{l}{I_x}, \\ g_\theta(X) &= \frac{l}{I_x}, \\ g_\psi(X) &= \frac{1}{I_x} \end{aligned}$$

since  $g_{\text{angle}}(X) = \text{diag}([g_\phi(X), g_\theta(X), g_\psi(X)])$ . This means that the angle controllers are stable, but what about the height controller?

For the height,

$$g_z(X) = \frac{\cos(\phi)\cos(\theta)}{m},$$

which can take positive and negative values since  $-1 \leq \cos \leq 1$ . The manufacturer of the AscTec Pelican quadrotor, however, limits the values of its angles  $\phi$ ,  $\theta$ , and  $\psi$  by  $-18^\circ$  and  $+18^\circ$  interval ( $-0.31$  to  $+0.31$  rad). This constraint always makes  $g_z(X)$  positive, which means that the sliding condition is satisfied if all the elements of matrix  $k$  are chosen positive.

Because it is impossible to measure the speeds of the quadrotor motors during flight, the term  $\gamma$  (Eqns. (4) to (6)) cannot be performed in the controller equations. We introduce the term  $\tilde{f}(X)$  which is identical to  $f(X)$  with no  $\gamma$  multiplier term, which will replace  $f(X)$  in the controller equations. The derivative of the Lyapunov function becomes

$$\begin{aligned} \dot{V} &= s(f(X) - \tilde{f}(X) - kg(X)\text{sign}(s)) \\ &= s(f(X) - \tilde{f}(X)) - ksg(X)\text{sign}(s) \quad (29) \\ &= s(f(X) - \tilde{f}(X)) - kg(X)|s|. \end{aligned}$$

If we can find a positive function  $F(X)$  with  $|f(X) - \tilde{f}(X)| \leq F(X)$ , then we can ensure the stability of the quadrotor by choosing

$$k \geq \frac{F(X) + \eta}{g(X)},$$

which leads to  $sF(X) - kg(X)|s| \leq -\eta|s|$ .  $F(X)$  is found by subtracting  $f(X)$  and  $\tilde{f}(X)$ , so all its elements are zero except

$$\begin{aligned} F_\phi(X) &= \frac{I_{\text{rotor}}}{I_x} \dot{\phi} \gamma \\ F_\theta(X) &= -\frac{I_{\text{rotor}}}{I_y} \dot{\theta} \gamma. \end{aligned}$$

The minimum accepted value of  $k$  is found by replacing its formula ( $k \geq (F(X) + \eta)/g(X)$ ) with the maximum value of  $F(X)$  ( $g(X)$  is constant). We have

$\gamma = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$  so its maximum value is  $\gamma_{\text{max}} = 2\Omega_{\text{max}} = 1046$  rad/sec. The minimum value of  $k_\phi$  is then

$$k_\phi \geq \frac{\frac{I_{\text{rotor}}}{I_x} \gamma_{\text{max}} \dot{\phi}}{\frac{l}{I_x}} = \frac{I_{\text{rotor}} \gamma_{\text{max}} \dot{\phi}}{l} = 0.4532 \dot{\phi}.$$

If we take the maximum possible angle change, we can say that no angle of the quadrotor will change from its minimum value to its maximum ( $-0.31$  rad to  $+0.31$  rad) in half a second. This means that the maximum possible rate of change of angles is  $\dot{\phi} = 0.62/0.5 = 1.24$  rad/sec;  $k_\phi$  is then  $k_\phi \geq 0.562$ , which is identical to what will be found for  $k_\theta$ . Note that, because  $F_z(X)$  and  $F_\psi(X)$  are zero, it is sufficient to choose  $k_z$  and  $k_\psi$  positive to ensure the stability of the controller as shown previously. ■

For the cascaded SMC, the stability test of the inner loop results in similar constraints as that of the regular controller. For the outer loop, the stability test follows the same steps as in the regular SMC.

**Proposition 2.** *The cascaded sliding mode controller is stable if the constants of the outer discontinuous controller are chosen positive.*

*Proof.* We have

$$\begin{aligned} \dot{V} &= s_{(\text{outer})} s_{(\text{outer})} = s_{(\text{outer})} (\dot{e} + c_{(\text{outer})} e) \\ &= s_{(\text{outer})} (\dot{X}_d - \dot{X} + c_{(\text{outer})} e). \end{aligned}$$

By replacing  $\dot{X}_d$  with its equivalent and discontinuous values, we end up with  $s_{(\text{outer})} s_{(\text{outer})} = s_{(\text{outer})} (-k_{(\text{outer})} \text{sign}(s_{(\text{outer})})) = -k_{(\text{outer})} |s_{(\text{outer})}|$  which is negative definite if  $k_{(\text{outer})}$  is chosen positive. The values of the matrices  $c$  and  $k$  shown in Table 2 ensure the stability of the regular and cascaded controllers because they are within the constraints calculated in this section. Moreover, the  $k$  values found by the ESA not only ensure the stability of the controller, but also are the optimal values giving the minimum possible error. ■

To study the stability of the controllers under a fault, the faulty control vector in (9) is used in the stability test. Following the procedure above with  $u_f(t) = F_a u(t)$ , the derivative of the Lyapunov function is found as  $\dot{V} = s(\ddot{X}_d - f(X) - g(X)(uF_a) + c\dot{e})$ .

Using the faulty SMC law and ensuring the reachability condition, the constraint on  $k$  to guarantee the controller stability under a fault is found as

$$k \geq \frac{s(1 - F_a)(\ddot{X}_d - f(X) + c\dot{e}) + \eta}{g(X)F_a|s|}. \quad (30)$$

Finding the constraint on  $k$  requires finding a maximum value of the function  $\ddot{X}_d - f(X) + c\dot{e}$  and use it

in Eqn. (30). Seeking simplicity and space, the fault range that the controllers can afford will be given in the following section, rather than in the solution of the equation above. A detailed study of controllers' stability under a fault will be interpreted in future works. Figures 7 and 8 show the phase-plane of the faulty quadrotor system controlled with both types of SMCs. The open discs in the portrait show the first 50 points, and the cross sign is the final point in the portrait. When faults are introduced, the system phase portrait no longer converges to the origin but to a close point in its vicinity (0.08 is the greatest value). This is still acceptable since faults introduce performance degradation to the system. The phase portrait shows that the system error is not increasing; it is not zero either but fixed at a constant value. This effect will be seen clearly in the quadrotor response in the next section.

## 5. Experiments and results

**5.1. Experiments.** The SIMULINK environment is used to check the fault tolerant performance of the designed controllers. The quadrotor is set to follow a continuous helix in the space and faults are injected to motors at different instants. The equation of the helix path is

$$x_d = 10 \cos(0.1t) + 10, \quad (31)$$

$$y_d = 10 \sin(0.1t) + 10, \quad (32)$$

$$z_d = 0.1t. \quad (33)$$

Both the sliding mode controllers designed in this paper are used to control the attitude and altitude variables ( $z$ ,  $\phi$ ,  $\theta$ , and  $\psi$ ) of the quadrotor. To make the quadrotor follow the desired path, a simple PD controller is used to calculate the position error and give the essential corrections. The output of the PD controller is then used to generate the desired attitude values, as shown in the equations associated with Eqn. (36):

$$\phi_d = \text{atan} \left( \frac{-\ddot{y}}{\sqrt{(\ddot{x}^2 + (\ddot{z} + g)^2)}} \right), \quad (34)$$

$$\theta_d = \text{atan} \left( \frac{\ddot{x}}{\ddot{z} + g} \right), \quad (35)$$

$$\psi_d = 0. \quad (36)$$

Here  $\ddot{x}$ ,  $\ddot{y}$ , and  $\ddot{z}$  are the outputs of the simple PD position controller

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

where  $i$  is  $x$ ,  $y$ , or  $z$ , and  $k_{ip} = 2$  and  $k_{id} = 3$  are respectively the proportional and derivative gains chosen by trial and error. It is important to emphasize that the height output of the PD controller  $\ddot{z}$  is used only to generate the desired angles and not to control the height of the quadrotor, which is done using the proposed controllers.

**5.2. Tuning controllers under a fault.** In the previous sections, regular and cascaded controllers are tuned in the fault-free situation. These controllers can handle good amount of fault as will be shown in the following section.

A natural question is as follows: Is it possible to improve the fault tolerance property of the controllers by tuning them with a faulty system? Can we embed a number of faults in the design of the controllers and have them more robust? To answer this question, the tuning process is repeated using a quadrotor with one faulty motor. The regular SMC is tuned with 10%, 20%, 30%, and 40% of faults injected at the start to the first motor. It was recognized that the controller tuned with a system with 30% and 40% faults exhibits the best robustness. The cascaded SMC was re-tuned under a quadrotor having 30% faults in its first motor. The best  $c$  and  $k$  matrix element values for the controllers tuned with faulty systems are shown in Table 6. Note that the ESA constants used in the new tuning process are identical with their values in the fault-free case (Tables 3–5). Finally, the fault used during tuning process is partial loss of speed of Motor 1.

**5.3. Results.** Once the system is made ready, we start testing its robustness against actuator faults. Two partial losses of efficiency faults are considered in this paper: the first one is partial loss of speed for one or two motors, and the second is partial loss in voltage supply affecting all the motors at once. Faults are injected at different instants while the quadrotor is set to follow the helical path. Faults are injected by scaling down the faulty motor speed by the error percentage in the case of partial speed losses, and scaling down the supply voltage of all motors in the case of a voltage supply fault. The motor speed drops down when the load torque increases because of rust or debris, or when the voltage across that motor changes as the internal resistance of the motor decreases. Any failure in the power supply or power unit will affect all the motors, resulting in a sudden voltage drop across all motors.

We first consider the regular SMC controlled quadrotor performing the helical path in the space. The regular SMC controlled system is able to handle a

Table 6. Matrix element values with controllers tuned under a fault.

Variable	$z$	$\phi$	$\theta$	$\psi$
$c$	6.2161	14.3027	7.8565	2.2437
$k$	177.3498	2.1076	9.9953	1.3583
$c_{(\text{inner})}$	17.1833	12.3679	10.1933	5.9480
$k_{(\text{inner})}$	1.1050	11.0206	11.5236	8.0855
$c_{(\text{outer})}$	24.4676	14.9194	2.9162	2.7811
$k_{(\text{outer})}$	1.5966	1.0052	2.5191	3.0196

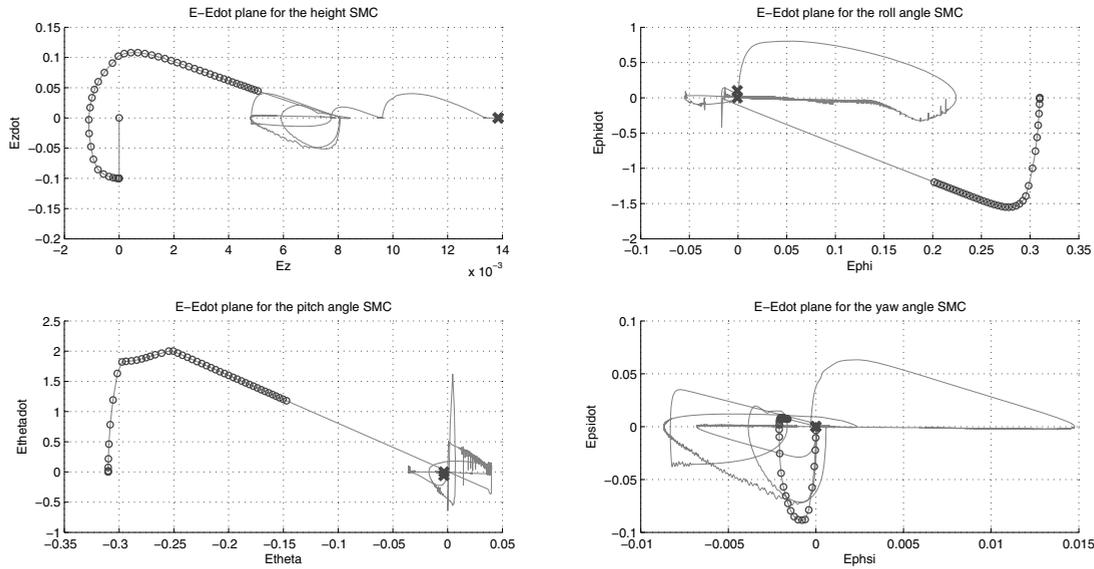


Fig. 7. Phase plane of a regular SMC controlled faulty quadrotor (two faults of 55% and 54%).

maximum fault of 55% in Motor 1 occurring at instant  $t = 40$  s, along with a fault of 40% in Motor 2 at  $t = 20$  s. This percentage rises remarkably when the controller is tuned with a faulty system. The faults becomes 57% and 56% for Motor 1 and Motor 2 respectively. The response of the quadrotor controlled with the regular SMC tuned under a fault is illustrated in Fig. 11. The quadrotor exhibits a small deviation from the desired path when the faults occurs. This deviation can be seen clearly in the angles of the quadrotor but is compensated directly by the controllers. Figure 9 shows respectively the motor speeds (not used in the control law) applied during flight for regular SMC tuned without faults. It can be clearly seen that motor speeds are realistic and stay within the physical limits of the quadrotor. One can recognize a sudden change in motor speeds and controls as the quadrotor compensates for the suddenly occurring faults. Moreover, the figures show that fault compensation does not occur directly at the fault injection instance ( $t = 20$  s and  $t = 40$  s) but after a short delay. This delay results in small deviation in the path remedied directly when the controls change.

Using the cascaded SMC tuned without a fault improves fault tolerance. The quadrotor can now handle up to 55% faults in Motor 1 occurring at instant  $t = 40$  s, along with 54% faults in Motor 2 at  $t = 20$  s is recognized even though no integral controller is used. Moreover, no deviation in the quadrotor path is recognized. This reveals the importance of the cascaded SMC where faults are compensated in the fast inner loop controlling the velocity system before they affect the quadrotor position. This analysis is supported by the figure showing the motor

speeds of the quadrotor (cf. Fig. 10). This figure shows that changes in motor speeds happen at the time the faults are injected ( $t = 20$  s and  $t = 40$  s), which means a fast compensation of any fault. As in the regular SMC, if the cascaded SMC is tuned under a fault, the system ability to handle faults rises. The controller can now handle up to 56% faults in both Motor 1 and Motor 2 occurring at  $t = 40$  s and  $t = 20$  s, respectively. Here also fault compensation is made quickly with no deviation in the path, as shown in Fig. 12. Both regular and cascaded controllers can handle a sudden decrease of 55% in all the motor speeds at  $t = 40$  s.

By comparing results found here with those of active fault tolerant sliding mode controllers of Merheb *et al.* (2014), one can recognize easily that passive FTC schemes still hold an important place in fault tolerant control. Active FTC of Merheb *et al.* (2014) outperforms the passive FTC regular SMC of Merheb *et al.* (2013) when one motor fault is considered. The maximum tolerated amount of the loss of effectiveness is 35% for passive and 55% for Active FTC. When passive FTC is upgraded with an integrator, its performance is improved and the amount of faults tolerated rises to 55%. Tuning passive controllers with a faulty system pushes their fault tolerant capacity to the limits. The passive FTC exceeds the active FTC scheme and tolerates up to 57% faults in one motor.

Note that the location of a fault in the tuning process might affect controllers robustness. Controllers can tolerate better the fault that was injected during their tuning. In our experiment, controllers tuned under a fault tolerate a considerable number of faults in Motor 1

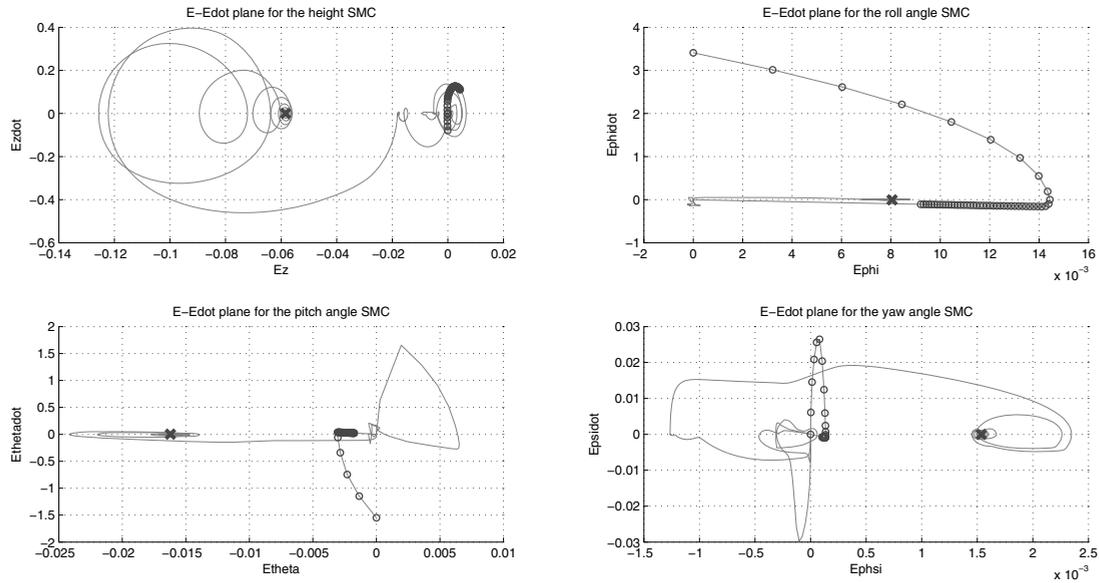


Fig. 8. Phase plane of a cascaded SMC controlled faulty quadrotor (two faults of 55% and 54%).

because they were tuned with this fault. The controller response would be very different if the fault were in Motor 4, for example. To solve this issue and improve fault tolerance of controllers, an active fault tolerant controller can be designed based on the passive FTC controllers proposed in this paper. Controllers are tuned under different fault locations and a bank of controllers is formed. A fault detection and identification (FDI) unit is used to detect any fault online. Whenever a fault is detected, the appropriate controller is chosen from the bank of controllers based on the fault location and its magnitude.

### 6. Conclusion

In this paper, we proposed two different sliding mode control (SMC) based passive fault tolerant controllers for the AscTec pelican quadrotor. The first one is an integrator augmented regular sliding mode controller designed and tuned in a fault free-situation. The second one is a cascaded sliding mode controller with an inner-loop SMC responsible for fast compensation of faults in the velocity loop, and an outer SMC loop controlling the height and attitude angles. The controllers are passive-type SMCs where no fault measurement or estimation is applied, and the robustness inherent in sliding mode control is used to compensate the faults.

Controller design is made easy by using the ESA, a bio-inspired stochastic search algorithm applied offline in fault-free and faulty situations to find the optimal values of design parameters of both the controllers. Stability analysis of the proposed controllers in a scalar and a

fault free case is provided. SIMULINK results show good fault tolerant performance of the controllers despite the under-actuated system used. With controllers tuned without faults, the cascaded SMC outperforms the regular SMC by compensating rapidly a higher fault percentage. Tuning the controllers using a faulty quadrotor system is seen to improve the fault tolerant capabilities of both SMCs, and the regular SMC tuned with fault matches the cascaded SMC tuned without a fault.

Using bio-inspired search algorithms is proved to be very important in SMC design: instead of choosing the discontinuous gain theoretically and arbitrarily following the reachability condition, the algorithm chooses values and tests them practically until an “optimal” value is found. Comparison between the passive and active FTC SMCs of Merheb *et al.* (2014) emphasizes the importance of passive fault tolerant controllers that compete with active schemes when unconventional techniques are used.

Future work should include the application of the proposed controllers to a real quadrotor testbed, a detailed study of controllers stability under faults, and the design of an active fault tolerant SMC. Results shown in this paper are believed to present the maximum fault value a passive fault tolerant control SMC can handle. To tolerate higher faults and severe actuator failures, over-actuated UAVs such as octocopters should be used.

### Acknowledgment

This research is supported by United Arab Emirates University, Al-Ain, UAE, research project no. 31N161.

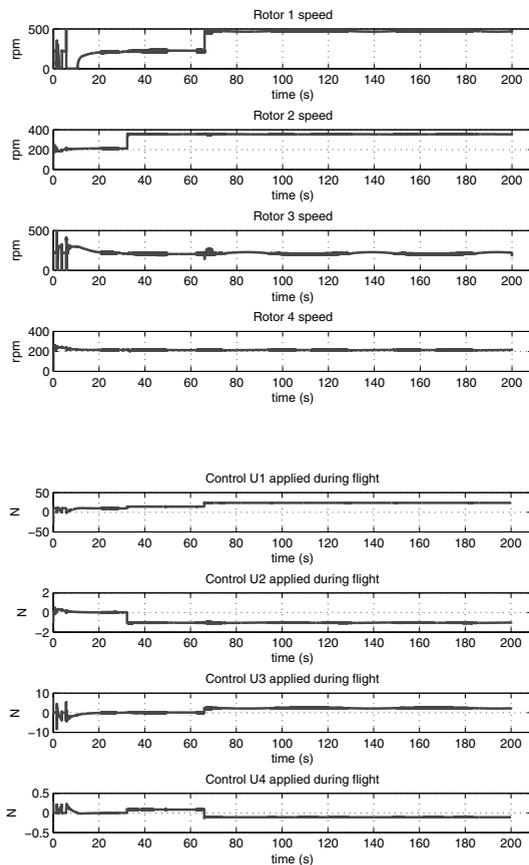


Fig. 9. Control signals and rotor speeds of regular SMC experiments (tuned with a fault).

## References

- Adigbli, P. (2007). Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques, *3rd US/European Competition and Workshop on Micro Air Vehicle Systems (MAV07)/European Micro Air Vehicle Conference and Flight Competition (EMAV2007), Toulouse, France*, pp. 1–9.
- Bouabdallah, S. (2007). *Design and Control of Quadrotors with Application to Autonomous Flying*, Ph.D. thesis, Ecole Polytechnique Federale De Lausanne, Lausanne.
- Bouadi, H., Bouchoucha, M. and Tadjine, M. (2007). Sliding mode control based on backstepping approach for an UAV type-quadrotor, *International Journal of Applied Mathematics and Computer Sciences* **4**(1): 12–17.
- Boudjedi, H., Yacef, F., Bouhali, O. and Rizoug, N. (2012). Dual neural network for adaptive sliding mode control of quadrotor helicopter stabilization, *International Journal of Information Sciences and Techniques* **2**(4): 1–14.
- Das, A., Lewis, F.L. and Subbarao, K. (2011). Sliding mode approach to control quadrotor using dynamic inversion, in A. Bartoszewicz (Ed.), *Challenges and Paradigms in Applied Robust Control*, InTech, Rijeka.
- Edwards, C., Alwi, H. and Tan, C.P. (2012). Sliding mode methods for fault detection and fault tolerant control with application to aerospace systems, *International Journal of Applied Mathematics and Computer Science* **22**(1): 109–124, DOI: 10.2478/v10006-012-0008-7.
- Fliess, M. and Join, C. (2013). Model free control, *International Journal of Control* **86**(12): 2228–2252.
- Fliess, M., Join, C. and Sira-Ramirez, H. (2008). Non-linear estimation is easy, *International Journal of Modelling Identification and Control* **4**(1): 12–27.
- Hao, L.-Y. and Yang, G. (2013). Robust fault tolerant control based on sliding mode method for uncertain linear systems with quantization, *ISA Transactions* **52**(5): 600–610.
- Jain, T., Yamé, J.J. and Sauter, D. (2012). Model-free reconfiguration mechanism for fault tolerance, *International Journal of Applied Mathematics and Computer Science* **22**(1): 125–137, DOI: 10.2478/v10006-012-0009-6.
- Li, T., Zhang, Y. and Gordon, B.W. (2012). 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* **227**(1): 12–23.

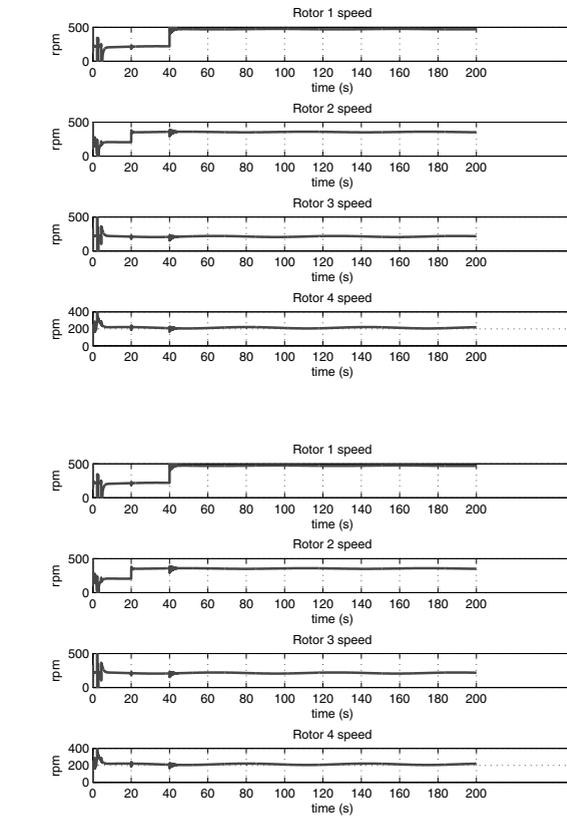


Fig. 10. Control signals and rotor speeds of cascaded SMC experiments (tuned with a fault).

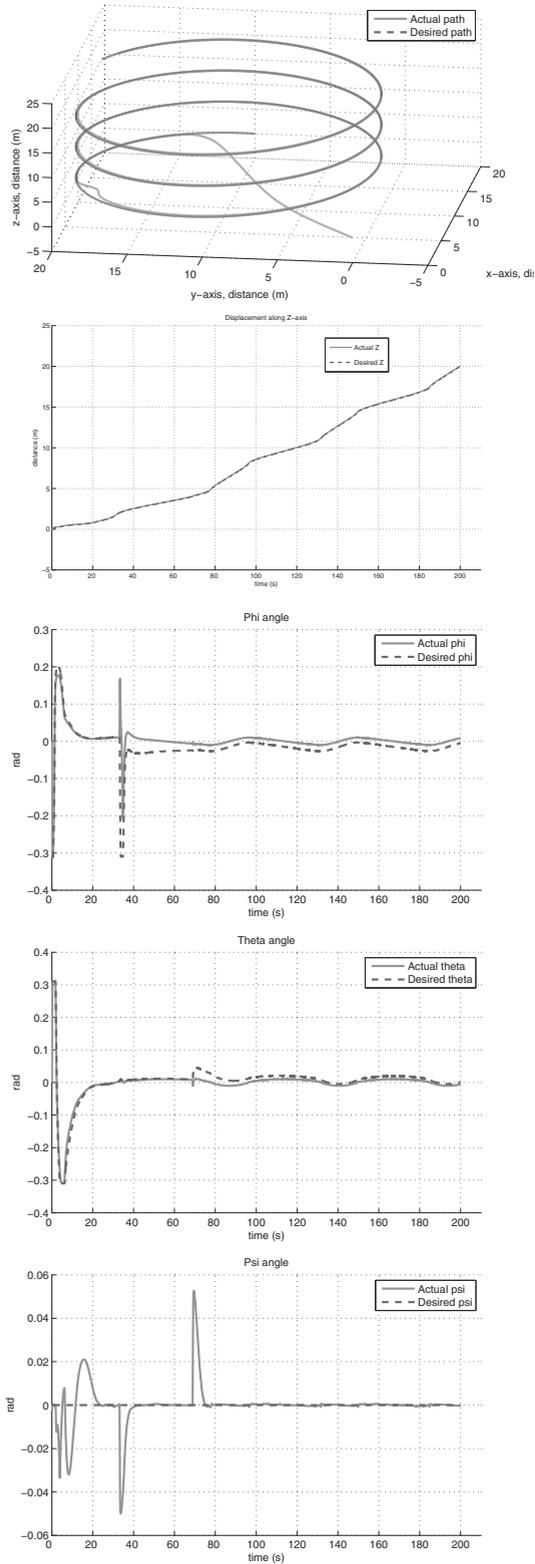


Fig. 11. Response of a regular SMC controlled quadrotor tuned under a fault (two faults: of 57% at 40 s and 56% at 20 s).

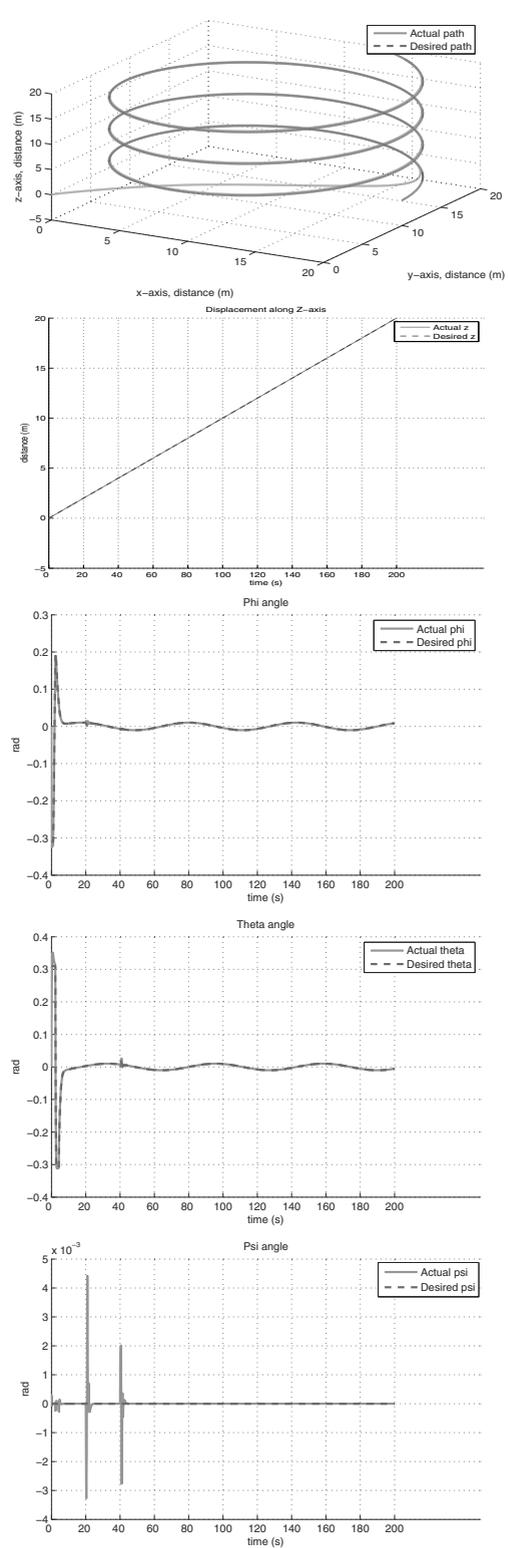


Fig. 12. Response of a quadrotor controlled with a cascaded SMC tuned under a fault (two faults: of 56% at 40 s and 56% at 20 s).

- Mahjoub, S., Mnif, F. and Derbel, N. (2011). Set point stabilization of a 2DOF underactuated manipulator, *Journal of Computers* **6**(2): 368–376.
- Mboup, M., Fliess, M. and Join, C. (2009). Numerical differentiation with annihilators in noisy environment, *Numerical Algorithms* **50**(4): 439–467.
- Merheb, A. and Noura, H. (2012). Novel bio-inspired stochastic tuning of a quadrotor PD controller, *2nd Annual Australian Control Conference (AUCC 2012), Sydney, Australia*, pp. 227–232.
- Merheb, A., Noura, H. and Bateman, F. (2013). Passive fault tolerant control of quadrotor uav using regular and cascaded sliding mode control, *2nd International Conference on Control and Fault-Tolerant Systems (SysTol'13), Nice, France*, pp. 330–335.
- Merheb, A., Noura, H. and Bateman, F. (2014). Active fault tolerant control of quadrotor uav using sliding mode control, *2014 International Conference on Unmanned Aircraft Systems (ICUAS14), Orlando, FL, USA*, pp. 156–166.
- Mnasri, C. and Gasmi, M. (2011). LMI-based adaptive fuzzy integral sliding mode control of mismatched uncertain systems, *International Journal of Applied Mathematics and Computer Science* **21**(4): 605–615, DOI: 10.2478/v10006-011-0047-5.
- Tang, Y. and Patton, R. (2012). Phase modulation of robust variable structure control for nonlinear aircraft, *UKACC International Conference on Control (CONTROL 2012), Cardiff, UK*.
- Wu, J., Weng, Z., Tian, Z. and Shi, S. (2008). Fault tolerant control for uncertain time-delay systems based on sliding mode control, *Kybernetika* **44**(5): 617–632.
- Xu, D., Jiang, B. and Shi, P. (2012). Nonlinear actuator fault estimation observer: An inverse system approach via a T-S fuzzy model, *International Journal of Applied Mathematics and Computer Science* **22**(1): 183–196, DOI: 10.2478/v10006-012-0014-9.
- Yu, X., Wei, S. and Guo, L. (2010). Cascade sliding mode control for bicycle robot, *International Conference on Artificial Intelligence and Computational Intelligence, Sanya, China*, pp. 62–66.
- Zhang, Y. and Chamseddine, A. (2012). Fault tolerant flight control techniques with application to a quadrotor UAV testbed, in D.T. Lombaerts (Ed.), *Automatic Flight Control Systems—Latest Developments*, InTech, Rijeka, pp. 119–150.



**Abdel-Razzak Merheb** received his Master's degree in electrical and electronics engineering/control and robotics from the TOBB University of Economics and Technology, Ankara, Turkey. He is now a Ph.D. student at Aix-Marseille University, a member of the LSIS Lab, France, and a lab instructor at Lebanese International University, Lebanon.



**Hassan Noura** is a professor at United Arab Emirates University, Department of Electrical Engineering (since 2007). He is the chair of the Electrical Engineering Department (since 2011). His research fields of interest are mainly control systems, fault diagnosis, prognostics, and fault tolerant control with application to a variety of industrial systems: steel industry, helicopters, overhead cranes, unmanned aerial vehicles, renewable energy systems, etc. He received his Master's and Ph.D. degrees in automatic control from University Henri Poincaré, Nancy 1 (now Lorraine University), France, in 1990 and 1993. He was an associate professor at the same university from 1994 to 2003. In September 2003, he joined Paul Cézanne University, Aix-Marseille III (now Aix-Marseille University), France, as a professor of control systems. He has supervised several graduate and Ph.D. students. He has participated in and led research projects in collaboration with industry in the fields of fault diagnosis and fault-tolerant control. He has authored and co-authored one book, one book chapter, and over 140 journal and conference papers.



**François Bateman** received his M.Sc. in electrical engineering in 1992 from Ecole Normale Supérieure de Cachan, France, and the Ph.D from Aix-Marseille University, Marseille, France, in 2008. He teaches in the French Air Force Academy of Salon de Provence and conducts his research at Aix-Marseille University. Bateman's research interests are in fault diagnosis and fault accommodation and their application to aircraft and helicopters.

Received: 21 April 2014

Revised: 17 October 2014

Re-revised: 14 February 2015