Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 4, 601–617
DOI: 10.34768/amcs-2022-0042

INTEGRATED FAULT–TOLERANT CONTROL OF A QUADCOPTER UAV WITH INCIPIENT ACTUATOR FAULTS

PAULIN KANTUE [a,*], JIMOH O. PEDRO [b]

[a]Research and Development Group
Uav4africa (Pty) Ltd
1 Cadogan Rd, Johannesburg, 2090, South Africa
e-mail: pkantue@gmail.com

[b]School of Mechanical, Industrial and Aeronautical Engineering
University of the Witwatersrand
1 Jan Smuts Ave, Johannesburg, WITS2050, South Africa
e-mail: jimoh.pedro@wits.ac.za

An integrated approach to the fault-tolerant control (FTC) of a quadcopter unmanned aerial vehicle (UAV) with incipient actuator faults is presented. The framework is comprised of a radial basis function neural network (RBFNN) fault detection and diagnosis (FDD) module and a reconfigurable flight controller (RFC) based on the extremum seeking control approach. The dynamics of a quadcopter subject to incipient actuator faults are estimated using a nonlinear identification method comprising a continuous forward algorithm (CFA) and a modified golden section search (GSS) one. A time-difference-of-arrival (TDOA) method and the post-fault system estimates are used within the FDD module to compute the fault location and fault magnitude. The impact of bi-directional uncertainty and FDD detection time on the overall FTC performance and system recovery is assessed by simulating a quadcopter UAV during a trajectory tracking mission and is found to be robust against incipient actuator faults during straight and level flight and tight turns.

Keywords: fault-tolerant control, quadrocopter, incipient actuator fault, radial basis function neural network.

1. Introduction

The development of reliable control systems, also known as fault-tolerant control systems (FTCSs), comprises the design and implementation of three separate components: the controller module, the fault detection and diagnosis (FDD) module and the controller reconfiguration module. Over the last two decades, the search for a design methodology to ensure that the synthesis of an FTCS achieves the desired performance, has been an important research area (Nett *et al*., 1988; Stoustrup *et al*., 1997; Fan *et al*., 2013; Liu *et al*., 2015; Liu and Yang, 2019; Salazar *et al*., 2020). The choice of the FDD method in relation to the fault type, the design of the RFC mechanism in relation to fault detection uncertainties and system rate of degradation, are some of the integration issues to be considered in order for an FTCS to be successful in a real-time environment.

A two-stage Kalman filter was used for simultaneous state and fault parameter estimation by Zhang and Jiang (1999; 2001). A multi-stage reconfiguration process was adopted to ensure that the estimation uncertainty of the actuator effectiveness did not degrade the controller reconfiguration process. The approach also reduced fault-induced transients and improves the post-fault system recovery. An eigenstructure assignment controller was used for the reconfiguration once faulty actuators were detected and diagnosed by quantifying control effectiveness factors. A nominal proportional-integral (PI) controller for a longitudinal vertical takeoff and landing (VTOL) aircraft model was augmented with a reconfigurable gain matrix computed such that the closed-loop eigenvalues of the estimated system resulted in a short transient response and small steady-state error. The effect of convergence rate and estimation accuracy was investigated by Wu *et al*. (2015). A fault detection and diagnosis scheme for an unmanned helicopter was

[*] Corresponding author

developed based on an extended set-membership filter. This method enabled simultaneous state and parameter estimation, which can further be used for controller reconfiguration.

The four-parameter problem solution was used to design an H_∞-based robust integrated controller by Marcos and Balas (2005). For feedback control and the diagnosis of faulty actuators and sensors, weight functions were specified. The optimisation routine, based on linear time-invariant (LTI) models obtained at various flight conditions, was used for a trade-off between controller robustness and fault diagnosis performance within a single problem formulation. A similar approach for the synthesis of an H_∞ output feedback controller using an unknown input observer was also investigated by Liu *et al*. (2012). An adaptive control law was implemented in an H_∞ sense without requiring the convergence of the fault parameters. This integrated approach was tested for a helicopter dynamic model with a random loss of the actuator effectiveness.

An approach using reinforcement learning and sliding mode control to synthesise a fault-tolerant controller was investigated by Li and Yang (2018). A specified integral-type sliding manifold was used to determine the sliding dynamics in the presence of actuator faults and disturbances. An optimal problem with a cost function reconstructed from a neural network was used to compute the control input within closed-loop bounds without *a-priori* knowledge of the system nonlinearities in the presence of the faults and disturbances. In a similar approach, the application of the sliding-mode fault-tolerant controller for a medium-scale unmanned autonomous helicopter with rotor flapping dynamics was studied by Yan *et al*. (2019). The disturbances in this case included wind gusts and the fault parameters were estimated using a combination of the extended state observer technique and an RBFNN. The closed-loop stability was proved through Lyapunov stability analysis.

The issue of control hardware degradation after the occurrence of an actuator fault combined with a sensor fault is investigated by Li *et al*. (2017). An integrated design of fault estimation and fault-tolerant control against simultaneous external disturbances, actuator and sensor faults was developed based on the Takagi–Sugeno fuzzy logic approach. Robustness against controller gain perturbations is achieved through a combination of an adaptive observer design and a non-fragile output feedback FTC design. This offline method was tested with a robotic manipulator without real-time environment applications. A similar adaptive fuzzy fault-tolerant controller under simultaneous actuator and sensor faults subject to a fixed-time stability of the closed-loop system was also investigated by Yang *et al*. (2020). The accommodation of additive and multiplicative faults for a slow-varying closed-loop system was achieved through a modified backstepping technique, resulting in the convergence time of the post-fault system being independent of the initial states.

A fault-tolerant scheme using RBFNNs was proposed by Shen *et al*. (2014). The integration of a fault detection threshold was used within a switching logic to select an active FTC scheme without an FDD component. Although fault location and classification was not investigated, the impact of FDD time delays on the selection of the appropriate RBFNN-based adaptive controller was analysed by Shen *et al*. (2017). The issue of fault detection time and the completion of the reconfiguration scheme was discussed by Yu and Jiang (2015). The implication of the computationally-expensive robust controller synthesis was highlighted as a weakness within an integrated FTCS context and a forgetting factor was suggested to increase the speed of the FDD scheme and improve the RFC performance by reducing the transient behaviour, even with an imprecise post-fault model.

Actuator saturation and rate limiting after a failure was investigated in the design of an FCTS using adaptive sliding mode control (ASMC) (Yu *et al*., 2017). Provided the remaining actuators were healthy, a fuzzy logic reconfiguration algorithm was combined with a zero-delay FDD system to synthesise an H_∞ and $\mu-$ controller. This resulted in an improved tracking performance of the post-fault system within the actuator constraints. The synthesis of a system reliability model, as a means of restoring the degraded performance of a faulty octocopter UAV, was investigated by Salazar *et al*. (2020). A control allocation scheme was developed which considers the load applied on each actuator over its mission life and the remaining actuators effectiveness based on possible load redistribution paths. This ensures a compromise between system reliability and controllability is achieved for a system subject to multiple actuator faults.

Similarly, such an integrated approach enabled the augmentation of a linear quadratic regulator controller to achieve system recovery without *a-priori* knowledge of the faulty system dynamics. This facilitated the implementation of such FTCS for real-world application, where the plant dynamics are often unknown or poorly described (Almutairi, 2016). The integrated approach to fault detection by considering the requirement to minimise false alarm rates through selecting an optimised false detection rate value was investigated by Chen *et al*. (2011).

The impact of bi-directional interactions between FDD and RFC within a real-time environment was investigated by Lan and Patton (2016) or Lan *et al*. (2017). It was shown that FTCS instability may occur if the system uncertainty, which causes the remaining healthy actuators to saturate, is ignored. Using a nonlinear 3-DOF helicopter system with saturated actuators, it was demonstrated that the lack of an integrated FTCS results in a sustained transient response combined with large overshoots due to poor estimation performance. The mitigation of such a problem was discussed by Prochazka *et al.* (2018). The use of an over-actuated system combined with a quadratic optimisation to minimise the control gain on the remaining actuators, proved to be an effective approach in compensating for the system dynamic response during the controller reconstruction process.

The impact of FDD uncertainties on the synthesis of an integrated FTCS was investigated by Rudin *et al.* (2020). Uncertainties in the form of (i) detection time delays (between a fault occurrence and its detection), (ii) undetected fault occurrences and (iii) detected non-faults were considered. An H_∞/μ synthesis is used to develop a set of the controllers with the reconfiguration switching mechanism designed such that the closed-loop system transient response is minimised through an optimisation algorithm of the objective cost function.

On the basis of the above literature and extending our previous works (Kantue and Pedro, 2018b; 2019; 2020), an integrated fault-tolerant control framework is proposed with considerations for a real-time environment and robustness against actuator fault estimation uncertainties. The development roadmap for such an integrated framework is shown in Fig. 1.

The fault diagnosis of incipient faults, while aiming to keep the time series information intact, has been investigated (Yang and Delpha, 2022). The local Mahalanobis distance method was used to develop a sensitive fault detection methodology applicable to non-Gaussian distributed data conditions. The real-time detection of incipient faults, irrespective of which system has been affected, is recognised for its practical importance in order to guarantee safe and optimal operating conditions and avoid the escalation of serious failures (Ji, 2021).

An incipient-type actuator fault for quadcopter UAV is considered and its online detection is achieved without *a-priori* knowledge of the system dynamics. Major contributions of this paper, in the context of the current literature, are as follows:

- The development of a nonlinear FDD framework by combining a multilateration approach and the real-time identification of a quadcopter incipient-type actuator fault using an RBFNN.

- The online computation of actuator fault uncertainties based on RBFNN prediction performance as part of an integrated FDD.

- The online computation of a controller reconfiguration cost function without *a-priori* knowledge of the post-fault dynamics as part of an integrated RFC.

- The development and real-time testing of a controller reconfiguration framework based on the extremum seeking algorithm under the presence of faults uncertainties.

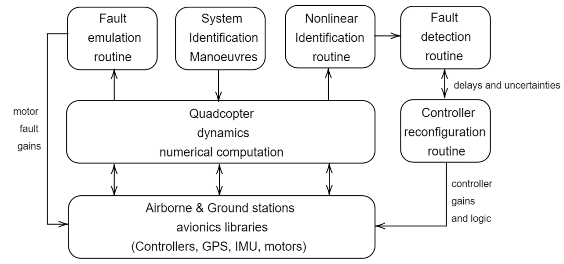

Fault emulation routine
System Identification Manoeuvres
Nonlinear Identification routine
Fault detection routine
delays and uncertainties
motor fault gains
Quadcopter dynamics numerical computation
Controller reconfiguration routine
controller gains and logic
Airborne & Ground stations avionics libraries (Controllers, GPS, IMU, motors)

Fig. 1. Technology roadmap for the proposed integrated fault-tolerant control framework.

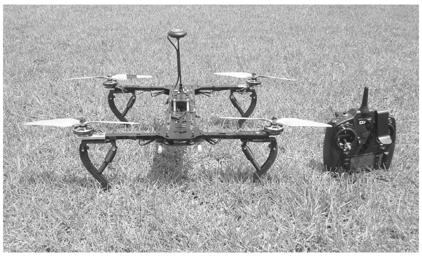


Fig. 2. Uav4africa research platform: H-1 quadcopter.

The rest of the paper is organised as follows. Section 2 describes the quadcopter system hardware and the dynamic model with incipient fault dynamics. The integrated FTCS is introduced in Section 3 and describes the integrated framework consisting of FDD and RFC modules. Section 4 discusses the results obtained, followed by conclusions and recommendations for future work in Section 5.

2. Quadcopter system

2.1. Airframe. The quadcopter system to test the real-time implementation of an integrated FTCS is the H-1 frame configuration developed by Uav4africa (Pty) Ltd (Kantue and Pedro, 2018a). The design was chosen for its stability in forward flight and a large fuselage area for the required sensors and auxiliary electronics. To enable non-destructive landing in the event of an unrecoverable system failure, a 3D-printed retractable landing gear has been installed (see Fig. 2). The actuation system is comprised of four RCTimer 510-620KV motors and inch propellers measuring 10 inches by 4.7 inches. A 4-cell 14.8V LiPo battery powers the flight controller and the powertrain.

2.2. Quadcopter dynamic model. By considering a coordinate system located at the quadcopter centre of gravity and aligned with the quadcopter body axes (shown in Fig. 3), the following equations of motion can be described (Gavrilets *et al.*, 2004; Mettler and Kanade, 2000; Kantue and Pedro, 2018b).

The equations of motion of a quadcopter can be expressed with respect to the body-fixed reference frame (Mettler and Kanade, 2000):

$$m\dot{\mathbf{v}} + m(\bar{\omega} \times \mathbf{v}) = \mathbf{F}, \tag{1}$$

$$\mathbf{I}\dot{\bar{\omega}} + (\bar{\omega} \times \mathbf{I}\bar{\omega}) = \mathbf{M}, \tag{2}$$

where the body-fixed linear and angular velocities are given by $\mathbf{v} = [u\ v\ w]^T$ and $\bar{\omega} = [p\ q\ r]^T$, respectively. External forces, such as gravity, aerodynamics and propulsion, are assumed to act at the quadcopter centre of gravity with a measured mass of m, and are defined as **F** = $[X\ Y\ Z]^T$. Based on geometric level arms and the mass moment of inertia defined as **I** = $[I_{xx}\ I_{yy}\ I_{zz}]$, external moments can be defined as **M** = $[L\ M\ N]^T$.

For a given rotor speed Ω and ambient density ρ, the generated propulsive force of the quadcopter rotor system is given as (Gavrilets *et al.*, 2004)

$$T_{p_i} = C_{T_i}\rho(\omega_{p_i}R_i)^2\pi R_r^2, \tag{3}$$

where the number of rotors is represented by $i = [1 - 4]$. The thrust coefficient of the i-th rotor can be expressed as

$$C_{T_i} = \frac{a_r\sigma_r}{2}\left(\theta_0\left(\frac{1}{3} + \frac{\mu_{ri}^2}{2}\right) + \frac{\mu_{z_{ri}} - \lambda_{0i}}{2}\right), \tag{4}$$

where the definition of the i-th rotor inflow velocity λ_{0i} can be expressed as

$$\lambda_{0i} = \frac{C_{T_i}}{2\eta_w\sqrt{\mu_{r_i}^2 + (\lambda_{0i} - \mu_{z_{ri}})^2}}, \tag{5}$$

$$\mu_{ri} = \frac{\sqrt{(u_i - u_{\text{wind}})^2 + (v_i - v_{\text{wind}})^2}}{\omega_{p_i}R_r}, \tag{6}$$

$$\mu_{z_{ri}} = \frac{w_i - w_{\text{wind}}}{\omega_{p_i}R_r}, \tag{7}$$

$$\sigma_r = \frac{2c_r}{\pi R_r}, \tag{8}$$

where the propeller lift-curve slope is a_r, c_r is the propeller blade root chord (this is approximated as a constant value), R_r and θ_0 are the propeller blade length and the approximated pitch angle, respectively (blade twist is assumed to be negligible). Here $\mu_{z_{ri}}$ and μ_{ri} represent the inflow velocities to be solved based on aerodynamic angles between the quadcopter body-axis and resulting velocity vector (Gavrilets, 2003).

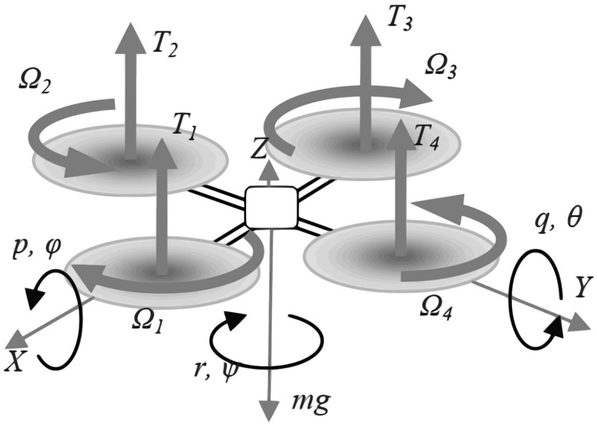


Fig. 3. Quadcopter propeller forces and moments.

Velocity components based on a local reference frame at the i-th rotor are denoted by u_i, v_i and w_i. These components can be computed as

$$u_i = u + S_{u_i}L_{fr}r\sin\theta_{fr}, \tag{9}$$

$$v_i = v + S_{v_i}L_{fr}r\cos\theta_{fr}, \tag{10}$$

$$w_i = w + R_{x_i}q + R_{y_i}p, \tag{11}$$

where L_{fr} and θ_{fr} are the linear and angular distances from the quadcopter centre of gravity to the frame rotor hub, respectively. $\bar{S}_u = [S_{u_1}, \ldots, S_{u_4}]$ is the distance scale factor for the i-th rotor. $\bar{R}_x$ and $\bar{R}_y$ are level arms resulting from the quadcopter angular motion along the body x-axis and y-axis, respectively.

The i-th rotor torque Q_r is computed as

$$Q_{p_i} = C_Q\rho(\omega_{p_i}R_r)^2\pi R_r^3, \tag{12}$$

where the torque coefficient is

$$C_Q = C_T(\lambda_{0i} - \mu_z) + \frac{C_{D_0}\sigma_r}{8}\left(1 + \frac{7}{3}\mu_{ri}^2\right). \tag{13}$$

For the purpose of developing an FTCS for a quadcopter with incipient actuator faults, the numerical computation of the H-1 quadcopter propeller dynamics was avoided by using wind tunnel data for a propeller measuring 25 cm long and 11.4 cm across (Brandt and Selig, 2011). Such wind tunnel data are plotted in Fig. 4. As expected, the propeller thrust C_T and torque C_Q coefficients are inversely proportional to the advance ratio (defined as the ratio between the forward speed and the propeller speed). This correlation between control effectiveness and forward flight is often not considered for most flight control laws integrated in commercially available flight controllers. By combining each rotor

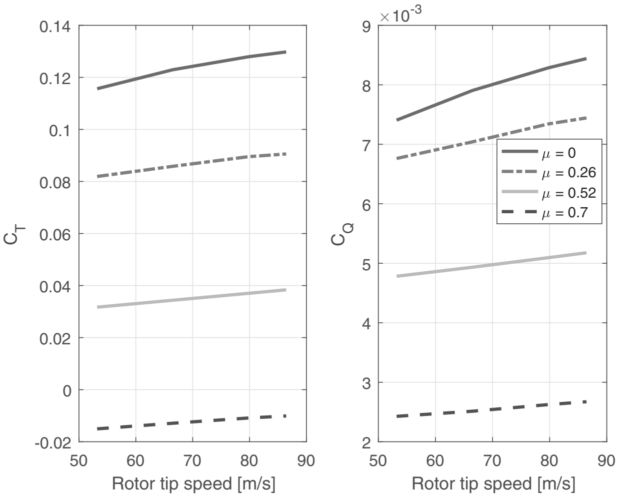


Fig. 4. Propeller thrust/torque coefficients for wind tunnel data.

flapping dynamics into a simplified tip-path-plane (TPP) model (excluding the dynamics of a feathering hinge motion), the following can be obtained (Gavrilets, 2003; Pedro and Kantue, 2011):

$$\dot{b}_{1i} = -p - \frac{b_{1i}}{\tau_e} - \frac{1}{\tau_e}\frac{\delta b_{1i}}{\delta \mu_v}\frac{v_i - v_{\text{wind}}}{\omega_{p_i} R_r}, \tag{14}$$

$$\dot{a}_{1i} = -q - \frac{a_{1i}}{\tau_e} - \frac{1}{\tau_e}\left(\frac{\delta a_{1i}}{\delta \mu}\frac{u_i - u_{\text{wind}}}{\omega_{p_i} R_r} + \frac{\delta a_{1i}}{\delta \mu_z}\frac{w_i - w_{\text{wind}}}{\omega_{p_i} R_r}\right), \tag{15}$$

the effective rotor time constant being τ_e. The impact of the TPP dihedral on the longitudinal dynamics is given as

$$\frac{\delta a_{1i}}{\delta \mu} = 2K_\mu \left(\frac{4\theta_0}{3} - \lambda_{0i}\right) \tag{16}$$

K_μ is a constant coefficient that introduces a restoring moment. The dihedral coefficients are defined as

$$\frac{\delta b_{1i}}{\delta \mu_v} = -\frac{\delta a_{1i}}{\delta \mu}, \tag{17}$$

$\delta a_{1i}/\delta \mu_z$ and $\delta b_{1i}/\delta \mu_v$ are the dihedral derivatives in the longitudinal and lateral axes, respectively. For model simplification, they are of equal magnitudes and have a destabilising effect.

The advancing blade produces more thrust during a heave movement resulting in a rotor hub moment. This effect can be described as

$$\frac{\delta a_{1i}}{\delta \mu_z} = K_\mu \frac{16\mu_{ri}^2}{(1 - \mu_{ri}^2/2)(8\,|\mu_{ri}| + a_r \sigma_r)}. \tag{18}$$

By considering the rotor flapping motion as the salient factor in the generation of rotor moments, aerodynamic moments generated by the quadcopter rotors can be represented by a linear model with a constant torsional stiffness value, K_β. This is shown in Fig. 5 and defined as

$$M_{k,\text{lon}} = K_\beta a_{1i}, \tag{19}$$

$$L_{k,\text{lat}} = K_\beta b_{1i}, \tag{20}$$

where $M_{k,\text{lon}}$ and $L_{k,\text{lat}}$ are the restoring moments due to the rotor flapping motion in the longitudinal and lateral axes, respectively. Additional moments are generated through the tilting of the i-th rotor due to the rotor flapping motion. Assuming that the rotor flapping angles are small, a linear approximation of the total moments about the quadcopter centre of gravity can be defined (N is the number of rotors):

$$L_p = \sum_{i=1}^{N} \left(K_\beta + T_{p_i} h_r\right) b_{1i}, \tag{21}$$

$$M_p = \sum_{i=1}^{N} \left(K_\beta + T_{p_i} h_r\right) a_{1i}, \tag{22}$$

$$N_p = \sum_{i=1}^{N} Q_{p_i}, \tag{23}$$

where the distance along the z-axis between the quadcopter centre of gravity and the rotor head is defined as h_r. Similarly, the rotor forces can be described as

$$X_p = -\sum_{i=1}^{N} T_{p_i} \sin(a_{1i}), \tag{24}$$

$$Y_p = \sum_{i=1}^{N} T_{p_i} \sin(b_{1i}), \tag{25}$$

$$Z_p = -\sum_{i=1}^{N} T_{p_i} \cos(a_{1i}) \cos(b_{1i}). \tag{26}$$

Given the above equations of motion, the additional dynamics of the incipient fault condition is defined through the model estimation of propeller angular speed Ω_i for the i-th affected rotor as a function of the faulty motor angular speed. The system identification of the propeller slippage faulty condition is described in Section 2.3.

2.3. Incipient fault model.

The most important part of a quadcopter is power electronics. Besides the power distribution board, the high-Watt battery and the electronic speed controller (ESC), the rotor system comprises a propeller tightened to the shaft of a three-phase brushless DC (BLDC) motor. This is shown in Fig. 6. Even though some rotors have a self-locking system, the presence of dust, grease, uneven mating surface and wear-and-tear can decrease the friction joint effectiveness causing a faulty

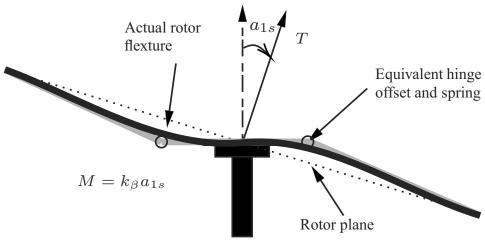


Fig. 5. Propeller flapping approximation.

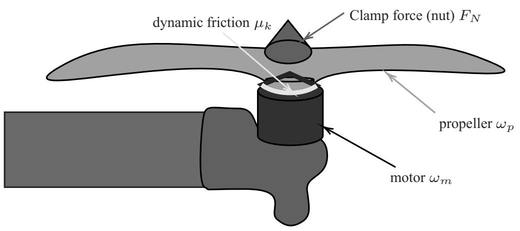


Fig. 6. Configuration of a quadcopter powertrain with an incipient fault condition.

quadcopter behaviour. If such a fault has a dynamic component or its amplitude increases with time, this is known as an incipient fault condition.

Given a motor torque on the i-th rotor defined as T_{m_i}, a fault condition can become incipient if the following conditions are true:

$$\omega_{m_i} - \omega_{p_i} > 0, \tag{27}$$

$$T_{m_i} \geq \mu_k(\xi_i) d_h (F_{N_i} - T_{p_i}), \tag{28}$$

where $\omega_{m_i}, \omega_{p_i}$ are the motor and propeller rotational speeds for the i-th rotor, respectively. Here d_h and F_{N_i} are the friction joint diameter and the friction joint clamping force, respectively. The propeller thrust is defined as T_{p_i}. $\mu_k(\xi_i)$ and ξ_i are the parameters defining the dynamic friction model for a Stribeck friction coefficient (Åström, 1998). Without access to the motor electrical parameters, such an incipient dynamic model can be identified through experimental methods as described by Kantue and Pedro (2019).

3. Integrated fault-tolerant control

3.1. Continuous forward algorithm. The continuous forward algorithm (CFA) was introduced by Peng *et al.* (2007), as a framework with analytical capabilities for both parameter optimisation and neural network construction. The CFA, unlike feedforward selection algorithms, has the ability to optimise the nonlinear parameters as the network architecture is adapted (Kantue and Pedro, 2018b; 2020). The main advantage of such a method includes an improved method for neural network modelling while achieving a low memory footprint per each network learning iteration. An input-output data set for a nonlinear RBFNN system is defined as

$$\hat{\mathbf{y}} = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}, \boldsymbol{\sigma}_i, \mathbf{c}_i) \tag{29}$$

where m is the number of hidden nodes, $\hat{y}$ and $\mathbf{x}$ are the RBFNN output and the input vector, respectively. $\phi_i(\mathbf{x}, \boldsymbol{\sigma}_i, \mathbf{c}_i)$ is the activation function of the i-th node of the hidden layer; $\mathbf{c}_i$, $\boldsymbol{\sigma}_i$ and w_i denote the node centre, width and linear output weight, respectively.

The optimal parameter values are obtained by minimising the sum squared error (SSE) defined as

$$\mathbf{J}(\mathbf{w}, \boldsymbol{\sigma}, \mathbf{c}) = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}). \tag{30}$$

The output vector is denoted as $\mathbf{y}$. A subset of k basis vectors exists from among M candidates such that the cost function can be minimised by computing network weights as

$$\mathbf{w} = (\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1} \mathbf{\Phi}_k^T \mathbf{y}, \tag{31}$$

where $\mathbf{\Phi}_k = [\phi_1, \dots, \phi_k]$ is a regressor subset to ensure the minimisation of the cost function defined in (30) can be expressed as a function of the regressor vector,

$$\mathbf{J}(\mathbf{\Phi}_k) = \mathbf{y}^T \left[\mathbf{I} - \mathbf{\Phi}_k \left(\mathbf{\Phi}_k^T \mathbf{\Phi}_k \right)^{-1} \mathbf{\Phi}_k^T \right] \mathbf{y}. \tag{32}$$

Further minimisation of the cost function can be achieved by specifying a basis vector $\forall \phi \in (\phi_{k+1}, \dots, \phi_M)$ as part of the regression matrix resulting in $\mathbf{\Phi}_{k+1} = [\mathbf{\Phi}_k, \phi]$. This cost function net reduction based on the updated regression matrix is defined as

$$\Delta \mathbf{J}_{k+1}(\phi) = \mathbf{J}(\mathbf{\Phi}_k) - \mathbf{J}([\mathbf{\Phi}_k, \phi]) \tag{33}$$

Based on the above cost function net reduction, a residual matrix exists such that (Li *et al.*, 2005)

$$\mathbf{R}_k = \begin{cases} \mathbf{I} - \mathbf{\Phi}_k \left(\mathbf{\Phi}_k^T \mathbf{\Phi}_k \right)^{-1} \mathbf{\Phi}_k^T, & 0 < k < M, \\ \mathbf{I}, & k = 0, \end{cases} \tag{34}$$

such that

$$\phi^{(k)} \triangleq \mathbf{R}_k \phi, \quad \mathbf{y}^{(k)} \triangleq \mathbf{R}_k \mathbf{y}, \tag{35}$$

where ϕ and $\mathbf{y}$ are a column vector and the output vector respectively such that $\phi^{(0)} = \phi$ and $\mathbf{y}^{(0)} = \mathbf{y}$. A recursive update of such vectors can be achieved as follows:

$$\phi^{(k)} = \phi^{(k-1)} - \frac{(\phi_k^{(k-1)})^T (\phi^{(k-1)})}{(\phi_k^{(k-1)})^T (\phi_k^{(k-1)})} \phi_k^{(k-1)} \tag{36}$$

Without a detailed analysis of the residual matrix properties defined by Li *et al.* (2005), the recursive computation of the node regressors can be applied for the k-th basis vector, which yields

$$\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} - \frac{(\phi_k^{(k-1)})^T(\mathbf{y}^{(k-1)})}{(\phi_k^{(k-1)})^T(\phi_k^{(k-1)})}\phi_k^{(k-1)}. \tag{37}$$

Similarly, the cost function net contribution is defined as

$$\Delta\mathbf{J}_{k+1}(\phi) = \frac{[(\mathbf{y}^{(k)})^T\phi^{(k)}]^2}{(\phi^{(k)})^T\phi^{(k)}}. \tag{38}$$

The implementation of (36) and (37) can be simplified by introducing an upper triangular matrix $\mathbf{A}$ of size $k \times M$ which consists of k cost function basis vectors. Such a matrix is defined as

$$\mathbf{A} \triangleq [a_{i,j}]_{k\times M}, \tag{39}$$

$$a_{i,j} = \begin{cases} 0, & j < i, \\ \phi_i^T\mathbf{R}_{i-1}\phi_i \\ = (\phi_i^{(i-1)})^T(\phi_i^{(i-1)}), & j = i, \\ \phi_i^T\mathbf{R}_{i-1}\phi_j \\ = (\phi_i^{(i-1)})^T(\phi_j^{(i-1)}), & i < j < M, \end{cases} \tag{40}$$

shown here in matrix form

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,M} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{k-1,M-k} & a_{k-1,M} \\ 0 & \cdots & 0 & 0 & a_{k,M} \end{bmatrix},$$

where

$$\mathbf{a}_y \triangleq [a_{i,y}]_{M\times 1}, \qquad a_{i,y} = \mathbf{y}^T\mathbf{R}_k\phi_i = (\mathbf{y}^{(k)})^T\phi_i^{(k)}, \tag{41}$$

$$\mathbf{d} \triangleq [d_i]_{M\times 1}, \qquad d_i = \phi_i^T\mathbf{R}_k\phi_i = (\phi_i^{(k)})^T\phi_i^{(k)}. \tag{42}$$

The parameter set of each hidden node (width and centre) can be expressed as

$$\phi(\mathbf{x}(t), \boldsymbol{\sigma}, \mathbf{c}) = \Phi(\mathbf{x}(t), \omega), \tag{43}$$

where

$$\omega = [\omega_0, \omega_1, \ldots, \omega_n] = [\boldsymbol{\sigma}, \mathbf{c}]. \tag{44}$$

The cost function net contribution can be defined as

$$\Delta\mathbf{J}_{k+1}(\omega) = \frac{C^2(\omega)}{D(\omega)}, \tag{45}$$

where

$$\begin{aligned} C(\omega) &= (\mathbf{y}^{(k)})^T\phi^{(k)}(\omega) && (46) \\ &= \sum_{t=1}^{N} y^{(k)}\phi^{(k)}(\mathbf{x}(t), \omega), \\ D(\omega) &= (\phi^{(k)}(\omega))^T\phi^{(k)}(\omega) && (47) \\ &= \sum_{t=1}^{N}(\phi^{(k)}(\mathbf{x}(t), \omega))^2, \end{aligned}$$

N being the number of training samples. The optimisation of each $(k+1)$-th added hidden node parameter set added to the regression matrix ensures that the cost function net contribution is maximised at each training iteration. This is achieved through a conjugate gradient approach by computing the derivative of the cost function net contribution (Kantue and Pedro, 2018b):

$$\begin{aligned} &\frac{\partial\Delta\mathbf{J}_{k+1}(\omega)}{\partial\omega_i} \\ &= \nabla\Delta\mathbf{J}_{k+1}(\omega), \\ &= \frac{2C(\omega)}{D(\omega)}\left(\mathbf{y}^{(k)} - \frac{C(\omega)}{D(\omega)}\phi^{(k)}(\omega)\right)^T \phi_{\omega_i}^{(k)}(\omega), \end{aligned} \tag{48}$$

$i = 0, 1, \ldots, n$ where $\phi_{\omega_i}^{(k)}$ is the impact factor of the k-th hidden node regressor based on the computed parameter set ω_i for a given i-th input vector, defined as

$$\begin{gathered} \phi_i^{(s)}(\omega) = \phi_i^{(s-1)}(\omega) - \frac{\partial a_{s,k+1}(\omega)}{\partial\omega_i}\frac{\phi_s^{(s-1)}}{a_{s,s}}, \\ s = 1, \ldots, k, \quad i = 0, 1, \ldots, n, \end{gathered} \tag{49}$$

where

$$\frac{\partial a_{s,k+1}(\omega)}{\partial\omega_i} = (\phi_s^{(s-1)}(\omega))^T\phi_i^{(s-1)}(\omega). \tag{50}$$

The algorithm can be initialised at $k = 0$ such that

$$\begin{aligned} \phi_{\omega_0}^{(0)}(\mathbf{x}(t), \omega) &= -2\omega_0\sum_{i=1}^{n}(x_i(t) - \omega_i)^2\phi(\mathbf{x}(t), \omega), \\ \phi_{\omega_i}^{(0)}(\mathbf{x}(t), \omega) &= 2\omega_0^2(x_i(t) - \omega_i)\phi(\mathbf{x}(t), \omega), \end{aligned} \tag{51}$$

$i = 1, \ldots, n$, where the node parameter set can be initialised as

$$\omega_0^{(0)} = \left[\sum_{i=1}^{n}\sum_{t=1}^{N}(x_i(t) - \omega_i^{(0)})^2/N\right]^{-1/2}. \tag{52}$$

This conjugate gradient method makes use of a line search iterative procedure such that gradient of the contribution of $\nabla\Delta\mathbf{J}_{k+1}(\omega_{k+1}^{(p)})$ results in a parameter set $\omega_{k+1}^{(p)}$ that maximises $\Delta\mathbf{J}_{k+1}(\omega_{k+1}^{(p)})$ and is considered an optimal value; $\phi(\omega_{k+1}^{(0)})$ is then computed along with the triangular matrix A resulting in the $(k+1)$-th hidden node being updated with $\phi(\omega_{k+1}^{(k)})$ as defined in (36) and its output vector can be computed as defined in (37). This results in the reduction in the SSE defined as

$$\mathrm{SSE}^{(k+1)} = \mathrm{SSE}^{(k)} - \Delta\mathbf{J}_{k+1}(\omega_{k+1}^{(k)}). \tag{53}$$

The above procedure is iterated until the net contribution of the cost function meets the condition of being below a defined threshold η (0.1 has been used) or the SSE is below 10^{-3}. The evaluation of the above-described CFA algorithm using quadcopter flight data is discussed by Kantue and Pedro (2018b). The following parameters can be defined for network validation and assess the network capturing of the underlying system dynamics:

$$\mathrm{NN}_{\text{bias}} = \Big(\frac{1}{N}\sum(Y' - Y)\Big)^2, \tag{54}$$

$$\mathrm{NN}_{\text{MSE}} = \frac{1}{N}\sum(Y' - Y)^2, \tag{55}$$

$$\mathrm{NN}_{\text{Var}} = NN_{\text{MSE}} - NN_{\text{bias}}, \tag{56}$$

where $\mathrm{NN}_{\text{bias}}$, NN_{MSE} and NN_{Var} are the network output bias, mean square error and variance, respectively. Y and Y' are the network predicted output and validation data set output, respectively.

The determination of the input/output mapping was achieved by selecting the RBF model structure through computing the collinearity amongst dependent and independent variables. The approach on data pre-conditioning and model structure size was investigated by Kantue and Pedro (2018b). The training and validation data sets were then extracted from that mapping and the RBF network was trained and validated using the modified CFA algorithm and the defined criteria, respectively.

3.2. Golden section search with multimodal capability. Given the construction of the CFA model identification method, the maximisation of the cost function net contribution, defined as $\Delta\mathbf{J}_{k+1}(\omega)$, will be achieved by finding the optimal value of $\omega_{k+1}^{(p)}$ through a line search procedure (Kantue and Pedro, 2020). If a function $f(x)$ is defined as unimodal, the maximum value of such a function $\max f(x)$ exists within a specified interval $[a, b]$. Such a maximisation problem can be solved by the golden section search (GSS) algorithm by finding a subinterval $[x_1, x_2]$. A detailed explanation of the GSS procedure can be found the work of Kantue and Pedro (2020). Given that the CFA algorithm introduces nonlinearities in the computation of the cost function net contribution such as $D(\omega) \to 0$:

$$\Delta\mathbf{J}_{k+1}(\omega) = \begin{cases} 0, & D(\omega) < 10^{-5}, \\ C^2(\omega)/D(\omega), & D(\omega) > 10^{-5}. \end{cases} \tag{57}$$

A local minimum is introduced resulting in a multimodal function.

To mitigate the limitations of the GSS algorithm, a meta-heuristic search was developed by Kantue and Pedro (2020). In summary, a function peak finder algorithm enables the computation of the GSS interval $[a, b]$ by using a moving window search space and monitors the line gradient within that window interval. The GSS algorithm is then executed within the specified interval of the highest peak and ensures that the maximum of the cost function net contribution $\Delta\mathbf{J}_{k+1}(\omega)$ is achieved.

3.3. Application of the time-difference-of-arrival-process. An incipient fault condition is based on the observation that the dynamic behaviour of a post-fault system is similar to the pre-fault system. This premise only holds provided that the controlling action or the external environment remain within certain bounds (Mien *et al.*, 2011). An FDD method comprised of the modified CFA algorithm described in Sections 3.1 and 3.2 and the concept of time-difference-of-arrival (TDOA) can be used to identify and locate an incipient actuator fault within the control action bounds to ensure that the performance of the post-fault system is maximised. The inclusion of the TDOA is complimentary to a data-driven estimation approach, as it is based on locating a signal source without *a-priori* knowledge of the system (Chen *et al.*, 2019).

By injecting a pulse width modulated (PWM) signal $\Delta\omega_{m_i}$ in a sequential manner and with an associated timestamp in each motor of a quadcopter configuration, as shown in Fig. 7, a localisation scheme can be developed using the TDOA method. Each motor acts as an emitter and the receiver of each emission is the inertial measurement unit (IMU) such that

$$S_i = [s_0, s_1, \dots, s_n] \tag{58}$$

where n represents the number of samples collected during the excitation of the i-th rotor with a PWM manoeuvre $\Delta\omega_m$. Ensuring that such a manoeuvre results in an optimal energy content and the description of its implementation in the Ardupilot software is described by Kantue and Pedro (2018b).

The TDOA relies on the premise that two emitting sources have been recorded and their respective timestamps (time of arrival or time of departure) are available. A TDOA square full-rank matrix $\overline{Q}$ (given

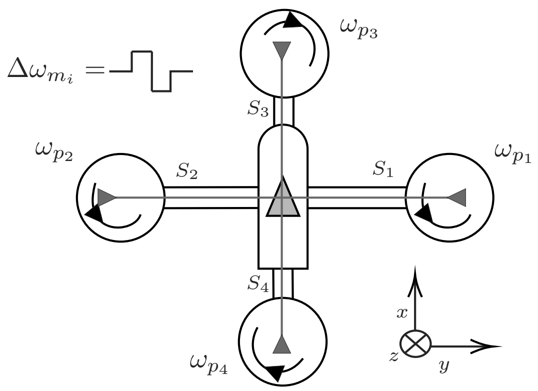


Fig. 7. TDOA concept for a quadcopter configuration.

$m = n$), can be defined as

$$\overline{Q} = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{m1} & S_{m2} & \dots & S_{mn} \end{bmatrix}, \tag{59}$$

where S_{mn} is the time difference cross-correlation between two signals as a function of the number of samples lagging or leading the reference signal defined as

$$S_{mn} = \begin{cases} Z(P_m, E_n), & m = n, \\ Z(P_m, P_n), & m \neq n, \end{cases} \tag{60}$$

where $Z(P_m, E_n)$ represents the cross-correlation time delay output between the predicted NN output of the m-th motor P_m and the estimated NN output of the n-th motor E_n. The diagonal of matrix $\overline{Q}$ represents the time delay cross-correlation when $m = n$. This is defined as

$$Z(P_m, E_n) = \arg\max_{t\,\in\,\mathbb{R}} \int_0^T P_m(t)E_n(t+\tau)\,\mathrm{d}t. \tag{61}$$

When $m \neq n$, we can also deduce that

$$Z(P_m, P_n) = \arg\max_{t\,\in\,\mathbb{R}} \int_0^T P_m(t)P_n(t+\tau)\,\mathrm{d}t, \tag{62}$$

where the time sample delay factor τ lies within $[-T; T]$ based on the following constraint:

$$P_n(t+\tau) = \begin{cases} 0, & \tau \leq t, \\ P_n(t+\tau), & \tau > t. \end{cases} \tag{63}$$

At each timestep τ and for $m = n$, the cross-correlation is defined as

$$\begin{aligned} &P_m(t)E_n(t+\tau) \\ &\quad = \frac{\sum_{t=0}^{T}\left(P_m(t) - \overline{P_m(t)}\right)\left(E_n(t+\tau) - \overline{E_n(t)}\right)}{\sum_{t=0}^{T}\left(P_m(t) - \overline{P_m(t)}\right)^2 \sum_{t=0}^{T}\left(E_n(t) - \overline{E_n(t)}\right)^2}, \end{aligned} \tag{64}$$

where $\overline{P_m(t)}$ and $\overline{E_n(t)}$ are the mean values. The localisation of the fault can be achieved through the index of column $\overline{Q}$ with the highest variance, defined as α, once the rank of $\overline{Q}$ is at least $n-2$. Such an index is written as

$$\alpha = \arg\max_{i\,\in\{1,\dots,n\}} \sigma_S^2(i), \tag{65}$$

$$\sigma_S^2(i) = \frac{1}{m}\sum_{z=1}^{m}\left(S_{zi} - \overline{S_i}\right), \tag{66}$$

where the variance of the i-th column for the matrix $\overline{Q}$ is denoted by $\sigma_S^2(i)$. The mean value and variance of the i-th column is defined as $\overline{S_i}$ and $\sigma_S^2(i)$. To ensure that the reconfiguration time is maximised, initial FDD information can be provided through fault localisation without the matrix $\overline{Q}$ being full rank. In addition, a level of uncertainty can be defined and associated with the estimated fault location. This is described in the next section.

3.4. Requirements from a reconfigurable controller. The augmented control of the remaining healthy rotors of a post-fault control system while the impact on system stability and performance is minimised, is the primary objective of the reconfigurable flight controller (RFC) (Xulin and Yuying, 2018). Given the increased transient behaviour of a post-fault model with multiple actuators, a control allocation-based reconfiguration scheme has the following integration requirements:

- maximise the reconfiguration time to ensure robustness against fault location and fault magnitude uncertainties;
- minimise the fault detection errors for optimal post-fault system performance.

To comply with the above requirements, the following mechanism has been developed. Consider a faulty nonlinear system

$$\dot{\mathbf{x}} = H_f(\mathbf{x}, \mathbf{u}), \tag{67}$$

$$\mathbf{y} = G_f(\mathbf{x}), \tag{68}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{y} \in \mathbb{R}^n$ is the output vector, $\mathbf{u} \in \mathbb{R}^n$ is the input vector and G_f, H_f

are unknown nonlinear functions describing the system faulty dynamics. A fault magnitude uncertainty factor $\mathbf{F}$ associated with the fault localisation index α can be defined as

$$\mathbf{F} = \begin{cases} 0, & \operatorname{rank} \overline{Q} \; < n-2, \\ 1 - e^{-\left(\eta \frac{F_\alpha}{\operatorname{rank}\,\overline{Q}}\right)}, & \operatorname{rank} \overline{Q} \; \geq n-2, \end{cases} \tag{69}$$

where

$$F_\alpha = \overline{S}_\alpha, \tag{70}$$

$\overline{S}_\alpha$ is the mean value of the α-th column of $\overline{Q}$. Provided the minimum rank of $\overline{Q}$ matrix has been achieved, a user-defined parameter $0 < \eta < 1$ can be defined such that $\mathbf{F}$ is close to zero when the matrix $\overline{Q}$ is close to full rank or the maximum variance column α is close to zero (no fault has occurred). With the correct selection of a suitable controller reconfiguration method, the above integration requirements can be achieved.

3.5. Extremum seeking control algorithm. Well-known control reconfiguration techniques such as gain scheduling, model reference adaptive control, eigenstructure assignment or dynamic inversion compute the required control action based on the *a-priori* knowledge or estimation of the underlying system dynamics. Such approaches are sensitive to fault uncertainties, model uncertainties and often require high computation resources for real-time implementation (Zhang *et al.*, 2013). A data-driven approach such as extremum seeking control is well-suited to be part of the integrated FTCS without the need for *a-priori* knowledge of the post-fault model. It can also be implemented as a control allocation mechanism under real-time constraints and its low computing requirements which make it well-suited for online system reconfiguration.

Consider the following state-space representation of a nonlinear model:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{71}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}, \tag{72}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{y} \in \mathbb{R}^q$ is the output vector, $\mathbf{u} \in \mathbb{R}^p$ is the input vector and $A_{n\times n}$, $B_{n\times p}$, $C_{q\times n}$ and $D_{q\times p}$ are state, input, output and feedthrough matrices, respectively. Assume that there exists a control law, parameterised by $\theta = [\theta_1, \dots, \theta_p]$, given as

$$\mathbf{u}_\theta \; = -KH(\theta, \mathbf{F})\mathbf{x} \tag{73}$$

The above state-space equation can be re-written as

$$\dot{\mathbf{x}} \; = \; A\mathbf{x} + B\mathbf{u}_\theta, \tag{74}$$

$$\mathbf{y}_\theta = C\mathbf{x} + D\mathbf{u}_\theta, \tag{75}$$

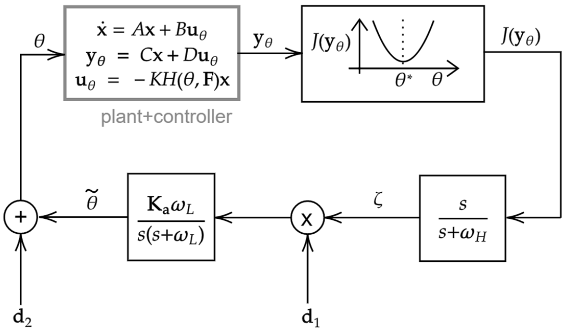


Fig. 8. Extremum seeking control framework.

such that the closed-loop system

$$(A - BKH(\theta, \mathbf{F}))\mathbf{x} = 0 \iff \mathbf{x} = L(\theta) \tag{76}$$

where K is the compensator, $H(\theta, \mathbf{F})$ is a control allocation based on a fault detection and diagnosis scheme and the fault uncertainty $\mathbf{F}$, which is unity when no fault has been detected. $L(\theta)$ is the system state equilibrium which is achieved through finding the extremum (in this case a minimum) value:

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^p} J(\mathbf{y}_\theta), \tag{77}$$

where $J(\mathbf{y}_\theta)$ is the objective cost function comprising of the output vector $\mathbf{y}_\theta$, such that the steady-state value of θ^* is obtained without knowledge of system matrices A, B, C and D. This minimisation feedback mechanism is known as extremum seeking (Yin *et al.*, 2018; Wang *et al.*, 2016). The ES control framework and its effect on a closed-loop system is shown in Fig. 8.

The ES process is to modulate the estimate $\tilde{\theta}$ with a periodic signal $\mathbf{d}_2$ such that a periodic response is created in $J(\theta)$. This is passed through a high-pass filter $s/(s + \omega_H)$ to remove the mean value. Another periodic signal, $\mathbf{d}_1$, demodulates the signal ζ and gets passed through a low-pass filter with integrator $\mathbf{K_a}\omega_L/(s(s + \omega_L))$ to update the estimate $\tilde{\theta}$. Given that the search is for a minimum, we have $\mathbf{K_a} < 0$. It should be noted that the parameter gradient can be defined as (Krstic and Wang, 2000)

$$\dot{\tilde{\theta}} \; = \; \mathbf{K_a} \left(\frac{\mathbf{d}_1 \mathbf{d}_2}{2} \right) \frac{\partial J}{\partial \tilde{\theta}} \tag{78}$$

for the user-defined choice of the periodic signal vectors

$$\mathbf{d}_1 = \; [a_1 \sin(\omega_1 t), \dots, a_n \sin(\omega_n t)], \tag{79}$$

$$\mathbf{d}_2 = \; [b_1 \sin(\omega_1 t + \phi_1), \dots, b_n \sin(\omega_n t + \phi_n)], \tag{80}$$

where ω_i and ϕ_i are the frequency and phase angle, respectively the adaptation gain vector $\mathbf{K_a} = [K_{a_1}, \dots, K_{a_n}]$, the low-pass filter frequency vector

$\omega_L = [\omega_{L_1}, \dots, \omega_{L_n}]$ and the high-pass filter frequency vector $\omega_H = [\omega_{H_1}, \dots, \omega_{H_n}]$ determine the speed of convergence of $\theta \to \theta^*$ and stability of the closed-loop system. Detailed analysis into the stability of the ES control algorithm has been carried out by Krstic and Wang (2000). Incorporating a fault emulation filter G_f, representing the estimated incipient fault dynamics, results in the stability criterion

$$(A - BG_f KH(\theta))\mathbf{x} \le \xi \iff \mathbf{x} = L_f(\theta), \qquad (81)$$

where ξ is a design parameter representing the new equilibrium state described by L_f. Given $\theta \in \mathbb{R}^p$, a real-time implementation of the ES control algorithm is required for a quadcopter system. A reduction of the optimisation parameter θ search space with dimension $p \times p$ is introduced in the next section.

3.6. Reconfigurable controller gain computation.

The quadcopter control input vector $\mathbf{u}$ can be defined as

$$\mathbf{u}_\theta = [\delta_t, \delta_a, \delta_e, \delta_r],$$

where $[\delta_t, \delta_a, \delta_e, \delta_r]$ is the throttle, aileron, elevator and rudder commands, respectively. Given a quadcopter '+' configuration as shown in Fig. 7 and a weight factor based on the ES optimisation vector $P(\theta, \mathbf{F})$, the mixing to each rotor command δ_{c_i} can be defined using the right-hand rule (abbreviated as P):

$$\begin{bmatrix} \delta_t \\ \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = \frac{1}{4} \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \\ -2P_1 & 2P_2 & 0 & 0 \\ 0 & 0 & -2P_3 & 2P_4 \\ 2P_1 & 2P_2 & -2P_3 & -2P_4 \end{bmatrix} \begin{bmatrix} \delta_{c_1} \\ \delta_{c_2} \\ \delta_{c_3} \\ \delta_{c_4} \end{bmatrix},$$

$$\mathbf{u}_\theta = H(\theta, \mathbf{F})\mathbf{u}_c,$$

where $P_i(\theta, \mathbf{F})$ is a weight factor of the i-th motor control command δ_{c_i} and based on a FDD fault uncertainty factor $\mathbf{F}$, described in Section 3.4. The optimisation parameter vector θ can be defined such that the ES objective cost function $J(\mathbf{y}_\theta)$ can be based on the same measured variables affected by the FDD mechanism with the associated uncertainties. This was achieved by considering the real-time implementation of the nominal autopilot architecture.

The ArduCopter software was modified by implementing the ES controller and interfacing it to the `AP_WPNav.cpp` routine. The ES controller was allowed to override the trajectory commands $\mathbf{T}_c$ with step functions commands $\mathbf{S}_c$ as inputs to the routine `AP_AttitudeControl.cpp`. This is shown in Fig. 9.

Given the attitude controller updating the motor outputs through routine `AP_MotorsMatrix.cpp` every k time-steps, the objective cost function is defined

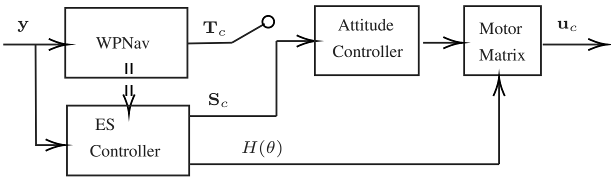


Fig. 9. ES controller within the ArduCopter software.

as follows:

$$J(\mathbf{y}_\theta)^k = \underbrace{\int_{k-2}^{k-1} \varepsilon_S(t)\,\mathrm{d}t}_{J(\mathbf{y}_\theta)_S} + \underbrace{\int_{k-1}^{k} \varepsilon_T(t)\,\mathrm{d}t}_{J(\mathbf{y}_\theta)_T}, \qquad (82)$$

where

$$\varepsilon_S(t) = \frac{V_0}{V_k} \sum (\mathbf{S}_c - \mathbf{E}_m)^2, \qquad (83)$$

$$\varepsilon_T(t) = \frac{V_0}{V_k} \sum (\mathbf{T}_c - \mathbf{E}_m)^2,$$

V_0 and V_k are the flight speeds at $k = 0$ and k steps, respectively. $\mathbf{S}_c$ is the vector of attitude step commands in the Euler roll, pitch and yaw angles. $\mathbf{T}_c$ is the vector of trajectory commands in the Euler roll, pitch and yaw angles. $\mathbf{E}_m$ is the vector of Euler roll, pitch and yaw angles measurements. At every k-th step, the motor weight factor matrix $H(\theta)$ was updated.

3.7. Hardware-in-the-loop model simulation.

The performance analysis of the FDD mechanism within a real-time environment to quantify its suitability for the chosen reconfiguration scheme is part of the development of an integrated FTCS. The method used to achieve this objective is using a hardware-in-the-loop simulation (HILS) model. This is shown in Fig. 10.

The CFA-based FDD scheme, as described in Section 3.1, was converted from Matlab scripts to executable code running on a 180 MHz microprocessor with a floating point unit known as Teensy 3.6. An additional integrated FTCS objective of having an FDD scheme supporting low-memory computation was achieved by splitting the CFA algorithm training, searching and prediction sub-routines across the CPU stack (local variables) and heap (global variables) memory allocations. A PC Windows running was used to pass pre-simulated training data to the microcontroller via USB and monitor the serial outputs.

4. Results

4.1. FDD performance.

In order to avoid parameter estimation inaccuracies and unnecessary model complexity, the collinearity amongst dependent and independent variables must be quantified with an RBFNN structure. The optimal design of such inputs is discussed by Kantue and Pedro (2018b). The CFA algorithm described in Section 3.1 was used for training the RBFNN model structure. The number of neurons and resultant SSE once ΔJ_{k+1} had reached zero, representing each motor, was then used for the TDOA algorithm. The GSS combined with the meta-heuristic algorithm was used to maximise the cost function net contribution $\Delta \mathbf{J}_{k+1}(\omega)$ on a data set of 352 points. This enables us to capture the underlying system dynamics through minimising the NN predicted output error compared with the validation dataset. Although external disturbances are not taken into consideration in the assessment of the CFA algorithm within this paper, such assessment was done by Kantue and Pedro (2018b).

The RBFNN input set was pre-selected as described by Kantue and Pedro (2018b). Based on the input/output signal correlation, the yaw rate r was chosen as the RBFNN output. Given the incipient fault type, this could be associated with the dynamic behaviour of the faulty propeller ω_p on the naturally unstable quadcopter yaw dynamics. A detailed analysis of the impact on the RBFNN variance and bias was made by Kantue and Pedro (2020). In summary, the robustness of the implemented TDOA to the RBFNN potential inability to capture the underlying dynamics is manifested through the computation of the TDOA matrix $\bar{Q}$. For a typical mission, the fault detection manoeuvres were activated when the GPS speed reached 3 m/s (around 73 s). The incipient fault model was introduced on motor 1 at the GPS speed of 5 m/s (around 75 s). These time events are illustrated on the PWM signals shown in Fig. 11. To ensure adequate trajectory tracking, a recovery period was introduced after all propellers were excited.

The TDOA matrix $\bar{Q}$ and the associated column variance used for FDD fault localisation, fault magnitude, and fault uncertainty computation, are shown in Figs. 12–15. Two incipient fault models were analysed. Low and high incipient fault magnitudes equal to 50% and 90% of motor efficiency, respectively. The FDD is able to correctly predict the fault localization through the column with the highest variance in the matrix $\bar{Q}$, which is shown in Figs. 12 and 14. FDD robustness to variance changes can also be seen as it increases from 250 to 6000. The fault uncertainty and its impact on the FDD detection time and overall RFC performance is shown in Figs. 13 and 15. Early detection (with a low $\operatorname{rank} \bar{Q}$ threshold and large uncertainty) is made independent of the incipient fault magnitude. The FDD mechanism of rejecting false positive cases, as shown with variance motor 3, also prevents the propagation of the fault uncertainty gain $\mathbf{F}$ to the RFC mechanism, ensuring the minimal transient behaviour for the activation of the RFC mechanism.

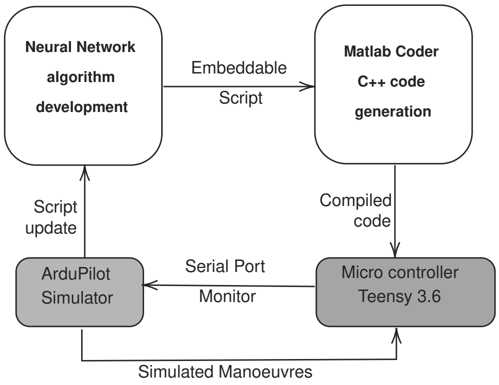


Fig. 10. HILS framework for the real-time testing of FDD algorithms.

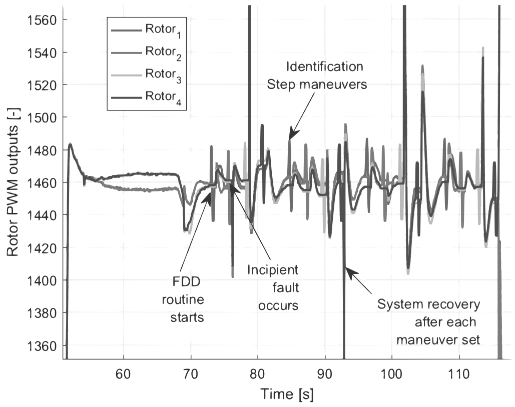


Fig. 11. PWM motor output during identification manoeuvres after the occurrence of an incipient fault.

4.2. RFC performance. The real-time implementation of the RFC mechanism was achieved through the modification of the ArduPilot software-in-the-loop (SITL) framework as described by Kantue and Pedro (2019). Two functions were developed directly in C++ and were executed at 400 Hz along with the rest of the software. Figure 16 illustrates the angle tracking performance between two phases within the context of the FTCS framework. The tuning of the extremum seeking controller was assisted with the low-pass filtering of the motor commands prior to controller reconfiguration. The feedback loop filters were designed as follows: high-pass filter bandwidth $\omega_H = 2$ rad/s, low-pass filter bandwidth $\omega_L = 4$ rad/s, the adaption gain $\mathbf{K_a} = 0.01$, the demodulation and modulation signal amplitudes are 0.05 and 0.01, respectively. The performance of those feedback filters is shown in Fig. 17. As expected, the high-frequency content of the modulated signal is filtered prior to the computation of the objective cost function.

The objective cost function $J(\mathbf{y}_\theta)$ and its tracking error components $J(\mathbf{y}_\theta)_S$ and $J(\mathbf{y}_\theta)_T$ are shown in Fig. 18. It can be seen that a minimum is reached with the controller remaining stable. The reconfiguration time

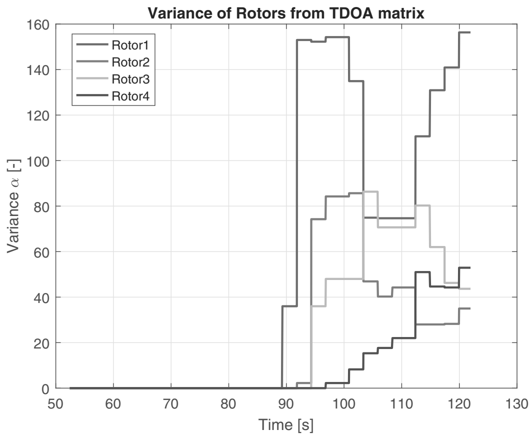
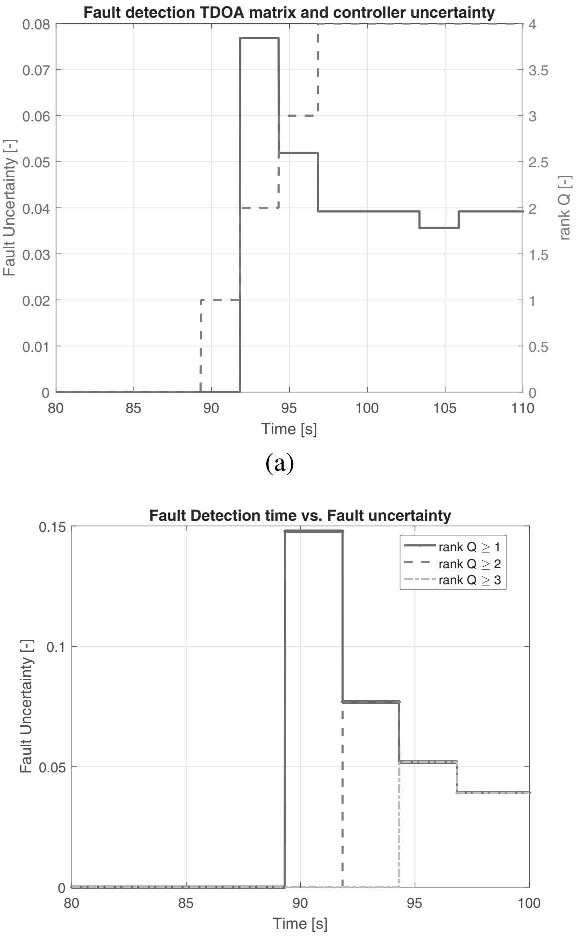


Fig. 12. Variance α of the output of the $\overline{Q}$ matrix during the execution of the FDD algorithm after the occurrence of a low magnitude incipient fault.



(a)

(b)

Fig. 13. FDD algorithm outputs: fault uncertainty (a) and uncertainty sensitivity (b) (a low magnitude incipient fault).

introduces a transient which coincides with the high rate of change in the optimisation parameter θ. Based on the FDD mechanism performance detailed in the previous section, the impact of a varying degree of the incipient fault magnitude on the reconfiguration time is through the fault uncertainty gain $\mathbf{F}$, but this is quickly dissipated as the $\operatorname{rank} \bar{Q}$ increases. The optimisation of the objective cost function and its impact on the reconfigured baseline case is analysed by comparing a trajectory tracking mission with or without an active FTCS. The system without an active reconfigurable controller, shown in Fig. 19, becomes unstable after each sharp turn due to the incipient rotor fault degrading the performance of the default controller. This degradation is completely removed with an active FTC shown in Fig. 20.

5. Conclusion

The development of an integrated approach to fault-tolerant controller design has been presented. An FDD mechanism against a specific type of the actuator faults, called incipient faults, has been constructed through the use of an RBFNN and a modified CFA with a multimodal capability. The integration requirements imposed on the FDD to minimise the detection time and be robust to false positives has been achieved through the use of the time-difference-of-arrival (TDOA) method. The secondary requirement to be fit-for-purpose to interface with an RFC mechanism in a real-time environment was achieved through considering an HILS environment. The approach of integrating the FDD fault uncertainty magnitude into an extremum seeking controller design resulted in maximising the reconfiguration time such that the optimisation of the controller gains could be achieved while minimising the transient behaviour during the reconfiguration. This has resulted in an integrated FTCS framework that has restored pre-fault tracking performance in a straight-level flight and during tight turns while being robust to FDD fault uncertainty and RFC transients.

Further investigations from this research include the real-time flight testing of such FTCS for various types of aircraft including fixed-wing ones and helicopters. A reduction in the variance of the estimated RBFNN outputs could improve the fault uncertainty factor and minimise the transient behaviour of the RFC objective cost function, further reducing the time span from when a fault has been detected to when the system has completed its controller gain reconfiguration.

References

Almutairi, S.H. (2016). *Optimal Fault-Tolerant Flight Control for Aircraft with Actuation Impairments*, PhD thesis, Cranfield University, Cranfield.

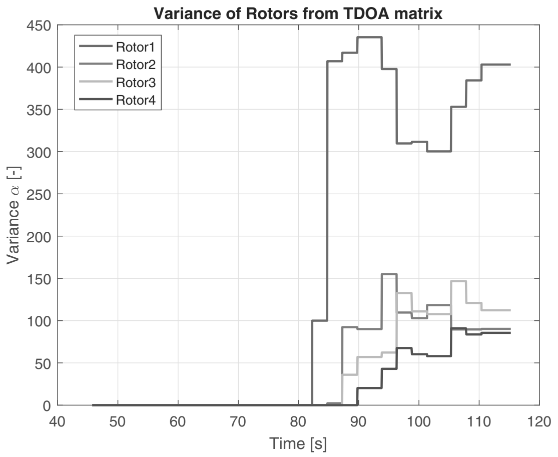
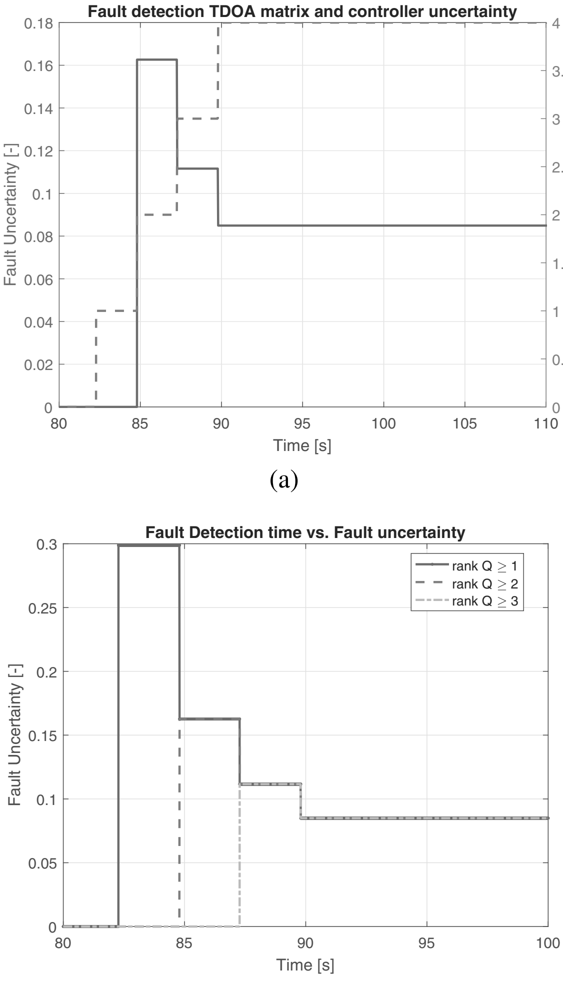


Fig. 14. FDD algorithm performance with the matrix value α for the matrix $\overline{Q}$ (a high-magnitude incipient fault).



(a)

(b)

Fig. 15. FDD algorithm outputs: fault uncertainty (a) and uncertainty sensitivity (b) (a high-magnitude incipient fault).

Åström, K.J. (1998). Control systems with friction, *Proceedings of the 4th International Conference on Motion and Vibration Control, Zurich, Switzerland*, pp. 25–32.

Brandt, J.B. and Selig, M.S. (2011). Propeller performance at low Reynolds numbers, *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, USA*, p. 1255.

Chen, W., Khan, A.Q., Abid, M. and Ding, S.X. (2011). Integrated design of observer based fault detection for a class of uncertain nonlinear systems, *International Journal of Applied Mathematics and Computer Science* **21**(3): 423–430, DOI: 10.2478/v10006-011-0031-0.

Chen, X., Wang, D., Yin, J. and Jia, C. (2019). Augmented Lagrange geolocation algorithm using TDOA measurements and calibration sources in the presence of satellite position errors, *AEU—International Journal of Electronics and Communications* **111**: 1–18.

Fan, J., Zhang, Y. and Zheng, Z. (2013). Adaptive observer-based integrated fault diagnosis and fault-tolerant control systems against actuator faults and saturation, *Journal of Dynamic Systems, Measurement, and Control* **135**(4): 041008.

Gavrilets, V. (2003). *Autonomous Aerobatic Maneuvering of Miniature Helicopters*, PhD thesis, Massachusetts Institute of Technology, Cambridge.

Gavrilets, V., Mettler, B. and Feron, E. (2004). Human-inspired control logic for automated maneuvering of miniature helicopter, *Journal of Guidance, Control, and Dynamics* **27**(5): 752–759.

Ji, H. (2021). Statistics Mahalanobis distance for incipient sensor fault detection and diagnosis, *Chemical Engineering Science* **230**: 116233.

Kantue, P. and Pedro, J.O. (2018a). Grey-box modelling of an unmanned quadcopter during aggressive maneuvers, *International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania*, pp. 640–645.

Kantue, P. and Pedro, J.O. (2018b). Nonlinear identification of an unmanned quadcopter rotor dynamics using RBF neural networks, *International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania*, pp. 292–298.

Kantue, P. and Pedro, J.O. (2019). Real-time identification of faulty systems: Development of an aerial platform with emulated rotor faults, *2019 4th Conference on Control and Fault Tolerant Systems (SysTol2019), Casablanca, Morocco,* pp. 20–25.

Kantue, P. and Pedro, J.O. (2020). Integrated fault detection and diagnosis of an unmanned aerial vehicle using time difference of arrival, *International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania*, pp. 336–342.

Krstic, M. and Wang, H.H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems, *Automatica* **36**(1): 595–601.

Lan, J. and Patton, R.J. (2016). A new strategy for integration of fault estimation within fault-tolerant control, *Automatica* **69**: 48–59.

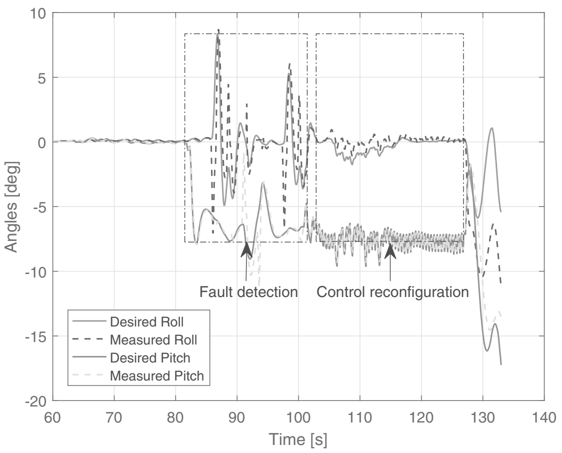


Fig. 16. Angles tracking during the fault-tolerant process.

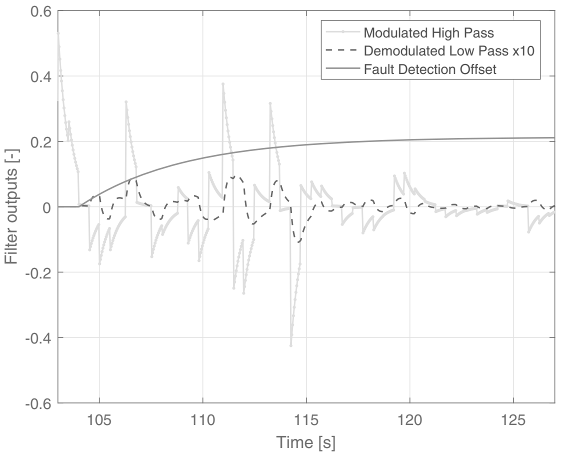


Fig. 17. RFC filter output.

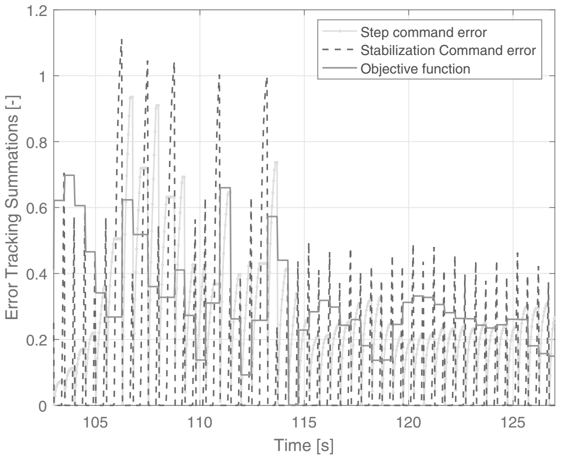


Fig. 18. RFC objective cost function.

Lan, J., Patton, R.J. and Zhu, X. (2017). Integrated fault-tolerant control for a 3-DOF helicopter with actuator faults and saturation, *IET Control Theory and Applications* **11**(14): 2232–2241.

Li, K., Peng, J.-x. and Irwin, G.W. (2005). A fast nonlinear model identification method, *IEEE Transactions on Automatic Control* **50**(8): 1211–1216.

Li, X., Lu, D., Zeng, G., Liu, J. and Zhang, W. (2017). Integrated fault estimation and non-fragile fault-tolerant control design for uncertain Takagi–Sugeno fuzzy systems with actuator fault and sensor fault, *IET Control Theory and Applications* **11**(10): 1542–1553.

Li, Y.-X. and Yang, G.-H. (2018). Adaptive integral sliding mode control fault tolerant control for a class of uncertain nonlinear systems, *IET Control Theory and Applications* **12**(13): 1864–1872.

Liu, L., Shen, Y. and Dowell, E.H. (2012). Integrated adaptive fault-tolerant H-infinity output feedback control with adaptive fault identification, *Journal of Guidance, Control, and Dynamics* **35**(3): 881–889.

Liu, L., Yu, Z. and Zhu, C. (2015). Integrated active fault-tolerant H-infinity control by designing an auxiliary self-examination plant, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015, Hong Kong, China*, pp. 234–239.

Liu, Y. and Yang, G.H. (2019). Integrated design of fault estimation and fault-tolerant control for linear multi-agent systems using relative outputs, *Neurocomputing* **329**: 468–475.

Marcos, A. and Balas, G.J. (2005). A robust integrated controller/diagnosis aircraft application, *International Journal of Robust and Nonlinear Control* **15**(12): 531–551.

Mettler, B. and Kanade, T. (2000). System identification modeling of a model-scale helicopter, *Robotics* **17**(6): 1–25.

Mien, V., Kang, H.J. and Ro, Y.S. (2011). A robust detection and isolation scheme for incipient and abrupt faults in robot manipulator using neural network, *Proceedings of the 6th International Forum on Strategic Technology, IFOST 2011, Harbin, China*, pp. 313–316.

Nett, C.N., Jacobson, C.A. and Miller, A.T. (1988). An integrated approach to controls and diagnostics: The 4-parameter controller, *1988 American Control Conference, Atlanta, USA*, pp. 824–835.

Pedro, J.O. and Kantue, P. (2011). Online aerodynamic parameter estimation of a miniature unmanned helicopter using radial basis function neural networks, *8th Asian Control Conference, Kaohsiung, Taiwan*, pp. 1170–1175.

Peng, J.-x., Li, K. and Irwin, G.W. (2007). A novel continuous forward algorithm for RBF neural modelling, *IEEE Transactions on Automatic Control* **52**(1): 117–122.

Prochazka, K.F., Eduardo, H. and Klein, S.R. (2018). Integrated fault-tolerant control of an over-actuated aircraft using optimal control allocation and robust sliding mode observers, *IEEE Conference on Control Technology and Applications, Copenhagen, Denmark*, pp. 171–178.

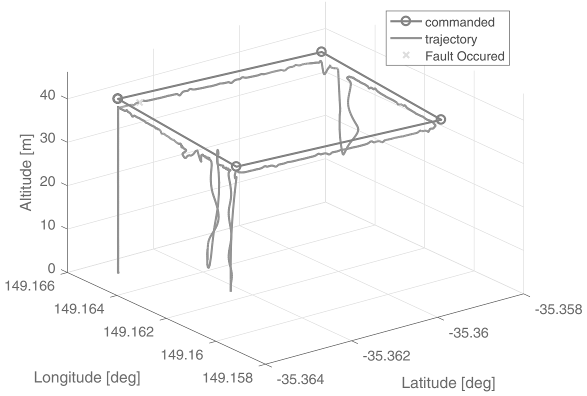


Fig. 19. Trajectory tracking: no active FTCS.

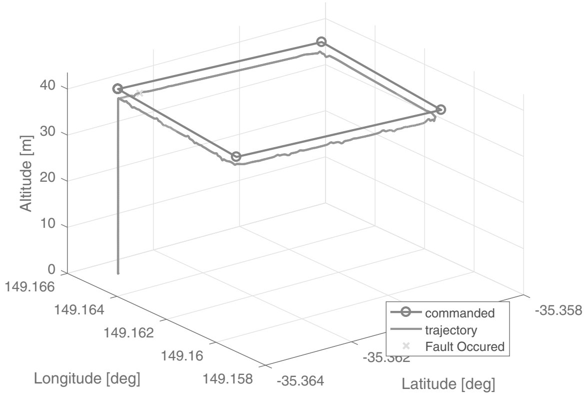


Fig. 20. Trajectory tracking: active FTCS.

Rudin, K., Ducard, G.J.J. and Siegwart, R.Y. (2020). Active fault tolerant control with imperfect fault detection information: Applications to UAVs, *IEEE Transactions on Aerospace and Electronic Systems* **56**(4): 2792–2805.

Salazar, J.C., Sanjuan, A., Nejjari, F. and Sarrate, R. (2020). Health-aware and fault-tolerant control of an octorotor UAV system based on actuator reliability, *International Journal of Applied Mathematics and Computer Science* **30**(1): 47–59, DOI: 10.34768/amcs-2020-0004.

Shen, Q., Jiang, B. and Shi, P. (2017). Adaptive fault tolerant control against actuator faults, *International Journal of Adaptive Control and Signal Processing* **31**(2): 147–162.

Shen, Q., Jiang, B., Shi, P. and Lim, C.C. (2014). Novel neural networks-based fault tolerant control scheme with fault alarm, *IEEE Transactions on Cybernetics* **44**(11): 2190–2201.

Stoustrup, J., Grimble, M. and Niemann, H. (1997). Design of integrated systems for the control and detection of actuator/sensor faults, *Sensor Reviews* **17**(2): 138–149.

Wang, L., Chen, S. and Ma, K. (2016). On stability and application of extremum seeking control without steady-state oscillation, *Automatica* **68**: 18–26.

Wu, C., Qi, J., Song, D., Qi, X. and Han, J. (2015). Simultaneous state and parameter estimation based actuator fault detection and diagnosis for an unmanned helicopter, *International Journal of Applied Mathematics and Computer Science* **25**(1): 175–187, DOI: 10.1515/amcs-2015-0013.

Xulin, L. and Yuying, G. (2018). Fault tolerant control of a quadrotor UAV using control allocation, *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018, Shenyang, China*, pp. 1818–1824.

Yan, K., Chen, M., Wu, Q. and Jiang, B. (2019). Extended state observer-based sliding mode fault-tolerant control for unmanned autonomous helicopter with wind gusts, *IET Control Theory and Applications* **13**(10): 1500–1513.

Yang, J. and Delpha, C. (2022). An incipient fault diagnosis methodology using local Mahalanobis distance: Detection process based on empirical probability density estimation, *Signal Processing* **190**: 1–13.

Yang, Y., Niu, Y. and Zhang, Z. (2020). Adaptive fuzzy fault-tolerant control for nonlinear systems under actuator and sensor faults: The practical fixed-time stability, *IET Control Theory and Applications* **14**(19): 3291–3300.

Yin, C., Dadras, S., Huang, X., Cheng, Y. and Malek, H. (2018). The design and performance analysis of multivariate fractional-order gradient-based extremum seeking approach, *Applied Mathematical Modelling* **62**: 680–700.

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

Yu, X., Liu, Z. and Zhang, Y. (2017). Fault-tolerant flight control design with finite-time adaptation under actuator stuck failures, *IEEE Transactions on Control Systems Technology* **25**(4): 1431–1440.

Zhang, Y., Chamseddine, A., Rabbath, C.A., Gordon, B.W., Su, C.Y., Rakheja, S., Fulford, C., Apkarian, J. and Gosselin, P. (2013). Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed, *Journal of the Franklin Institute* **350**(9): 2396–2422.

Zhang, Y. and Jiang, J. (1999). Design of integrated fault detection, diagnosis and reconfigurable control systems, *Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, USA*, Vol. 4, pp. 3587–3592.

Zhang, Y. and Jiang, J. (2001). Integrated active fault-tolerant control using IMM approach, *IEEE Transactions on Aerospace and Electronic Systems* **37**(4): 1221–1235.

Paulin Kantue received his BSc and MSc degrees in aeronautical engineering at the University of the Witwatersrand, Johannesburg, in 2007 and 2011, respectively. He is currently a PhD candidate in aeronautical engineering from the University of the Witwatersrand. He is a co-founder and the head of research for an unmanned systems engineering company called Uav4africa. He also works as a senior flight control laws engineer at Lilium GmbH, Munich, Germany. His research interests include application flight control techniques for unmanned systems, nonlinear system identification and active fault tolerant control.

Jimoh Olarewaju Pedro received his MSc and PhD degrees in aeronautical engineering from the Warsaw University of Technology in 1986 and 1992, respectively. He was a post-doctoral associate at the Institute of Aviation in Warsaw in 1993–1994. He is currently a visiting associate professor at the School of Mechanical, Industrial and Aeronautical Engineering, University of the Witwatersrand, Johannesburg, South Africa, after retiring in December 2021. He has published over 90 articles in refereed journals and conference proceedings. His research interests include applications of soft computing techniques to mechatronic systems and aerospace vehicles, trajectory optimization, and applications of robust control theory to rigid/flexible aerospace vehicles.

Received: 23 October 2021
Revised: 31 March 2022
Accepted: 4 May 2022