Journal of Intelligent and Robotic Systems manuscript No.
(will be inserted by the editor)

An Intelligent Hybrid Artificial Neural Network-Based Approach for Control of Aerial Robots

Siddharth Patel · Andriy Sarabakha · Dogan Kircali · Erdal Kayacan

Received: date / Accepted: date

Abstract In this work, a learning model-free control method is proposed for accurate trajectory tracking and safe landing of unmanned aerial vehicles (UAVs). A realistic scenario is considered where the UAV commutes between stations at high-speeds, experiences a single motor failure while surveying an area, and thus requires to land safely at a designated secure location. The proposed challenge is viewed solely as a control problem. A hybrid control architecture – an artificial neural network (ANN)-assisted proportional-derivative controller – is able to learn the system dynamics online and compensate for the error generated during different phases of the considered scenario: fast and agile flight, motor failure, and safe landing. Firstly, it deals with unmodelled dynamics and operational uncertainties and demonstrates superior performance compared to a conventional proportional-integral-derivative controller during fast and agile flight. Secondly, it behaves as a fault-tolerant controller for a single motor failure case in a coaxial hexacopter thanks to its proposed sliding mode control theory-based learning architecture. Lastly, it yields reliable performance for a safe landing at a secure location in case of an emergency condition. The tuning of weights is not required as the structure of the ANN controller starts to learn online, each time it is initialised, even when the scenario changes – thus, making it completely model-free. Moreover, the simplicity of the neural network-based controller allows for the implementation on a low-cost low-power onboard computer. Overall, the real-time experiments show that the proposed controller outperforms the conventional controller.

Siddharth Patel · Dogan Kircali
School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU), 50 Nanyang Avenue, Singapore, 639798
E-mail: PATE0006@e.ntu.edu.sg, dkircali@ntu.edu.sg

Andriy Sarabakha
School of Mechanical and Aerospace Engineering (MAE), Nanyang Technological University (NTU), 50 Nanyang Avenue, Singapore, 639798
E-mail: andriy001@e.ntu.edu.sg

Erdal Kayacan
Department of Engineering, Aarhus University, Aabogade 34, Aarhus, 8000, Denmark
E-mail: erdal@eng.au.dk

Keywords Artificial neural networks · Sliding mode control · Unmanned aerial vehicles · Fault tolerant control · Fast and agile manoeuvres

1 Introduction

Nowadays, the use of UAVs by amateurs with minimal piloting skills, skilled hobbyists, and licensed pilots for purposes of aerial photography, drone racing, and hobby, continues to grow [1]. There are two major types of users which lead to the innovation and development of cutting-edge technology in this sector – military and consumer industry. The former is mainly interested in long-range endurance missions, and – thus, usually prefers fixed-wing UAVs, while the latter leans more toward the vertical take-off and landing (VTOL) capable multicopter UAVs. This modern-day marvel is improving progressively and it is becoming accessible to most of the people. Furthermore, besides leisure activities, these UAVs have also started to assume an important role in a wide range of applications like – transportation [2], surveying and mapping [3], and search and rescue [4].

The above-mentioned applications are possible with the development of sophisticated flight control systems. Generally, these controllers are used for the control of multicopter UAVs and they can be classified into two categories – model-based and model-free. The model-based controller requires an accurate knowledge of the system model. The conventional model-based controllers, like a proportional-integral-derivative (PID) controller, are commonly used in the industry, and their modified variants, like nonlinear PID control systems, are also proposed [5]. Other advanced model-based controllers, like the dynamic feedback linearized controller [6], and the model predictive controller [7], also provide necessary performance but their benefits start to diminish in certain situations with uncertainties and nonlinearities. Moreover, since they are model-based, a slight change in the system configuration (internal or external uncertainty) can lead to a significant degradation in the performance of the controller. Moreover, tuning the parameters for different scenarios is a time-consuming process.

Typical operations of VTOL aircraft are generally slow and non-aggressive, where the UAV operates at near-hover conditions which do not pose a significant challenge for their control. The applications mentioned above are facile if accomplished with the aid of advanced flight control systems. This may include making fast decisions and movements for collision avoidance, during high-speed flight, or during a motor failure scenario. In such cases, the controller has to react instantaneously given the time-critical state of the situation. The widespread use of autonomous vehicles requires high standards as the UAV reaches the boundaries of performance. Furthermore, safety and reliability become crucial to the successful mission execution. This calls for more complex and reliable control approaches to oversee such aggressive manoeuvres, maintain stability, and ensure safety and reliability of the UAV. Model-free control algorithms are more generic approaches to the control problems since they learn the system dynamics online. These intelligent controllers are well versed and their uses in the UAVs are studied by researchers all around the world [8–10].

For the real-world deployment of multicopters, the robustness to an actuator failure is one of the fundamental safety requirement. Given the increasing use of UAVs, some accidents due to sensor/actuator failure are inevitable. Therefore, the

development of fault-tolerant controllers (FTCs), that can manage the actuator failure, is becoming more demanding. The aim is to minimize the impact on the system failure from such faults [11]. In the literature, two alternatives exist to manage motor failure: active and passive [12]. A nonlinear model predictive control approach is exhibited in [13] as an active fault-tolerant control method. In the active FTC approach, fault detection and identification are crucial[14]. Faults can be detected using the inertial measurements [15], or by generating residuals [16], among others. Another way of actuator/sensor fault detection and diagnosis is shown in [17], as the authors use different Kalman filter techniques along with numerous control approaches like – PID control, gain-scheduled PID control, fuzzy gain-scheduled PID control, versions of model reference adaptive control – as fault-tolerant control methods. A drawback of active FTC is a possible deterioration in performance due to an unexpected behaviour because of the failure [18].

The passive FTC approach requires neither fault detection nor controller reconfiguration. These controllers are designed to be robust against a class of presumed faults [19]. A strategy used in [20] is to let the quadcopter yaw freely about an axis and control the roll, pitch, and translation by tilting this axis with respect to the body. In this particular case, authors investigated lose of one, two, and even three propellers while still maintaining position in space. Thus, passive FTC approach can be exploited with the use of an adaptive controller which "understands" that the system behaviour has changed and takes necessary actions. The use of intelligent controllers increases the reliability of the UAV in case of faults. The use of learning-based controller like neural networks, that are model-free by definition, helps to develop such adaptive controllers [21].

Similar to any other robotics application, a trade-off between robustness and performance exists in a UAV application where researchers mostly prefer safe and robust controller tuning, when dealing with more aggressive and accurate controllers. The operation of UAV near the boundaries has always been an alluring research topic in the literature, as shown in [22], that multi-flips are performed using a simple learning strategy and the first-principles model. A manoeuvre regulation perspective follows a geometric path with a certain velocity in [23]. Since the path to be followed is not a time-based reference state, unlike in trajectory tracking, a linear quadratic regulator-based controller ensures the exact path-following to perform such space-dependent manoeuvres. As an alternative solution, the aerodynamic effects of blade flapping and thrust variations on a rotor at higher angle of attacks are studied and are used to develop control techniques for operations in high-speed aggressive manoeuvres [24]. The authors present a novel feedback linearization controller to take into account such aerodynamic disturbances. A hardware solution to the mentioned problem, used in [25], is the implementation of the variable pitch rotors. This method expands the normal rotor operation regime by varying the blade pitch which provides better thrust vectoring to achieve agile UAV manoeuvres.

Feasible aggressive trajectories emulating constrained indoor environment are designed in [26, 27]. The algorithm generates trajectories in real-time to ensure safe passage through corridors and satisfy constraints on velocities and accelerations. A nonlinear controller based on the changing dynamics of the UAV and errors in the model is developed which ensures the tracking of desired states in the three-dimensional space. A simple model describing the essential dynamics of the system is used in an iterative learning algorithm to perform an aggressive motion [28]. The

knowledge obtained from the successful trajectories is used to reduce the transients when performing similar subsequent manoeuvres.

The results in [29] shows that a similar adaptive fuzzy-neuro system designed to handle large number of uncertainties in the operation of the system. The applications involving fast and agile flights present a challenge to the use of conventional controllers. Thus, this work presents an ANN-based controller implemented for FTC to tackle the single motor failure case. This approach is trained by sliding mode control (SMC) theory-based learning algorithm which works in parallel with a simple conventional proportional-derivative (PD) controller. Apart from using the ANN controller as FTC, it is also used in the trajectory tracking problem of the UAV in case of fast and agile flight manoeuvres. Other than the literature above, researchers analyze the planning and perception aspects to enable agile navigation using onboard sensors such as stereo cameras [30], a single camera and IMU [31]. However, our work focuses on the control part, as mentioned earlier, to cope with the uncertainties from the unmodelled dynamics of the system and external effects. We show that an accurate trajectory tracking for agile flights is achieved in the outdoor environment. Moreover, the proposed ANN structure is computationally light to be implemented on a low-cost onboard computer.

In this work, we show a unique implementation of an ANN-assisted control method, which enables both fast flight and agile manoeuvres, allows us to obtain accurate trajectory tracking results without compromising the robustness and safety in the system, in such varied scenarios. The experimental scenario is designed to verify the performances of an ANN-assisted control method under various challenging operational conditions, such as, fast flight, agile manoeuvres and motor failure. The UAV is launched from the ground station for the surveillance of a distant area. Then, it commutes between the base and destination with a high-speed flight, reaches the surveillance area, starts the tracking of the agile section of the trajectory, experiences a single motor failure during the inspection, and lands safely at an identified secure location. A safe landing is as important as the mission for the safety of the UAV, people, and property, or safeguarding any data which was recorded during flight. The UAV is compelled to land at a safe site in emergency situations like the motor failure scenario discussed before. The ANN quickly adapts to the changing conditions and performs better than conventional controllers. This is a major advantage of this approach since being model-free, there is no need for the remapping of control allocation matrix. The chosen scenario is not arbitrary in the sense that the possibility of having an actuator failure, while following such agile and challenging paths, is high in UAVs. The onboard implementation of the approach on the "single-board low-cost computers" also opens the doors to autonomous flights. Another advantage of this work is that all the experiments were conducted in the outdoor environment using the real-time kinematic global positioning system (RTK-GPS), which can be used in real-world situations.

The contributions of this study can be summarized as follows:

- an ANN-assisted control method is applied to a challenging scenario with fast flight, agile manoeuvres and motor failure;
- a comparison of the ANN with traditional model-based controllers is carried out showing the superiority of the proposed learning control approach;

– the real-time applicability of the proposed learning control method is elaborated using an on-board computer on an aerial robot.

This work is structured as follows: Section 2 gives a brief description of the dynamical model of the coaxial hexacopter. In Section 3, an overview of the proposed control scheme is presented. In Section 4, the results of the experiments are described. The work is concluded in Section 5.

2 System model

The platform used is a customised Y6 coaxial hexacopter with six rotors attached to three arms. The differential thrust produced by each rotor is used for basic control of the hexacopter. The upper and lower rotors rotate clockwise and counter-clockwise, respectively. All the rotors produce lift with the thrust vector in the vertical direction, which in turn translates to roll and pitch motions by tilting the thrust vector. Figure 1 shows the considered reference frames and sign conventions for the forces (F_i), torques (τ_i) and rotational speeds (Ω_i) of the rotors, where i denotes the rotor number. A brief description of the dynamics and kinematics of the coaxial hexacopter, as in [32], is presented here.

The world-fixed inertial frame or Earth frame is $\mathcal{F}_E = \{\mathbf{x}_E, \mathbf{y}_E, \mathbf{z}_E\}$ and the body-fixed frame is $\mathcal{F}_B = \{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$. The dynamics of the system are defined by assuming the UAV as a rigid body with origin at the centre of gravity of the UAV. The rolling (τ_p), pitching (τ_q), and yawing (τ_r) moments and total thrust (T), as shown in the figure, are produced by the six rotors at any given time. Considering the dynamics, the system is underactuated and there are four control inputs ($T,\ \tau_p,\ \tau_q,\ \tau_r$) which are expressed as:

$$T = F_1 + F_2 + F_3 + F_4 + F_5 + F_6. \tag{1}$$

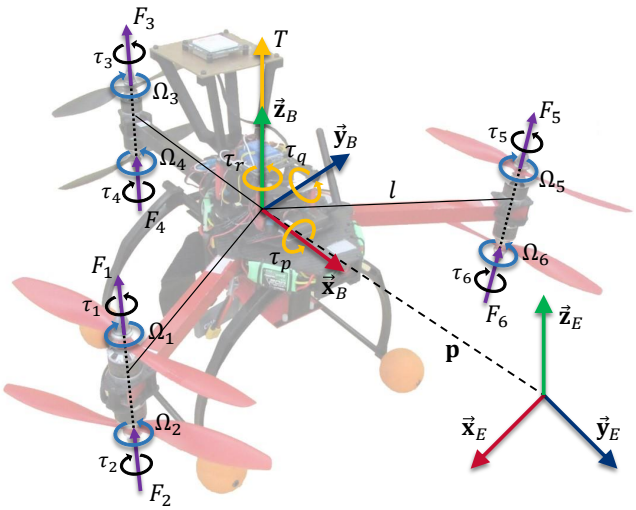

Fig. 1 Coordinate reference frame and sign conventions

The moments acting on the UAV with moment arms $\left(l,\ \frac{\sqrt{3}}{2}l,\ \frac{1}{2}l\right)$ are:

$$\begin{bmatrix}\tau_p\\ \tau_q\\ \tau_r\end{bmatrix} = \begin{bmatrix}(F_5+F_6-F_1-F_2)\frac{\sqrt{3}}{2}l\\ (F_3+F_4)l-(F_1+F_2+F_5+F_6)\frac{1}{2}l\\ \tau_1+\tau_3+\tau_5-\tau_2-\tau_4-\tau_6\end{bmatrix}. \tag{2}$$

The position and orientation of the UAV are defined by vectors, $\begin{bmatrix}x\ y\ z\end{bmatrix}^T \in \mathbb{R}^3$ and $\begin{bmatrix}\phi\ \theta\ \psi\end{bmatrix}^T \in \mathbb{R}^3$ in $\mathcal{F}_E$. The time derivatives of these state vectors, $\mathbf{v} = \begin{bmatrix}\dot{x}\ \dot{y}\ \dot{z}\end{bmatrix}^T$ and $\omega = \begin{bmatrix}\dot{\phi}\ \dot{\theta}\ \dot{\psi}\end{bmatrix}^T$, gives the translational and rotational kinematic equations, which are described as:

$$\begin{cases}\mathbf{v} &= \mathbf{R}\mathbf{v}_B\\ \omega &= \mathbf{T}\omega_B\end{cases}, \tag{3}$$

where $\mathbf{v}_B$ and ω_B are linear and angular velocities given by vectors $\begin{bmatrix}u\ v\ w\end{bmatrix}^T \in \mathbb{R}^3$ and $\begin{bmatrix}p\ q\ r\end{bmatrix}^T \in \mathbb{R}^3$ in $\mathcal{F}_B$, respectively. Moreover, $\mathbf{R}$ and $\mathbf{T}$ are the rotation matrices [33], for transformation from $\mathcal{F}_E$ to $\mathcal{F}_B$ using the Z –Y –X Euler angle rotation sequence, given by:

$$\mathbf{R} = \begin{bmatrix}c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi\\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi\\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta\end{bmatrix}, \tag{4}$$

$$\mathbf{T} = \begin{bmatrix}1 & s_\phi t_\theta & c_\phi t_\theta\\ 0 & c_\phi & -s_\phi\\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta}\end{bmatrix}, \tag{5}$$

where $s_\square$, $c_\square$, and $t_\square$ are $\sin\square$, $\cos\square$, and $\tan\square$, respectively. Finally, the kinematic equations in (3) can be framed as:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \mathbf{R}\begin{bmatrix}u\\ v\\ w\end{bmatrix}, \quad \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \mathbf{T}\begin{bmatrix}p\\ q\\ r\end{bmatrix}. \tag{6}$$

The rigid body dynamic equations are derived using the Newton-Euler formulation [34] in the body frame and translate to the final form as:

$$\begin{cases}m\mathbf{I}_3\dot{\mathbf{v}}_B &= F-(\omega_B\times m\mathbf{v}_B)\\ \mathbf{I}\dot{\omega}_B &= \tau-(\omega_B\times\mathbf{I}\,\omega_B)\end{cases}, \tag{7}$$

where m is mass, $\mathbf{I}_3$ is a 3×3 Identity matrix and $\mathbf{I}$ is the inertia matrix.

External force F and torque τ are expressed as:

$$\begin{cases}F = F_{total} - \begin{bmatrix}-mg\sin\theta\\ mg\cos\theta\sin\phi\\ mg\cos\theta\cos\phi\end{bmatrix}\\ \tau = \begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix}\end{cases}, \tag{8}$$

where $F_{total} = \begin{bmatrix} 0 & 0 & T \end{bmatrix}^T$. Finally, the dynamic equations in (7) are derived as:

$$\begin{cases} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} F - \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}^{-1} \tau - \begin{bmatrix} \frac{I_z - I_y}{I_x} qr \\ \frac{I_x - I_z}{I_y} pr \\ \frac{I_y - I_x}{I_z} qr \end{bmatrix} \end{cases} . \tag{9}$$

Thus, the kinematic and dynamic differential equations in (6) and (9) describe the general mathematical model of the hexacopter. The forces and reaction torques exerted by the rotors, rotating at Ω_i angular velocities, can be formulated as:

$$\begin{cases} F_i &= K_F \Omega_i^2 \\ \tau_i &= K_\tau \Omega_i^2 \end{cases} , \tag{10}$$

where K_F and K_τ are termed as force and torque coefficients and are modeled based on the motor-propeller combination.

Equations (1) and (2) can be augmented together and expressed in the form:

$$\boldsymbol{\nu} = \begin{bmatrix} T & \tau_p & \tau_q & \tau_r \end{bmatrix}^T = \mathbf{B}\mathbf{u}, \tag{11}$$

where $\mathbf{u} = \begin{bmatrix} \Omega_1^2 & \Omega_2^2 & \Omega_3^2 & \Omega_4^2 & \Omega_5^2 & \Omega_6^2 \end{bmatrix}^T$ is a column vector of actuator inputs (six rotors) and $\mathbf{B} \in \mathbb{R}^{4 \times 6}$ is the control allocation matrix given by:

$$\mathbf{B} = \begin{bmatrix} K_F & K_F & K_F & K_F & K_F & K_F \\ -K_F \frac{\sqrt{3}}{2} l & -K_F \frac{\sqrt{3}}{2} l & 0 & 0 & K_F \frac{\sqrt{3}}{2} l & K_F \frac{\sqrt{3}}{2} l \\ -K_F \frac{1}{2} l & -K_F \frac{1}{2} l & K_F l & K_F l & -K_F \frac{1}{2} l & -K_F \frac{1}{2} l \\ K_\tau & -K_\tau & K_\tau & -K_\tau & K_\tau & -K_\tau \end{bmatrix} . \tag{12}$$

As seen in (12), the control allocation matrix $\mathbf{B}$ is not a square matrix for a (coaxial) hexacopter, since it has four control inputs and six rotors. However, the pseudo-inverse $\mathbf{B}^+$ exists for any matrix $\mathbf{B}$, if it has full rank [35]. In particular, $\mathbf{B}^+$ can be computed as:

$$\mathbf{B}^+ = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T. \tag{13}$$

In our case, $\mathbf{B}$ will have full rank if and only if all of the following conditions will verify: $l \neq 0$, $K_F \neq 0$ and $K_\tau \neq 0$. The arm length, force and moment coefficients of UAV are physical quantities of the hexacopter and they cannot be equal to 0. Therefore, $\mathbf{B}$ will always have full rank and a pseudo-inverse of the matrix $\mathbf{B}$ exists:

$$\mathbf{B}^{+} = \frac{1}{6}\begin{bmatrix} \frac{1}{K_F} & \frac{-\sqrt{3}}{K_F l} & \frac{-1}{K_F l} & \frac{1}{K_\tau} \\ \frac{1}{K_F} & \frac{-\sqrt{3}}{K_F l} & \frac{-1}{K_F l} & \frac{-1}{K_\tau} \\ \frac{1}{K_F} & 0 & \frac{2}{K_F l} & \frac{1}{K_\tau} \\ \frac{1}{K_F} & 0 & \frac{2}{K_F l} & \frac{-1}{K_\tau} \\ \frac{1}{K_F} & \frac{\sqrt{3}}{K_F l} & \frac{-1}{K_F l} & \frac{1}{K_\tau} \\ \frac{1}{K_F} & \frac{\sqrt{3}}{K_F l} & \frac{-1}{K_F l} & \frac{-1}{K_\tau} \end{bmatrix}. \tag{14}$$

Remark: The dynamic equations derived are coupled, nonlinear, and the system to be controlled is underactuated.

3 Control architecture

The concept of neural network-based control is to train a network of neurons to be able to mimic the actions of a system, just like a human brain [36]. The neural networks are being used for the control of UAVs for over a decade. Recent development in neural network-based adaptive flight control may be applied to control a UAV where the reference commands can be the position, velocity, and attitude [37]. Traditionally, the process to train a neural network consists of four steps. Firstly, the training data is collected during the manual flight of the UAV. Then the neural network is trained using this collected data. The third step is to get the parameters from the trained network for the use in real-time flight. Finally, the neural network can use the onboard sensor data and control the flight. These steps are first carried out in simulation before trying on a real-time flight. In contrast to this approach, the neural network used in this work can learn online from scratch and adapt to different situations – thus, minimizing the model error, and and the stability of the system is ensured using a conventional PD controller. The block diagram showing the real-time implementation giving the overall view of the system integration is shown in Fig. 2.

3.1 Artificial neural networks

ANNs are highly regarded for their learning ability from input-output data. An inter-connected structure of neurons receives the input, processes it, and generates an output depending on the input and internal state. In a general ANN structure, the neurons are linked together, as shown in Fig. 3. The neural network is organized into an input layer, a hidden layer, and an output layer. To the output of each neuron is given a weight (v_i or w_i), which is updated in the learning process by a set of learning rules. The evaluation of the distance from the sliding-mode manifold determines these weights.

The ANN structure used for the design of the controller in this work has two input neurons ($n_1 = 2$), one output neuron ($n_3 = 1$), and nine neurons ($n_2 = 9$) in the hidden layer. The hidden layer defines the learning capabilities of the ANN. Typically, a large number of neurons in the hidden layer should be able to provide better convergence values to give desired outputs [38]. However, a smaller number of neurons may result in better generalisation in terms of different

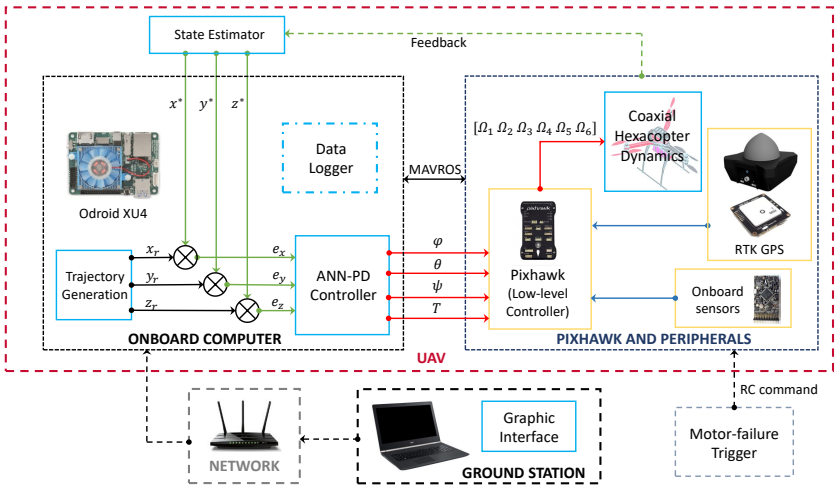

Fig. 2 Overall control architecture of the controller for the UAV

situations observed during tests, so like most engineering problems, a trade-off exists and an optimal number of neurons are selected. The neural networks with a single hidden layer are universal approximators [39], i.e., provided sufficient neurons are available, the network can be trained online to learn any measurable function and is capable of reducing model error [37]. The position error e and its time derivative $\dot{e}$ are given to the developed ANN controller as two inputs, i.e., $\mathbf{x} = [e \quad \dot{e}]^T$. The output control signal from ANN is computed as a linear combination of each input:

$$u_{\text{ANN}} = \frac{\sum_{i=1}^{n_2} h_i w_i}{\sum_{i=1}^{n_2} h_i} = \sum_{i=1}^{n_2} \bar{h}_i w_i, \tag{15}$$

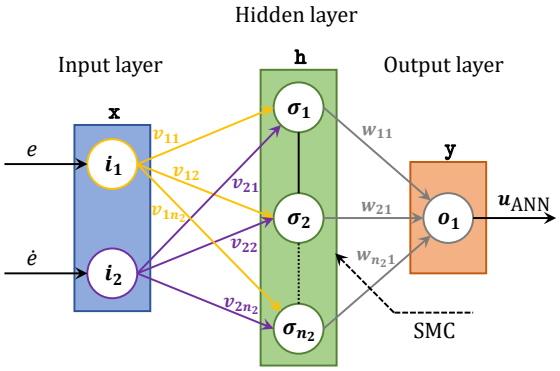

Fig. 3 Structure of a three-layer artificial neural network

where $\bar{h}_i$ is the normalized value of the output from the i^{th} neuron in the hidden layer:

$$\bar{h}_i = \frac{h_i}{\sum_{j=1}^{n_2} h_j} \quad \forall i \in [1, n_2], \tag{16}$$

and,

$$h_i = \sigma\left(\sum_{j=1}^{n_1} x_j v_{ji}\right) \quad \forall i \in [1, n_2], \tag{17}$$

where $k = 1, 2$ and $\sigma(\square)$ is a scalar activation function.

In the proposed control scheme, the ANN works in parallel with a conventional PD controller as shown in Fig. 4. The PD controller ensures the stability of the system in the initial phase of learning process and acts as a feedback part of the controller providing sufficient time for ANN to initialize its learning process [21]. Thus, ANN will learn the system dynamics (UAV dynamics, in this case) and take over the control of the system. With its adaptive learning rates, ANN is very fast to learn and can instantaneously contribute to better performance, i.e., trajectory tracking accuracy, in our case. The control signal u given to the system is calculated as follows:

$$u = u_{\text{PD}} - u_{\text{ANN}}, \tag{18}$$

where u_{PD} and u_{ANN} are the control signals generated by PD and ANN controllers, respectively. The general PD control law is described as follows:

$$u_{\text{PD}} = k_p e + k_d \dot{e}, \tag{19}$$

where e and $\dot{e}$ are the position feedback error and its time derivative, respectively; k_p and k_d are termed as proportional and derivative gains, respectively, which are some positive constants. The ANN controller generates the control signal u_{ANN}, i.e., $y = u_{\text{ANN}}$, as output.

According to the control scheme in Fig. 4, it is assumed that the two input signals, $e(t)$ and $\dot{e}(t)$, and their respective time derivatives, $\dot{e}(t)$ and $\ddot{e}(t)$, will never reach infinite values [40]. Hence, they are bounded by finite real constants as follows:

$$\begin{cases} |e(t)| \leq B_e \\ |\dot{e}(t)| \leq B_{\dot{e}} \\ |\ddot{e}(t)| \leq B_{\ddot{e}} \end{cases} \quad \forall t, \tag{20}$$

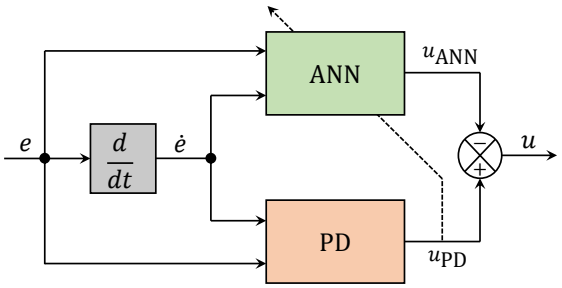

Fig. 4 Control scheme: ANN in parallel with PD controller

where $B_e > 0$, $B_{\dot{e}} > 0$ and $B_{\ddot{e}} > 0$ are some real constants. Similarly, the weight coefficient can be considered bounded such that they minimize the errors, i.e.:

$$\begin{cases} |W| & \leq B_W \\ |V| & \leq B_V \end{cases}, \tag{21}$$

where $B_W > 0$ and $B_V > 0$ are some real constant. From (20) and (21), it is evident that $u_{\mathrm{ANN}}(t)$ and $\dot{u}_{\mathrm{ANN}}(t)$ are also bounded signals:

$$\begin{cases} |u_{\mathrm{ANN}}(t)| \leq B_u \\ |\dot{u}_{\mathrm{ANN}}(t)| \leq B_{\dot{u}} \end{cases} \quad \forall t, \tag{22}$$

where $B_u > 0$ and $B_{\dot{u}} > 0$ are some real constants.

3.2 Sliding mode control theory-based training algorithm

An SMC parameter adaptation scheme is used for the learning process for ANN. The SMC framework is designed by selecting a suitable sliding manifold that will ensure desired system dynamics. Moreover, to fulfill the sliding mode constraints/conditions, a dynamic feedback adaptation mechanism or an online learning algorithm for ANN parameters has to be designed. The SMC provides robustness to parameter uncertainties and external disturbances – thus, is a widely used control method for nonlinear systems applications.

The difference between the measured output of the system and the output of the ANN can be defined as a time-varying sliding surface. It is guaranteed that the system will be on the sliding surface, under certain conditions [41].

A time-varying sliding surface S_{PD} can describe the zero value of the learning error coordinate $u_{\mathrm{PD}}(t)$ by using the theory of SMC [42]:

$$S_{\mathrm{PD}}(u_{\mathrm{ANN}}, u) = u_{\mathrm{PD}}(t) = u_{\mathrm{ANN}}(t) + u(t) = 0. \tag{23}$$

Using the condition in (23), the ANN is trained to obtain the desired response such that it becomes a nonlinear regulator that assists the conventional PD controller. Thus, the sliding surface for the nonlinear system under control is [43]:

$$S(e, \dot{e}) = \dot{e} + \lambda e, \tag{24}$$

where λ is a positive constant parameter which determines the slope of the sliding surface.

A sliding motion will occur on the sliding manifold $S_{\mathrm{PD}}(u_{\mathrm{ANN}}, u) = u_{\mathrm{PD}}(t) = 0$ after a finite time t_h, if the condition $S_{\mathrm{PD}}(t)\dot{S}_{\mathrm{PD}}(t) = u_{\mathrm{PD}}(t)\dot{u}_{\mathrm{PD}}(t) < 0$ is satisfied for all t such that $[t, t_h) \subset (-\infty, t_h)$ in some nontrivial semi-open subinterval of time [44].

The use of sign() function guarantees the convergence of the sliding surface to zero in finite time. Consequently, $u_{\mathrm{ANN}}(t)$ is constrained to perfectly follow the desired output signal $u(t)$ for all $t > t_h$. The time instant t_h is the hitting time for the learning error $\tau(t) = 0$. For an arbitrary initial condition $\tau(0)$, $\tau(t)$ will eventually converge to a small neighbourhood of zero during a finite time t_h.

The adaptation laws for the parameters of the ANN are given as follows:

$$\begin{cases} \dot{w}_i & = -\alpha \dfrac{\sum_{j=1}^{n_1} x_j}{n_2|\mathbf{x}|} \text{sign}(u_{\text{PD}}) \\ \dot{\alpha} & = 2\gamma |u_{\text{PD}}| \end{cases} \quad \forall i \in [1, n_2], \tag{25}$$

where $\gamma > 0$ is a sufficiently large positive learning rate which satisfies the condition $\gamma > B_{\dot{u}}$. The pseudo-code of the neural network training is presented in Algorithm 1. The reader can refer to [45] for the stability proof of the proposed learning algorithm.

Remark: In the adaptation laws in (25), the learning rate α is variable and its value evolves during the learning process. This adaptation law allows choosing a small initial value for α which, consequently, grows during the training phase.

4 Experimental results

The experimental scenario is designed to verify the performances of an ANN-assisted control method under various challenging operational conditions, such as, fast flight, agile manoeuvres and motor failure. The real-time tests are conducted with a coaxial hexacopter, shown in Fig. 5, to validate the performance of the proposed controller for the different phases in the described scenario. Extensive simulation studies are carried out to choose from different trajectories and to test the controller. The `Gazebo` simulation environment is an open source tool used to create this realistic scenario of flying UAV with motor failure in the simulation. Due to its "robust physics engine" and high-graphics robot simulation ability, it is an ideal choice for most of the researchers in the robotics community. One of the

Algorithm 1: Online training of ANN.

Input: e, $\dot{e}$, u_{PD}
Output: u_{ANN}
Data: n_2, γ
Result: ANN learns the system dynamics and controls the system online
begin
  Get n_2 and γ
  ANN $\leftarrow$ ConstructNetworkLayers(2, n_2, 1)
  $w \leftarrow$ InitializeWeights()
  $\alpha \leftarrow \alpha_0$
  repeat
    Get e, $\dot{e}$ and u_{PD}
    $\mathbf{x} \leftarrow [e, \dot{e}]$
    $\dot{w}_i \leftarrow -\alpha \dfrac{\sum_{j=1}^{2} x_j}{n_2|\mathbf{x}|} \text{sign}(u_{\text{PD}}) \quad \forall i \in [1, n_2]$ by using (25)
    $\dot{\alpha} \leftarrow 2\gamma |u_{\text{PD}}|$ by using (25)
    $h_i = \sigma\left(\sum_{j=1}^{2} x_j v_{ji}\right) \quad \forall i \in [1, n_2]$ by using (17)
    $\bar{h}_i \leftarrow \frac{h_i}{\sum_{j=1}^{n_2} h_j} \quad \forall i \in [1, n_2]$ by using (16)
    $u_{\text{ANN}} \leftarrow \sum_{i=1}^{n_2} \bar{h}_i w_i$ by using (15)
    Send u_{ANN} to the system
  until *Stop*
end

milestones of this paper is the development of the `gazebo_motor_failure_plugin` in the `Gazebo` simulator. The plugin fails the motor of the UAV model in-flight using the robot operating system (`ROS`)/`Gazebo` supported commands. This plugin helped us to contribute to the `sitl_gazebo` repository of the open source Autopilot stack group – PX4 Pro Drone Autopilot, on GitHub: `https://github.com/PX4/sitl_gazebo`.

The Odroid-XU4 is used as the low-cost and low-power onboard computer. All the codes are executed on this computer and are written in C++ with the commands sent over the `ROS` network – thus, making the system autonomous. The RTK GPS is used to provide the real-time UAV position: x, y and z coordinates with an accuracy of 20cm at 5Hz. The position information together with the data from the inertial measurement unit is fed into the local position estimator which estimates the pose of UAV. This information is used by the controller to compute the control signal and provide it to the UAV. A 5GHz wireless network is used to communicate with the UAV. An additional motor failure circuitry is added on the UAV consisting of a relay, which helps to trigger the motor failure on-demand from the radio transmitter. The manual as well as automated failure signals can be sent to the relay, via-flight controller, to fail the motor in-flight.

Distinctive trajectories are defined for the outdoor tests based on the requirements of the application defined earlier. The tracking performance is determined for each type of controller. The results are compared with the widely known position controller of the autopilot stack – Pixhawk [46,47] – (referred as PID_{FCU}) and a conventional PID position controller (referred as PID_{pos}) sending attitude-setpoints – roll, pitch, and yaw angles – and thrust commands. Note that, the gains used for the PID controller are the same for all the scenarios. A long straight manoeuvre is chosen to track the motion at high-speeds reaching 20m/s for the fast flight of the UAV. Then the UAV starts the mapping trajectory with 3m-by-5m dimensions, after reaching the destination area, at a speed of 2m/s. During this part of the trajectory, the UAV experiences a motor failure and, yet, continues to complete the trajectory and land safely at the end. It is to be noted that the experiments were conducted with average wind gusts of 5m/s.

Remark: The goal here is to achieve a greater performance using the proposed controller in challenging and previously unknown trajectories, for which a perfectly tuned set of PID gains are not determined. The ANN-assisted conservative PD

Fig. 5 Experimental setup: coaxial hexacopter

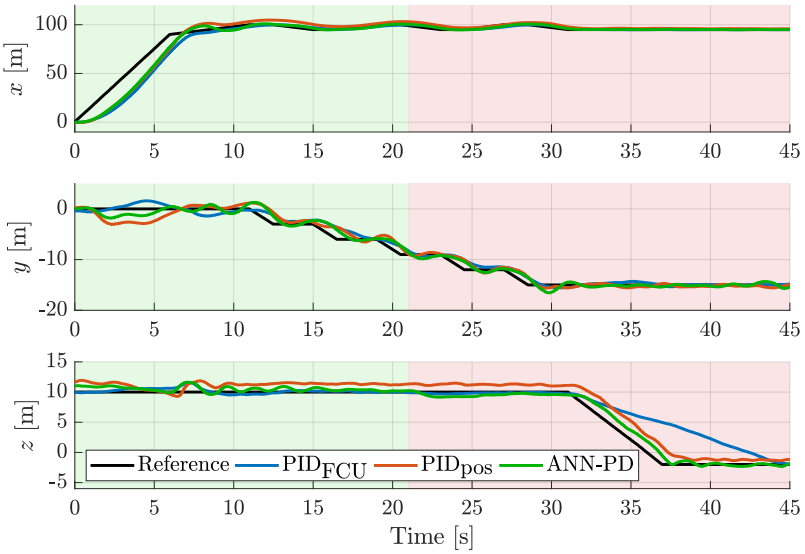

Fig. 6 Position tracking performance of various controllers for an experimental scenario with fast flight, agile manoeuvres and motor failure (green-shaded portion indicates normal operational conditions, while red-shaded portion indicates the motor failure)

controller is employed to perform trajectory tracking better than equally well-tuned PID for the particular case. The advantage is that since the ANN starts learning online each time from scratch when initialized, it converges faster and better when compared to other controllers as shown in the results.

The plot in Fig. 6 shows the trajectory tracking of the UAV in 3-dimensional space over time. The plots are shaded in two colors, the initial green phase shows the motors are running properly and the red phase starting at 21s mark shows the flight with motor failure. The motor failure is triggered when the UAV is following the mapping part of the trajectory. A slight change in z axis seen from the plot is the initial drop in height due to the instantaneous loss of thrust. The ANN controller learns it as a disturbance and compensates for the loss. The integral term of the PID controller also tries to minimize the steady-state error but ANN-PD is more effective. The UAV lands at the end with the motor failure state and a huge lag in the PID_{FCU}'s capability to land can be seen. On the other hand, the ANN-based controller and PID_{pos} are more effective at landing compared to the former. It is to be noted that the landing height is not exactly 0m, but slightly below that, as the field where the experiments are carried out is not an even surface - thus, the landing point is below the datum of the takeoff point.

The high speeds, experiencing motor failure, and the wind gusts exert tremendous stresses on the rotors and the UAV inertial dynamics, and thus small deviations from the trajectory are inevitable. The overview of the trajectory as seen from the top is depicted in Fig. 7. It can be noted that the PID_{pos} has more deviations from the actual path and the ANN controller follows the sharp bends more effectively, thus minimizing the overall error. In the trajectory tracking problem,

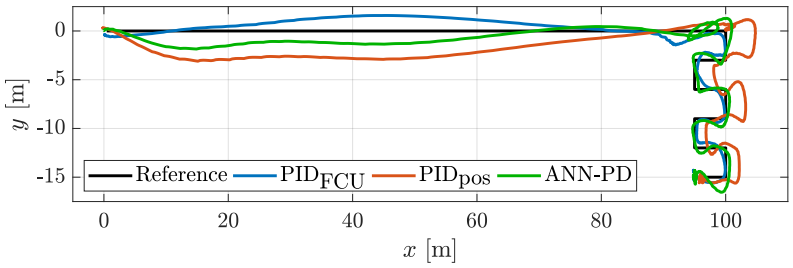

Fig. 7 Top view comparison of various controllers for an experimental scenario with fast flight, agile manoeuvres and motor failure

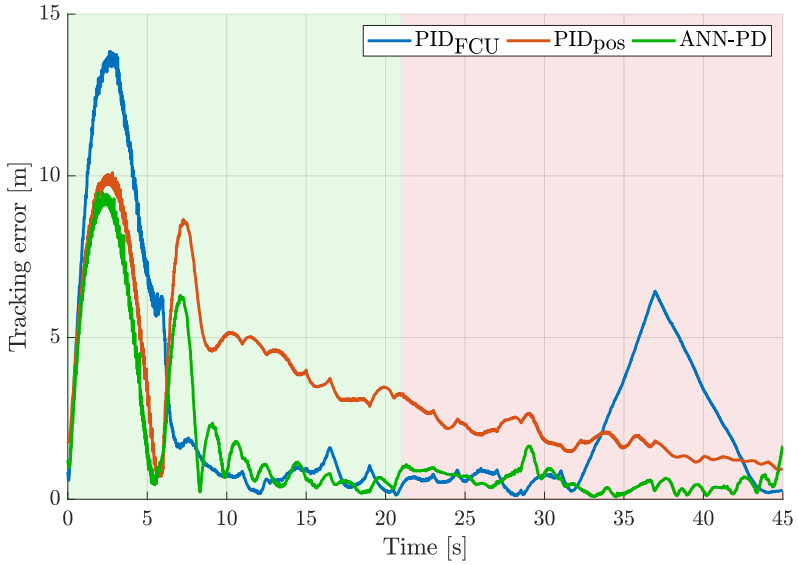

Fig. 8 Tracking error comparison of various controllers (green-shaded portion indicates normal operational conditions, while red-shaded portion indicates the motor failure)

the Euclidean error is usually calculated to determine the controller's performance, but it may penalize the algorithm as it takes into account the time delay in following the trajectory and not how accurately it is following [48]. Thus, the overall tracking error in the x, y, and z axes showing the UAV's capability of following the actual path is shown in Fig. 8. The ANN controller is able to achieve an overall improvement of 41% and 55% when compared to the PID_{FCU} and PID_{pos}, respectively, for the entire stretch of trajectory. Keeping in mind the high speeds and the high attitude angles achieved during the trajectory, the error for ANN is quite small.

The ground speed of the UAV, during the trajectory, for the various considered controllers is shown in Fig. 9. The ANN accelerates faster and thus tracks the trajectory better than the other two controllers. The acceleration plot is also shown in Fig. 9. The UAV follows the mapping part of the trajectory at a speed of 2m/s

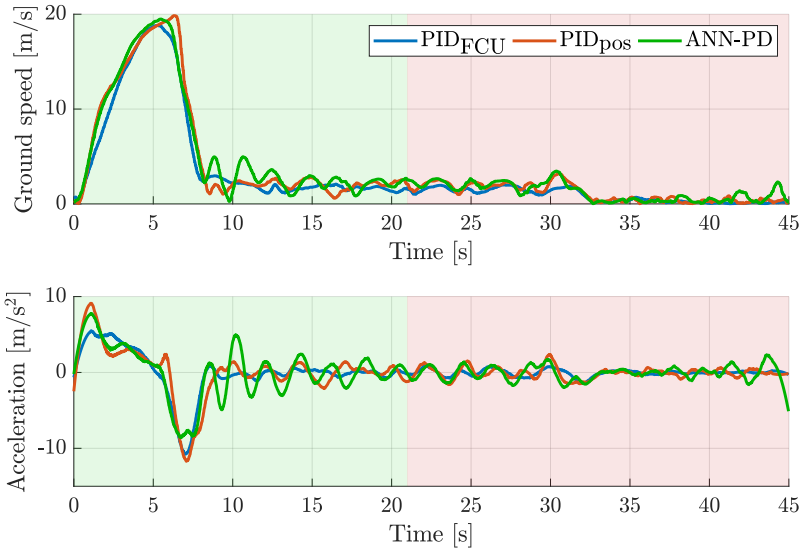

Fig. 9 Ground speed and acceleration of different controllers (green-shaded portion indicates normal operational conditions, while red-shaded portion indicates the motor failure)

with a motor failure. The ANN controller working in parallel with a PD controller is able to stabilize the flight during this scenario and still able to follow the desired path closely. The UAV lands at the end of the trajectory at a defined location.

The control output of the ANN controller for the x, y, and z axes is shown in Fig. 10. Note the sudden increase in the control output of z axis as the motor is failed at the instant of 21s. The performance characteristics of the controllers are given in Table 1.

5 Conclusions

In this work, an ANN-assisted PD controller is proposed for the control of the UAV for various challenging conditions. A fast flight manoeuvre at speeds in excess of 18m/s is performed to show the superior performance of the proposed controller. The UAV experiences a single motor failure while performing a pre-defined task and the controller handles the failure ensuring the safety of the mission as well as UAV. The UAV lands at a predefined location, with the motor failure, at the

Table 1 Comparison metrics of controllers

Controller	PID_{FCU}	PID_{pos}	ANN-PD
MAE (m)	5.564	6.653	4.288
Max. speed (m/s)	18.812	19.862	19.489
Max. acceleration (m/s^2)	5.508	9.125	7.759

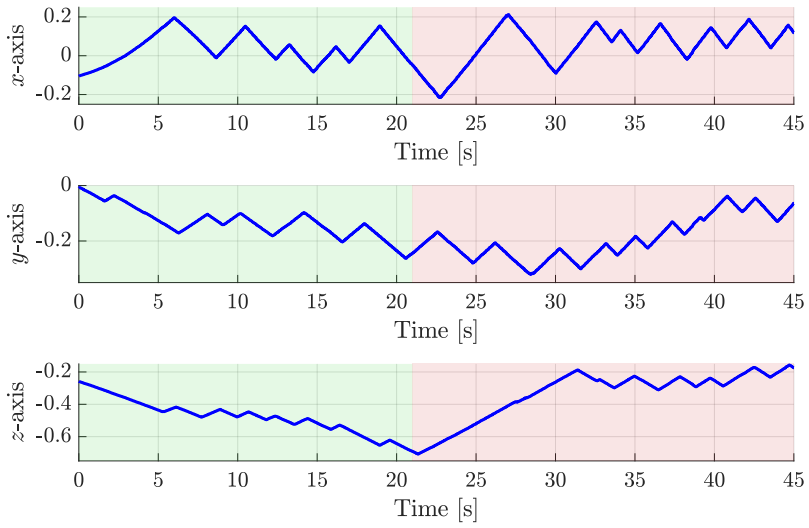

Fig. 10 Control output of ANN controller for x, y, and z axes (green-shaded portion indicates normal operational conditions, while red-shaded portion indicates the motor failure)

end of the trajectory to protect it from a potential crash. The model-free nature of the controller helps in accurate trajectory tracking even for high speed and agile manoeuvres. The advantage of the proposed controller is that it does not need a well-tuned set of PD gains as it learns online and improves the performance metrics while following the trajectory. Moreover, the proposed controller is computationally cheap to be implemented on the onboard computer of UAV. The real-time experiments are carried out in the outdoor environment with the use of RTK-GPS for localization. We show that for all the phases of the considered scenario, the proposed controller outperforms the conventional PID controllers. The average improvement of the ANN-based controller is above 40%.

Acknowledgements This work was partially financially supported by the Singapore Ministry of Education (RG185/17). In addition, this research was also partially supported by Aarhus University, Department of Engineering (28173).

References

1. Kehlenbeck, A.: Quaternion-based control for aggressive trajectory tracking with a micro-quadrotor UAV. Ph.D. thesis, University of Maryland, College Park (2014)
2. Loianno, G., Spurny, V., Thomas, J., Baca, T., Thakur, D., Hert, D., Penicka, R., Krajnik, T., Zhou, A., Cho, A., Saska, M., Kumar, V.: Localization, Grasping, and Transportation of Magnetic Objects by a Team of MAVs in Challenging Desert-Like Environments. IEEE Robotics and Automation Letters **3**(3), 1576–1583 (2018)
3. Siebert, S., Teizer, J.: Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. Automation in Construction **41**, 1 – 14 (2014)
4. Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I.L., Ruess, F., Suppa, M., Burschka, D.: Toward a Fully Autonomous UAV: Research Platform for

Indoor and Outdoor Urban Search and Rescue. IEEE Robotics Automation Magazine **19**(3), 46–56 (2012)
5. Goodarzi, F., Lee, D., Lee, T.: Geometric nonlinear PID control of a quadrotor UAV on SE(3). In: 2013 European Control Conference (ECC), pp. 3845–3850 (2013)
6. Mistler, V., Benallegue, A., M'Sirdi, N.K.: Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In: Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591), pp. 586–593 (2001)
7. Slegers, N., Kyle, J., Costello, M.: Nonlinear model predictive control technique for unmanned air vehicles. Journal of guidance, control, and dynamics **29**(5), 1179–1188 (2006)
8. Vachtsevanos, G., Tang, L., Reimann, J.: An intelligent approach to coordinated control of multiple unmanned aerial vehicles. In: Proceedings of the American Helicopter Society 60th Annual Forum, Baltimore, MD (2004)
9. Santos, M., López, V., Morata, F.: Intelligent fuzzy controller of a quadrotor. In: 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, pp. 141–146 (2010)
10. Achtelik, M., Bierling, T., Wang, J., Höcht, L., Holzapfel, F.: Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties. In: Infotech@ Aerospace 2011, p. 1485 (2011)
11. Blanke, M., Frei, W.C., Kraus, F., Patton, J.R., Staroswiecki, M.: What is Fault-Tolerant Control? IFAC Proceedings Volumes **33**(11), 41 – 52 (2000). 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes 2000 (SAFEPROCESS 2000), Budapest, Hungary, 14-16 June 2000
12. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews in Control **32**(2), 229 – 252 (2008)
13. Mehndiratta, M., Kayacan, E.: Reconfigurable Fault-tolerant NMPC for Y6 Coaxial Tricopter with Complete Loss of One Rotor. In: 2018 IEEE Conference on Control Technology and Applications (CCTA), pp. 774–780 (2018)
14. Eren, U., Prach, A., Koçer, B.B., Raković, S.V., Kayacan, E., Açıkmeşe, B.: Model predictive control in aerospace systems: Current state and opportunities. Journal of Guidance, Control, and Dynamics pp. 1–25 (2017)
15. Frangenberg, M., Stephan, J., Fichter, W.: Fast actuator fault detection and reconfiguration for multicopters. In: AIAA Guidance, Navigation, and Control Conference, p. 1766 (2015)
16. Saied, M., Lussier, B., Fantoni, I., Francis, C., Shraim, H., Sanahuja, G.: Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 5266–5271 (2015)
17. Zhang, Y., Chamseddine, A., Rabbath, C., Gordon, B., Su, C.Y., Rakheja, S., Fulford, C., Apkarian, J., Gosselin, P.: Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed. Journal of the Franklin Institute **350**(9), 2396 – 2422 (2013)
18. Rotondo, D., Nejjari, F., Puig, V.: Robust Quasi–LPV Model Reference FTC of a Quadrotor UAV Subject to Actuator Faults. International Journal of Applied Mathematics and Computer Science **25**(1), 7–22 (2015)
19. Verhaegen, M., Kanev, S., Hallouzi, R., Jones, C., Maciejowski, J., Smail, H.: Fault Tolerant Flight Control - A Survey, pp. 47–89. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
20. Mueller, M.W., D'Andrea, R.: Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 45–52 (2014)
21. Sarabakha, A., Imanberdiyev, N., Kayacan, E., Khanesar, M.A., Hagras, H.: Novel Levenberg-Marquardt based learning algorithm for unmanned aerial vehicles. Information Sciences **417**, 361 – 380 (2017)
22. Lupashin, S., Schöllig, A., Sherback, M., D'Andrea, R.: A simple learning strategy for high-speed quadrocopter multi-flips. In: 2010 IEEE International Conference on Robotics and Automation, pp. 1642–1648 (2010)
23. Spedicato, S., Notarstefano, G., Bülthoff, H.H., Franchi, A.: Aggressive Maneuver Regulation of a Quadrotor UAV, pp. 95–112. Springer International Publishing, Cham (2016)
24. Huang, H., Hoffmann, G.M., Waslander, S.L., Tomlin, C.J.: Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3277–3282 (2009)

25. Pretorius, A., Boje, E.: Design and Modelling of a Quadrotor Helicopter with Variable Pitch Rotors for Aggressive Manoeuvres. IFAC Proceedings Volumes **47**(3), 12208 – 12213 (2014). 19th IFAC World Congress
26. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. The International Journal of Robotics Research **31**(5), 664–674 (2012). DOI 10.1177/0278364911434236
27. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation, pp. 2520–2525 (2011)
28. Purwin, O., D'Andrea, R.: Performing aggressive maneuvers using iterative learning control. In: 2009 IEEE International Conference on Robotics and Automation, pp. 1731–1736 (2009)
29. Kayacan, E., Kaynak, O., Abiyev, R., Tørresen, J., Høvin, M., Glette, K.: Design of an adaptive interval type-2 fuzzy logic controller for the position control of a servo system with an intelligent sensor. In: International Conference on Fuzzy Systems, pp. 1–8 (2010)
30. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor. In: Robotics: Science and Systems, vol. 1. Citeseer (2013)
31. Loianno, G., Brunner, C., McGrath, G., Kumar, V.: Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU. IEEE Robotics and Automation Letters **2**(2), 404–411 (2017)
32. Sarabakha, A., Kayacan, E.: Y6 tricopter autonomous evacuation in an Indoor Environment using Q-learning algorithm. In: 2016 IEEE 55th Conference on Decision and Control (CDC), pp. 5992–5997 (2016)
33. Mahony, R., Kumar, V., Corke, P.: Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. Robotics Automation Magazine, IEEE **19**(3), 20–32 (2012)
34. Bouabdallah, S., Siegwart, R.: Design and Control of a Miniature Quadrotor, pp. 171–210. Springer Netherlands, Dordrecht (2007)
35. Penrose, R.: A generalized inverse for matrices. Mathematical Proceedings of the Cambridge Philosophical Society **51**(3), 406–413 (1955)
36. Dunfied, J., Tarbouchi, M., Labonte, G.: Neural network based control of a four rotor helicopter. In: Industrial Technology, 2004. IEEE ICIT '04. 2004 IEEE International Conference on, vol. 3, pp. 1543–1548 Vol. 3 (2004)
37. Johnson, E., Kannan, S.: Adaptive flight control for an autonomous unmanned helicopter. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 4439 (2002)
38. Buskey, G., Wyeth, G., Roberts, J.: Autonomous helicopter hover using an artificial neural network. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), vol. 2, pp. 1635–1640 vol.2 (2001)
39. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5), 359 – 366 (1989)
40. Yildiz, Y., Sabanovic, A., Abidi, K.: Sliding-Mode Neuro-Controller for Uncertain Systems. IEEE Transactions on Industrial Electronics **54**(3), 1676–1685 (2007)
41. Kayacan, E., Khanesar, M.A.: Chapter 7 - Sliding Mode Control Theory-Based Parameter Adaptation Rules for Fuzzy Neural Networks. In: E. Kayacan, M.A. Khanesar (eds.) Fuzzy Neural Networks for Real Time Control Applications, pp. 85 – 131. Butterworth-Heinemann (2016)
42. Efe, M.Ö.: Sliding Mode Control for Unmanned Aerial Vehicles Research, pp. 239–255. Springer International Publishing, Cham (2015)
43. Imanberdiyev, N., Kayacan, E.: A fast learning control strategy for unmanned aerial manipulators. Journal of Intelligent & Robotic Systems (2018)
44. Erdal, K., Okyay, K.: Sliding mode control theory-based algorithm for online learning in type-2 fuzzy neural networks: application to velocity control of an electro hydraulic servo system. International Journal of Adaptive Control and Signal Processing **26**(7), 645–659 (2012). DOI 10.1002/acs.1292
45. Kayacan, E., Kayacan, E., Khanesar, M.A.: Identification of Nonlinear Dynamic Systems Using Type-2 Fuzzy Neural Networks – A Novel Learning Algorithm and a Comparative Study. IEEE Transactions on Industrial Electronics **62**(3), 1716–1724 (2015)
46. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6235–6240 (2015). DOI 10.1109/ICRA.2015.7140074

47. Meier, L., Tanskanen, P., Heng, L., Lee, G.H., Fraundorfer, F., Pollefeys, M.: PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. Autonomous Robots **33**(1), 21–39 (2012)
48. Wilburn, B.K., Perhinschi, M.G., Moncayo, H., Karas, O., Wilburn, J.N.: Unmanned aerial vehicle trajectory tracking algorithm comparison. International Journal of Intelligent Unmanned Systems **1**(3), 276–302 (2013)