International Journal of Automation and Computing

18(1), February 2021, 124-141
DOI: 10.1007/s11633-020-1251-2

Design of an Executable ANFIS-based Control System to Improve the Attitude and Altitude Performances of a Quadcopter Drone

Mohammad Al-Fetyani[1] Mohammad Hayajneh[2] Adham Alsharkawi[1]

[1] Department of Mechatronics Engineering, The University of Jordan, Amman11942, Jordan

[2] Department of Mechatronics Engineering, The Hashemite University, Zarqa13133, Jordan

Abstract: Nowadays, quadcopters are presented in many life applications which require the performance of automatic takeoff, trajectory tracking, and automatic landing. Thus, researchers are aiming to enhance the performance of these vehicles through low-cost sensing solutions and the design of executable and robust control techniques. Due to high nonlinearities, strong couplings and under-actuation, the control design process of a quadcopter is a rather challenging task. Therefore, the main objective of this work is demonstrated through two main aspects. The first is the design of an adaptive neuro-fuzzy inference system (ANFIS) controller to develop the attitude and altitude of a quadcopter. The second is to create a systematic framework for implementing flight controllers in embedded systems. A suitable model of the quadcopter is also developed by taking into account aerodynamics effects. To show the effectiveness of the ANFIS approach, the performance of a well-trained ANFIS controller is compared to a classical proportional-derivative (PD) controller and a properly tuned fuzzy logic controller. The controllers are compared and tested under several different flight conditions including the capability to reject external disturbances. In the first stage, performance evaluation takes place in a nonlinear simulation environment. Then, the ANFIS-based controllers alongside attitude and position estimators, and precision landing algorithms are implemented for executions in a real-time autopilot. In precision landing systems, an IR-camera is used to detect an IR-beacon on the ground for precise positioning. Several flight tests of a quadcopter are conducted for results validation. Both simulations and experiments demonstrated superior results for quadcopter stability in different flight scenarios.

Keywords: Quadcopter, proportional integral derivate (PID) control, fuzzy control, adaptive neuro-fuzzy, altitude control, attitude control.

1 Introduction

Due to its simple construction and low-cost, as well as the massive development of its flight control systems, quadcopters have been extensively used in many civilian and military fields in the past two decades. Although the quadcopter system is nonlinear and under-actuated, making it difficult to achieve its dynamics stability and to improve its accuracy to track the required trajectory, linear-based controllers such as proportional integral derivate (PID) have obtained satisfactory results in controlling its movements[1−3].

However, linear controllers are limited by their inability to reach quadcopter stability in large scopes around the equilibrium point, preventing them from flying aggressively. In addition, the presence of external disturbances increases the deviation of the flight path as long as the quadcopter is operating. Therefore, many researchers have improved the work of the PID controller by tuning its gains online using classical fuzzy logic systems for stabilizing a quadcopter and controlling its attitude for trajectory tracking[4−7]. Others have recently applied fuzzy logic controllers to improve the trajectory followed by a quadcopter depending on factors such as the curvature of the path and the velocity of the quadcopter[8]. In other words, they applied the fuzzy logic to tune the distance and the velocity of a virtual point on the curvature as well as pitch and roll speeds of the quadcopter.

The problem in the fuzzy logic controller listed above may no longer be appropriate when there is a change or variation in system parameters or working conditions, which needs to be redesigned and adjusted frequently to deal with various uncertainties in the system[9]. In addition, this method requires considerable experience in designing and tuning IF-THEN rules that represent nonlinear approximations of the system[10, 11]. The approach to addressing this issue is to automatically adjust rule conditions and input/output parameters through learning algorithms. Therefore, the fuzzy logic could be combined with the learning ability of neural network methods in a form called adaptive neuro-fuzzy inference sys-

Research Article
Manuscript received April 22, 2020; accepted August 28, 2020; published online November 6, 2020
Recommended by Associate Editor Jie Zhang

tem (ANFIS). Some successful applications of the ANFIS approach in motion control can be found in [12−14]. The ANFIS approach has also found success in some other areas[15−17].

Few works in the literature have used an ANFIS controller to control the movement of the quadcopter. One of these works compared the results of the controller on the attitude of the quadcopter with respect to the performance of linear system recognition techniques without any validation of how the results improve the overall performance[18]. In other side, the ANFIS technique is applied in [10, 19, 20] to stabilize a quadcopter attitude and/or altitude, and it has proved better results than those produced by a PID controller, especially when applying external disturbances. However, the attempts in [10, 19, 20] discussed theoretical aspects and did not go beyond this to the practical application of the claimed solutions.

The aim of this paper is to apply ANFIS to control the altitude and attitude of a quadcopter. This work is completed in two main stages. The first stage includes simulation studies of quadcopter models in Matlab Simulink. In simulations, the performance of a well-trained ANFIS controller is compared to a classical proportional-derivative (PD) controller and a properly tuned fuzzy logic controller. The ANFIS controller is tested under several different flight conditions including external disturbances. Furthermore, aerodynamic effects are modeled and also simulated, to ensure more practical behavior of the quadcopter. The second stage focuses on practical implementation and experimental validation of the proposed controllers. More precisely, a C++ model of the proposed controllers is generated, using an embedded coder, by a supported package in Matlab Simulink. The attitude and altitude models are carefully customized to be run by PX4 autopilot, and to use on-board sensor data and other calculations at run time. In addition, the quadcopter is equipped with a vision-based system that includes an IR camera and beacon for precision landing purposes. Hence, a precision landing algorithm is adopted to estimate the relative horizontal positions of the drone with respect to the IR beacon coordinates. It is necessary to mention that the construction of a quadcopter underwent several examinations to ensure the integrity of the components and to reach reliable long-term flights.

The remainder of this paper is organized as follows. Section 2 briefly describes the quadcopter mathematical model which is essential in simulation and control design. Furthermore, this section discusses the adopted strategy for precision landing. Section 3 discusses the control design of a PD controller, fuzzy-PD controller and ANFIS-PD controller. Also, Section 3 provides a simple PD position controller and complementary filters for attitude and position estimations. This is followed by Section 4, where the hardware and software implementations of a quadcopter system are discussed for real flight experiments. Then the simulation and experimental results and the main findings are discussed in Section 5. Finally, some concluding remarks are given in Section 6.

2 System overview

This section mentions the quadcopter′s dynamic and kinematic models which are adopted in this work in simulation and in the design of the ANFIS-based attitude and altitude controllers. This section also presents a strategy that has been built for precision landing purposes.

2.1 Quadcopter model

Quadcopter dynamics are easily derived through a Newton-Euler method since the quadcopter is assumed to be a rigid body. The quadcopter equations of motion including the drag force are expressed in (1). Details of the derivation can be found in [21].

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{T}{m}\begin{bmatrix}C_\psi S_\theta C_\phi + S_\psi S_\phi\\ S_\psi S_\theta C_\phi - C_\psi S_\phi\\ C_\theta C_\phi\end{bmatrix} - \frac{1}{m}\begin{bmatrix}R_x\dot{x}\\ R_y\dot{y}\\ R_z\dot{z}\end{bmatrix} + \frac{1}{m}\begin{bmatrix}0\\0\\c_z\end{bmatrix} \tag{1}$$

where $C_\psi = \cos(\psi)$, $C_\phi = \cos(\phi)$, $S_\theta = \sin(\theta)$, $S_\psi = \sin(\psi)$, $S_\phi = \sin(\phi)$, m is the total mass of the quadcopter and g is the gravitational acceleration.

Euler angular velocities expressed in the inertial frame can be related to the body angular velocities expressed in the body frame by the following transformation matrix:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & S_\phi T_\theta & C_\phi T_\theta\\ 0 & C_\phi & -S_\phi\\ 0 & S_\phi/C_\theta & C_\phi/C_\theta\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix} \tag{2}$$

where $T = \tan(\theta)$, p, q and r are the angular velocities expressed in the body frame.

The quadcopter equations of angular motion can be expressed in the body frame as follows:

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}(I_{yy} - I_{zz})qr/I_{xx}\\ (I_{zz} - I_{xx})pr/I_{yy}\\ (I_{xx} - I_{yy})pq/I_{zz}\end{bmatrix} + \begin{bmatrix}\tau_\phi/I_{xx}\\ \tau_\theta/I_{yy}\\ \tau_\psi/I_{xx}\end{bmatrix} \tag{3}$$

where I_{xx}, I_{yy} and I_{zz} are the inertial moments of the quadcopter in the corresponding body frame axes.

2.2 Vision-based landing

Unlike the traditional landing method used in unmanned aerial vehicles (UAVs), which is based on the location from GPS, the vision based approaches provide precision landing on a specific spot accurate within centimeters. In this case, the UAV camera detects a landing spot, or marker, which defines the relative distance between the vehicle and the marker frames. Particularly in this work, an IR-lock camera mounted on a quadcopter detects an IR-beacon on the ground as illustrated in Fig. 1. Hence, the position of the beacon with respect to the camera frame C is calculated by a vision algorithm while the movements of the quadcopter are defined to the start point inertial frame I.

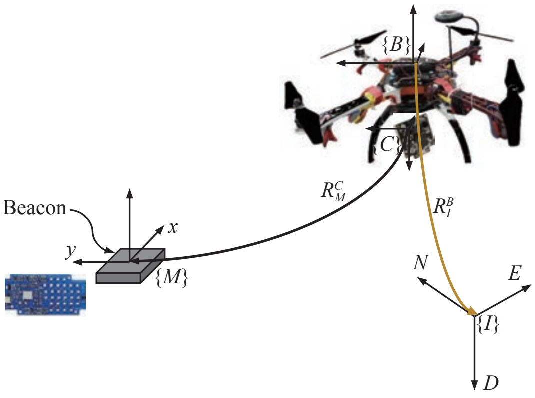


Fig. 1 Coordinate systems of vision-based landing

Once the IR-beacon is detected by the IR-camera, the drone attempts to move laterally (i.e., in x, y directions) towards the center of the beacon frame and starts precision landing from there. While the drone is descending, the ANFIS-attitude controller compensates for the effect of wind and the inherent instability. In this work, the adopted landing strategy is described by Fig. 2. The IR-beacon can be detected by the drone′s camera from a height of 10 m. Therefore, the landing process begins after the drone approaches the landing station. At this stage, the loiter mode is activated, where it keeps the drone hovering over the station and allows it to search for the beacon. If the beacon is detected within 40 s, the precision landing controller is activated, otherwise a GPS-based landing is performed. In the case of precision landing, the drone′s lateral positions are always estimated with respect to beacon frame which is ideally approaching zero. The drone will keep moving horizontally and the precision landing process will be executed continuously as long as the height is large (i.e., higher than 0.5 m). When the drone is near the ground with a distance of less than 0.5 m, the descending is handled with a special technique that ensures safe landing and disarms the motors.

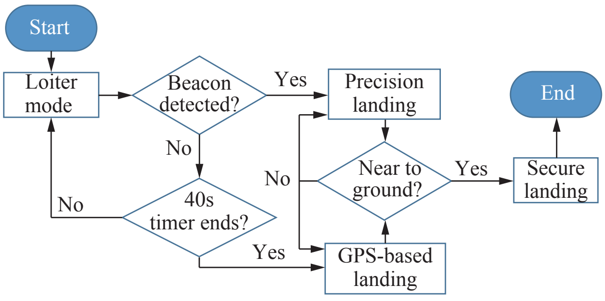


Fig. 2 Vision-based landing strategy

3 Control design

The quadcopter is an unstable system and requires closed-loop feedback to obtain stability. The quadcopter is controlled using the four motor angular velocities (ω_1, ω_2, ω_3 and ω_4), which are responsible for the six states of the quadcopter, linear positions and angular positions. The quadcopter can be stabilized by controlling four states, altitude (z) and attitude (ϕ, θ and ψ).

The motor angular velocities can be obtained through inverse dynamics[22], thus the system has to be simplified to provide an inverse model that is easy to deal with. In order to do so, the movement of the quadcopter is considered to be near the hovering state, in which small angular changes occur, and the rotational matrix between Euler angular velocities and the body angular velocities is deemed as close to identity.

Hence, the simplified quadcopter equations used in the control design are given as

$$\begin{aligned}\ddot{z} &= -g + (\cos(\theta)\cos(\phi))\frac{T}{m}\\ \ddot{\phi} &= \frac{\tau_\phi}{I_{xx}}\\ \ddot{\theta} &= \frac{\tau_\theta}{I_{yy}}\\ \ddot{\psi} &= \frac{\tau_\psi}{I_{zz}}.\end{aligned} \tag{4}$$

3.1 PID controller

A classical PID controller is still one of the most widely used controllers due to its simple structure and ease of implementation[23, 24]. One way to implement a PID controller is to use a parallel PID structure, which allows for a complete decoupling of proportional, integral and derivative actions. The parallel PID structure can be described by the following:

$$\begin{aligned}e(t) &= x_d(t) - x(t)\\ u(t) &= K_P e(t) + K_I\int_0^t e(\tau)\mathrm{d}\tau + K_D\frac{\mathrm{d}e(t)}{\mathrm{d}t}\end{aligned} \tag{5}$$

where $e(t)$ is the difference between the desired state $x_d(t)$ and the current measured state $x(t)$, $u(t)$ is the control signal, K_P is the proportional coefficient, K_I is the integral coefficient and K_D is the derivative coefficient.

Given the system double integrator behavior described in (4), four PD controllers are required to stabilize the four desired states of the quadcopter, altitude (z) and attitude (ϕ, θ and ψ), which are derived from (4) as follows:

$$\begin{aligned}T &= (g + K_{z,D}(\dot{z}_d - \dot{z}) + K_{z,P}(z_d - z))\frac{m}{C_\phi C_\theta}\\ \tau_\phi &= (K_{\phi,D}(\dot{\phi}_d - \dot{\phi}) + K_{\phi,P}(\phi_d - \phi))I_{xx}\\ \tau_\theta &= (K_{\theta,D}(\dot{\theta}_d - \dot{\theta}) + K_{\theta,P}(\theta_d - \theta))I_{yy}\\ \tau_\psi &= (K_{\psi,D}(\dot{\psi}_d - \dot{\psi}) + K_{\psi,P}(\psi_d - \psi))I_{zz}.\end{aligned} \tag{6}$$

The values obtained from (6) are used to calculate the four angular velocities, which are the inputs to the quadcopter. The angular velocities can be expressed as

$$\begin{aligned}\omega_1^2 &= \frac{T}{4k} - \frac{\tau_\theta}{2kl} - \frac{\tau_\psi}{4b}\\ \omega_2^2 &= \frac{T}{4k} - \frac{\tau_\phi}{2kl} + \frac{\tau_\psi}{4b}\\ \omega_3^2 &= \frac{T}{4k} + \frac{\tau_\theta}{2kl} - \frac{\tau_\psi}{4b}\\ \omega_4^2 &= \frac{T}{4k} + \frac{\tau_\phi}{2kl} + \frac{\tau_\psi}{4b}.\end{aligned} \tag{7}$$

3.2 Fuzzy-PD controller

Fuzzy logic is a multi-valued logic that deals with degrees of membership and degrees of truth. The process of designing a fuzzy logic controller is performed in three main steps[18]. The first step is called fuzzification in which the inputs are mapped onto membership functions. The membership values are then quantified from a rule base. This step is called rule evaluation. The last step is called defuzzification in which membership values are converted into crisp outputs.

A fuzzy-PD controller is a combination between a classical PD controller and a fuzzy logic controller. The purpose of the fuzzy-PD controller is to establish a varying coefficient for the PD controller rather than a fixed coefficient which allows having an optimal output based on a set of rules[25].

Fuzzy membership functions, for both the inputs and the output, of the fuzzy-PD controller are shown in Fig. 3. Rule base for both the inputs and the output are presented in Table 1. Each input is divided into five fuzzy sets as in [7], i.e., negative big (NB), negative small (NS), zero (Z), positive small (PS) and positive big (PB). The output is also divided into five fuzzy sets, i.e., fast negative (FN), slow negative (SN), slow (S), slow positive (SP) and fast positive (FP). Fuzzy rule surfaces for the altitude and the attitude are presented in Fig. 4.

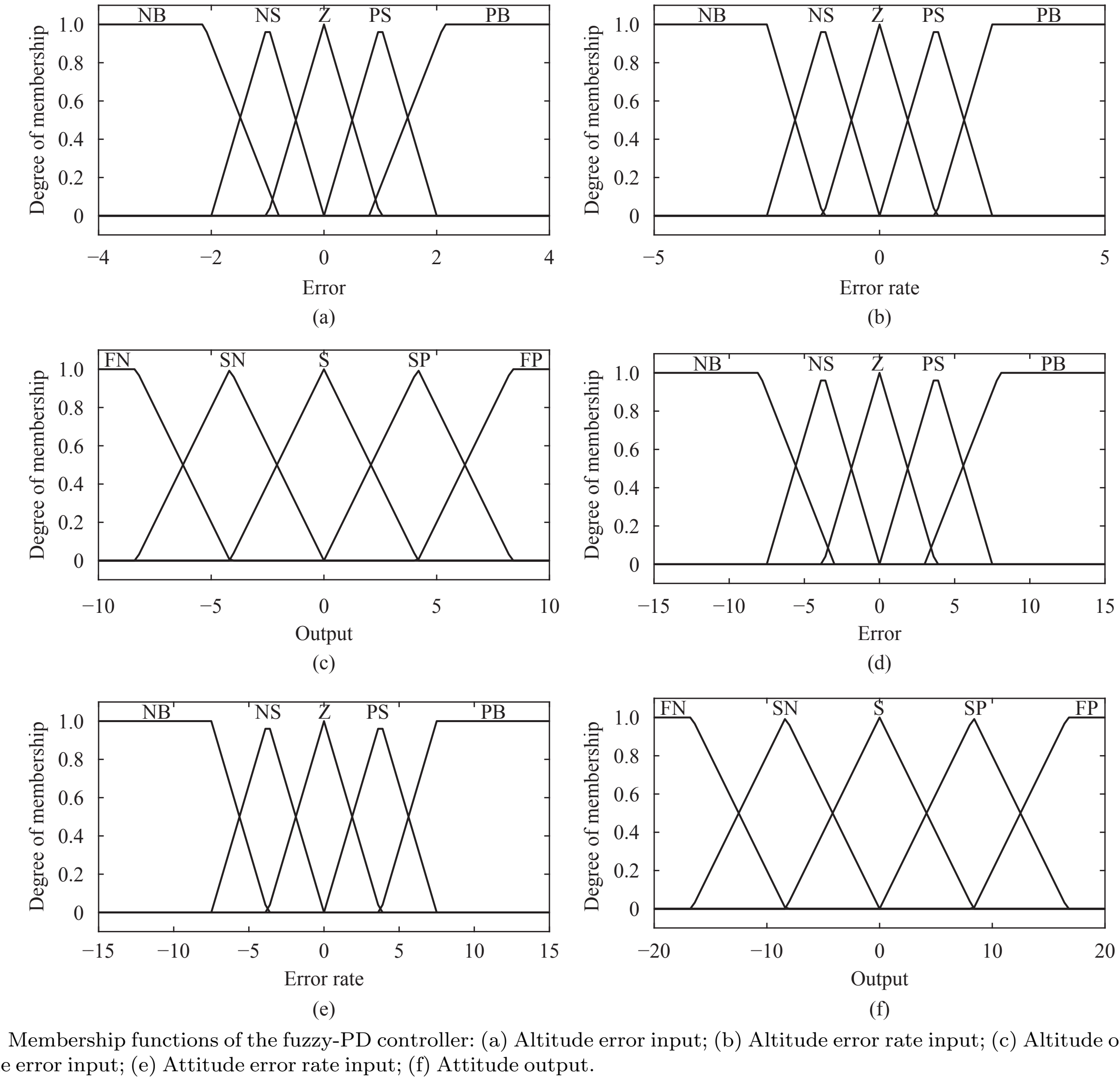


Fig. 3 Membership functions of the fuzzy-PD controller: (a) Altitude error input; (b) Altitude error rate input; (c) Altitude output; (d) Attitude error input; (e) Attitude error rate input; (f) Attitude output.

Table 1 Rule base of the fuzzy-PD controller

		Error				
		NB	NS	Z	PS	PB
Error rate	NB	SN	FP	FP	FP	FP
	NS	FN	SN	SP	SP	FP
	Z	FN	SN	S	SP	FP
	PS	SN	S	SN	S	FP
	PB	S	FN	FN	SN	S

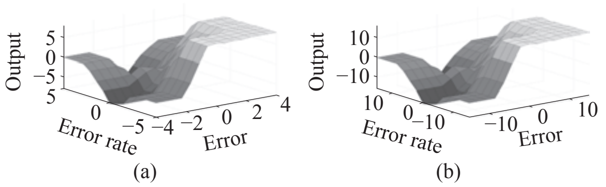


Fig. 4 Rule base surfaces of the fuzzy-PD controller: (a) Altitude; (b) Attitude.

3.3 ANFIS-PD controller

Designing a fuzzy logic controller can be a complicated process especially for a complex system. Given a set of input/output training data, ANFIS is an approach that can be used to easily obtain a properly tuned fuzzy logic controller[19]. The strength of the ANFIS approach comes from the combined advantages of the fuzzy inference system (FIS) and neural networks (NN).

In this paper, the ANFIS architecture used for the quadcopter is presented in Fig. 5.

The controller has to learn only once by a set of training data. The training data set for the altitude is obtained by simulating the quadcopter when controlled by a classical PD controller. A similar process is repeated for the attitude control.

A neuro-fuzzy designer is used for training and designing the ANFIS-PD controller. Membership functions obtained after the training are presented in Fig. 6. The rule base of the ANFIS-PD controller is presented in Table 2. The output equations of the altitude controller can be found in (A1) in Appendix and the output equations of the attitude controllers are found in (A2) in Appendix. Fuzzy rule surfaces for the altitude and attitude are illustrated in Fig. 7.

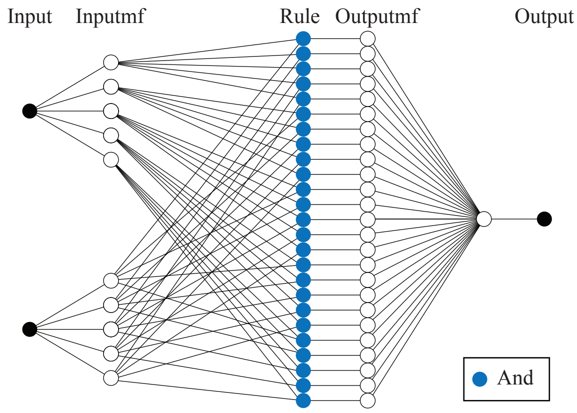


Fig. 5 ANFIS multi-layer architecture

3.4 Position controller

The attitude and altitude controllers, which have been discussed previously in this section, are essential for overcoming the inherent instability of the quadcopter drone. To control the movement of the vehicle horizontally, a closed loop position controller is needed to approach the desired waypoint. In the quadcopter case, the horizontal motion is dependent with the roll and pitch states. Quadcopter positions can be easily defined by GPS location readings. However, positions from GPS have low accuracy, and are not suitable for precise landing purposes on a specific point. Therefore, a vision based landing process is adopted in this work as described in Section 2. An outer loop PD controller is adopted for both GPS-based and vision-based systems during the landing process as

$$u(t)=K_Pe(t)+K_D\frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{8}$$

where $e(t)=p_i-p_o$ and $\dot{e}(t)=\dot{p}_i-\dot{p}_o$ are error position and velocity state vectors respectively. The state vector is expressed by the deference between the reference states $p_o=[x_o\ y_o]^{\mathrm{T}}$ and $\dot{p}_o=[\dot{x}_o\ \dot{y}_o]^{\mathrm{T}}$, and the target states p_i and $\dot{p}_i$. The output of the proposed controller is the control command $u(t)=[u_\theta\ u_\phi]^{\mathrm{T}}$ that is provided to the attitude controller. u_θ and u_ϕ allow the quadcopter to adjust its horizontal positions according to the position estimates from GPS or vision system. K_P and K_D are proportional and derivative gains respectively tuned for better tracking performance.

3.5 Attitude and position estimations

Due to noises and biases in cheap sensors which are used in low cost embedded systems, sensor fusion techniques are used for estimates of the drone′s attitude and position. This work adopts complementary filtering algorithms for state estimations. A complementary filter is used to fuse between an accelerometer, magnetometer and gyroscope measurements for better attitude estimation. A second filter is described for altitude estimation using accelerometer readings and barometer measurements. A third filter combines the GPS information and an accelerometer for better horizontal position estimations.

For the sake of keeping things short, these filters will be discussed briefly. For more details about design and implementation, you can read our previous works[26, 27].

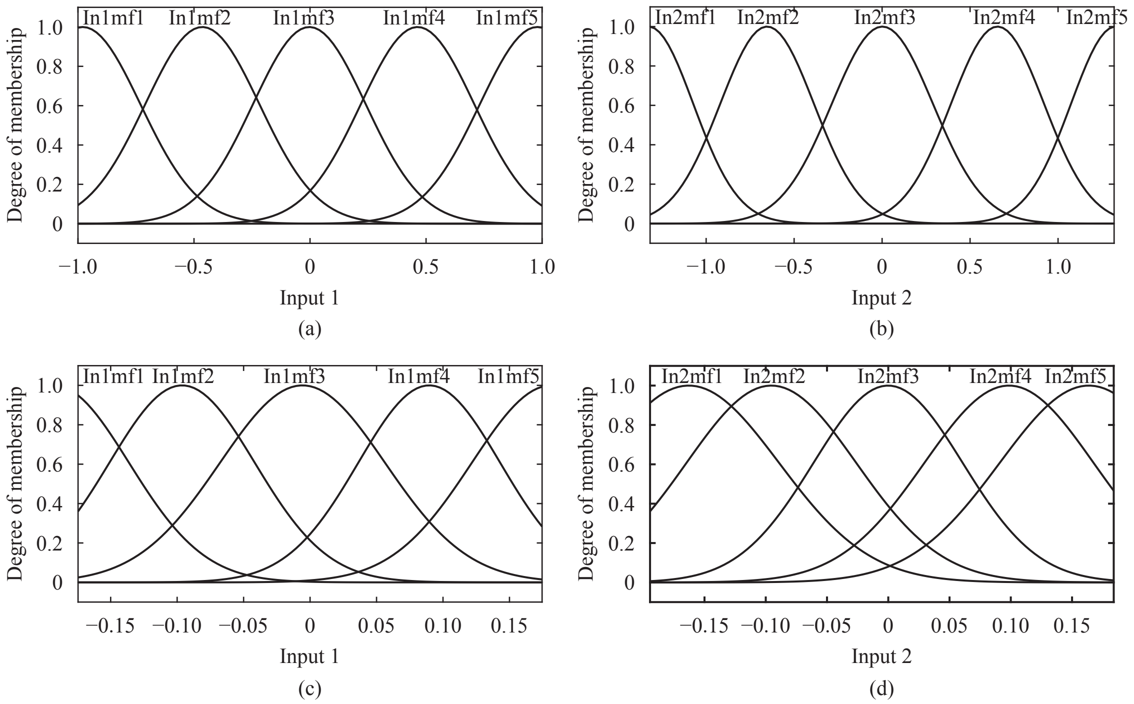


Fig. 6 Membership functions of the ANFIS-PD controller: (a) Altitude error input; (b) Altitude error rate input; (c) Attitude error input; (d) Attitude error rate input.

Table 2 Rule base of the ANFIS-PD controller

		Input1 (Error)				
		In1mf1	In1mf2	In1mf3	In1mf4	In1mf5
Input2 (Error rate)	In2mf1	Out1mf1	Out1mf6	Out1mf11	Out1mf16	Out1mf21
	In2mf2	Out1mf2	Out1mf7	Out1mf12	Out1mf17	Out1mf22
	In2mf3	Out1mf3	Out1mf8	Out1mf13	Out1mf18	Out1mf23
	In2mf4	Out1mf4	Out1mf9	Out1mf14	Out1mf19	Out1mf24
	In2mf5	Out1mf5	Out1mf10	Out1mf15	Out1mf20	Out1mf25

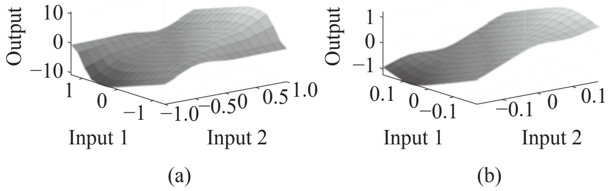


Fig. 7 Rule base surfaces of the ANFIS-PD controller: (a) Altitude; (b) Attitude.

3.5.1 Attitude estimation

The attitude estimation could be integrated from angular speeds measured by a gyroscope. But the presence of biases and noises in readings leads to divergence in these interactions. Therefore, the gyroscope readings are combined with accelerometer's readings for better estimations in roll (ϕ) and pitch (θ) angles, and with magnetometer measurements to estimate the yaw angle (ψ). The complementary filter that is adopted for aforementioned purposes is implemented as follows:

$$\left\{\begin{array}{l} \dot{\hat{q}} = \dfrac{1}{2}\hat{q} \otimes P\Big(\Omega - \hat{b} + K_P \displaystyle\sum_{i=1}^{n} k_i v_i \times \hat{v}_i\Big) \\ \dot{\hat{b}} = -K_I \displaystyle\sum_{i=1}^{n} k_i v_i \times \hat{v}_i \end{array}\right. \tag{9}$$

where $\hat{q}$ and $\hat{b}$ are the attitude estimation in quaternions representation and the gyro biases respectively. For better performance, K_P, K_I and k_i are tuned properly. Ω is the vector of rate gyro measurements. The vector v is measured in body frame and vector $\hat{v}$ is its observation in the inertial frame. Two vectors in the inertial frame can be observed in embedded systems which are the Earth's gravitational and magnetic field measured by an accelerometer and magnetometer respectively.

3.5.2 Altitude and horizontal positions estimations

Trajectory tracking in automatic missions requires robust position estimations using on-board sensors. Due to noisy and low frequency GPS and barometer sensors, a complementary filter is implemented to optimally estimate a drone's 3-D positions as follows:

$$\dot{\hat{p}} = \hat{v} + \frac{1}{2}\int (Qa + ge_3)\mathrm{d}x + k_2(p - \hat{p}) \qquad (10)$$

where p is the position vector which represents two raw measurements of GPS horizontal positions and one barometer measurement for the vehicle′s altitude. The position estimation is a function of $\hat{v}$ which is the estimated velocity introduced by the following relation:

$$\begin{cases} \dot{\hat{v}} = k_1(v - \hat{v}) + ge_3 + Qa \\ \dot{Q} = QS(\Omega - \hat{b}_\Omega) + k_v(v - \hat{v})a^{\mathrm{T}} \end{cases} \qquad (11)$$

where $\Omega \in \mathbf{R}^3$ and $a \in \mathbf{R}^3$ are raw measurements of angular speeds and linear accelerations measured in body frame. The vector $v \in \mathbf{R}^3$ includes the actual velocities measured in the inertial frame by GPS, the receiver and pressure sensor. Moreover, $Q \in \mathbf{R}^{3\times3}$ is a virtual rotation matrix and $ge_3 = [0,\ 0,\ g]^{\mathrm{T}}$ is the local gravity vector in the inertial north-east-down (NED) frame. The best performance of this filter can be achieved with good gains tuning of k_1, k_2 and k_v.

4 Real-time implementations

A UAV quadcopter (see Fig. 8) is used in this work to implement and verify the performance of the control algorithms just discussed. The full implementation of the ANFIS-based controllers and the visual-feedback precision landing controller are executed by a Pixhawk autopilot as shown in Fig. 9. The quadcopter is also equipped with an IR-lock sensor used for precise position estimation during the landing process. Furthermore, a radio telemetry unit is connected in order to send and receive flight parameters (e.g., flight state, waypoints, flight modes, $\cdots$, etc.) to and from the ground control station (GCS).

Due to the diversity of UAV models in market, it is worth mentioning the characteristics of the UAV hardware and software architectures adopted in implementations which are briefly demonstrated in Fig. 9.

4.1 Quadcopter

The flight control algorithms on the proposed quadcopter (see Fig. 8) are executed by a Pixhawk (PX4) autopilot as an open-source and low-cost flight control unit. The Pixhawk autopilot includes different on-board sensors (i.e., inertial measurements unit (IMU), barometer) and supports external sensors such as GPS, ultrasonic, and IR-lock camera. Sensor data are fed to control algorithms running on a ARM Cortex-M microprocessor based on a NuttX real-time operating system (RTOS). In this work, ANFIS based control algorithms are adopted for attitude, and altitude stability and automatic landing. These algorithms also perform an allocation matrix to provide motors with corresponding pulse width modulation (PWM) outputs. Via the Micro air vehicle link (MAV Link) as a light weight massage protocol, the Pixhawk autopilot transmits flight information to the GCS such as attitude, position, and flight conditions, and receives data from the GCS such as path waypoints. The Pixhawk is connected also with a light-weight and low-cost IR-lock camera which is used for precision landing.

4.2 IR-lock camera

The IR-lock camera, as shown in Fig. 9, is connected

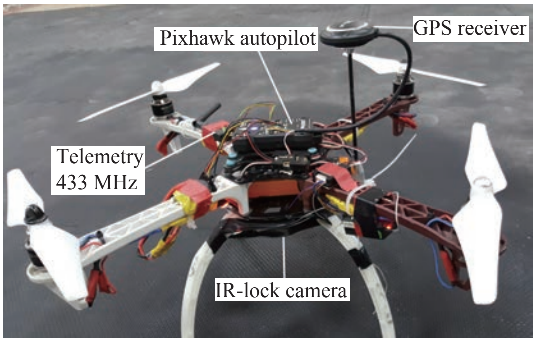


Fig. 8 Quadcopter

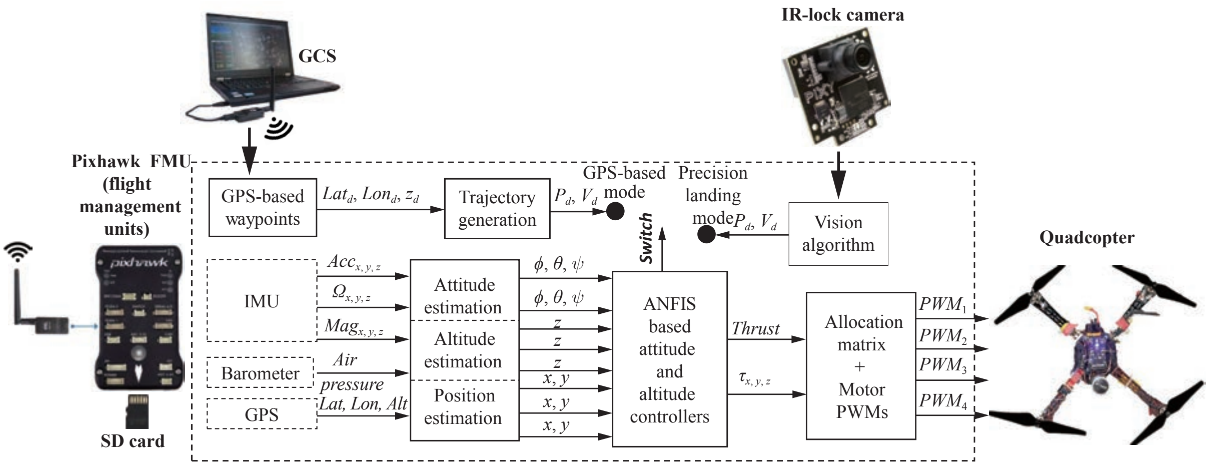


Fig. 9 UAV hardware and software architecture

to the quadcopter for precision landing purposes. Furthermore, an IR-beacon is mounted on a center of the landing point and is filtered by the IR camera to define its position in the 2D image frame. A vision algorithm is implemented to estimate horizontal coordinates, x_i and y_i, which are measured in pixel units. The x and y components are fused to calculate the estimated positions and velocities with respect to the beacon frame. This system provides high precise landing, within 20 cm accuracy, that allows the vehicle to approach the center of the beacon horizontally while descending toward a specific landing station.

4.3 Ground control station

The ground control station consists of a high-speed processing laptop computer and a radio controller. The laptop is connected to the quadcopter by a radio telemetry module for flight data and parameter transmission. The computer also runs a graphical user interface software (i.e., mission planner) that allows a user to monitor flight states and to change parameters. Furthermore, a Matlab software with PX4-Simulink support package is installed with required code compilers on the computer.

4.4 PX4-Simulink support package

The PX4-Simulink support package in Matlab allows you to customize different algorithms in Simulink blocks that leverage Pixhawk sensor data and other calculations at run time. Moreover, a C++ code can be generated from Simulink models and deployed to the Pixhawk flight management unit (FMU) using the PX4 Toolkit. This package also supports a sensor/peripheral block library that allows reading on-board sensor data such as inertial measurements, GPS, and vehicle estimation from the IR-lock sensor, writing PWM signals to the motor′s electronic speed control (ESC), and activating analog to digital converter (ADC) and serial Rx/Tx.

According to the official package website[28], the package is specifically tailored for the Pixhawk flight management units. Therefore, the generated code has been optimized for this particular flight controller. During the deployment process, no problems were encountered, either in terms of memory usage or computations. For other hardware, however, one can easily extract the fuzzy control system since the equations for the membership functions and the rules are explicitly provided, each in a separate function by the compiler.

5 Simulation and experimental results

This section discusses the results of simulation and flight experiments on a quadcopter drone. The characteristics of the used quadcopter (see Fig. 8) are experimentally calculated and presented in Table 3 for better implementation.

In the first stage, the quadcopter model and different controllers are implemented for simulation in Matlab 2018b, in order to compare the performance of the different controllers in this paper. The simulation is performed in three different cases, and parameter values of the adopted PD controllers are chosen from [2] which are presented in Table 4.

The feasibility of running the controllers through embedded systems depends on the control signals, which are the values of the angular velocities of the motors. Thus, the control signals during the simulation are considered.

5.1 Case 1

The first case is taken from [2]. The initial conditions of the quadcopter, linear positions and angular positions, are $[x, y, z]^{\mathrm{T}} = [0, 0, 1]^{\mathrm{T}}$ in meters, $[\phi, \theta, \psi]^{\mathrm{T}} = [10, 10, 10]^{\mathrm{T}}$ in degrees. The desired state of the quadcopter is hovering, thus $[z, \phi, \theta, \psi]^{\mathrm{T}} = [0, 0, 0, 0]^{\mathrm{T}}$. The simulation runs for 5 s. The output graphs for the altitude and attitude during the simulation are presented in Fig. 10. Control signals during the simulation are presented in Fig. 11.

Table 4 PD controller parameter values

Parameter	Value	Parameter	Value
$K_{z,D}$	2.50	$K_{z,P}$	1.5
$K_{\varphi,D}$	1.75	$K_{\varphi,P}$	6.0
$K_{\theta,D}$	1.75	$K_{\theta,P}$	6.0
$K_{\psi,D}$	1.75	$K_{\psi,P}$	6.0

Table 3 Quadcopter parameters

Parameter	Value	Unit	Parameter	Value	Unit
g	9.810	m/s^2	I_{xx}	0.04856	kg·m^2
Mass, m	1.568	kg	I_{yy}	0.04856	kg·m^2
Arm length, 1	0.225	m	I_{zz}	0.08801	kg·m^2
R_x	0.50	kg/s	Drag coefficient, k	2.980×10^{-6}	
R_y	0.50	kg/s	Thrust coefficient, b	1.140×10^{-7}	
R_z	0.50	kg/s			

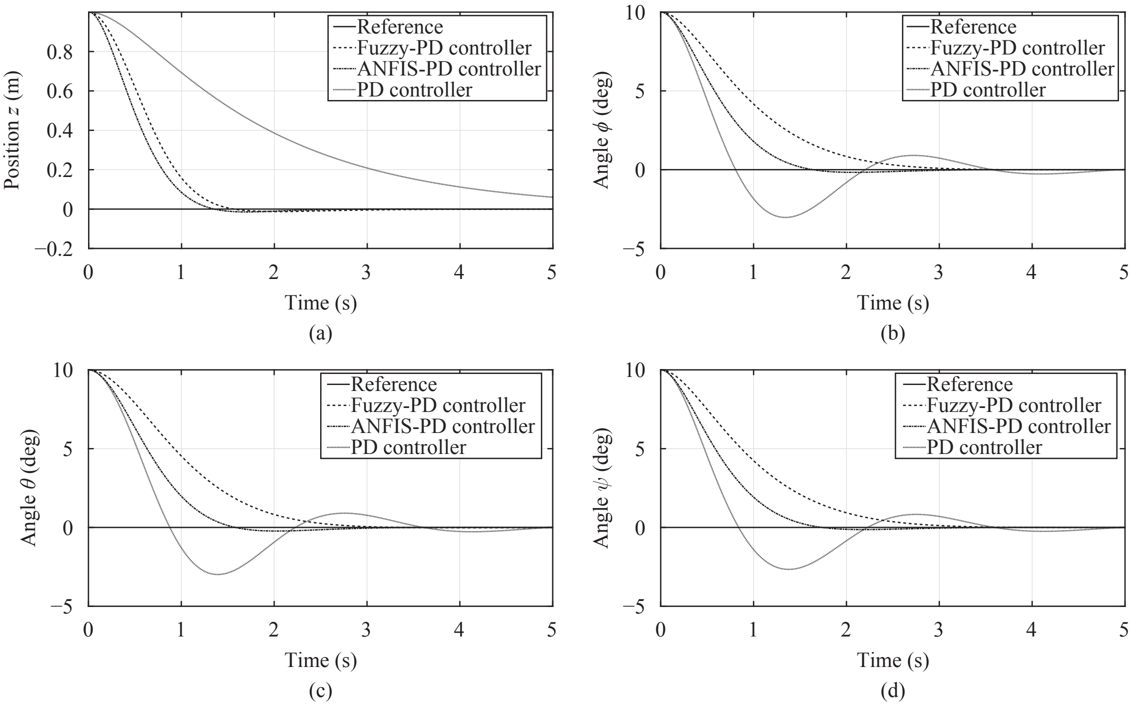


Fig. 10 Simulation results for Case 1: (a) Position z; (b) Angle ϕ; (c) Angle θ; (d) Angle ψ.

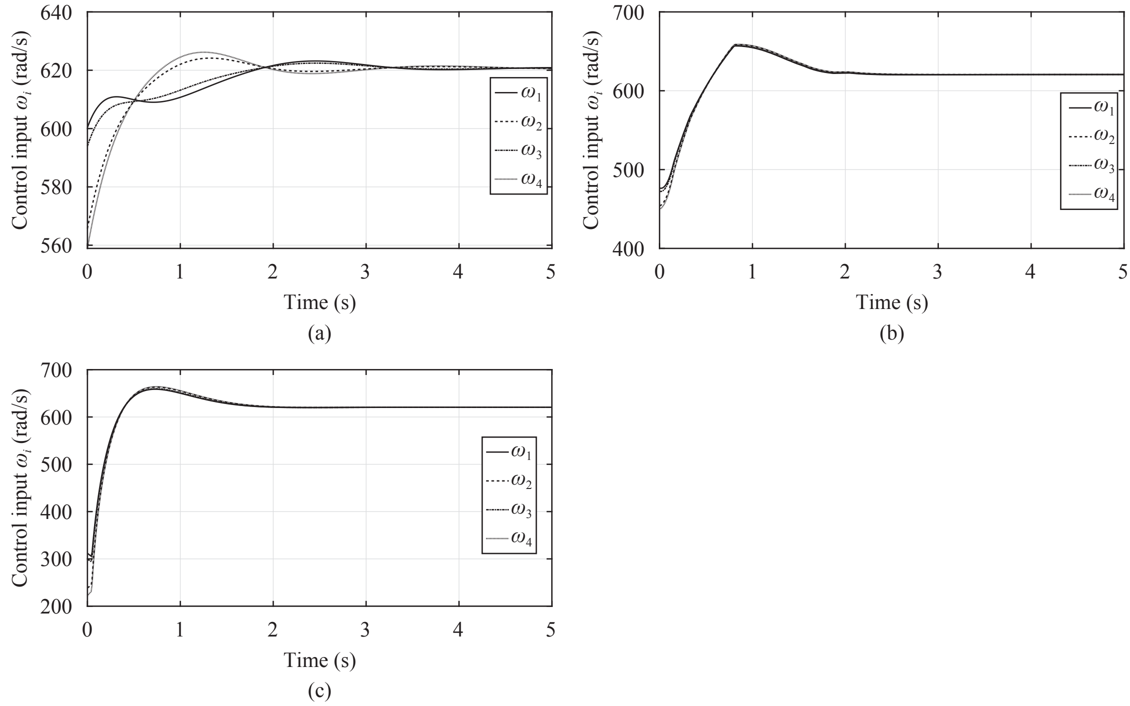


Fig. 11 Control inputs for Case 1: (a) PD controller; (b) Fuzzy-PD controller; (c) ANFIS-PD controller.

In this case, it can be clearly seen that the ANFIS-PD controller is the fastest to reach the desired state with negligible overshoot and feasible control signals. Although the fuzzy-PD controller may not be the best controller in this case, it still shows good results when compared to the PD controller. The oscillatory behavior of the PD controller can be easily observed when controlling the attitude along with a slow settling time. Values of the settling time and overshoot for each of the controllers are presented in Table 5.

The computational costs, for this case, are computed in the simulation environment. The average execution time for the PD and ANFIS controllers has been estimated to be 6 s and 16 s respectively. However, the execution of the ANFIS model in real-time processor did not cause an overrun. The processor in the autopilot was act-

Table 5 Case 1: Response characteristics of the three controllers

		PD	Fuzzy-PD	ANFIS-PD
Settling time (s)	Position z	>5	1.39	1.22
	Angle ϕ	4.43	2.68	1.48
	Angle θ	4.47	2.60	2.26
	Angle ψ	4.40	2.78	1.55
Overshoot peak value	Position z	–	−0.01	−0.01
	Angle ϕ	−3.03	−0.01	−0.17
	Angle θ	−2.99	−0.02	−0.23
	Angle ψ	−2.66	−0.01	−0.14

ively operating the control system for a very responsive flights as shown by experiments in Section 5.4. On the other hand, in our experimental setup, we used the motor from [29], which can give a maximum power of 109 W with full throttle. After the angular velocities are converted to corresponding PWM signals, the average PD and ANFIS PWM signal that is required from the motors are found to be 2 000, which corresponds to 100% throttle, and 1 621, which can be approximated to 65% throttle, respectively. As a result, the PD controller consumes 109 W while the ANFIS controller only consumes 42.2 W.

5.2 Case 2

In this case, the desired state is time-varying, in which the quadcopter is expected to follow several ramp functions. This is illustrated in Table 6. The simulation runs for 30 s. Altitude and angular positions during the simulation are presented in Fig. 12. Control signals during the simulation are presented in Fig. 13.

It is quite clear in this case that the PD controller shows an oscillatory behavior when trying to control the angular positions ϕ and ψ in the time interval between 10 s and 15 s. The reason for this is the fact that the PD controller is tuned to a near hovering state. Similar to the first case, the ANFIS-PD controller is still the fastest to reach a time-varying desired state when compared to both controllers, the PD controller and fuzzy-PD controller.

5.3 Case 3

The third case investigates the performance of the three controllers in tracking a desired altitude in the presence of disturbances. The simulation runs for 10 s. At the beginning of the simulation, the quadcopter is expected to track a sinusoidal target and then maintain zero altitude. During the simulation, an external disturbance force is added at the beginning of the simulation and towards the end of the simulation. This is illustrated in Fig. 14(a). The performance of the three controllers along with the control signals are presented in Figs. 14(b) and 15, respectively.

It is rather obvious here that the ANFIS-PD controller is coping quite well with the external disturbance force. For clarity, Table 7 shows the deviation from the desired target for the three controllers through mean square error (MSE) and mean absolute error (MAE).

5.4 Experimental flight tests

The proposed ANFIS controllers, attitude and position estimators, and the precision landing algorithm are executed and verified experimentally by a Pixhawk embedded system on top of a quadcopter as discussed in Sections 3 and 4. Several experiments were conducted on the quadcopter to prove the system's efficiency and reliability.

First, the ANFIS-based attitude and the altitude controllers are tested in manual mode, where a user sends commands by a remote controller to control the position of the drone in the space. The commands in manual mode are: throttle signal to control the altitude, the roll, pitch and yaw signals to control the drone's attitude. Fig. 16(a) illustrates the flight path of the quadcopter in the manual mode. The 39-seconds flight experiment shows great

Table 6 Case 2: Quadcopter time-varying desired state

Time t (s)	Position z (m)	Angle ϕ (deg)	Angle θ (deg)	Angle ψ (deg)
$0 \leq t \leq 5$	$\frac{1}{5}t$	0	0	$\frac{20}{5}t$
$5 \leq t \leq 10$	1	$\frac{40}{5}(t-5)$	0	$-\frac{50}{5}(t-5)+20$
$10 \leq t \leq 15$	$\frac{1}{5}(t-10)+1$	40	$-\frac{80}{5}(t-10)$	−30
$15 \leq t \leq 20$	$-\frac{1}{5}(t-15)+2$	40	−80	−30
$20 \leq t \leq 25$	1	$-\frac{40}{5}(t-20)+40$	−80	−30
$25 \leq t \leq 30$	$-\frac{1}{5}(t-25)+1$	0	$\frac{80}{5}(t-25)-80$	$\frac{30}{5}(t-25)-30$

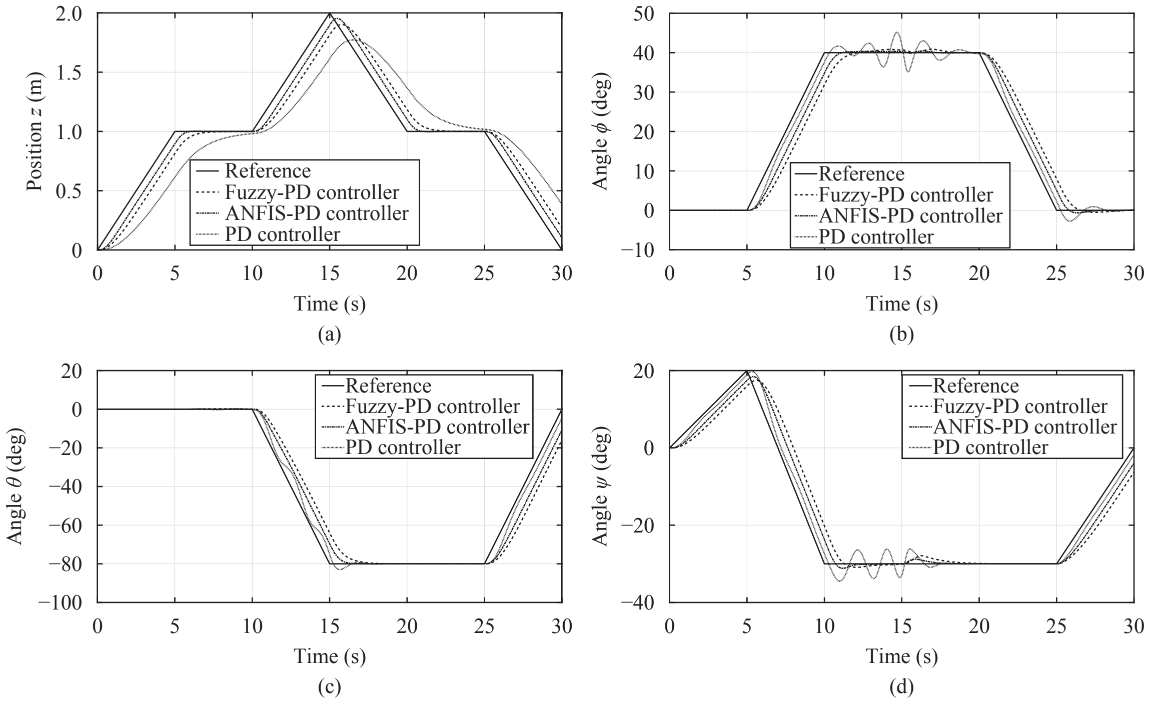


Fig. 12 Simulation results for Case 2: (a) Position z; (b) Angle ϕ; (c) Angle θ; (d) Angle ψ.

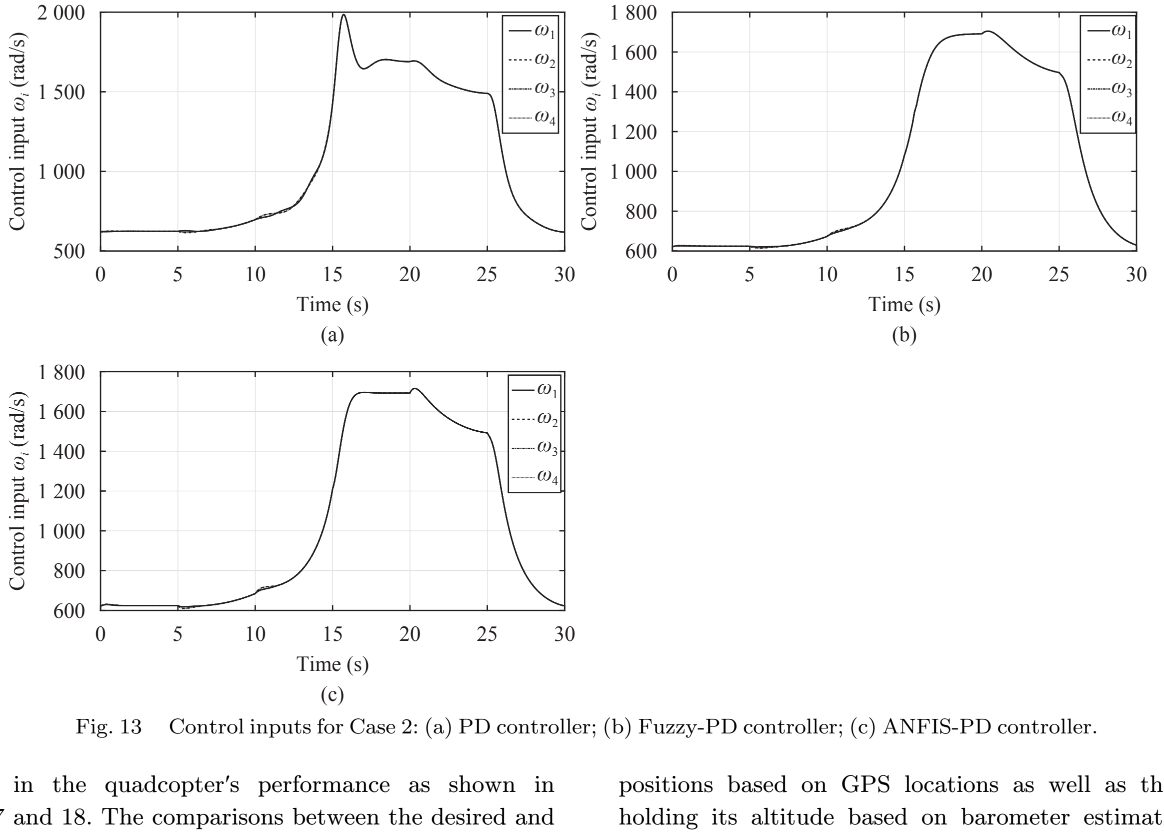


Fig. 13 Control inputs for Case 2: (a) PD controller; (b) Fuzzy-PD controller; (c) ANFIS-PD controller.

results in the quadcopter′s performance as shown in Figs. 17 and 18. The comparisons between the desired and the estimated (current) angles of orientation (i.e., ϕ, θ, and ψ) of the drone are illustrated in Fig. 17. The angles are presented in degrees and the MSEs for each orientation angle are 0.23°, 0.38° and 0.018° respectively.

Secondly, the drone′s performances are tested in loiter mode, where the drone automatically keeps its horizontal positions based on GPS locations as well as the drone holding its altitude based on barometer estimations. In this mode, the position controller sends the desired angles, and height commands to the attitude and the altitude controllers which keep the drone hovering over a certain point. In this case, the attitude and the altitude controllers are primarily responsible for maintaining the drone′s balance. Fig. 19 shows excellent results in atti-

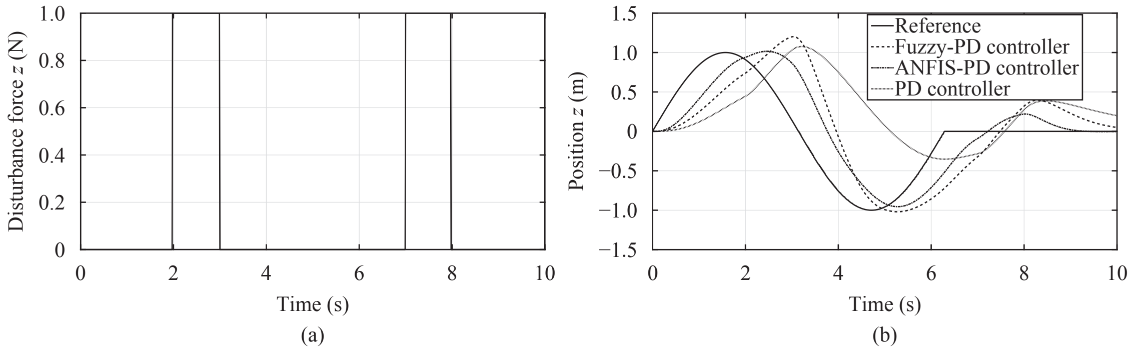


Fig. 14 Simulation results for Case 3: (a) External force; (b) Altitude.

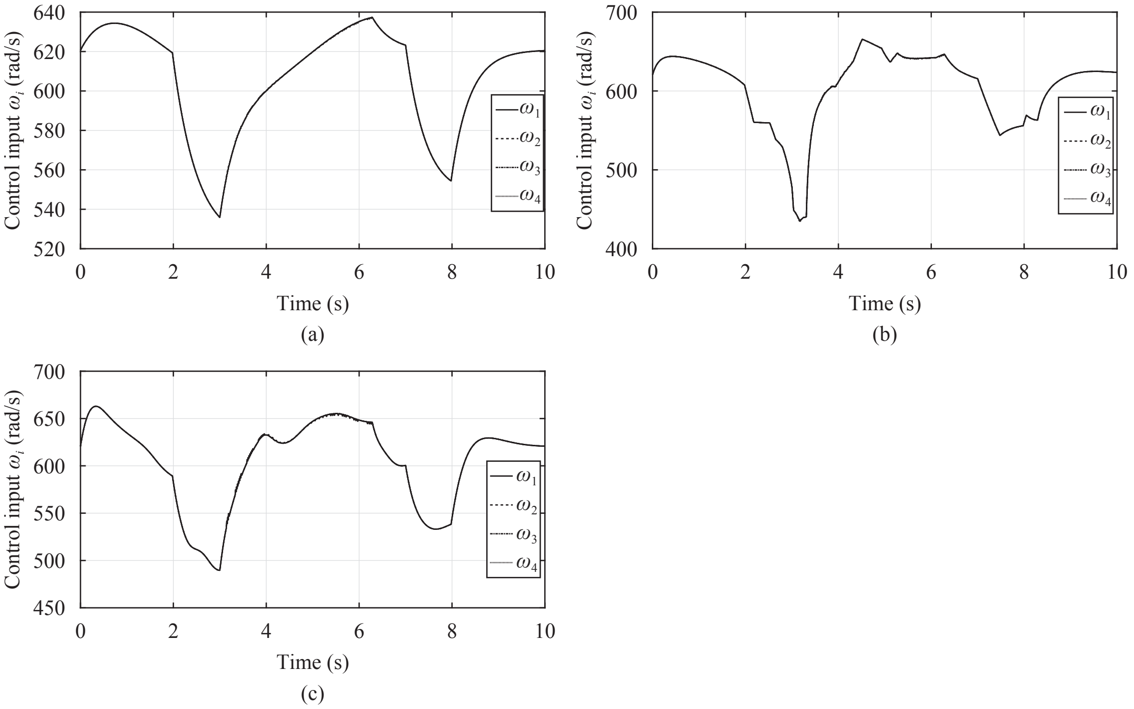


Fig. 15 Control inputs for Case 3: (a) PD controller; (b) Fuzzy-PD controller; (c) ANFIS-PD controller.

Table 7 Case 3: Performance evaluation

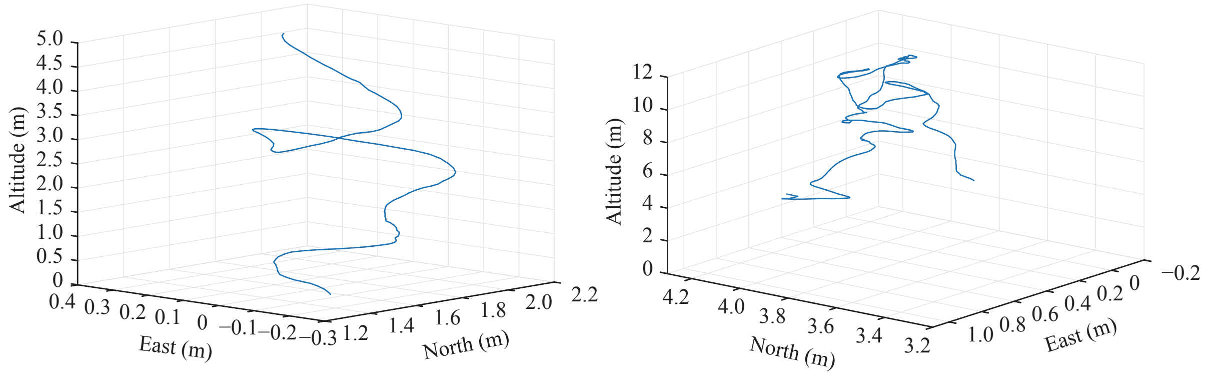
	MSE	MAE
PD controller	0.514 6	0.570 6
Fuzzy-PD controller	0.251 8	0.407 1
ANFIS-PD controller	0.118 6	0.271 1

tude with MSEs less than 1°. Furthermore, the altitude controller responds promptly to the throttle commands to reach the desired altitude with MSE $= 0.12$ m as shown in Fig. 18. The 3-D path of the quadcopter during this experiment is illustrated in Fig. 16. The deviations in the horizontal location are due to the low accuracy provided by GPS.

Thirdly, the proposed ANFIS controller and the estimation system are tested for automatic precision landing. In this experiment, an IR camera in the drone detects an IR beacon on the ground to land over it automatically and precisely. In this experiment, as shown in Fig. 20, the drone takes off and flies in loiter mode to reach a specific altitude. When the camera detects the beacon, the precision landing mode is activated and the horizontal positions of the quadcopter are calculated with respect to the center of the beacon. The quadcopter adjusts its horizontal positions to align, as far as possible, the center of the drone with the center of the beacon. Fig. 20 shows how the drone is landed precisely near to the beacon within less than 20 cm error.

Fig. 21 shows how effectively the attitude and altitude controllers perform in this mode. The overall MSEs for ϕ and θ are 0.13° and 0.34° respectively, and the MSE in altitude is 0.036 m.

Finally, a relatively long-term experiment was performed on the drone, in order to examine the performance of the approved system in changing flight conditions, as well as to identify the maximum capabilities of the drone. In this experiment, the flight duration was about 15 min, including a number of take offs and land-



(a) (b)

Fig. 16 flight tests: (a) Manual flight mode; (b) Loiter flight mode.

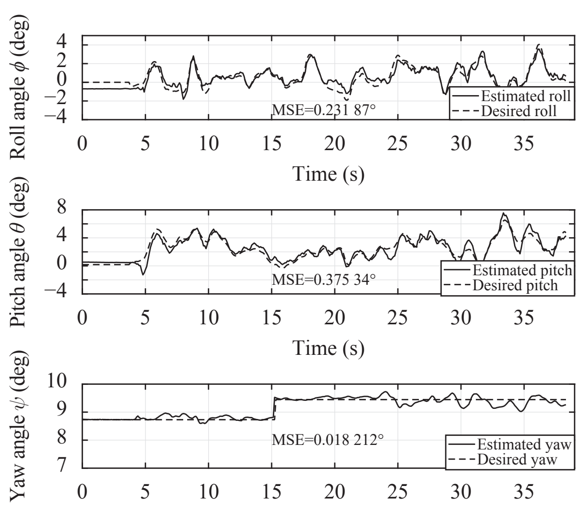


Fig. 17 Quadcopter's attitude in manual mode

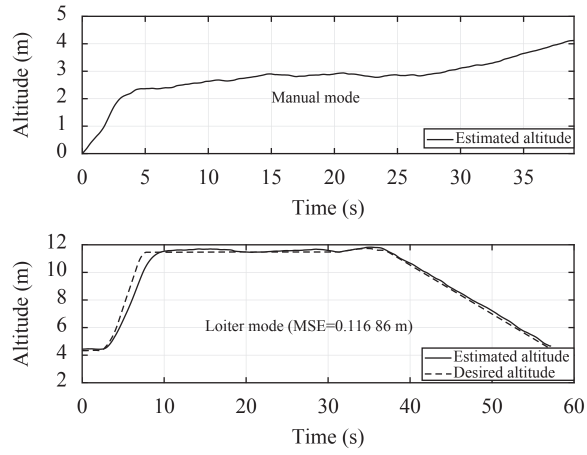


Fig. 18 Quadcopter's attitude in loiter mode

ings, travelling long distances at different speeds, and reaching high altitudes. Also, some intentional variables were introduced to the drone, such as changing the propeller size, adding additional loads, changing the flight modes (i.e., stabilize, loiter, land, auto). In addition, some variables of the surrounding environment were taken into account while analyzing the drone's performance which are listed in Table 8.

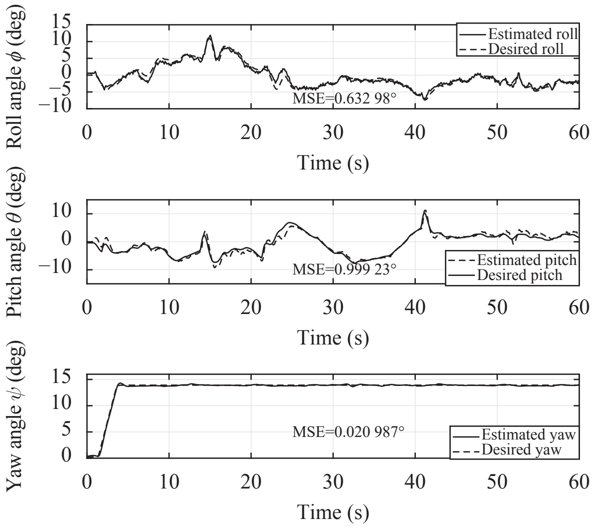


Fig. 19 Quadcopter's altitude in manual and loiter modes

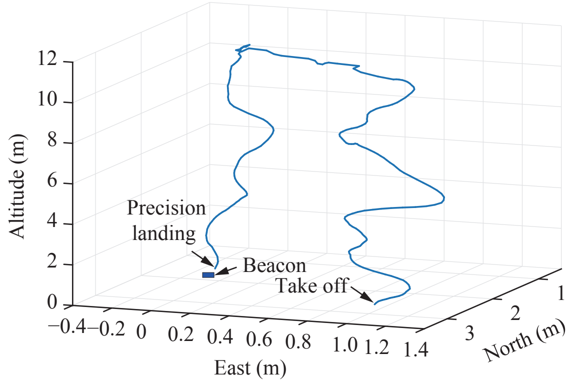


Fig. 20 Quadcopter path in precision landing

During the 15-minute flight experiment, the adopted system approved reliable performance with the various conditions mentioned above. To analyze the performance of the drone, two portions of the flight results were studied during two different periods due to the large amount of information. Figs. 22 and 23 illustrate the drone's velocities and attitude in three directions (North East, and

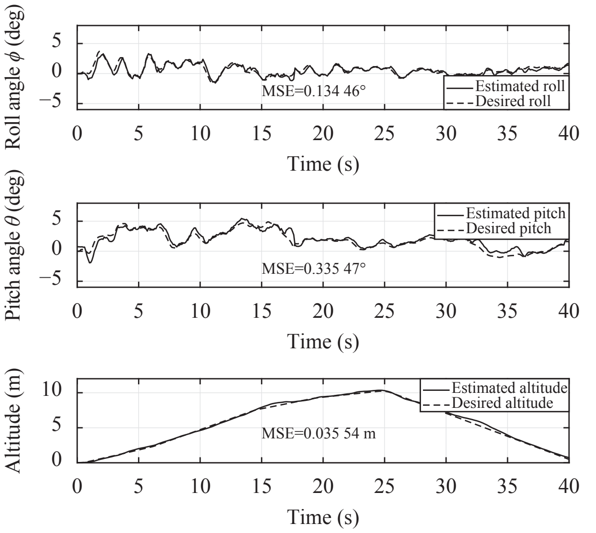


Fig. 21 Attitude and altitude of the quadcopter during precision landing.

Table 8 Environmental conditions during flight time.

	Range
IMU temperature (°C)	59–60
Barometer temperature (°C)	54–60
External temperature (°C)	37–40 forecast
Wind speed (km/h)	15–20 forecast
Number of satellites	6–8

Up) respectively. Fig. 22 shows the drone's ability to travel at high speeds up to 10 m/s in longitudinal directions and up to 4 m/s in vertical directions. Moreover, Fig. 23 proves the stability in large attitudes where roll and pitch angles reach more than 30°, as well as a fast rate in yaw angle.

6 Conclusions

An adaptive neuro-fuzzy control system has been successfully developed to control the altitude and attitude of a quadcopter. Comparing ANFIS with PD and fuzzy controllers, simulation results have clearly demonstrated the feasibility and effectiveness of the proposed control approach in terms of reference tracking and disturbance rejection.

The ANFIS controller outperformed a classical PD controller and a properly tuned fuzzy logic controller in three different yet commonplace scenarios. In fact, the performance of the ANFIS controller developed in this paper can be considered satisfying when compared to the linear controllers reported in [30–32] and the non-linear controllers reported in [7, 33, 34].

This work went beyond simulation to practical application of the quadcopter, where a complete flight system is designed and implemented in a Pixhawk autopilot using a PX4-Simulink support package. The flight system contained ANFIS controllers, attitude and position estimations, and vision-based precision landing algorithms which are successfully deployed in the autopilot. Several flight tests were conducted and the results showed that the controllers performed perfectly in achieving the attitude and altitude stability of the quadcopter. Furthermore, a long flight experiment was conducted, and the performance was analyzed under various conditions. The results prove the efficiency of the adopted system and the flight reliability with integrity of the components.

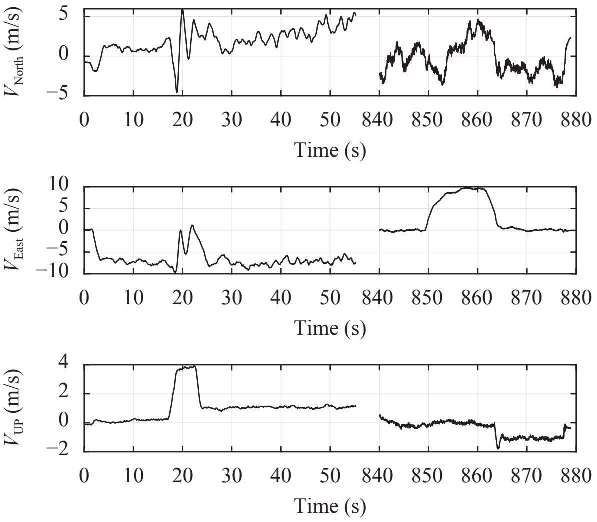


Fig. 22 Drone's velocities in high responsive flight

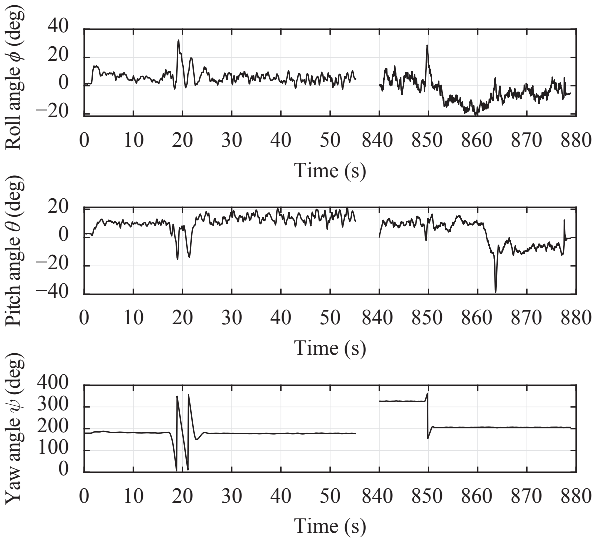


Fig. 23 Drone's attitude in high responsive flight

Although the results of this work demonstrated satisfying performances of the Pixhawk-based quadcopter, running a high-computational demand image algorithms along with the ANFIS controller may reduce the efficiency of the autopilot performance. This issue forces us to simplify the precision landing controller using low-computational IR-lock sensors. Therefore, this leads us to overcome this issue in future work by using an external board, such as a Raspberry Pi, for purposes of handling the computational demand that is required by the image processing techniques. This solution will also allow us to implement the ANFIS position controller for better trajectory tracking and landing.

Future work could also focus on some of the most common commercial applications and uses of quadcopters. Regarding the control system, it is worth exploring the idea of multiple-input multiple-output ANFIS controllers and performing comparisons with other nonlinear controllers often used in flight control systems.

Acknowledgements

The authors acknowledge the Hashemite University for providing the financial and technical supports for this project. Also the authors thank all colleagues and students at Jordan University and at the Hashemite University for their valuable assistance.

Appendix

The followings are the output equations of the ANFIS controller.

$$\begin{aligned}
&out1mf1 = 5.634\,5x^2 + 5.480\,6x + 5.083\,1\\
&out1mf2 = 5.320\,5x^2 + 0.000\,3x + 3.609\,0\\
&out1mf3 = 0.084\,6x^2 + 0.133\,4x + 0.770\,6\\
&out1mf4 = -0.005\,3x^2 - 0.175\,3x + 3.381\,8\\
&out1mf5 = 4.747\,4x^2 + 3.135\,3x - 0.209\,9\\
&out1mf6 = -0.011\,6x^2 + 0.780\,9x - 0.013\,8\\
&out1mf7 = 0.214\,7x^2 + 3.640\,3x + 0.000\,3\\
&out1mf8 = 5.308\,2x^2 + 5.096\,1x + 5.477\,8\\
&out1mf9 = 5.643\,0x^2 - 5.265\,9x - 4.460\,2\\
&out1mf10 = -4.502\,9x^2 - 0.648v4x - 0.000\,0\\
&out1mf11 = -0.046\,4x^2 - 4.593\,6x - 3.522\,2\\
&out1mf12 = -1.401\,7x^2 - 0.037\,3x + 0.286\,8\\
&out1mf13 = -4.390\,6x^2 - 4.718\,0x - 4.194\,7\\
&out1mf14 = 0.267\,4x^2 - 0.055\,7x - 1.400\,1\\
&out1mf15 = -3.186\,8x^2 - 4.706\,5x + 0.015\,2\\
&out1mf16 = -0.000\,0x^2 - 0.663\,9x - 4.523\,0\\
&out1mf17 = -4.455\,1x^2 - 5.264\,9x - 5.370\,3\\
&out1mf18 = -4.768\,8x^2 - 5.217\,9x - 5.332\,9\\
&out1mf19 = -0.000\,3x^2 + 2.697\,6x - 3.885\,6\\
&out1mf20 = -3.205\,6x^2 - 3.354\,0x - 0.066\,9\\
&out1mf21 = 0.014\,9x^2 + 0.249\,7x + 0.027\,9\\
&out1mf22 = -0.320\,4x^2 + 0.232\,4x + 0.078\,6\\
&out1mf23 = 3.607\,6x^2 + 3.107\,2x + 3.861\,0\\
&out1mf24 = -2.793\,5x^2 + 0.000\,3x + 5.321\,0\\
&out1mf25 = 5.202\,8x^2 + 4.742\,2x + 5.343\,5
\end{aligned}\tag{A1}$$

$$\begin{aligned}
&out1mf1 = 0.479\,0x^2 + 0.007\,3x + 0.177\,8\\
&out1mf2 = 0.331\,0x^2 + 0.124\,3x - 0.063\,9\\
&out1mf3 = 0.255\,8x^2 + 0.045\,4x + 0.038\,1\\
&out1mf4 = 0.046\,8x^2 - 0.115\,0x + 0.234\,8\\
&out1mf5 = 0.179\,7x^2 + 0.161\,8x - 0.231\,4\\
&out1mf6 = 0.018\,9x^2 - 0.068\,8x + 0.123\,9\\
&out1mf7 = 0.146\,9x^2 + 0.139\,6x + 0.120\,3\\
&out1mf8 = 0.342\,0x^2 + 0.138\,2x + 0.167\,5\\
&out1mf9 = 0.363\,9x^2 - 1.439\,7x + 0.837\,9\\
&out1mf10 = -0.099\,7x^2 - 0.109\,2x - 0.030\,3\\
&out1mf11 = -1.456\,1x^2 + 0.424\,1x + 0.183\,6\\
&out1mf12 = -0.010\,9x^2 + 0.062\,8x - 0.3723\\
&out1mf13 = 0.272\,7x^2 - 1.450\,6x + 0.351\,5\\
&out1mf14 = -0.505\,7x^2 + 0.011\,3x - 0.068\,0\\
&out1mf15 = 0.307\,7x^2 + 0.338\,7x - 1.143\,6\\
&out1mf16 = -0.026\,5x^2 - 0.090\,3x - 0.242\,3\\
&out1mf17 = 0.982\,8x^2 - 1.354\,2x - 0.325\,0\\
&out1mf18 = -1.040\,9x^2 - 1.155\,8x - 1.833\,3\\
&out1mf19 = -0.695\,1x^2 + 0.029\,1x - 0.406\,3\\
&out1mf20 = -0.495\,0x^2 - 1.425\,0x + 0.017\,0\\
&out1mf21 = 1.050\,1x^2 + 0.036\,5x + 0.004\,3\\
&out1mf22 = -0.038\,4x^2 - 1.071\,0x + 0.259\,5\\
&out1mf23 = 1.181\,9x^2 + 0.322\,1x + 0.329\,0\\
&out1mf24 = -0.236\,4x^2 + 0.660\,1x + 1.837\,9\\
&out1mf25 = 1.115\,7x^2 + 1.027\,9x + 0.249\,9.
\end{aligned}\tag{A2}$$

References

[1] S. Bouabdallah, A. Noth, R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Sendai, Japan, pp. 2451–2456, 2004. DOI: 10.1109/IROS.2004.1389776.

[2] T. Luukkonen. Modeling and control of quadcopter. *Journal of American Society for Mass Spectrometry,* vol. 22, pp. 1134–1145, 2011

[3] P. E. I. Pounds, D. R. Bersak, A. M. Dollar. Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control. *Autonomous Robots*, vol. 33, no. 1–2, pp. 129–142, 2012. DOI: 10.1007/s10514-012-9280-5.

[4] D. K. Tiep, Y. J. Ryoo. An autonomous control of fuzzy-PD controller for quadcopter. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 2, pp. 107–113, 2017. DOI: 10.5391/IJFIS.2017.17.2.107.

[5] F. Santoso, M. A. Garratt, S. G. Anavatti. Fuzzy logic-based self-tuning autopilots for trajectory tracking of a low-cost quadcopter: A comparative study. In *Proceedings of International Conference on Advanced Mechatron-*

ics, Intelligent Manufacture, and Industrial Automation, IEEE, Surabaya, Indonesia, pp. 64–69, 2015. DOI: 10.1109/ICAMIMIA.2015.7508004.

[6] B. E. Demir, R. Bayir, F. Duran. Real-time trajectory tracking of an unmanned aerial vehicle using a self-tuning fuzzy proportional integral derivative controller. *International Journal of Micro Air Vehicles*, vol. 8, no. 4, pp. 252–268, 2016. DOI: 10.1177/1756829316675882.

[7] M. Rabah, A. Rohan, Y. J. Han, S. H. Kim. Design of fuzzy-PID controller for quadcopter trajectory-tracking. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 3, pp. 204–213, 2018. DOI: 10.5391/IJFIS.2018.18.3.204.

[8] P. Garcia-Aunon, M. S. Peñas, J. M. de la Cruz García. Parameter selection based on fuzzy logic to improve UAV path-following algorithms. *Journal of Applied Logic*, vol. 24, pp. 62–75, 2017. DOI: 10.1016/j.jal.2016.11.025.

[9] E. Kayacan, R. Maslim. Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions. *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 339–348, 2017. DOI: 10.1109/TMECH.2016.2614672.

[10] A. Dorzhigulov, B. Bissengaliuly, B. F. Spencer Jr, J. Kim, A. P. James. ANFIS based quadrotor drone altitude control implementation on raspberry pi platform. *Analog Integrated Circuits and Signal Processing*, vol. 95, no. 3, pp. 435–445, 2018. DOI: 10.1007/s10470-018-1159-8.

[11] D. Domingos, G. Camargo, F. Gomide. Autonomous fuzzy control and navigation of quadcopters. *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 73–78, 2016. DOI: 10.1016/j.ifacol.2016.07.092.

[12] S. Kundu, D. R. Parhi. Reactive navigation of underwater mobile robot using ANFIS approach in a manifold manner. *International Journal of Automation and Computing*, vol. 14, no. 3, pp. 307–320, 2017. DOI: 10.1007/s11633-016-0983-5.

[13] O. Salah, A. A. Ramadan, S. Sessa, A. A. Ismail, M. Fujie, A. Takanishi. ANFIS-based sensor fusion system of sit- to-stand for elderly people assistive device protocols. *International Journal of Automation and Computing*, vol. 10, no. 5, pp. 405–413, 2013. DOI: 10.1007/s11633-013-0737-6.

[14] M. Gheisarnejad, M. H. Khooban. Supervised control strategy in trajectory tracking for a wheeled mobile robot. *IET Collaborative Intelligent Manufacturing*, vol. 1, no. 1, pp. 3–9, 2019. DOI: 10.1049/iet-cim.2018.0007.

[15] M. Gheisarnejad, M. H. Khooban, J. Boudjadar. Adaptive network based fuzzy inference system for frequency regulation in modern maritime power systems. In *Proceedings of the 5th International forum on Research and Technology for Society and Industry*, IEEE, Florence, Italy, pp. 302–307, 2019. DOI: 10.1109/RTSI.2019.8895596.

[16] M. Gheisarnejad, P. Karimaghaee, J. Boudjadar, M. H. Khooban. Real-time cellular wireless sensor testbed for frequency regulation in smart grids. *IEEE Sensors Journal*, vol. 19, no. 23, pp. 11656–11665, 2019. DOI: 10.1109/JSEN.2019.2934599.

[17] M. Panda, B. Das, B. Subudhi, B. B. Pati. A comprehensive review of path planning algorithms for autonomous underwater vehicles. *International Journal of Automation and Computing*, vol. 17, no. 3, pp. 321–352, 2020. DOI: 10.1007/s11633-019-1204-9.

[18] F. Santoso, M. A. Garratt, S. G. Anavatti. Adaptive neuro-fuzzy inference system identification for the dynamics of the AR.drone quadcopter. In *Proceedings of International Conference on Sustainable Energy Engineering and Application*, IEEE, Jakarta, Indonesia, pp. 55−60, 2016. DOI: 10.1109/ICSEEA.2016.7873567.

[19] S. Rezazadeh, M. A. Ardestani, P. S. Sadeghi. Optimal attitude control of a quadrotor UAV using adaptive neuro-fuzzy inference system (ANFIS). In *Proceedings of the 3rd International Conference on Control, Instrumentation, and Automation*, IEEE, Tehran, Iran, pp. 219−223, 2013. DOI: 10.1109/ICCIAutom.2013.6912838.

[20] S. Khatoon, I. Nasiruddin, M. Shahid. Design and simulation of a hybrid PD-ANFIS controller for attitude tracking control of a quadrotor UAV. *Arabian Journal for Science and Engineering*, vol. 42, no. 12, pp. 5211–5229, 2017. DOI: 10.1007/s13369-017-2586-z.

[21] T. Bresciani. Modelling, Identification and Control of a Quadrotor Helicopter, Master dissertation, Lund University, Sweden, 2008.

[22] D. Kotarski, Z. Benić, M. Krznar. Control design for unmanned aerial vehicles with four rotors. *Interdisciplinary Description of Complex Systems*: *INDECS*, vol. 14, no. 2, pp. 236–245, 2016.

[23] N. S. Bao, X. Ran, Z. F. Wu, Y. F. Xue, K. Y. Wang. Research on attitude controller of quadcopter based on cascade PID control algorithm. In *Proceedings of the 2nd Information Technology, Networking, Electronic and Automation Control Conference*, IEEE, Chengdu, China, pp. 1493−1497, 2017. DOI: 10.1109/ITNEC.2017.8285044.

[24] G. Ononiwu, O. Onojo, O. Ozioko, O. Nosiri. Quadcopter design for payload delivery. *Journal of Computer and Communications*, vol. 4, no. 10, pp. 1–12, 2016. DOI: 10.4236/jcc.2016.410001.

[25] E. Kuantama, T. Vesselenyi, S. Dzitac, R. Tarca. PID and fuzzy-PID control model for quadcopter attitude with disturbance parameter. *International Journal of Computers Communications & Control*, vol. 12, no. 4, pp. 519–532, 2017. DOI: 10.15837/ijccc.2017.4.2962.

[26] M. Hayajneh, M. Melega, L. Marconi. Design of autonomous smartphone based quadrotor and implementation of navigation and guidance systems. *Mechatronics*, vol. 49, pp. 119–133, 2018. DOI: 10.1016/j.mechatronics.2017.11.012.

[27] M. R. Hayajneh, A. R. E. Badawi. Automatic UAV wireless charging over solar vehicle to enable frequent flight missions. In *Proceedings of the 3rd International Conference on Automation, Control and Robots*, Prague, Czech Republic, pp. 44−49, 2019.

[28] MATLAB. Px4 autopilots support from embedded coder. [Online], Available: https://www.mathworks.com/hardware-support/px4-autopilots.html, 2020.

[29] T-motor air 2213 920kv brushless motor, [Online], Available: http://dekasto.com/catalog/view/364.

[30] C. S. Subudhi, D. Ezhilarasi. Modeling and trajectory tracking with cascaded PD controller for quadrotor. *Procedia Computer Science*, vol. 133, pp. 952–959, 2018. DOI: 10.1016/j.procs.2018.07.082.

[31] G. Szafranski, R. Czyba. Different approaches of PID control UAV type quadrotor. In *Proceedings of International Micro Air Vehicle Conference and Flight Competition*, Delft, Netherlands, 2011.

[32] Z. Mustapa, S. Saat, S. H. Husin, N. Abas. Altitude controller design for multi-copter UAV. In *Proceedings of International Conference on Computer, Communications, and Control Technology*, IEEE, Langkawi, Malaysia, pp. 382–387, 2014. DOI: 10.1109/I4CT.2014.6914210.

[33] J. A. Paredes, C. Jacinto, R. Ramírez, I. Vargas, L. Trujillano. Simplified fuzzy-PD controller for behavior mixing and improved performance in quadcopter attitude control systems. In *Proceedings of IEEE ANDESCON*, IEEE, Arequipa, Peru, 2016. DOI: 10.1109/ANDESCON.2016.7836217.

[34] E. S. Filatova, A. V. Devyatkin, A. I. Fridrix. UAV fuzzy logic stabilization system. In *Proceedings of IEEE International Conference on Soft Computing and Measurements*, IEEE, St. Petersburg, Russia, pp. 132–134, 2017. DOI: 10.1109/SCM.2017.7970517.

Mohammad Al-Fetyani received the B. Sc. degree in mechatronics engineering with an excellent grade at The University of Jordan, Jordan in 2020. He is currently working as a research assistant at The University of Jordan and intends to complete his postgraduate studies.

His research interests include robotics and intelligent systems.

E-mail:m4bh@hotmail.com

ORCID iD: 0000-0002-7360-2403

Mohammad Hayajneh received the B. Sc. and M. Sc. degrees in mechatronics engineering from Jordan University of Science and Technology, Jordan in 2010 and 2012. He received the Ph. D. degree in automatic control and operational research from University of Bologna, Italy in 2016. He is currently an assistant professor in mechatronics engineering at The Hashemite University, Jordan.

His research interests include the design and development of control and navigation methods in aerial and ground robots and their applications.

E-mail: mhayajneh@hu.edu.jo (Corresponding author)

ORCID iD: 0000-0001-6869-1862

Adham Alsharkawi received the B. Sc. degree in machatronics engineering from Tafila Technical University, Jordan in 2010. He received the M. Sc. degree in advanced control and system engineering from University of Manchester, UK in 2013. He received the Ph. D. degree in automatic control and systems engineering from University of Sheffield, UK in 2017. He is currently an assistant professor in Mechatronics Engineering Department, The University of Jordan, Jordan. He has been working in projects related to wheeled mobile robots, quadcopters, and solar thermal power plants.

His interests include system dynamics, automatic control, artificial intelligence, and solar energy.

E-mail: a.sharkawi@ju.edu.jo