Arabian Journal for Science and Engineering (2019) 44:2627–2639
https://doi.org/10.1007/s13369-018-3330-z

RESEARCH ARTICLE - SYSTEMS ENGINEERING

Fuzzy Logic-Based Robust and Autonomous Safe Landing for UAV Quadcopter

Muhammad Talha[1] · **Furqan Asghar**[2] · **Ali Rohan**[1] · **Mohammed Rabah**[1] · **Sung Ho Kim**[1]

Received: 29 November 2017 / Accepted: 17 May 2018 / Published online: 6 June 2018

Abstract
Recent years, unmanned aerial vehicles (UAVs) have been used to perform various tasks such as surveillance, monitoring, rescue, photography, and security. Quadcopter configuration of UAV is most common due to its simplicity, stability, and versatile controllability. Quadcopter landing is most crucial part of the overall operating process. It needs a lot of practice and effort for a safe landing. Therefore, a fuzzy logic-based safe landing system is proposed in this paper. This paper includes a landing system based on laser rangefinder, Arduino Mega MCU, and Pixhawk flight controller. The lookup table technique is used to implement fuzzy logic inside Arduino Mega. This technique takes very small execution time for data processing in fuzzy logic, which is essential for high-speed data processing and updating. Autonomous landing process can be triggered and override at any time using remote controller. Furthermore, various tests are performed on a quadcopter to verify the feasibility of proposed algorithm.

Keywords UAV automation · Fuzzy logic · Quadcopter auto-landing · Pixhawk · Arduino Mega

1 Introduction

Quadcopter, also known as quadrotor helicopter, is a type of UAV that consists of four upward rotors which help quadcopter for any kind of maneuvers within its flying region. Radiofrequency (RF)-based remote controller (RC) is used for attitude control. Operating range of quadcopter depends on the maximum RC range. Autonomous UAVs have been hot research topic in recent years, and autonomous control systems are being developed for deploying weapons, reconnaissance, surveillance, coastal security, and other hazardous tasks. In past years, many researches are done for UAV automation [1–3]. Global positioning system (GPS) combined with inertial measurement unit (IMU) sensors is most commonly used for navigation and automatic flight control which provides the information about UAV position with 7.8 m accuracy.

Automatic flight control system can be divided into three parts: (1) takeoff, (2) task to perform, and (3) landing. Safe takeoff does not require any complicated procedure; it just needs roll and pitch value to stabilize the UAV quadcopter and altitude information to achieve the desired altitude; next step is to increase the throttle only to reach the required altitude. The second step is performing a required task such as surveillance, monitoring, or moving to some other destination, and these tasks can be performed using GPS information. A lot of techniques and strategies have been developed by several researchers in this area. UAV quadcopter landing is most tricky part of automation or manual operation. Only a trained operator can land a UAV quadcopter safely, or it needs very precise and efficient algorithm for autonomous landing. Generally, a cushion of air builds up underneath the craft as it lands, and this has to be spilled, by slowing down the rotors at a controlled rate, before it can settle on the ground. Any mistakes tend to lead to the helicopter tipping over, which is catastrophic, as a large amount of energy stored in the spinning blades will then be dissipated by smashing the whole craft (and probably any equipment on board) to pieces. GPS data are not sufficient to provide required precision. Therefore, some additional sensors are required such as camera,

---

✉ Muhammad Talha
engr.talha72@gmail.com

Furqan Asghar
furqan.asghar@hotmail.com

[1] School of electronics and information engineering, Kunsan National University, Kunsan, Republic of Korea

[2] School of electrical engineering, The University of Faisalabad, Faisalabad, Punjab, Pakistan

LIDAR, to measure correct position on the platform and altitude of a UAV [4]. The camera is responsible for detecting docking station or landing platform, whereas LIDAR calculates the altitude of UAV [5].

Recently, there have been two different approaches used for safe landing, (1) position control approach and (2) velocity control approach. Position control approach sends altitude commands slightly below the current altitude until UAV reaches ground level [6]. Lidar or sonar sensors are used to get altitude information. Velocity control technique involves a constant downhill velocity command mechanism until the ground level is reached [7,8]. Some other researchers have proposed computer vision-based landing platform detection [9,10]. A camera is mounted on UAV to detect the landing platform, and the auto-landing process is triggered after successful detection of landing platform [11]. GPS-based docking system is also proposed by few researchers [12]. In this approach, GPS is used to detect the position of docking station using the coordinates. However, GPS accuracy is limited to 7.8 m (4 m RMS); therefore, a vision-based landing platform detection system is used from that point. The auto-landing process is initiated when landing platform is successfully detected [13,14].

Autonomous landing and takeoff require some important factors to be taken into consideration.

- Accurate measurement of altitude between UAV and ground platform.
- Accurate measurement of landing velocity to avoid high-velocity impact with the ground.

Moreover, conventional auto-landing (position control and velocity control) techniques have some drawbacks. Position control technique offers a secure landing approach but landing process is too slow, which leads to high power consumption issue, whereas velocity control approach does not provide any safety assurance and quick velocity decrease might crash the UAV. In [15], a Kalman filter-based auto-landing system is proposed. This system used Kalman filter to estimate the position and velocity of quadcopter with respect to target. This technique is quite effective but Kalman filter implementation is quite hard, and it takes large computational time to provide an estimation value, whereas in quadcopter systems, we need to make the landing process as quick as possible to save the battery life for further operations. In [16,17], a vision-based system is proposed for auto-landing. This system lacks of effectiveness in real-time application due it time-consuming image processing algorithms. Therefore, an efficient closed-loop control system is required to assure safe and quick landing process. PID controllers are most common controllers to be used, but a fixed gain of PID controllers cannot provide immediate response to overcome nonlinear thrust effect (ground effect) with decreasing altitude and gravity in landing process. Thrust strength directly depends on altitude, thrust effects faster at a lower altitude and slower at high altitude. Moreover, PID gain tuning is a crucial part and needs a lot of effort for optimal gain tuning.

Therefore, considering aforementioned drawbacks, a fuzzy logic-based optimized landing controller is proposed in this paper. The proposed fuzzy logic controller has its own rule base which can successfully eliminate the nonlinear ground effect. Therefore, the proposed fuzzy rule which can be considered as a hybrid of position and velocity control algorithm provides a fast and autonomous landing performance.

In Sect. 2, the practical configuration of our test system is discussed in detail. Section 3 explains the conventional landing control algorithms and some important parameters needed to be considered during controller design. The proposed fuzzy logic landing technique is explained in Sect. 4 in detail. Section 5 includes simulation and experimental results of the proposed technique and comparison with ordinary PID controller.

2 Configuration of Safe Landing System for Quadcopter

The quadcopter UAV used in this research consists of the quadcopter, Pixhawk-v2 flight controller, Arduino Mega MCU, which implements the fuzzy logic controller and LIDAR-Lite-v3 rangefinder for measuring the altitude of the quadcopter. Block diagram of the overall system is shown in Fig. 1.

2.1 Pixhawk Flight Controller

Pixhawk is ARM Cortex MCU base flight controller that works as the brain of UAV and controls the attitude and stability of quadcopter. Pixhawk is fully equipped flight controller and is capable of interfacing with a variety of sensors and modules such as GPS, RF modules, barometer, IMU sensors, compass, LIDAR, camera. This controller can handle up to eight electronic velocity controllers (ESCs) and

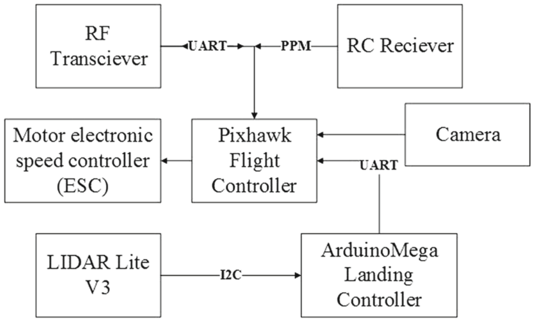


Fig. 1 Block diagram of the proposed system

is also compatible with various UAV configurations, i.e., helicopter, tri-copter, quadcopter, and hex-copter. This controller has Arducopter open-source code program written in C++, which is free to use and modify. Pixhawk controller has five UART ports for communication and telemetry. Mission planner and QGroundControl are the software used for calibration, telemetry, minor gain tunings, and other purposes. A Linux-based operating system Ubuntu is used for code editing and programming the pixhawk MCU. UART-E of pixhawk is used for communication between pixhawk flight controller and Arduino Mega landing controller.

2.1.1 Arduino Mega Landing Controller

Arduino Mega MCU with LIDAR-Lite-v3 rangefinder is used to implement fuzzy logic-based auto-landing mechanism. LIDAR is a low-power and lightweight 40-m rangefinder. This rangefinder is used due to its lower noise, high efficiency, and high range. LIDAR is installed under the quadcopter facing down to measure the altitude. Arduino Mega is programmed to calculate altitude and vertical velocity of the quadcopter. These two parameters are input parameters of the fuzzy logic controller. The fuzzy logic controller outputs throttle adjustment value, which is transferred to flight controller via UART communication protocol.

3 Conventional Landing Control Algorithms

Quadcopter landing is most crucial part in quadcopter automation due to various factors such as gravitational pull, thrust variations, payload, ground effect, and wind effect. All landing controllers (position control, velocity control) directly or indirectly control the thrust values for autonomous landing. Selection of proper thrust value for a certain quadcopter is very important for proper flight and control [18]. Some important factors need to be considered carefully to design auto-landing algorithm.

3.1 Ground Effect

In all types of UAVs, ground effect is the increased lift (force) near the ground in comparison with a higher altitude. UAV vertical drag with respect to altitude can be divided into two parts (1) IGE (in ground effect) and (2) OGE (out of ground effect). IGE is a condition where the downwash of air from the main rotor is able to react with a hard surface (the ground) and give a useful reaction to the helicopter in the form of more lift force available with less engine power required. OGE is the opposite of the above, where there are no hard surfaces for the downwash to react against. For example, a UAV hovering 150 ft above the ground will be in an OGE condition and will require more power to maintain a constant altitude than if it was hovering at 15 ft. Hence, a helicopter will always have a lower OGE ceiling than IGE due to the amount of engine power available [19].

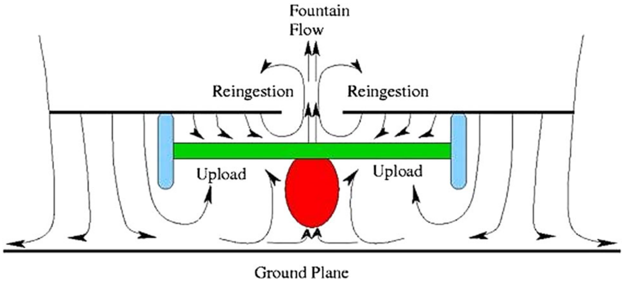


Fig. 2 Quadcopter thrust effect under IGE

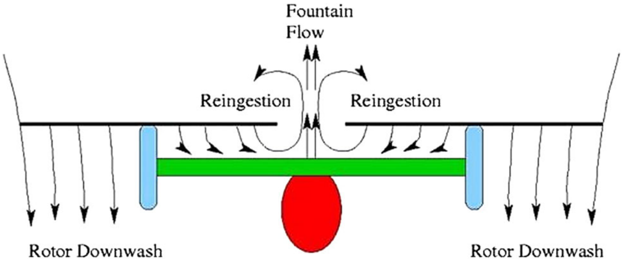


Fig. 3 Quadcopter thurst effect under OGE

IGE and OGE effect in quadcopter UAV systems is shown in Figs. 2 and 3. IGE effect is a most important issue to be considered while designing an auto-landing controller mechanism. Therefore, the designed controller must be intelligent enough to overcome this scenario during the landing process. Ordinary PID controllers are not capable of handling such effect due to its fixed gains. Therefore, quadcopter starts hovering at a certain height from the ground and landing process gets too slow, which is not efficient and consumes a lot of battery power.

3.2 Conventional Landing Controllers

There are two main types of conventional landing control techniques described earlier in Sect. 1: (1) position control and (2) velocity control.

Position control technique uses position commands and sends slightly lower position reference than the previous position until the ground level is approached. This control technique is safe and does not require any intelligent control algorithm to be implemented. However, this landing approach is too slow because there is no velocity control option available. Furthermore, some safety issues are also present because there is no option to monitor landing velocity. If the landing velocity is too fast, quadcopter might get crashed or too slow process can make the drone hovering above the ground. The velocity control technique constantly transmits downhill velocity commands until quadcopter reaches ground safely. Similar kinds of drawbacks are present in this case such as safety issues and slow landing. PID controller is another option to make landing controller. PID controller uses distance data from LIDAR sensors and controls the throttle values until distance reaches zero. This controller is quite safer than the previous control techniques. However, PID gain tuning is a challenging task; optimum gain tuning is very hard to get the best response. Moreover, PID controllers have fixed gains, which make the landing process very slow when quadcopter reaches near the ground due to IGE. This makes the quadcopter hover at some altitude near the ground, and landing process gets very slow from this point. Therefore, an intelligent controller is needed to make the landing process fast and safe for maximum efficiency. Fuzzy logic-based landing control technique is proposed is Sect. 4.

4 Fuzzy Logic-Based Landing Controller

The fuzzy logic was proposed by R. Lotfi Zadeh of the University of California at Berkeley in the 1960s. Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1. It is employed to handle the concept of partial truth, where the truth value may range between completely true and false. By contrast, in Boolean logic, the truth values of variables may only be the integer values 0 or 1. Furthermore, when linguistic variables are used, these degrees may be managed by specific (membership) functions. This controller works on the basis of the degree of truth. The fuzzy logic process is performed in three steps named as fuzzification, rule base, and defuzzification.

4.1 Control Methodology

The fuzzy logic controller is best option to perform auto-landing operation due to its time-optimal control over PID controllers. The fuzzy logic can handle the complex control situation with great precision.

A fuzzy logic-based landing controller is designed and proposed with hybrid position and velocity control algorithm. The proposed fuzzy logic controller consists of two inputs and one output. Altitude and vertical velocity are taken as inputs, and throttle adjustment is taken as output for this fuzzy logic controller as shown in Fig. 4. Throttle adjustment values are added or subtracted from throttle percentage value inside flight controller control to adjust the landing velocity and direction. Equation 1 describes the throttle adjustment commands interaction with actual throttle values.

$$\text{Trottle}_{\text{input}} = \text{Trottle}_{\text{pre}} \pm \text{Trottle}_{\text{adj}} \qquad (1)$$

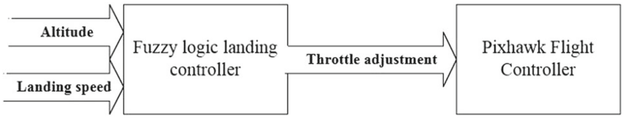


Fig. 4 Block diagram for fuzzy logic auto-landing controller

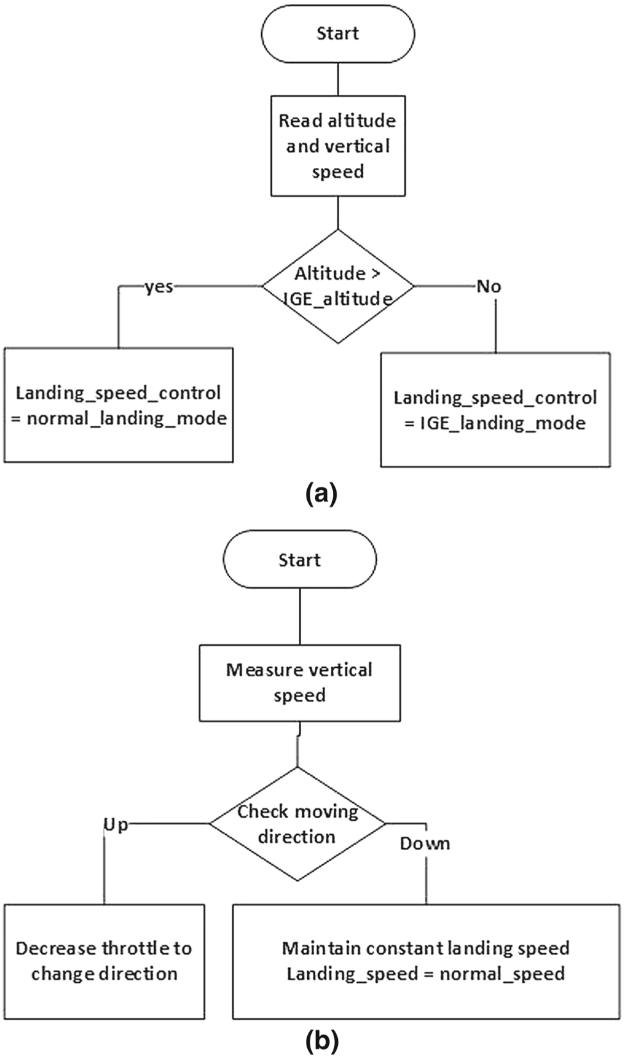


Fig. 5 **a** Flowchart for altitude control. **b** Flowchart for landing velocity control

where, $\text{Trottle}_{\text{input}}$ = Throttle value to flight controller; $\text{Trottle}_{\text{pre}}$ = Previous throttle input; $\text{Trottle}_{\text{adj}}$ = Throttle adjustments values from landing controller.

Each input and output parameter has five membership functions. The universe of discourse of each parameter should be carefully selected regarding minimum and maximum limits. IGE is a most important factor to be considered during the designing process of the fuzzy rule base. An IGE threshold altitude level is chosen, and fuzzy logic landing control will increase the throttle adjustments for smooth landing when IGE threshold is reached. Flowchart for fuzzy logic-based landing control is shown in Fig. 5. Designed landing system can be divided into two parts, (1) altitude control (Fig. 5a) and (2) landing velocity control (Fig. 5b).

Altitude control is similar to position control which keeps measuring its current altitude using LIDAR rangefinder and input data to fuzzy logic. The fuzzy logic rule base is designed to generate throttle adjustment commands to control the landing velocity of the quadcopter. Landing velocity control is same as velocity control technique, which uses landing velocity input and altitude information to generate throttle adjustment values. The vertical velocity of the quadcopter and moving direction are measured using LIDAR inside Arduino Mega MCU at every 200 ms.

These data are then inputted to the fuzzy controller, which checks the moving direction and velocity and generates output throttle adjustment commands to make the landing direction downwards and keep the landing velocity constant. Fuzzy logic landing algorithm can be described as shown in Fig. 5. The auto-landing process is triggered via RC whenever required, and it can be overridden at any time. When the auto-landing is triggered, landing controller measures the altitude, vertical velocity, and direction. These parameters are inputted to the fuzzy logic controller. The fuzzy logic system checks the altitude, vertical velocity, and direction, and it generates the throttle adjustment commands to move the quadcopter downwards with constant velocity. If the altitude reaches its IGE threshold limit, some special commands are generated to make the landing process faster.

The fuzzy logic rule base is designed to control the landing velocity, direction, and IGE during the auto-landing process. When the auto-landing process is triggered via RC, the fuzzy logic controller measures altitude, vertical velocity, and vertical direction of the quadcopter. The following are the control algorithms which fuzzy logic will run during the auto-landing process.

- If a quadcopter is above IGE threshold, normal landing process will run.
- The controller will check the moving direction of the quadcopter, if quadcopter is moving upwards, it will send high negative throttle adjustment values to change the direction.
- Landing velocity will be set to a constant value, if landing velocity increases or decreases, the fuzzy logic controller will use throttle adjustments to make it constant.
- When quadcopter reaches IGE range, high gain landing commands will be triggered to overcome the IGE.
- Throttle value will be set to zero as quadcopter reaches the ground.

4.2 Controller Design

The fuzzy logic controller is a complex mathematical controller, which requires a lot of effort to implement on MCU. The main problem in a fuzzy logic implementation using MCU is the long execution time. Its long execution cycle delays the output, which slows down the response. In the landing process, we need very fast execution time, which a normal fuzzy logic controller cannot produce. Therefore, a lookup table technique is used to implement the fuzzy logic on Arduino Mega MCU. This technique produces very fast execution due to very simple mathematics. This implementation technique for fuzzy logic is divided into two parts: (1) a fuzzy logic controller design in MATLAB/Simulink and (2) acquisition of input and output data to build lookup table. The complete process is described below in detail.

4.2.1 Fuzzy-Based Controller Design in MATLAB/Simulink

Fuzzy input–output membership functions and rule base are designed in MATLAB/Simulink using the Simulink fuzzy toolbox. The universe of each input and output is carefully selected according to desired ranges. This fuzzy logic controller has two inputs and one output, and 30 rules are made for optimum landing control.

Input and output membership functions are shown in Fig. 6.

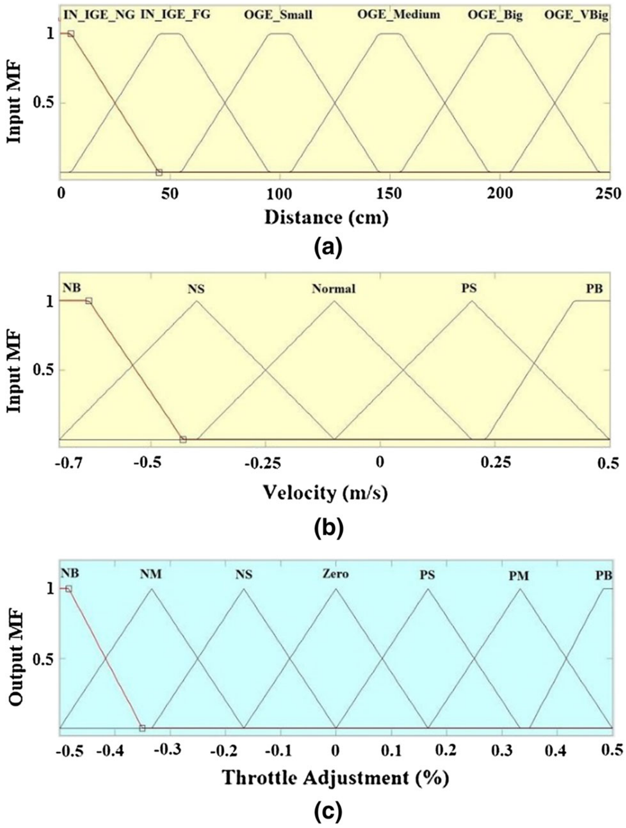


Fig. 6 Input and output membership functions **a** distance, **b** velocity, **c** throttle adjustment

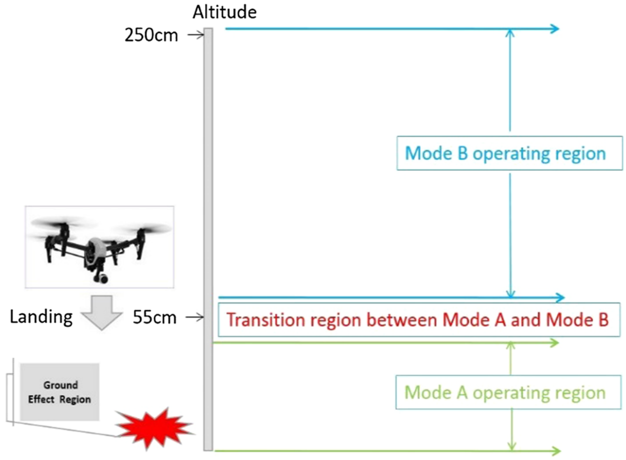


Fig. 7 Operating regions of proposed fuzzy logic landing controller

The universe of discourse of each membership function defines the operation range of those specific fuzzy linguistic variables. These ranges are adjustable and act similar to PID gains. These fuzzy linguistic variables are defined as IN_IGE_NG (inside IGE range near ground), IN_IGE_FG (inside IGE range far ground), OGE_Small, OGE_Medium, OGE_Big and OGE_VBig for the corresponding distance input as in Fig. 6a. Fuzzy linguistic variables for the second input in vertical velocity are defined as NB (negative big), NS (negative small), normal, PS (positive small), PB (positive big) as shown in Fig 6b. Velocity input parameters are the most important parameter in this controller which controls the landing velocity and direction at the same time. The fuzzy set "Normal" indicates the normal landing velocity for safe landing. Whereas in Fig. 6c, NB, NM, NS and PS, PM, PB are fuzzy linguistic functions for landing velocity input. Negative velocity like NB and NS indicates high-speed landing velocities, and positive velocity indicates the upward moment of the quadcopter. Equation (2) explains landing velocity calculation formula.

$$\mathrm{Vel}_{\mathrm{land}} = \frac{\mathrm{Alt}_{\mathrm{current}} - \mathrm{Alt}_{\mathrm{pre}}}{\Delta t} \tag{2}$$

where $\mathrm{Vel}_{\mathrm{land}}$ = Landing velocity; $\mathrm{Alt}_{\mathrm{current}}$ = Current altitude; $\mathrm{Alt}_{\mathrm{pre}}$ = Previous altitude; Δt = time difference.

Distance range is taken from 0–250 cm, and any value above 250 cm will be taken as 250 cm for fuzzy logic input. IGE range is fixed to 100 cm for this quadcopter system. IGE range depends on quadcopter size and value of thrust produced by its propellers, and the IGE range will increase for high thrust-producing quadcopters. The IGE range is adjustable, and it can be set according to the quadcopter specification. In this designed landing controller, IGE is divided into two separate operating regions. First is IN_IGE_FG and second is IN_IGE_NG; IGE starts getting effective from the boundary region of IN_IGE_FG, and this affect keeps on increasing exponentially as altitude decreases. Two-step division of IGE expands the control option to make the controller precise and reliable over different scenarios. Landing velocity ranges from -0.7 to 0.5 m/s, the "±" sign indicates moving direction, negative means moving downward, and positive means moving upward. Throttle adjustments are throttle percentage values that need to be added or subtracted from actual throttle percentage in flight controller ranging from -0.5 to 0.5. These ranges for each parameter can be changed in fuzzy logic, or gains can be used for more precise tuning. Making the rules base is most crucial part of the fuzzy logic controller, and the whole function of this controller depends on the rule base. Table 1 shows the 30 rules designed to control the auto-landing operation. The ordinary fuzzy logic controller works similar to PID controller with fixed gains. Therefore, to implement the proposed landing algorithm, the rule base is modified to overcome the PID controller drawbacks. This rule base works in two different modes, Mode A and Mode B.

Mode A This is normal operation mode when quadcopter altitude is more than IGE range. Here, vertical velocity and direction of the quadcopter are monitored and controlled to land with a constant velocity (-0.2 m/s) within the normal range. In this case, throttle adjustment values work between NS to PS unless until any uncertain change in velocity happens; such as quadcopter is landing with very high velocity or quadcopter is going upward as shown in Table 1(A). NM and

Table 1 Fuzzy logic rule base for safe landing

Velocity	Altitude					
	A		B			
	IN_IGE_NG	IN_IGE_FG	OGE_Small	OGE_Medium	OGE_Big	OGE_VBig
NB	PB	PB	PM	PM	PM	PM
NS	PB	PM	PS	PS	PS	PS
Normal	Zero	Zero	Zero	Zero	Zero	Zero
PS	NB	NM	NS	NS	NS	NS
PB	NB	NB	NM	NM	NM	NM

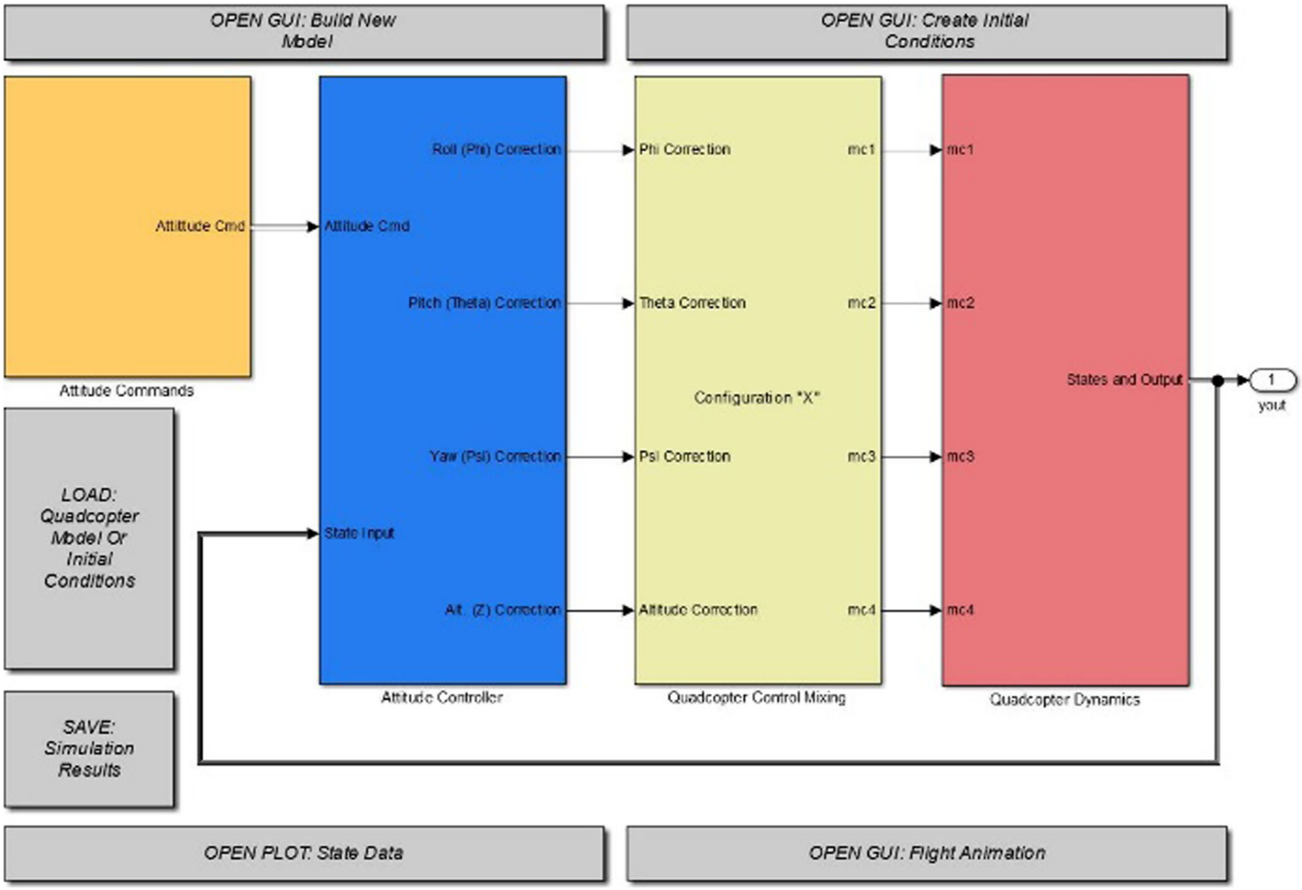


Fig. 8 Quadcopter simulation system

PM will be used to change the landing velocity and direction in this case.

Mode B The second scenario is when quadcopter is within IGE range, throttle adjustments are then shifted to NB and PB for a quick change in throttle values to keep the landing velocity constant to -0.2 m/s as thrust gets more effective within IGE range as shown in Table 1(B). NS and PS do not provide enough thrust changes to keep the landing velocity constant under IGE range. To improve the controller performance to next level, landing controller operation under IGE range is divided into two steps. First is IN_IGE_FG; quadcopter starts entering inside IGE range in this region. Therefore, throttle adjustment commands are shifted toward higher gain values to provide enough throttle adjustments to overcome IGE. Second region is IN_IGE_NG; IGE gets more effective in this region which delays the landing time and keeps the quadcopter hovering for long time before complete landing. To overcome this problem, gains are shifted more toward high values which makes the throttle adjustment commands to work between NB and PB values.

PB and NB landing velocity inputs will get invalid due to very low altitude; therefore, only PB and NB throttle adjustment commands will perform rest landing operation. The transition region between Mode A and Mode B needs to be smooth to avoid any kind of abrupt speed change. This transition response depends on the overlap region between IN_IGE_FG and OGE_Small fuzzy linguistic variables. Figure 7 explains operation regions for Mode A, Mode B and a transition region between Mode A and Mode B.

5 Simulation and Experimental Studies

5.1 Simulation Studies

MATLAB/Simulink-based quadcopter simulation system is used to test and perform a comparative study between PID and proposed fuzzy logic controller. This simulation system is available on MATLAB file exchange, and it is free to use. The basic purpose of this system is to study the behavior of a quadcopter system and how different parameters affect the quadcopter flight. In this simulation system, quadcopter parameters can be changed including flight controller, quadcopter configuration, and initial conditions, and a GUI is provided to demonstrate actual moments of quadcopter in three-dimensional space.

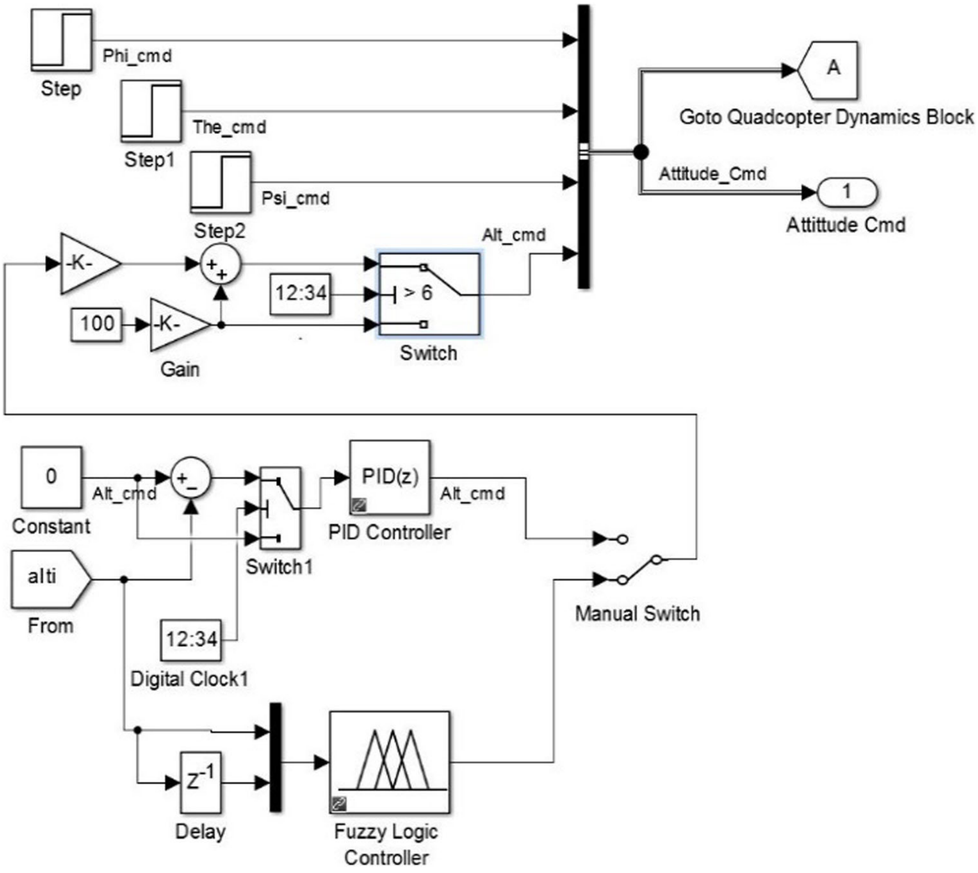


Fig. 9 Auto-landing controller implementation

The full system is shown in Fig. 8. An additional block is added to introduce IGE during the landing process. After at certain altitude, this block boosts the thrust effect exponentially with respect to altitude which reduces the landing speed and acts as real IGE in a practical scenario.

The attitude control block in Fig. 9 sends the roll, pitch, yaw, and altitude commands. Auto-landing controller is implemented in attitude control block as shown in Fig. 8. First, the PID controller-based auto-landing process is made and tested after optimum gain tuning. Then, the proposed fuzzy logic-based system is implemented and tested for the same model. Equation (3) shows general PID controller equation.

$$\mathrm{PID} = K_{\mathrm{p}} e\left(t\right) + K_i \int_0^t e\left(\tau\right) \mathrm{d}\tau + K_{\mathrm{d}} \frac{\mathrm{d}e\left(t\right)}{\mathrm{d}t} \tag{3}$$

Here, K_{p} is proportional gain, K_{i} is integral gain, and K_{d} is differential gain. The term 'e' represents the error. PID gain tuning is challenging process. It requires a lot of hit and trails to achieve perfect gain values. Figure 10 shows the response of PID landing controller on different gains. It can be seen that landing speed changes by changing the gain values. "Red line" shows the fastest response as compared to other responses. However, landing speed for this response is too fast which will eventually crash the quadcopter. The negative bump in this figure (red circle) shows the bumping of quadcopter due to high-speed impact with the ground. However, other responses are safer but landing process gets very slow due to IGE as quadcopter reaches near the ground. The best possible response of PID landing controller is selected after testing different gain values.

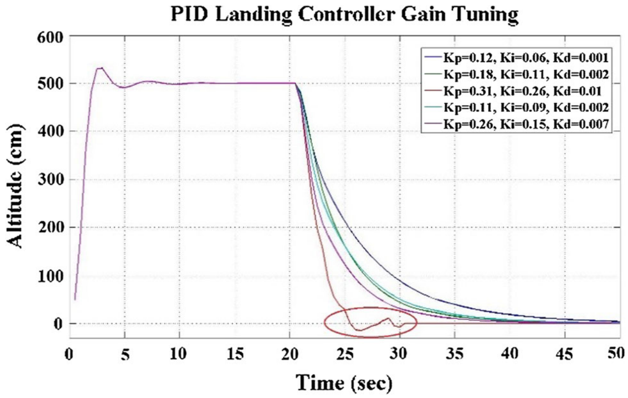


Fig. 10 PID landing control gain tuning

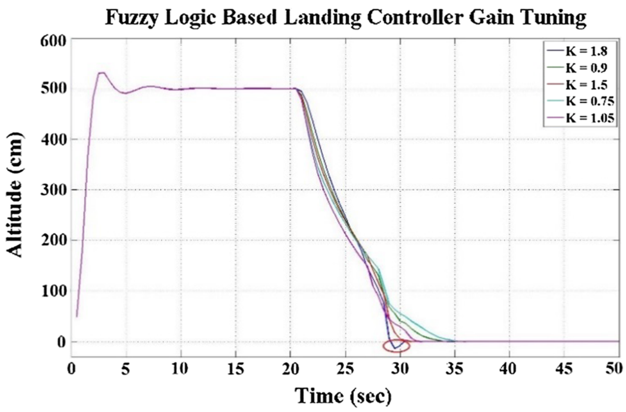


Fig. 11 Fuzzy logic landing controller gain tuning

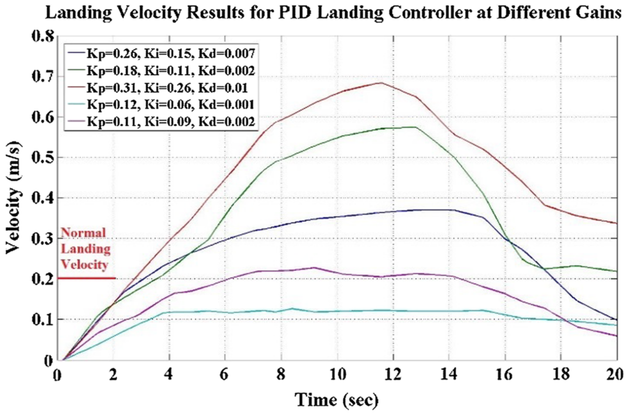


Fig. 12 Landing velocities of PID landing controller at different gains

As mentioned earlier, the output of fuzzy logic controller (throttle adjustment) can be multiplied by a gain value to tune the best response from fuzzy logic-based landing controller. Figure 11 shows the fuzzy logic landing controller response at different gain values. Higher gain value improves the response time because desired landing frequency is achieved faster but very high gain value disturbs the landing under IGE threshold. Therefore, the correct gain value should be chosen for the best response as it can be seen in the figure, the result with a gain value of 1.05 shows the best response. Higher gain values than this cause bouncing (red circle in Fig. 11) or very fast landing speed under IGE threshold, and smaller gain value shows a slow response. Equation (4) explains the fuzzy logic output calculation by COG (center of gravity) technique. In this equation, $\text{Throttle}_{\text{adj}}$ is fuzzy logic output value, K is the throttle adjustment gain, i indicates the input number, μi represents corresponding membership value, μ is fuzzy input, and n is a total number of inputs.

$$\text{Throttle}_{\text{adj}} = K\left(\frac{\sum_{i=1}^{n} \mu_i \mu\left(i\right)}{\sum_{i=1}^{n} \mu\left(i\right)}\right) \tag{4}$$

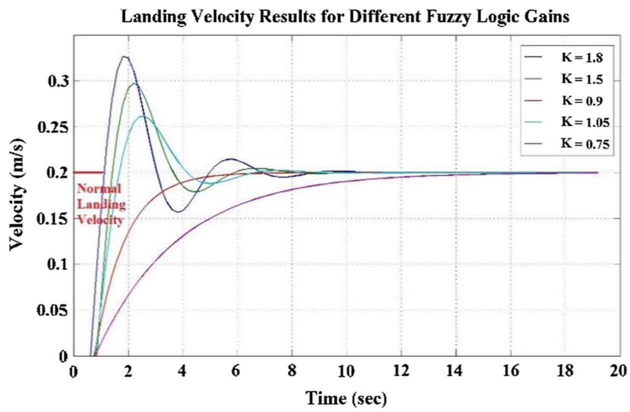


Fig. 13 Landing velocity response for fuzzy logic landing controller at different gains

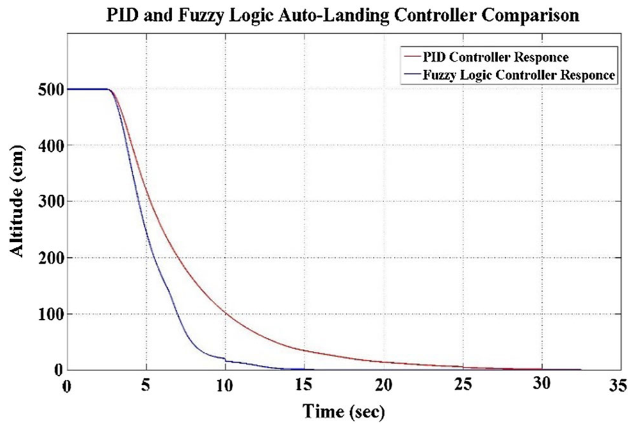


Fig. 14 Response comparison of PID and fuzzy landing controller

Landing velocity control is the second property of proposed fuzzy logic landing controller. Figure 12 shows landing velocity response over time at different gains for PID landing controllers. It can be seen that landing velocity amplitude increases with high gains and decreases with lower gain values. Furthermore, landing velocity for PID controller shows dynamic behavior over time; initially, landing velocity increases, and later it starts decreasing due to ground effect. PID controller is unable to keep the landing velocity constant, which causes safety issues due to fast landing velocity if high gains are chosen and it takes longer landing time when gains are small, whereas proposed fuzzy logic-based landing controller keeps landing velocity constant at the desired value. It can be seen in Fig. 13 that fuzzy logic landing control sets the landing velocity at a constant value and changing the gain value affects the settling time. Figure 13 shows that lower gain slows down the settling time and higher gains add initial high-velocity spike. Perfect gain value can be chosen with slight tuning.

A comparison between PID and proposed fuzzy logic landing controller is shown in Fig. 14. We can see that PID

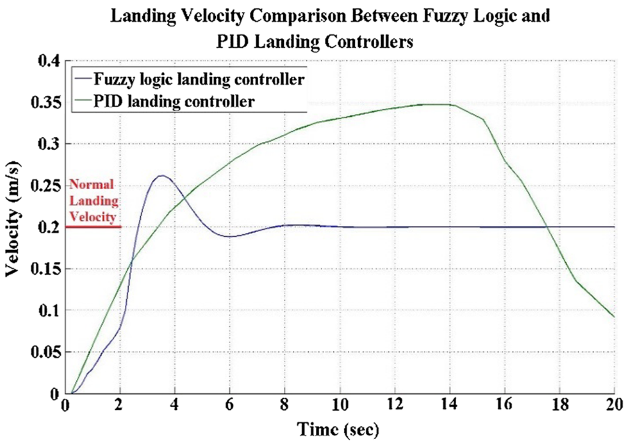
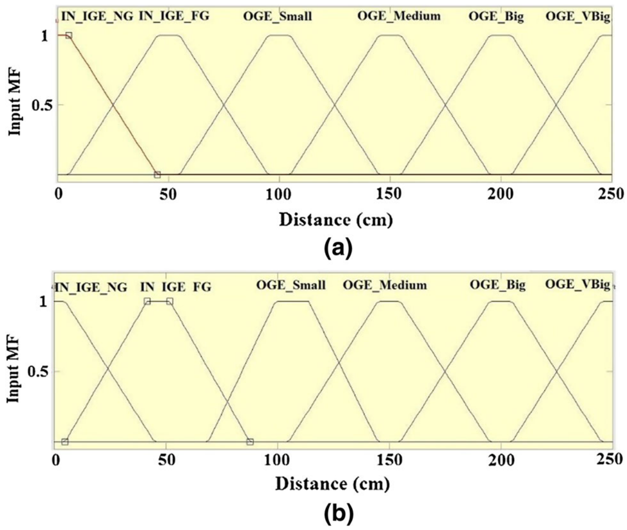


Fig. 15 Landing velocity response comparison of PID and fuzzy logic controllers

controller shows a slow response as altitude decreases due to its fixed gains. PID controller is unable to handle IGE, which makes the landing process slower, whereas fuzzy logic controller shows much fast response due to its speed and altitude control strategy and it covers the IGE efficiently according to designed rule base, which makes the landing process faster as compared to PID. Moreover, speed control strategy ensures the safety of quadcopter during landing. Figure 14 shows the auto-landing response comparison between PID landing controller and fuzzy logic landing controller.

Here, it can be seen that PID controller landing velocity gets slower under IGE, whereas the fuzzy logic controller manages to control and set the landing velocity fast enough for quick landing. PID controller-based auto-landing technique takes almost 25 s for complete safe landing process from 500-cm altitude level, whereas fuzzy logic-based auto-landing controller reduced the landing time to 14 s. Landing speed comparison is shown in Fig. 15. Here, PID-based landing system shows very dynamic and unstable velocity response, whereas fuzzy logic landing controller provides steady-state landing velocity, which makes the landing process smooth and quicker than PID. These simulation results provide satisfactory information to prove the supremacy of the proposed technique over ordinary one. However, experimental results are needed to verify the practical feasibility of proposed auto-landing algorithm in the physical environment.



(a)

(b)

Fig. 16 **a** 50% overlap b/w IN_IGE AND ODE Small. **b** 30% overlap b/w IN_IGE AND ODE Small

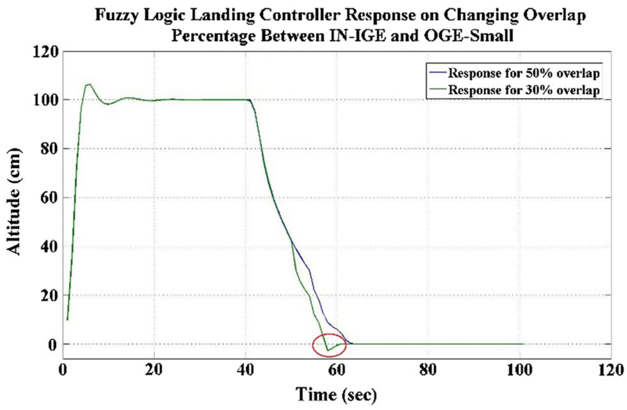


Fig. 17 Fuzzy logic response for 50 and 30% overlap between IN_IGE and OGE_Small

The overlap region between IN_IGE_FG and OGE_Small fuzzy linguistic variables is responsible for controlling the transition between Mode A and Mode B. A 50% overlap is used for smooth transition between both modes. Change in the overlap region between IN_IGE_FG and OGE_Small changes the transition response. In Fig. 16a, 50% overlap is shown, whereas Fig. 16b shows 30% overlap region. Figure 16 shows the change in response when overlap region is changed between IN_IGE_FG and OGE_Small fuzzy linguistic variables. It can be seen in Fig. 17 that 30% overlap increases the landing speed in Mode B, which causes bumping response (red circle) due to high-speed impact with the ground. Similar response will occur if overlap region is changed between IN_IGE_FG and IN_IGE_NG.

5.2 Experimental Studies

A quadcopter UAV system is used for experimental verification of designed fuzzy logic landing controller. Pixhawk is the main flight controller, which includes built-in IMU sensor, compass, and barometer, and it is capable of holding more devices such as GPS, external compass, RF modules for telemetry, LIDAR, etc. It can handle up to six ESCs for hex-copter configuration also. An RF transceiver is used to send altitude information to PC at every 500 ms to store the data in the auto-landing process. The fuzzy logic controller is a complicated controller, which takes long computation

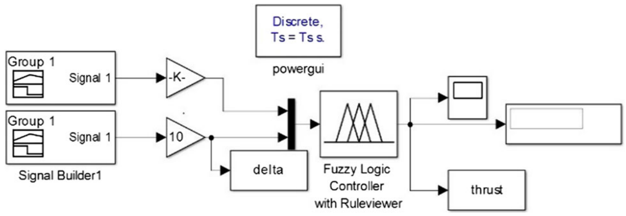


Fig. 18 Simulink block for generation of lookup table

time. It is possible to implement fuzzy logic controller on most high-end MCUs, but the problem occurs when we need a high-speed response from the controller similar to quadcopter landing scenario. Fuzzy logic controller design is a time-consuming process when it comes to real-time calculation. Therefore, a lookup table-based fuzzy logic controller is designed to make the landing process faster and efficient.

5.2.1 Lookup Table Designing

A Simulink-based fuzzy logic system is designed to generate output data from the fuzzy logic controller. Figure 18 shows the Simulink-based fuzzy logic system with virtual inputs. These inputs are selected according to all possible variations. Variable velocity signal is made ranging from − 0.7 to 0.5 m/s using signal builder block, and output throttle adjustment values are saved for 0–250-cm altitude with a 5-cm difference. Conclusive lookup table has been created with 25 different velocity values aligned in rows with 52 different speeds as columns, which made lookup table of 1300 values.

This lookup table is written inside the Arduino Mega-based landing controller. The landing controller measures the altitude and velocity using LIDAR and then rounds off the data to exactly match the lookup table inputs. These values are then used as input for the lookup table-based fuzzy logic controller, which generates the corresponding throttle adjustment output for these specific inputs, and these data are further sent to Pixhawk flight controller via UART communication to update the attitude commands for a safe landing. These instructions are updated every 200 ms.

Landing response comparison between PID landing controller and the proposed fuzzy logic landing controller is shown in Fig. 19. As we can see, altitude is measured in cm unit via LIDAR rangefinder and data are recorded every 500 ms. Altitude information shows the response of the proposed fuzzy logic-based landing controller and PID-based controller in practical quadcopter system, and it can be seen that fuzzy logic-based landing process is quicker and more efficient as compared to PID controller as it was in simulation test system. In the start, the quadcopter is made to hover at an altitude slightly above 500 cm. At this point, the auto-landing process is triggered via RC and quadcopter initiates the landing process as shown in Fig. 19. The same process is repeated for PID-based auto-landing and fuzzy logic-based auto-landing algorithm. Results show that PID controller takes 30 s for a successful landing and fuzzy logic-based landing controller takes about 20 s from 500 to 0 cm altitude. These results prove the feasibility of proposed controller in the physical environment. The physical configuration of quadcopter experimental system is shown in Fig. 20, and photographs in Fig. 21 are taken during hovering and landing process.

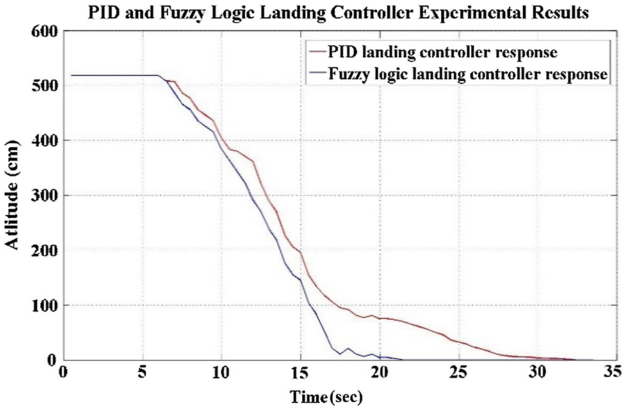


Fig. 19 Experimental results of fuzzy logic controller and PID controller

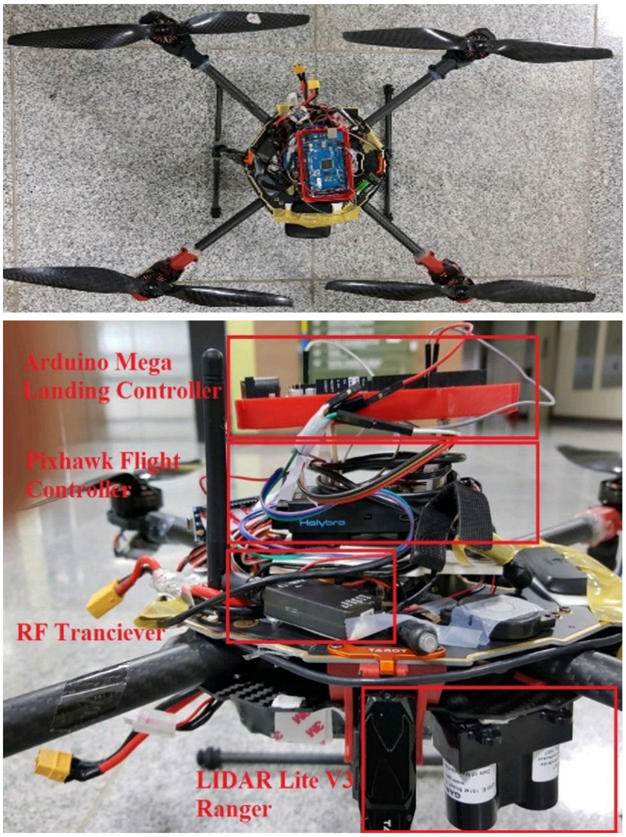


Fig. 20 Quadcopter experimental setup

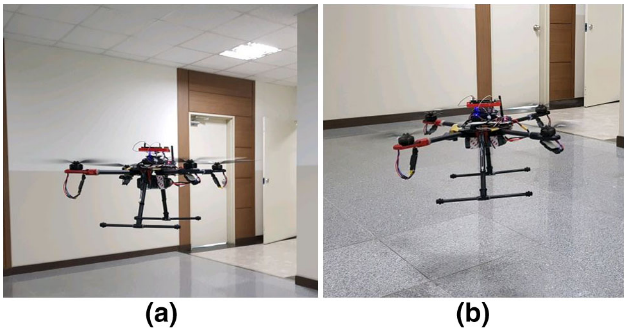


Fig. 21 **a** Hovering, **b** landing

Table 2 Comparison table

	Settling time (s)	Sum square error (SSE)
PID landing controller	26	22.01
Fuzzy logic landing controller	16	19.43

Settling time and sum square error (SSE) are calculated to verify the effeteness of proposed landing controller over the conventional one. Comparison table between PID landing controller and fuzzy logic landing controller is shown in Table 2. It can be seen that PID landing controller settling time and SSE are significantly greater than fuzzy logic landing controller. This argument proves the supremacy of proposed fuzzy logic controller over conventional PID landing controller.

6 Conclusion

A fuzzy logic-based position and speed control auto-landing technique has been proposed in the paper. This system uses real-time position and velocity information to control the attitude of quadcopter UAV system. Velocity control algorithm ensures the secure landing and prevents the quadcopter to hit the ground at high speed. Position control measures the altitude, and commands are generated to overcome IGE near ground level for smooth and quick landing. This combination of position and speed control increases the efficiency by minimizing the landing time and provides the safety assurance as compared to conventional controllers. Lookup table-based fuzzy logic technique improved the execution time as compared to a normal fuzzy technique, which enhanced the response time during landing. A simulation platform is used to compare the proposed landing controller with conventional PID controller. Results showed significant improvement over the previous technique. Furthermore, the proposed technique was tested on a practical quadcopter system to verify its practical ability in the physical environment. Results are provided to verify the practical implementation in real time. It can be seen that proposed landing controller is faster and much safer as compared to conventional techniques. This approach is applicable to all types of copters without any major changes.
In the future, an IR camera will be used to detect the platform to pinpoint the landing position for a quadcopter.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest regarding the publication of this paper.

References

1. Luukkonen, T.: Modelling and control of quadcopter. Independent research project in applied mathematics. Espoo, August 22 (2011)
2. Huang, H.; Hoffmann, G.M.; Waslander, S.L.; Tomlin, C.J.: Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: IEEE International Conference on Robotics and Automation, pp. 3277–3282 (2009)
3. Lavi, B.: Human body detection and safety care system, for flying robot. In: Second International Conference on Advanced Information Technologies and Applications (ICAITA), pp. 329–337 (2013)
4. Jeffrey, J.; Yanev, A.; Biaz, S.; Murray, C.: UAV Quadcopter Landing on a Displaced Target. Technical Report #CSSE14-05
5. Sefidgari, B.L.; Shamchi, S.P.: Auto-landing process for autonomous flying robot by using image processing based on edge detection. In: CS and IT-CSCP, pp. 361–368 (2014)
6. Bouabdallah, S.: Design and Control of Quadrotors with Application to Autonomous Flying. Ph.D Thesis, Ecole Polytechnique Federale de Lausanne, pp. 129–136 (2007)
7. Mellinger, D.: Trajectory Generation and Control for Quadrotors. Phd's Thesis, University of Pennsylvania, pp. 136–152 (2012)
8. Yingcai, B.; Haibin, D.: Implementation of autonomous visual tracking and landing for a low-cost quadrotor. Optik Int. J. Light Electron Opt. **124**, 3296–3300 (2013)
9. Charland-Arcand, G.: Controle non lineaire par backstepping d'un helicoptere de type quadrotor pour des applications autonomes. Master's thesis, Ecole de technologie sup'erieure, pp. 165–180 (2014)
10. Gomes Carreira, T.: Quadcopter Automatic Landing on a Docking Station. Master's thesis, TECNICO LISBOA, pp. 63–95 (2013)
11. Chamseddine, A.; Charland-Arcand G.; Akhrif O.; Gagne, S.; Gagnon, F.; et Couillard, D.: Optimal position seeking for unmanned aerial vehicle communication relay using only signal strength and angle of arrival. In: IEEE 53rd Annual Conference on Decision and Control (CDC) (2014)
12. Zhu, H.; Yuan, S.; Shen, Q.: Vision/GPS-based docking control for the UAV autonomous aerial refueling. In: Guidance, Navigation and Control Conference (CGNCC), IEEE Chinese (2016)
13. Beck, H.; Lesueur, J.; Charlan-Archand, G.; Akihrif, O.; Gagon, S.; Gagon, F.; Couillard, D: Autonomous takeoff and landing of a quadcopter. In: International Conference on Unmanned Aircraft Systems (ICUAS) (2016)
14. Khan, M.: Quadcopter flight dynamics. Int. J. Sci. Technol. Res. **10**(8), 1–5 (2014)
15. Borowcky, A.; Nuguyrn, D.-T.; Nguyen, D.Q.; Saussie, D.; Le N.J.: Autonomous landing of a multirotor micro air vehicle on

a high velocity ground vehicle. In: IFAC-PapersOnLine, vol. 50, no 1 (2017)

16. Borshchova, I.; O'young, S.: Marker-guided auto-landing on a moving platform. Int. J. Intell. Unmanned Syst. **5**(1), 24–28 (2017)
17. Nguyen, P.H.; Kim, K.W.; Lee, Y.W.; Park, K.R.: Remote Marker-Based Tracking for UAV Landing Using Visible-Light Camera Sensor. In: Sensors/MDPI (2017)
18. Torres, C.; Wechsler, A.; Diaz, R.; Vargas, C.; Shinde, P.; Tosunoglu, S: Quadcopter with Vertical Landing Payload System. Department of Mechanical and Materials Engineering, Florida International University
19. Radhakrishnan, A.: An Experimental Investigation of Ground Effect on a Quad Tilt Rotor in Hover and Low-Velocity Forward Flight. Ph.D. Thesis, University of Maryland, College Park (2006)